



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Název projektu	Rozvoj vzdělávání na Slezské univerzitě v Opavě
Registrační číslo projektu	CZ.02.2.69/0.0./0.0/16_015/0002400

Informační a internetové technologie

Distanční studijní text

Josef Botlík

Karviná 2021



**SLEZSKÁ
UNIVERZITA**
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

Obor: 061 Informační a komunikační technologie (ICT)

Klíčová slova: Architektura počítačů, mobilní ICT, komunikace, sítě, internetové technologie, služby Internetu, WWW, HTML, XML

Anotace: Stručná anotace studijní opory (1 až 2 odstavce)

Cílem studijní opory je průřezově seznámit studenty s informačními a komunikačními technologiemi se zaměřením na služby Internetu. Studenti se v úvodu seznámí s architekturou počítačů, současným rozdělením počítačového trhu. CISC a RISC architekturou, základní koncept hardware si objasní pomocí Von Neumannovo schématu a Harvardské architektury. Dále se seznámí se standardy současných PC založených na modularitě, nechybí náznak vývojových tendencí a trendů, srovnání platform a OS, vývoj, trendy. Aktuální nabídka a parametry počítačových komponentů, procesory, paměti, periferie, rozhraní, zobrazovací hardware, vývoj hardware. Studenti se stručně seznámí s principy skalárních počítačů, superskalárních počítačů či kvantového počítače a s trendy v segmentu superpočítačů.

Dále je obsahem výukového materiálu úvod do problematiky mobilních informačních technologií, seznámení s problematikou OS pro tablety, PDA či mobilní telefony na platformě Android a Windows.

V oblasti komunikací a sítí jsou v tomto studijním materiálu vysvětleny základy sítí, síťový a komunikační hardware, bezdrátové technologie a platformy, vývojové trendy v oblasti komunikací IT, IoT, IoE, IoX. Dále se studenti seznámí se základy internetových technologií a službami Internetu. Ve skriptech najdou studenti úvod do komunikací v síti, seznámí se s protokoly, ISO OSI modelem a TCP/IP protokolem se zaměřením na služby Internetu plynoucí z aplikačních protokolů. Vysvětleny budou základní příkazy (ipconfig, tracert, ping, arp), služby pro přenos souborů a vzdálená správa. Studenti se rovněž seznámí s pilíři a organizací Internetu, správou a organizací domén. či bezpečností na Internetu. Dále je součástí výukového textu úvod do WebMasteringu. administrace domén, registrace domén. webhostingu a. webových technologií. Je zde seznámení s WWW serverem, publikováním www stránek na www serveru, správa a konfigurace lokálních serverů Apache a Xitami.

V souvislosti se základními technologiemi pro tvorbu WWW budou studenti seznámeni se značkovacím jazykem HTML, XML, se strukturou HTML dokumentu, základními značkami a tagy, s atributy či elementy. Studenti najdou ve skriptech i úvod do dynamiky ve www stránkách, JavaScriptu a CSS. Nechybí ani seznámení s OnLine a WYSIWYG editory.

Autor: **Ing. Josef Botlík**

Obsah

ÚVODEM.....	7
RYCHLÝ NÁHLED STUDIJNÍ OPORY.....	8
1 ARCHITEKTURA POČÍTAČŮ	9
1.1 Hardware, software	9
1.2 Software	10
1.3 Operační systémy.....	13
1.4 Počítače, východiska, koncepty	15
1.5 Instrukční sada, programování.....	24
1.5.1 Další související pojmy.....	28
1.5.2 Klasifikace počítačů.....	29
1.5.3 Superpočítače	32
2 VÝVOJOVÉ TRENDY A MOBILNÍ IT	35
2.1 Základní komponenty PC	36
2.1.1 Základní deska	37
2.1.2 Procesor.....	39
2.2 Technologické trendy v existujícím hardware.....	45
2.2.1 Vývojové trendy v oblasti PC.....	45
2.2.2 Trendy a inovace v segmentu procesorů.....	50
2.3 Vývoj a inovace zařízení založených na existujících technologiích.....	52
2.3.1 Komunikace pro podporu IoT, IoE.....	52
2.3.2 Rozšířená a virtuální realita	55
2.4 Vývoj nových technologií.....	60
2.4.1 Qubit	60
2.4.2 Kvantový počítač	62
2.4.3 Další počítačové technologie	65
3 KOMUNIKACE, SÍŤ.....	72
3.1 Principy komunikace v sítích.....	72
3.1.1 Komunikační protokoly	73
3.2 Konfigurace připojení	74
3.2.1 Příkazový řádek a základní příkazy	77
3.3 Typy sítí dle jejich dosahu	82

3.3.1	WAN (Wide Area Network).....	82
3.3.2	MAN (Metropolitan Area Network).....	82
3.3.3	LAN (Local Area Network).....	82
3.3.4	PAN (Personal Area Network)	83
3.4	Nejčastější síťové architektury v prostředí internetu	83
3.4.1	Architektura klient/server	83
3.4.2	Architektura P2P	84
4	ZÁKLADY INTERNETOVÝCH TECHNOLOGIÍ, SLUŽBY INTERNETU	85
4.1	Služby Internetu	86
4.2	Protokol TCP/IP.....	86
4.2.1	TCP a UDP protokoly	88
4.2.2	IP protokol	89
4.2.3	Aplikační protokoly	90
4.3	Služby založené na aplikačních protokolech	91
4.3.1	FTP.....	92
4.3.2	Telnet	96
4.3.3	Poštovní služby	97
4.3.4	HTTP.....	100
4.4	Organizace v prostředí internetu	100
4.5	Internetové standardy.....	102
4.5.1	RFC (REQUEST FOR COMMENT) DOKUMENTY	103
4.6	Síťové modely.....	104
4.6.1	ISO/OSI model.....	105
4.6.2	TCP/IP model.....	107
5	PILÍŘE A ORGANIZACE INTERNETU, WWW	109
5.1	Tři pilíře Internetu.....	109
5.1.1	Komunikační protokoly	110
5.1.2	Fyzická infrastruktura	110
5.1.3	Domény.....	115
5.2	Fungování internetu	115
5.2.1	Služby ISP.....	117
5.2.2	Tier.....	118
5.3	Identifikace souborů a počítačů	118

5.3.1	URI, URL, URN	119
5.3.2	Doména, IP adresa	119
5.3.3	WWW klient	121
5.3.4	WWW server	122
6	ZÁKLADNÍ TECHNOLOGIE PRO TVORBU WWW	124
6.1	Princip Hypertextu	124
6.2	Služba WWW	125
6.2.1	Základy tvorby webu	126
6.2.2	Internetová stránka	127
6.2.3	Typy webových prezentací	127
6.3	Technologie pro vytváření webových stránek	127
6.3.1	Nástroje pro tvorbu webových prezentací	128
6.4	Práce s hypertextem	129
6.4.1	HTML	129
6.4.2	Verze jazyka HTML	131
6.4.3	XML	133
6.4.4	Formátování v HTML, kaskádové styly	134
6.4.5	JavaScripty	136
6.5	HTML	138
6.5.1	HTML struktura	139
6.5.2	Textové tagy	140
6.5.3	Layout	142
6.5.4	Tabulky	142
6.5.5	Seznamy	143
6.5.6	Multimédia	143
6.5.7	Formuláře	144
6.5.8	Ostatní tagy	145
6.5.9	Zastaralé tagy (podporované z důvodů kompatibility)	146
6.6	Kaskádové styly	146
6.6.1	Charakteristika CSS3	146
6.6.2	Příkazy a klíčové výtazy	147
6.7	JavaScript	156
6.7.1	Úvod do JavaScriptu	157

6.7.2	Charakteristiky jazyka	157
6.7.3	Objektový model.....	158
6.7.4	Objekt window.....	159
6.7.5	Vlastnosti a metody objektu window.....	159
6.7.6	Metody	162
6.7.7	základní datové typy, proměnné, funkce, operátory	162
6.7.8	Proměnné	163
6.7.9	Funkce.....	164
6.7.10	Operátory	165
6.7.11	Větvení.....	166
7	WEBDESIGN, WEBMARKETING, OPTIMALIZACE.....	168
7.1	Trendy současného webdesignu	169
7.1.1	Vývoj webdesignu	169
7.2	Základní pravidla webdesignu	170
7.2.1	Nástroje pro optimalizaci.....	171
	LITERATURA	172
	SHRNUTÍ STUDIJNÍ OPORY	174
	PŘEHLED DOSTUPNÝCH IKON.....	175

ÚVODEM

Současná doba digitálního zpracování informací vyžaduje nejen standardní znalost práce s PC, ale komplexně informačními a komunikačními technologiemi (ICT) se zaměřením na mobilitu a komunikace. Je vhodné se nejdříve seznámit s východisky, protože klasické chápání významu „počítač“ je dnes potlačeno. Je problematické si uvědomit hranici, kde začíná či končí třída zařízení označovaná jako počítač, tablet či mobilní telefon, je problematické stanovit hranice a pravidla pro kategorizaci. Proto je nutné seznámit se i se základy počítačů, principy a zásadami fungování, které umožní pochopit souvislosti, ale i predikovat tendence a trendy potencionálního vývoje. Je neoddiskutovatelné, že jeden ze současných trendů je mobilita, která je však plně závislá nejen na hardware, ale především na formách komunikace a na prostředí a službách dostupných v Internetu.

Studijní opora je určena především studentům oboru Manažerská informatika, lze ji však doporučit všem, kteří se chtějí seznámit s informačními technologiemi, zejména pak je určena pro studenty předmětu Informační a internetové technologie. Obecně je studijní opora určena studentům prvního ročníku ekonomické fakulty s cílem zvládnout základní informace o službách Internetu. Nejsou kladeny žádné požadavky na předchozí znalosti, předpokládá se však aktivní práce s příslušným prostředím, ať už se jedná o aktivní komparaci nabídky hardware pomocí stránek prodejců, či praktické odzkoušení uváděných služeb, např. registrace domény.

Při studiu mohou studenti používat standardní distanční prvky výuky, především jsou ve výukovém textu následující prvky:

- Otázky – shrnují okruh látky a slouží k ověření znalostí dílčí části kapitoly.
- Odpovědi – slouží k ověření znalostí, jsou uvedeny na konci kapitoly.
- Pro zájemce – tato část rozšiřuje výklad.
- Průvodce studiem – pomáhá studentům uvědomit si souvislosti nebo upřesňuje probírané téma.
- K zapamatování – upozorňuje studenty na důležité učivo.
- Samostatný úkol, Námět na tutoriál a Úkol k zamyšlení – slouží studentovi k zamyšlení, uváděný problém bude rozebrán na tutoriálu, v textu není k němu odpověď.

Mimo tyto prvky obsahuje standardně každá kapitola:

- Rychlý náhled kapitoly, ve kterém se seznámí s obsahem učiva v kapitole.
- Cíle kapitoly, kde se student ve stručných bodech dozví, co by se měl v kapitole naučit nebo k čemu je kapitola určena.
- Čas potřebný ke studiu. Uvádí orientační časový fond, který by student měl věnovat problematice kapitoly pro její úspěšné zvládnutí. Do fondu je počítána i samostatná práce a řešení příkladů.
- Shrnutí kapitoly, ve kterém je zopakováno, co bylo cílem a co by měl student umět po studiu dané kapitoly.

Text je pro přehlednost doplněn margináliemi, které umožní rychlejší orientaci ve studijním materiálu.

RYCHLÝ NÁHLED STUDIJNÍ OPORY

Dostává se vám do rukou výukový text, pomocí kterého se máte seznámit s technickými prostředky a informačními technologiemi používanými v podnikatelské praxi. Vzhledem k tomu, že informační technologie a výpočetní technika se vyvíjí obrovským tempem, doporučuji v průběhu studia doplňovat informace na Internetu. Porovnávejte a doplňujte své poznatky především ze stránek prodejců hardware a ze stránek uvedených v textu, část textu i obrázků je převzata (případně převzata a upravena) ze stránek www.alza.cz a www.svethardware.cz. V těchto skriptech se taky vychází z těchto zdrojů. Vzhledem k tempu vývoje těchto technologií mohou být už při vašem studiu některé údaje zastaralé, a to i přesto, že jsem se snažil v době psaní těchto skript vycházet z nejnovějších technologií. Část skript se věnuje počítačům PC, jejich vývoji a architektuře. Počítače tvoří a nadále budou tvořit jádro technických prostředků informačních systémů. Vše ostatní se ve větším či menším měřítku odvíjí od počítače, jeho komponentů, periférií či vstupů a výstupů.

Současné trendy v oblasti hardware plynou z krátkodobých a dlouhodobých koncepcí předních výrobců hardware, potřeby trhu (požadavků zákazníků) a technologických inovací. Trendy lze shrnout do tří oblastí: vývoj a modifikace existujícího hardware, vývoj a inovace zařízení založených na existujících technologiích a vývoj nových technologií. Současné tendence nelze zúžit na vývoj hardware PC, protože technologie ICT se prolínají do různých platforem. Zejména v době Internetu věcí (Internet of Things, IoT) a Internetu všeho (Internet of Everything, IoE), je standardem nejen komunikace zařízení mezi sebou, ale i jejich řízení a komunikace s člověkem prostřednictvím internetu, tedy komunikace věcí, lidí a procesů. Ta vyžaduje sofistikované nástroje a technologie, ve své podstatě to znamená distribuci procesorů, paměti a bezdrátových technologií do komunikujících zařízení. Předpokládané vývojové trendy v oblasti IT budou tedy směřovány především do roviny komunikací, mobility a virtuální reality, do roviny virtuálního, propojeného a komunikujícího světa. S oblastí komunikací vyvstává i otázka, jaké technologie budou ovlivňovat funkcionalitu Internetu. Pro zvládnutí služeb Internetu je nutné se seznámit se základními pravidly a principy fungování Internetu tak, aby bylo možné v maximální míře využívat služeb, které Internet poskytuje, se zaměřením na publikování a tvorbu www stránek. Studenti se seznámí se zásadami fungování základních aplikačních protokolů a jimi podporovanými aplikacemi. Naučí se pracovat se základními diagnostickými nástroji, nutnými pro zajištění fungující komunikace. Současně se seznámí se základy server-klient architektury, se vzdáleným přístupem pomocí FTP protokolu, se zásadami webhostingu a hostingů internetových portálů. Naučí se používat vybrané webové technologie tak, aby mohli plnohodnotně pracovat s vybranými internetovými systémy a využívat internet jako moderní prostředek komunikace a prostředí prezentace. V rovině tvorby www stránek se student seznámí se základními technologiemi a jazyky, seznámí se s hypertexty, s HTML, XML, dále s CSS a JavaScriptem. Tvorbu www stránek je prezentována na různých úrovních, od textového editoru, přes Wysiwyg editory až po online projekty.

1 ARCHITEKTURA POČÍTAČŮ

RYCHLÝ NÁHLED KAPITOLY



Cílem kapitoly je prohloubit znalostí studentů v oblasti hardware a vývojových trendů. Studenti se seznámí se základními koncepty počítačů, upřesní si základní pojmy hardware a software. Seznámí se s počítačovou architekturou, s koncepty počítačové architektury a s vývojovými trendy.

CÍLE KAPITOLY



Cílem kapitoly je seznámit studenty se s modulární koncepcí PC, se základním hardwarem. Dále je cílem ukázat rozdíly mezi počítačovými architekturami, seznámit nejen s klasickými procesory PC ale i s ARM procesory, jako platformě mobilních technologií. Cílem je rovněž objasnit vztahy mezi hardwarem a softwarem a nastínit vývojové trendy. Student pochopí vztahy mezi komponenty PC a rozdíly mezi platformami ICT. Po absolvování kapitoly student rovněž porozumí základním konceptům možných vývojových tendencí a platformem, jako uhlíkové či kvantové počítače.

KLÍČOVÁ SLOVA KAPITOLY



Hardware, Software, Von Neumannovo schéma, Harvardská architektura, CISC, RISC, ARM, CPU, Skalární architektura, Superskalární architektura, alternativy křemíkové technologie, Superpočítače, Kvantové počítače.

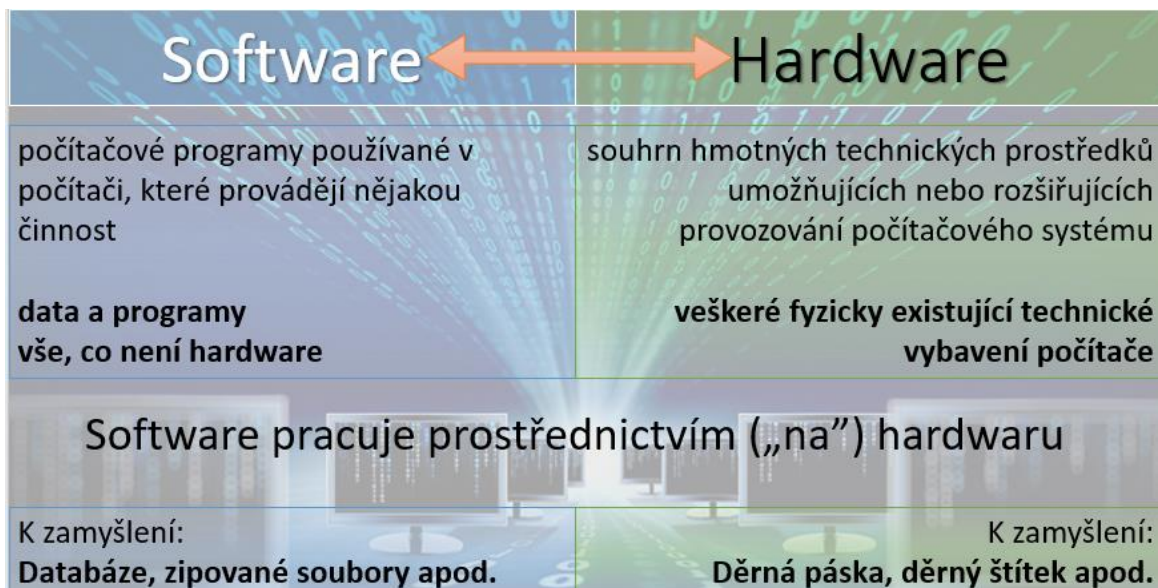
1.1 Hardware, software

Jak bylo uvedeno dříve, je problematické rozhodnout, kde lze hovořit o počítači a kde o jiných typech zařízení, například o mobilním telefonu. Proto je praktičtější hovořit ve dnešní době spíše o informačních a komunikačních technologiích než o počítači. Existuje však několik základních kritérií, která umožní specifikaci počítačů a ICT. Vycházejí z konceptu Turingova stroje, základní architektura je dána Von Neumannovým schématem a

pracují na principu programového vybavení (software), které je spouštěno na specifickém fyzickém zařízení (hardware).

Někteří autoři zmiňují, že přesná definice pojmů software a hardware je problematická. Jako hardware bývá označován souhrn hmotných technických prostředků umožňujících nebo rozšiřujících provozování počítačového systému. Je to tedy veškeré fyzicky existující technické vybavení počítače. Je zřejmé, že v některých případech nelze tuto definici účelně aplikovat, Například děrná páska, děrný štítek apod. je „hmatatelná“, tedy fyzická součást, je však spíše „nosičem dat“ nebo „záznamem software“. Nicméně, při srovnání s klasickým hardware najdeme analogii s paměťovými komponenty.

Obdobně, jako software bývají označovány počítačové programy používané v počítači, které provádějí nějakou činnost. I v tomto případě existuje několik nejasností, například v kontextu dat a programů. Někteří autoři namítají, zda je například komprimovaný (například zipovaný) program ještě programem, či jsou to pouhá data. Obdobně, program posílaný mailem jako příloha může být chápán jako data. Rovněž programová sekvence, která není přímo spustitelná (například makro napsané ve Wordu v prostředí Visual Basic nelze spustit v jiném prostředí) nemůžeme vždy chápat jako software. Někteří autoři proto označují za software vše, co není hardwarem.



Obrázek 1: Vztah mezi Hardware a Software

1.2 Software

Klasifikace software lze provádět podle více kritérií, zpravidla se však používá následující rozdělení:

- Podle funkce (použití)
- Podle práv (způsob licencování)

ROZDĚLENÍ PODLE FUNKCE

Podle funkce (použití) lze rozdělit software na systémové a aplikační programové vybavení.

Systémové programové vybavení je v informatice speciální (základní) programové vybavení, které umožňuje uživateli s počítačem pracovat, ovládat ho, spouštět aplikace a podobně. Systémový software se nachází na pomezí hardware a aplikačního software a běžný uživatel s ním přímo nepracuje (na rozdíl od aplikací). Mezi systémový software patří operační systém, pomocné nástroje a firmware.

- Firmware – software obsažený v hardware (BIOS, firmware vstupně-výstupních zařízení jako tiskárny, CD/DVD mechaniky, grafické a zvukové karty, ...).
- Operační systém – spravuje počítač, vytváří prostředí pro programy ,
 - jádro operačního systému - (včetně ovladačů zařízení),
 - pomocné systémové nástroje – pro správu operačního systému (formátování disků, nastavení oprávnění, utility, ...).

Mimo to lze do systémového software zařadit i **middleware**, což je software, který je mezi operačním systémem a aplikacemi, které jsou v něm spuštěné. Middleware v podstatě funguje jako skrytá transakční vrstva a umožňuje komunikaci a správu dat pro distribuované aplikace. Použití middlewaru umožňuje uživatelům provádět požadavky, jako je odesílání formulářů ve webovém prohlížeči nebo povolení webovému serveru vracet dynamické webové stránky na základě profilu uživatele.

Aplikační programové vybavení (zkráceně aplikace) - v informatice programové vybavení počítače, které umožňuje provádět nějakou užitečnou činnost. Aplikace využívají pro interakci s uživatelem grafické nebo textové rozhraní, případně příkazový řádek.

Aplikace se může skládat z několika počítačových programů. Umožňuje uživateli vykonávat nějakou užitečnou činnost, například:

- kancelářské aplikace: textový editor, tabulkový procesor, prezentační program,
- grafické programy: vektorový grafický editor, bitmapový grafický editor, CAD,
- vývojové nástroje: vývojové prostředí (pro tvorbu software), překladač, programovací programy, kompilátory, interprety, skriptovací jazyky
- zábavní software: počítačové hry, přehrávače digitálního zvuku a videa apod.,
- Antivirové a bezpečnostní programy – antivirus, firewall
- Webová aplikace (Web Application) - www prohlížeče, komunikátory, internetové aplikace (bankovníctví apod.)
- Účetní a ekonomické programy
- Utility – správci souborů

- atd. (různá rozdělení podle druhu, účelu, vzhledu, funkčnosti).

Pokud se na aplikační programové vybavení podíváme úžeji z pohledu ekonomických, zejména firemních aplikací, pak lze rozlišit například:

- Systémy správy podnikových zdrojů – ERP
- Systémy pro řízení výroby – APS
- Systémy pro správu lidských zdrojů – HRM
- Systémy pro řízení vztahů se zákazníky – CRM
- Systémy správy obsahu – ECM
- Ekonomické a účetní systémy
- Systémy pro zpracování plateb
- Systémy pro správu majetku
- Systémy pro správu IT
- Systémy pro podporu a řízení procesů – BPM
- Systémy pro řízení a plánování údržby – EAM
- Manažerské informační systémy, Business Intelligence
- Business Process Management nástroje
- GIS (Geografický informační systém) (Geographic Information System)

ROZDĚLENÍ PODLE PRÁV

Komerční programy -kupujeme (přesněji Licenci na jejich užívání) podobně jako jiné zboží. Tj. vybereme si program v obchodě, v katalogu apod., zaplatíme a program nainstalujeme na svém počítači. Součástí dodávky programu bývá vlastní program na nosiči (nejčastěji CD), manuál (návod k jeho užívání) technická podpora k programu, realizovaná buď pomocí e-mailu nebo přes telefon.(tzv. hotline). K takto zakoupenému programu máme většinou nárok na zlevněný upgrade, tj. přechod na novou verzi programu za nižší cenu.

OEM software - programy dodávané s počítačem, případně s nějakým technickým (hardwarovým) dílem. OEM program je většinou levnější (poměrně výrazně) než stejný program komerční. Jeho užívání je však vázáno na díl, ke kterému byl zakoupen. Nebývá k němu manuál, není nárok na technickou pomoc ani na levný upgrade na vyšší verzi programu.

Demoverze a zkušební verze - jsou plné nebo redukované verze programu, které zpravidla mají zablokované ukládání dat, případně fungují jen po určitou dobu. Slouží k vyzkoušení funkcí programu před jeho zakoupením.

Shareware - plně fungující programy s omezením. Po uplynutí doby je nutno zaplatit nebo program z počítače vymazat.

Freeware - můžeme zdarma používat i šířit. Nesmíme jej ale měnit ani používat ve svých vlastních programech apod.

GNU/GPL - je druh licence zajišťující zcela volný přístup k programům, šířeným pod touto licencí. GPL znamená „General Public Licence“ tedy „Obecná veřejná licence“. S takovýmto programem musí být šířen i jeho zdrojový kód, každý, kdo má příslušné znalosti, může program upravovat a vylepšovat.

Legální software. Každý program, který je nainstalován na počítači, musí mít platnou licenci. U freeware a programů šířených pod GNU/GPL licencí to je zajištěno automaticky, u ostatních programů musíme prokázat zakoupení licence

1.3 Operační systémy

Jedná se o základní softwarové vybavení počítače. Je to v podstatě množina programů, utilit, ovladačů a dalších softwarových komponent, které umožňují počítač ovládat. Tvoří rozhraní mezi aplikačními (uživatelskými) programy a hardwarem.

Mezi základní funkce OS patří organizace přístupu k datům, spouštění aplikací, řízení jejich průběh, přidělování hardwarových prostředků (čas a místo v paměti a na procesoru, přístup k periferním zařízením, přístup k datovým souborům atd.) činnost poskytování služeb aplikačním programům. Zavádí se do operační paměti hned po zapnutí počítače ze zvláštní oblasti pevného disku, Boot Sectoru, kde jsou fyzicky uloženy jeho soubory. Do Boot Sectoru se OS uloží jenom při instalaci. Když se OS zkopíruje z jiného PC klasickým způsobem, počítač ho nenajde a nenačte do paměti, a nebude tedy s ním možno pracovat. Je to soubor programů, které řídí činnost technického vybavení počítače a umožňují a usnadňují činnost aplikací. Hlavním úkolem OS je zprostředkovat vazbu mezi technickým vybavením počítače a aplikacemi. OS poskytuje aplikacím služby a aplikace voláním těchto služeb ovládají technické vybavení počítače. Aplikace by neměly mít přímý přístup k technickému vybavení počítače.

FUNKCE OPERAČNÍHO SYSTÉMU

- Řídí a spravuje přístup ke zdrojům výpočetního systému – rozdělování času procesoru, přidělování operační paměti, přístup k vnějším paměťům (disk, CD ROM, DVD), správa komunikace s perifériemi (klávesnice, myši, tiskárny, scanery atd.).
- Organizuje přístup k datům – zamezení neoprávněného přístupu.
- Řídí zpracování úloh – jejich přípravu, plánování a průběh tak, aby byla zajištěna maximální efektivita jejich zpracování.
- Podporuje komunikaci s uživatelem – provádění uživatelem zadaných příkazů a spouštění aplikací, informuje uživatele o vzniklých problémech či chybách.

- Způsob, jakým uživatel dává pokyny pro OS a v jaké formě dostává zprávy o výsledku provedené operace, je označován jako uživatelské rozhraní (User Interface).

Existují v podstatě dvě formy uživatelského rozhraní – textové uživatelské rozhraní, např. MS-DOS, uživatel zadává pokyny prostřednictvím tzv. příkazového řádku, grafické uživatelské rozhraní (Graphical User Interface, GUI) – převážná většina současných operačních systémů, obvykle též možnost zapisovat příkazy v příkazovém řádku.

DĚLENÍ OS:

- podle distributora a verzí (Windows, Linux, IOS, Android apod.),
- podle hardware (PC, servery, mobilní zařízení, průmyslové automaty),
- podle počtu uživatelů (více uživatelské, jedno uživatelské),
- podle počtu úloh (více úlohové, jedno úlohové) - multitasking, preemptivní multitasking.

V některých případech lze dále rozlišit klasifikaci:

- nesíťové/síťové
- podle úrovně sdílení CPU
- pole velikosti HW
- podle počtu procesorů
- podle míry distribuovanosti
- podle funkcí

Současný podíl operačních systémů na trhu v segmentu počítačů PC ukazuje obrázek 2.

	1. čtvrtletí	2. čtvrtletí	3. čtvrtletí	4. čtvrtletí
Windows	87,50%	81,70%	78,90%	76,70%
Chrome OS	5,30%	10,00%	11,50%	14,40%
macOS	5,80%	7,60%	8,40%	7,70%

Obrázek 2: Podíly dodávek operačních systémů v roce 2020 (podle. www.msn.com)

1.4 Počítače, východiska, koncepty

Východisko konceptu počítačů je definováno jako **Turingův stroj** (TS), což je teoretický model počítače popsáný matematikem Alanem Turingem.

Skládá se z procesorové jednotky, tvořené konečným automatem, programu ve tvaru pravidel přechodové funkce a nekonečné pásky pro zápis mezivýsledků. Využívá se pro modelování algoritmů v teorii vyčíslitelnosti. Jedná se o jednoduchý stavový automat s konečným počtem stavů, o jednoduché zařízení, kterým prochází nekonečná páska rozdělená na políčka, přičemž stroj vždy čte pouze jediné. Přečte na ní zapsaný symbol a podle stavu, v němž se nachází, udělá nějakou akci – symbol vymaže, převine pásku doprava či doleva, zapíše jiný symbol, případně změní svůj stav. Pomocí takto jednoduchého automatu lze generovat jakékoliv představitelné konečné i nekonečné posloupnosti čísel či symbolů.

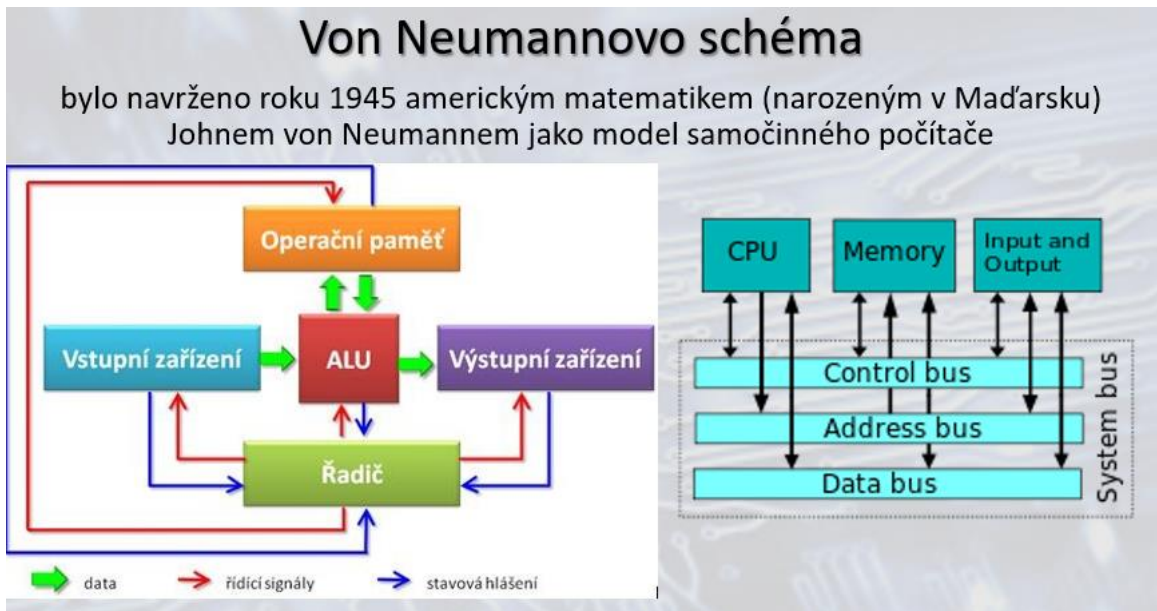
V roce 1945 navrhl a popsal John von Neumann první počítač s uloženým programem, který byl později postaven pod názvem EDVAC (Electronic Discrete Variable Automatic Computer). Tím dal základ dnes běžně používanému pojmu "počítač s von Neumannovo architekturou". Na myšlenku, že počítače by mohly být řízeny programem, přišel více než sto let před Johnem von Neumannem Angličan Charles Babbage. Konkrétní způsob realizace této myšlenky se ovšem vyvíjel postupně. První představa byla taková, že počítač má svůj program zaznamenaný na nějakém vnějším záznamovém médiu (děrné pásce či děrných štítcích), postupně jej načítá a ihned provádí. To ale přináší komplikace – například provedení skoku znamená přemotání děrné pásky na požadované místo (analogicky pro štítky), což trvá nezanedbatelnou dobu. Zdlouhavé a problematické je pak i volání podprogramů, eventuální rekurze při takovémto volání apod. Hlavní vinu na tom má především sekvenční charakter paměti, používané pro uchovávání programu, a s tím souvisejí postupné načítání té části programu, která je zapotřebí. Proto John von Neumann navrhl uchovávat program takovým způsobem, aby byl celý trvale k dispozici (tj. aby se nemusel postupně načítat), a aby bylo možné v něm libovolně (a hlavně "okamžitě") „skákat“, navrhl tedy uchovávání celého programu v paměti počítače, a navíc v takové paměti, která nemá sekvenční charakter, ale chová se jako paměť s tzv. přímým přístupem.

Velmi zajímavým momentem bylo také zpochybnění dosud zastávané představy o tom, že program a data jsou principiálně odlišné objekty, které v zásadě není možné jakkoli míchat (a tudíž ani uchovávat společně). Snad nejvýznamnější myšlenkou celé von Neumannovy architektury je právě zásada, že program i data jsou ve své podstatě jedno a totéž (a to posloupnost bitů), a o jejich povaze rozhoduje pouze způsob, jakým jsou interpretovány. Tato zásada pak von Neumannovi umožnila navrhnout, aby počítač měl jedinou paměť (tzv. operační paměť), ve které by se společně uchovávaly jak data, tak i celé programy. Na základě tohoto teoretického modelu navrhl Von Neumann základní počítačové komponenty a pravidla komunikace mezi nimi. Základem počítače je ALU (aritmeticko logická jednotka), řadič, který určuje kdy a v jakém pořadí se zpracovávají data a instrukce, paměť, ve které se data a instrukce uchovávají a vstupně/výstupní zařízení, pomocí kterých

systém komunikuje s okolím. Jednotlivé komponenty komunikují mezi sebou prostřednictvím sběrnice, která se dělí podle typu přenášených informací. Tyto moduly tvoří základ architektury současných počítačů.

Struktura počítače je nezávislá na typu řešení úlohy a je řízena obsahem paměti, instrukce a operandy jsou uloženy v téže paměti, paměť je rozdělena do buněk stejné velikosti, jejichž pořadová čísla se používají jako adresy, program je tvořen posloupností elementárních příkazů (instrukcí), v nichž zpravidla není obsažena hodnota operandu (uvádí se pouze jeho adresa), takže | program se při změně dat nemění; instrukce se provádějí jednotlivě v pořadí, v němž jsou zapsány do paměti, změna provádění instrukcí se vyvolá instrukcí podmíněného a nepodmíněného skoku, pro reprezentaci instrukcí a čísel (operandů, výsledků, adres apod.) se používají dvojkové signály a dvojková číselná soustava.

Architektura počítačů (automatů) je tedy dána Von Neumannovou architekturou, dále byla modifikována jako Harvardská architektura, popřípadě Modifikovaná Harvardská architektura.



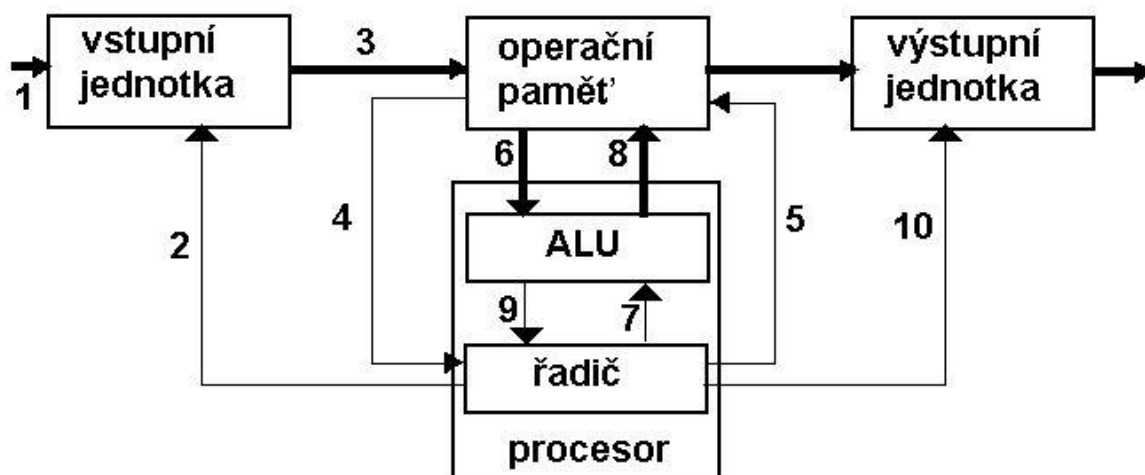
Obrázek 3: Von Neumannovo schéma

- Operační paměť: slouží k uchování zpracovávaného programu, zpracovávaných dat a výsledků výpočtu
- ALU - Arithmetic-logic Unit (aritmeticko logická jednotka): jednotka provádějící veškeré aritmetické výpočty a logické operace. Obsahuje sčítačky, násobičky (pro aritmetické výpočty) a komparátory (pro porovnávání)
- Řadič: řídicí jednotka, která řídí činnost všech částí počítače. Toto řízení je prováděno pomocí řídicích signálů, které jsou zasílány jednotlivým modulům. Reakce na řídicí signály, stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí stavových hlášení
- Vstupní zařízení: zařízení určená pro vstup programu a dat.

- Výstupní zařízení: zařízení určená pro výstup výsledků, které program zpracoval

Princip činnosti počítače podle von Neumannova schématu:

- Do operační paměti se pomocí vstupních zařízení přes ALU umístí program, který bude provádět výpočet.
- Stejným způsobem se do operační paměti umístí data, která bude program zpracovávat
- Proběhne vlastní výpočet, jehož jednotlivé kroky provádí ALU. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Mezivýsledky výpočtu jsou ukládány do operační paměti.
- Po skončení výpočtu jsou výsledky poslány přes ALU na výstupní zařízení.



Chceme-li si tedy popsat princip práce, pak má následující strukturu, vlastní práce počítače podle tohoto principu pak probíhá následovně:

1. Úloha se zadá vstupní jednotce v podobě dat a programu.
2. Řadič iniciuje vstupní jednotku, aby úlohu přijala.
3. Vstupní jednotka převede podle vnitřních pravidel systému zadání do řady nul a jedniček, data a program převede do buněk operační paměti (které si lze představit jako adresovatelné přihrádky). Program je řada instrukcí popisujících postup při řešení úlohy.
4. Řadič převezme z operační paměti první instrukci s pokynem, co má udělat.
5. Řadič vydá příkaz operační paměti, která data z buněk na určených adresách má vydat aritmetické jednotce.
6. Paměť předá data z uvedených adres do registru aritmetické jednotky.
7. Řadič přikáže aritmetické jednotce, aby provedla s daty, která naběhla do registru, operaci podle instrukce č.1 (např. aby sečetla, logicky porovnála apod.).
8. Výsledek výpočtu, či logického porovnání se uloží do operační paměti.
9. Řadič je informován o uložení předchozí operace, což je podmínkou k zahájení následující operace, ke které si krokem 4 vyžádá v programu následující instrukci, a tím se začne opakovat cyklus kroků 4 až 8 tak dlouho, až se dospěje k vyřešení úlohy.

10. Řadič zapojí výstupní jednotku, která předá výsledek úlohy. Přitom se předpokládá, že adresa, kde je výsledek úlohy v operační paměti uložen, byla přikázána krokem 5.

Principy zavedené von Neumannem znamenaly změnu k přístupu zpracování dat ale taky například ke tvorbě programů, které si zpravidla ani neuvědomíme.

Například programovací jazyky ve fázi překladu programu přistupují k programu jako k datům, ale tato samá data mohou být záhy spuštěna jako program (tj. mohou být interpretována jako program). Pokud bychom trvali na tom, že data a program jsou něco v principu jiného, musel by překladač a sestavující program vygenerovat přeložený program do paměti pro data, a odsud by pak musel být tento program explicitně přenesen do paměti pro programy, odkud by teprve mohl být spuštěn.

Další zásadní myšlenkou celé von Neumannovy koncepce je sekvenční způsob fungování počítače. Jednotlivé strojové instrukce, tvořící program, se provádějí postupně tak, jak jsou umístěny za sebou (až na případné skoky, volání a návraty z podprogramů). Úkol, je rozdělen na posloupnost kroků a tyto se provádí postupně – nikoli souběžně. Pro člověka, který takovýto program píše, je jeho dekompozice do posloupnosti sekvenčně prováděných kroků mnohem přirozenější a jednodušší než rozklad do takových činností, které se mohou provádět souběžně, mohou se pro-línat v čase, a je nutné dbát na jejich správnou návaznost.

Čistě sekvenční charakter je ovšem to, co je von Neumannově architektuře snad nejvíce vyčítá-no. Důvodem je skutečnost, že sekvenční charakter výpočtů neumožňuje zrychlení, které by bylo možné, kdyby činnosti bylo možné provádět souběžně (paralelně). Ve dnešní době vidíme návrat k paralelnímu zpracovávání úloh především na základě více procesorových systémů. Objevily se i různé alternativní architektury (tzv. ne-von Neumannovské), které s paralelní prací počítají. Prosadily se však jen v dosti speciálních aplikacích a ukazují, že není těžké zkonstruovat paralelní počítač ale jeho paralelismus využít. Problém je v samotném člověku. Lidskému myšlení je bližší sekvenční postup než velká míra paralelismu.

Velmi důležitou myšlenkou, která se prosadila v rámci von Neumannovy architektura, je zásada, že počítač by se neměl přizpůsobovat potřebám konkrétní aplikace svou vnitřní strukturou, ale pouze programem. Jinými slovy, vnitřní struktura počítače by se neměla měnit a měla by být maximálně univerzální. Naopak programy by měly být modifikovatelné.

Tato zásada má tu výhodu, že veškerou složitost, spojenou s řešením určitého zadání, převádí pouze na složitost návrhu programu (a nikoli na složitost návrhu vnitřní struktury počítače), řeší ji tedy jen v softwaru, a nikoli v hardwaru. I v oblasti návrhu hardwaru samozřejmě existují různé metody a nástroje, umožňující vyrovnat se s velkou složitostí řešeného úkolu, ale na poli softwaru jsou analogické metody a nástroje mnohem výkonnější, lacinější i rychlejší, a v důsledku toho umožňují zvládnout i složitější a náročnější úkoly.

Jinou otázkou je ovšem efektivnost tohoto řešení. Počítač von Neumannovy koncepce je zcela záměrně navržen tak, aby na něm šlo naprogramovat prakticky cokoli. Na druhé straně hardware, který je navržený potřebám nějakého úzce vymezeného okruhu aplikací, bude zcela zákonitě efektivnější. Příkladem může být skutečnost, že úspěšné šachové počítače nejsou univerzálními počítači, ale počítači úzce specializovanými.

Von Neumannovo schéma má základní nedostatek, v paměti se provádějí operace i ukládají data, což může vést k malé produktivitě využití procesoru a paměti. Proto bylo navrženo schéma, ve kterém se oddělila část paměti pro data a pro programy.

Počítačová architektura, která fyzicky odděluje paměť programu a dat a jejich spojovací obvody se nazývá Harvardská architektura. Název pochází z počítače Harvard Mark I, který byl postaven na této architektuře.



Obrázek 4: Harvardská architektura

- rozdělen blok paměti na dva bloky – zvlášť paměť pro programy a zvlášť paměť pro data.
- koncepcí dovoluje používat pro paměť programu například paměti typu ROM (Read Only Memory) a umožňuje v podstatě zdvojnásobení velikosti paměti oproti von Neumannově architektuře při stejné veliké adresové sběrnici

U harvardské architektury není potřeba mít paměť stejných parametrů a vlastností pro data a pro program. Paměti mohou být naprosto odlišné, mohou mít různou délku slova, časování, technologii a způsob adresování.

V některých systémech se pro paměť programu používá typ paměti ROM (read only memory), přičemž paměť dat vyžaduje typ paměti RWM (Read-Write Memory). Dvojí paměť umožňuje paralelní přístup k oběma paměťem, což zvyšuje rychlost zpracování. Umístění programu v paměti ROM může významně přispět k bezpečnosti systému (program nelze modifikovat). Pro velmi rychlé zpracování velkého toku dat se čím dál více používají počítače s paralelním zpracováním, založené na Harvardské architektuře. U této architektury, díky odděleným propojovacím obvodům a separátní paměti programu a dat, může procesor zároveň číst/zapisovat z/do paměti programu a paměti dat najednou.

Harvardská architektura byla dále upravena jako koncept nazvaný Modifikovaná Harvardská architektura, který kombinuje funkce Harvardské architektury a von Neumannovy architektury. V tomto konceptu je sice oddělena paměť dat a paměť programu, využívají se však společná data a adresová sběrnice. Rychlé moderní procesory spojují obě architektury. Uvnitř procesoru je použita Harvardská architektura, kde se paměť Cache (velmi rychlá,

statická paměť typu RAM, umístěná blízko jádra procesoru) dělí na paměť instrukcí a paměť pro data. Celý procesor se „z venku“ chová jako procesor s architekturou Von Neumanna, protože načítá data i program z hlavní paměti na jednou.

Architektura tak umožňuje snadný přenos dat mezi rozdělenými paměťmi. Umožňuje zacházet s instrukcemi jako s daty, tj. přenést část kódu do paměti dat. Příkladem použití této architektury je rodina jednočipových procesorů, používaných zejména:

- u signálových procesorů,
- mobilních telefonů
- řídicích jednotek automobilů, aut. praček, myček atd.
- profesionálních audio/video stanice

Jednou z nejvíce kritizovaných vlastností von Neumannovy architektury je její čistě sekvencí charakter, který nepočítá s tím, že by se provádělo více činností souběžně. To souvisí s představou, že procesor (resp. řadič jako část procesoru) bude bezprostředně řídit skutečně všechny prováděné činnosti, včetně vstupně výstupních operací.

Toto omezení však začalo již velmi brzy vadit, protože rychlost procesoru i většiny ostatních částí počítače prakticky od začátku řádově převyšovala rychlost všech vstupně/výstupní zařízení. Pokud by ale procesor (resp. řadič) skutečně přímo řídil takováto vstupně/výstupní zařízení, musel by se nutně přizpůsobovat jejich rychlosti. Efekt by pak byl takový, že drtivou většinu času by procesor musel trávit čekáním na připravenost těchto zařízení.

Přítom bezprostřední řízení jednotlivých periférií je vesměs natolik jednoduché, že je lze svěřit nepřilíš složitému jednoúčelovému obvodu. Proto se začaly k jednotlivým periferním zařízením přidávat samostatné řadiče (nejčastěji označované jako tzv. řídicí jednotky), které převzaly úlohu bezprostředního řízení těchto zařízení. Procesor těmto řídicím jednotkám vždy zašle explicitní pokyn k zahájení určité vstupně/výstupní operace či její dílčí části, a řídicí jednotka pak určitou dobu pracuje nezávisle na procesoru, dokud zadaný pokyn nedokončí. Protože tato řídicí jednotka pracuje nezávisle na rychlosti procesoru, hovoříme o asynchronním průběhu vstupně/výstupních operací

Při použití řídicí jednotky u vstupně/výstupních zařízení musí být vyřešeno, jak se má řídicí jednotka zachovat v okamžiku, kdy dokončí dříve zadaný úkol, a chce oznámit, že je schopna převzít nový úkol a jak má řešit situaci, kdy dojde k nějaké nestandardní situaci, a řídicí jednotka nemůže pokračovat ve své práci. Rovněž může nastat případ, že potřebuje spolupráci procesoru k tomu, aby přenesla nějaká data z/do operační paměti. Pokud bychom tento problém vyřešili tak, že se procesor sám průběžně dotazuje řídicí jednotky na její stav (iniciativa je na straně procesoru), tak popíráme účel samostatné práce řídicí jednotky. Procesor sice již nemusí bezprostředně řídit průběh vstupně/výstupní operace, ale místo toho se musí neustále ptát řídicí jednotky, zda už je hotova.

Aby řídicí jednotka měla možnost komunikace s procesorem, aby se mohla přihlásit k procesoru a vynutit si jeho pozornost – přinutit jej přerušit provádění právě probíhajícího programu, a místo toho vyvolat jiný program (tzv. obslužný program), který zjistí příčinu, zajistí vše potřebné, a pak zase vrátí řízení zpět původně přerušnému programu, byl zaveden princip přerušení.

K ZAPAMATOVÁNÍ



IRQ je zkratka z anglického Interrupt ReQuest (česky požadavek na přerušení). Označuje signál, kterým požádá zařízení (např. klávesnice, časovač atd.) procesor o věnování pozornosti, tedy požádá o přerušení probíhajícího procesu za účelem provedení důležitější akce.

Mechanismus přerušení spolu s asynchronními vstupně/výstupními operacemi odlehčil procesoru a umožnil mu věnovat se provádění jiných programů. Stále však procesoru mnoho času zabírá například veškerý přesun dat mezi pamětí a vstupně/výstupním zařízením, protože procesor má jako jediný přístup k operační paměti, přestože se tak děje na základě požadavků řídicí jednotky periferie, v rámci obslužných programů přerušení.

Pokud byly obsluhované periferie relativně pomalé, a přenášených dat poměrně málo – jako například při tisku na tiskárně – procesor to nijak zvlášť nezatěžovalo. Ovšem v případě rychlých periferií, které vyžadují přenosy relativně velkých objemů dat (jako například disky) došlo k situaci kdy procesor byl velmi často přerušován za účelem přenosu dat z/do periferie a míra jeho schopnosti vykonávat vedle toho ještě i jiný program byla tudíž značně omezená, respektive procesor nestačí přenášet data z/do periferie potřebnou rychlostí.

Ve snaze ještě více odlehčit procesoru se přišlo na řešení zabudovat do počítače specializovaný subsystém, tzv. **kanál**, určený výhradně k zajišťování přenosů dat mezi operační pamětí a periferními zařízeními (resp. jejich řídicími jednotkami). Periferie se pak se svými žádostmi o jednotlivé přenosy dat nemusí obracet na procesor (prostřednictvím přerušení), ale místo toho se obrací přímo na paměť prostřednictvím kanálu.

Aby kanál mohl plnit svou funkci a přenášet data mezi pamětí a periferními zařízeními, musí se dělit s procesorem o právo přístupu k operační paměti, čímž procesor ztrácí výhradní komunikaci s pamětí v celém počítači. Kanál ke své funkci vyžaduje přímý přístup do paměti (Direct Memory Access **DMA**).

Kanál tedy musí být relativně inteligentním zařízením, které dokáže korektně soutěžit s procesorem o získání přístupu k paměti. V zájmu dosažení maximální autonomie kanálů jsou tyto vybavovány vlastním programem (tzv. kanálovým programem), na jehož základě jim mohly být předepsány rozsáhlejší vstupně/výstupní operace, sekvence takovýchto operací apod. Na procesor se pak kanál obrací jen v okamžiku, kdy dokončil provádění celého kanálového programu, nebo v okamžiku výskytu nějaké nestandardní situace (chyby a pod.).

První kanál ve výše naznačeném smyslu byl použit v roce 1956 v počítačích UNIVAC. Koncepce kanálů pak dlouhou dobu dominovala především v oblasti tzv. střediskových počítačů (mainframes). U minipočítačů se původně schopností přímého přístupu do paměti vybavily přímo řídicí jednotky jednotlivých periferních zařízení. U současné mikroprocesorové techniky se začaly používat samostatné speciální obvody, vybavené schopností přímého přístupu do paměti, ale bez možnosti používání kanálových programů (jsou řízeny údaji, zapisovanými přímo do jejich interních registrů). Tyto tzv. DMA řadiče pak mohou zajišťovat potřebné přenosy dat z/do paměti pro více jednotlivých periférií.



K ZAPAMATOVÁNÍ

DMA (Direct Memory Access) - přímý přístup do paměti je schopnost sběrnice posílat data ze zařízení (CD-ROM, Pevný disk apod.) bez účasti procesoru. K funkci je zapotřebí řadič DMA, který zajišťuje přenos po sběrnici přímo mezi vstupně-výstupními zařízeními a pamětí. Toto se děje bez ukládání dat v pomocných registrech a bez účasti procesoru.

Základní odlišnosti dnešních počítačů od von Neumannova schématu:

Podle von Neumannova schématu počítač pracuje vždy nad jedním programem. Toto vede k velmi špatnému využití strojového času. Ve dnešní době počítač zpracovává paralelně více programů zároveň - tzv. **multitasking** a může disponovat i více než jedním procesorem – tzv. **multiprocessing**.

Požadavky uživatelů rostly rychleji než technologické možnosti výrobců. Nejmarkantnější byl rozpor mezi požadavky uživatelů a aktuálními možnostmi výrobních technologií v oblasti paměti. To bylo v principu možné řešit přidáváním dalších a dalších paměťových modulů (v rámci konstrukčních omezení).

Tento problém je řešen tak, že uživatelům se předstírá, že mají k dispozici větší objem operační paměti, zatímco skutečný objem této paměti větší není.

Podstata řešení spočívá v oddělení toho, co uživatelé a jejich programy "vnímají" (tj. toho, co si uvědomují, resp. s čím si myslí, že pracují), od toho, s čím pracují doopravdy.

Vedle skutečné a reálně existující paměti se tedy zavedla ještě další paměť, která je pouze iluzí, předkládanou uživatelům a jejich programům (a která reálně neexistuje, a je pouze předstírána – proto se jí také říká virtuální). **Virtuální paměť** může být výrazně větší než fyzická paměť. Uživatel si myslí, že pracuje s velmi velkou pamětí, zatímco ve skutečnosti pracuje s pamětí mnohem menší.

Princip spočívá v tom, že do skutečně existující paměti se průběžně přesouvají právě jen části programů a dat, se kterými program v danou chvíli pracuje, zatímco zbývající části mohou být uloženy například na disku. Rozsah virtuální paměti (se kterou si uživatel myslí, že pracuje), je pak dán kapacitními možnostmi disku, a nikoli skutečným objemem reálně existující paměti.

Vše musí být zařízeno tak, aby veškeré přesouvání dat a programů mezi operační pamětí a diskem (v současnosti SSD disky nebo paměti Optane) bylo pro uživatelské programy plně transparentní (tj. neviditelné) - což ale není principiální problém zajistit.

Existuje více různých technik pro realizaci virtuální paměti: nejpoužívanější z nich je tzv. **stránkování (paging)**, které mezi operační pamětí a diskem přesouvá celé tzv. stránky (vždy stejně velké, např. 1 kB, 2 kB, 4 kB apod.). Další používanou technikou pro realizaci virtuální paměti je tzv. **segmentace (segmentation)**.

Pro kompenzaci rozdílných rychlostí, např. procesoru, paměti, periférií se využívá režimu, kdy obsah té části operační paměti, se kterou právě probíhající program pracuje, se přesune do tzv. **vyrovnávací paměti (cache)**, která je umístěna mezi procesorem a operační pamětí (respektive mezi jinými, různě rychlými zařízeními), je výrazně menší než operační paměť, ale výrazně rychlejší. Jakmile probíhající program vysloví požadavek na přístup do operační paměti, budou se příslušná data hledat nejprve v této malé a rychlé vyrovnávací paměti, a teprve v případě neúspěchu se bude "sahat" až do skutečné operační paměti.

Vyrovnávací paměti se začaly znovu využívat, když rychlost procesorů vzrostla natolik, že předběhla možnosti polovodičových pamětí. V současné době se implementují vyrovnávací paměti, i do samotných procesorů (jako tzv. interní cache paměti).

Spuštěný a zpracovávaný počítačový program který je umístěn v operační paměti počítače v podobě strojových instrukcí, které vykonává procesor v určitém sledu se nazývá **proces**. Ten obsahuje nejen kód vykonávaného programu, ale i dynamicky měnící se data, které se zpracovávají. Jeden program může v počítači běžet jako více procesů s různými daty (např. vícekrát spuštěný webový prohlížeč zobrazující různé stránky). Správu procesů vykonává operační systém, který zajišťuje jejich oddělený běh, přiděluje jim systémové prostředky počítače a umožňuje uživateli procesy spravovat (spouštět, ukončovat atp.).

Vlákno (vlákno řízení, thread)

Vláknem se označuje odlehčený proces, který snižuje režie operačního systému při střídání procesů na procesoru. Proces se tedy může dělit na vlákna, která pak pomocí sdílené paměti usnadňují komunikaci – vlákno vlastní paměť nemá, sdílí ji společně s ostatními vlákny běžícími ve stejném procesu. Vlákna tak mohou přistupovat ke stejným datům současně. To ovšem může přinést riziko tzv. souběhu (race condition). Záleží také na tom, jestli operační systém vlákna podporuje nebo ne. Pokud nepodporuje, každý proces má pouze jedno vlákno.

Výhody: program, kde se počítá určitý výpočet, a přitom se výsledek zobrazuje v podobě grafu. Výpočet je však náročný a trvá systému několik vteřin, než jej vypočte. Pokud by byla aplikace pouze jednoprosesová a tento proces měl jen jedno vlákno, ve kterém by se vše vykonávalo popořádku (posloupnost instrukcí), pak by se program na několik vteřin "seknul" a až po dokončení výpočtu by se zase dostalo na vykreslení. Přitom vlákno nemusí ani odchyťávat stisknuté klávesy a kliknutí myši. Pokud jedno vlákno bude počítat výsledek a druhé bude jenom vykreslovat graf, pak se může graf vykreslovat třeba pomalu (vlákna se střídají, takže se chvíli počítá a chvíli vykresluje), ale program stále poběží viditelně i pro uživatele (i v prvním případě běžel, ale nic viditelného se s ním nedělo).

Z hlediska spravování vláken OS můžeme vlákna rozdělit na: - vlákna na uživatelské úrovni (ULT) - správu vláken provádí tzv. vláknová knihovna na úrovni aplikačního procesu, jádro operačního systému o jejich existenci neví, - vlákna na úrovni jádra (KLT) - správa vláken je řízena OS



K ZAPAMATOVÁNÍ

Cache je označení pro vyrovnávací paměť používanou ve výpočetní technice. Je zařazena mezi dva subsystémy s různou rychlostí a vyrovnává tak rychlost přístupu k informacím. Cache může být vytvořena programově vymezením určité části operační paměti pro potřeby vyrovnávací paměti (např. disková cache v operačním systému), nebo hardwarově paměťovými obvody (např. pro potřeby procesoru).

1.5 Instrukční sada, programování

Každý počítač má vždy definován určitou množinu strojových instrukcí, které je schopen provádět, tzv. instrukční soubor. Programy, které je takovýto počítač schopen přímo vykonávat, pak musí být sestaveny právě a pouze z těchto strojových instrukcí. Přitom každý počítač má obecně jiný instrukční soubor, jiné způsoby adresování atd.

Programátor, který svůj program, sestavuje přímo z jednotlivých strojových instrukcí zapisovaných v tzv. jazyku symbolických adres neboli assembleru, samozřejmě musí tyto strojové instrukce znát, a s nimi musí dosti podrobně znát i další více či méně detaily, jako

skladbu registrů, způsoby adresování, významy příznaků apod. Programátor, který programuje na úrovni strojového kódu, si tedy musí být vědom, že pracuje na určitém konkrétním typu počítače, a musí respektovat jeho specifika.

Tento problém se odstranil pokusem vytvořit takový programovací jazyk, který bude nezávislý na konkrétním instrukčním souboru, a nebude od programátora očekávat znalost konkrétního počítače, pro který bude program psát. Z tohoto pohledu tedy strojově nezávislý jazyk může být jazykem vyšší úrovně než strojově závislý jazyk. Příkazy strojově nezávislého jazyka nemohou odpovídat jednotlivým strojovým instrukcím konkrétních počítačů (jako je tomu u assembleru jako strojově závislého jazyka). Strojově nezávislý jazyk vyžaduje buď existenci **kompilátoru**, který v něm zapsané programy přeloží do proveditelného tvaru, tedy do tvaru programu, sestaveného ze strojových instrukcí konkrétního počítače nebo bude vyžadovat tzv. **interpret**, který bude postupně brát jednotlivé příkazy strojově nezávislého jazyka, a bude průběžně zajišťovat jejich provádění (bude interpretovat, co jednotlivé příkazy požadují, a příslušné činnosti provede sám).

Strojově-nezávislý programovací jazyk je obecně nazývaný programovací jazyk a převádí tvar programu „srozumitelný pro člověka“ do tvaru, který je „srozumitelný pro počítač“, přesněji řečeno, pro příslušnou instrukční sadu.

Mezi první programovací jazyky patřil např. FORTRAN (od: FORmula TRANslation), LIPS (LISt Processor) pro umělou inteligenci, v roce 1960 vznikl jazyk COBOL (COmmon Business Oriented Language) a v témže roce univerzální programovací jazyk ALGOL (ALgorithmic Language, z roku 1961 pochází první programovací jazyk, určený pro počítačové simulace (jazyk GPSS, neboli General Purpose Systems Simulator). Jazyk BASIC (Beginner's All-purpose Symbolic Instruction Code) pochází z roku 1964, jazyk Pascal pochází z roku 1971. V roce 1974 pak vylepšením nepřítel známého programovacího jazyka B vznikl jazyk C, který je po různých modifikacích používán dodnes.

Jak bylo zmíněno, program převádí požadované úkony do instrukcí, které umí procesor interpretovat. Podle zpracování instrukcí lze dále rozlišovat čtyři základní principiální architektury:

- CISC (Complex Instruction Set Computer),
- RISC (Reduced Instruction Set Computer),
- VLIW (Very Long Instruction Word) a
- MISC (Minimum Instruction Set Computer).

V praxi se vychází zpravidla z následujících architektur:

CISC – Complex Instruction Set Computer

Procesor s velkým množstvím strojových instrukcí (řádově stovky) a relativně malým počtem registrů (jejich počet obvykle nepřesahuje 30). Procesory CISC mají různě dlouhé

strojové instrukce, jejichž vykonání trvá různě dlouhou dobu. Paradoxně se tak může stát, že operace provedená složenou instrukcí (například násobení) může být nahrazena sledem jednodušších strojových instrukcí (sčítání a bitové posuvy), které mohou být ve výsledku vykonány rychleji, než hardwarově implementovaná složená varianta.

Označení CISC bylo zavedeno jako protiklad až poté, co se prosadily procesory RISC, které mají instrukční sadu naopak maximálně redukovanou (pouze jednoduché operace, tj. žádné složené, jsou stejně dlouhé a jejich vykonání trvá stejnou dobu).

Obvyklou chybou je domněnka, že procesory CISC mají více strojových instrukcí než procesory RISC. Ve skutečnosti nejde o absolutní počet, ale o počet různých druhů operací, které procesor sám přímo umí vykonat na hardwarové úrovni (tj. již z výroby). Procesor CISC tak může například paradoxně obsahovat pouze jednu strojovou instrukci pro danou operaci (např. logická operace), zatímco procesor RISC může tuto operaci obsahovat jako několik strojových instrukcí, které stejnou operaci umí provést nad různými registry.

Na celé vývojové řadě je možno pozorovat technický i technologický vývoj. Patrný je postupný přerod typického procesoru CISC v procesory, které jsou konstruovány jako procesory RISC a přitom si zachovávají úplnou zpětnou kompatibilitu.

Standardně jsou procesory CISC typickým zástupcem používaným pro počítače PC, dominantními výrobci jsou Intel a AMD.

RISC – Reduced Instruction Set Computer

Procesor s redukovanou instrukční sadou, jejichž návrh je zaměřen jednoduchou, vysoce optimalizovanou sadou strojových instrukcí, která využívá jen velmi malého množství nejčastěji užívaných, instrukcí. Téměř všechny instrukce potřebují stejnou dobu pro své vykonání, obvykle jen jeden cyklus pro své vykonání, na rozdíl od CISC, kde se doba vykonání instrukce liší podle vykonávané instrukce, počet instrukcí a způsobů adresování je malý, ale zůstává úplný, aby bylo možno provést vše (na rozdíl od CISC). Instrukce jsou vytvořeny pomocí obvodu, mají jednotný formát – délku i obsah, každý strojový cyklus znamená dokončení jedné instrukce, používá se zřetěžené zpracování instrukcí. Procesory RISC přenášejí složitost technologického řešení do programu (překladače).

Z RISC procesorů vycházejí **ARM** procesory. ARM je architektura procesorů vyvinutá v Británii firmou ARM Limited. Starší obchodní název architektury ARM je Advanced RISC Machine, původní název je Acorn RISC Machine. Tato architektura způsobila v několika směrech revoluci v informačních technologiích. ARM je používána díky své nízké spotřebě elektrické energie zejména v mobilních zařízeních (mobilní telefony, tablety). Globálně je ARM nejpočetněji zastoupenou architekturou mikroprocesorů.

PROUDOVÉ ZPRACOVÁNÍ (PIPELINING)

Ve snaze co možná nejvíce zvýšit výpočetní výkon svých procesorů, začali výrobci hledat skryté rezervy. Jednou z vlastností bylo, že provádění jednotlivých strojových instrukcí probíhá v určitých fázích: nejprve musí dojít k načtení instrukce z operační paměti, pak k jejímu dekódování, přípravě jejích operandů, pak může následovat výkonná fáze. Dále může ještě následovat fáze ukládání cílových operandů (výsledků).

Konkrétní počet fází přitom není příliš podstatný (navíc je na každém počítači různý). Podstatná je spíše skutečnost, že různé fáze obvykle provádí různé části procesoru. Jednotlivé fáze jedné a téže instrukce přitom na sebe sekvenčně navazují. To znamená, že v každém okamžiku se užitečné činnosti věnuje jen jedna v více částí procesoru, zatímco ty ostatní neúčelně zahálí.

Problém tedy byl, jak trvale vytižít všechny části procesoru. Provádět souběžně více různých fází jedné a téže instrukce není dost dobře možné, protože tyto na sebe příčinně navazují. Tvůrci počítačů ovšem přišli na řešení, nemohou-li současně prováděné fáze patřit jedné a téže instrukci, může se provádět více instrukcí.

Tato myšlenka znamená, že v procesoru se v každém okamžiku bude nacházet více strojových instrukcí v různém stádiu rozpracovanosti. Například, současně s výkonnou fází jedné instrukce se budou připravovat operandy následující instrukce, a současně s tím bude z paměti načítána ještě jedna další instrukce. Provádění jednotlivých instrukcí se tak bude v čase překrývat. Doba, potřebná k provedení jedné strojové instrukce, se tím sice vůbec nezkrátí, ale za jednotku času bude dokončeno několikanásobně víc strojových, podle toho, kolik instrukcí se bude najednou překrývat. Výsledný mechanismus se nazývá proudové zpracování (**pipelining**). Dnes je tento mechanismus v nějaké podobě implementován téměř v každém procesoru (resp. mikroprocesoru).

Samotná myšlenka překrývání instrukcí v procesoru (tj. proudového zpracování) je relativně starého data. Poprvé byl tento mechanismus použit již v 60. letech

ARCHITEKTURA x86

Dá se konstatovat, že v některých případech existují v názvosloví nesrovnalosti. V některých případech bývá jako architektura označován základní koncept (např. Harvardská architektura), v některých případech je spojována s instrukční sadou (RIC architektura), podle jiných klasifikací vychází z délky zpracovávané instrukce (32 bitová architektura, 64 bitová architektura apod.). Ve smyslu architektury se často setkáte s pojmem x86 nebo 64, x86-64 apod. Jedná se o označování používané pro počítače PC (personal computer), označující rodinu instrukčních sad pro procesory, označení x86 též definuje hardwarovou počítačovou platformu (architekturu). Architektura x86 bývá označovaná jako IBM PC, byla použita v osobních počítačích od firmy IBM, její typové označení vychází z názvu procesoru Intel 8086 (tehdejší označování procesorů bylo, zjednodušené řečeno, založeno na

procesorech Intel, základní řady byly, zjednodušeně, i8080, i8086, i80286, i80386, i80484, dále následovalo značení Pentium. Od řady 80386 došlo k odklonu AMD od této architektury). X86 je v současnosti chápána jako 32 bitová architektura, někdy se značí x86-32 nebo IA-32. Označení x64 nebo x86-64 se používá pro 64 bitovou architekturu (podle některých zdrojů se lze setkat i s označením AMD64, EM64T, IA-32E apod.). Tato architektura je zpětně kompatibilní s 32 a 16 bitovou архитектурou.

Instrukční sady x86 a ARM byly vyvinuty se stejným záměrem – vytvořit výkonnou a zároveň úspornou instrukční sadu, kterou lze použít v domácích počítačích. Z počátku jasně dominovala architektura x86 a ARM se tak soustředil především na jednoúčelové vestavěné systémy (například bankomaty, autopiloty nebo kalkulačky). Mezi vestavěné systémy však řadíme i smartphony, tablety nebo mobilní elektroniku, a to i přesto, že slouží rozhodně k více než jednomu účelu. Ještě před několika málo lety vypadala situace následovně – procesorové jednotky pro počítače dominoval Intel s instrukční sadou x86 (ať už v 32bitové nebo 64bitové verzi) a drtivou většinu chytrých mobilních přístrojů obstarávaly procesory s instrukční sadou od ARMu (v roce 2010 měl ARM ve prodáváných smartphonech více než 95% podíl). V současnosti ARM proniká do oblasti širokého spektra počítačů, od tabletů přes tablet PC, Surface (tablet PC firmy Microsoft) až po superpočítače a v segmentu superpočítačů začíná mít dominantní postavení. V současnosti (rok 2021) je nejrychlejším superpočítačem japonský Fugaku. debutoval v roce 2020 a stal se nejrychlejším superpočítačem na světě v seznamu TOP500 v červnu 2020 a stal se také první počítač založený na architektuře ARM. Od února 2021 je Fugaku nejrychlejší superpočítač na světě. Dá se konstatovat, že architektura ARM úplně převzala mobilní sféru a směřuje do serverů. Energetická efektivita je hlavním argumentem. V segmentu mobilních zařízení chce Apple budoucí Macy postavit na architektuře ARM místo x86-64 a také Microsoft vyvíjí Windows 10 pro ARM. I díky tomu, že se ARM soustředil na vestavěné systémy, je jeho instrukční sada výrazně jednodušší a také úspornější. Instrukční sada Intelu nabízí výrazně vyšší výkon.

1.5.1 DALŠÍ SOUVISEJÍCÍ POJMY

V souvislosti se způsobem zpracování se setkáme s pojmy vektorový a skalární procesor a architektura. Převážná část osmibitových a šestnáctibitových mikroprocesorů i první generace mikroprocesorů třicetidvoubitových byla postavena na takzvané skalární architektuře. Tímto termínem jsou označovány mikroprocesory, které v jednu taktu načtou maximálně jednu instrukci a provedou na základě jejího operačního kódu pouze jednu aritmetickou či logickou operaci s jednoduchou (skalární) hodnotou. Vektorový procesor je procesor navržený tak, aby dokázal vykonávat matematické operace nad celou množinou čísel v daném čase. Je opakem skalárního procesoru, který vykonává jednu operaci s jedním číslem v daném čase. Jejich koncept pochází z návrhu superpočítačů, kde našly i své první uplatnění a to v 80. a 90. letech. Dnes se využívají například jako CPU, GPU a APU. V současnosti je používána superskalární architektura, Superskalární architektura (superskala-

rita) je v informatice jedním ze způsobů zvyšování výkonu procesoru. Superskalarita umožňuje v jednom taktu zpracovat více strojových instrukcí zároveň, protože některé části procesoru jsou duplikovány, například matematický koprocessor (FPU) nebo aritmeticko-logická jednotka (ALU). Od vícejádrových procesorů se superskalarita liší tím, že je zvětšen počet pouze některých částí procesoru.

Výkon lze navyšovat rovněž paralelním způsobem zpracování jednotlivých instrukcí. Tzv. **Flynnovo roztřídění**, třídí paralelní systémy vzhledem k počtu toků dat a instrukcí. Používá jednoduché logiky značení a skládání názvu:

- SI - Single Instruction stream nebo-li jeden tok instrukcí (též systém s jedním programem)
- MI - Multiple Instrukcion stream nebo-li víceprogramový tok instrukcí (též systém provádějící několik programů)
- SD - Single Data stream neboli systém s jedním tokem dat
- MD - Multiple Data stream neboli systém s vícenásobným tokem dat.

Kombinací těchto zkratk pak dostáváme 4 základní skupiny paralelních systémů:

- SISD – počítač, který zpracovává data sériově podle jednoho programu (von Neumannova typu)
- SIMD – víceprocesorový počítač řízený společným programem. Všechny procesory sice provádí stejnou instrukci ve stejném čase, ale každý s jinými daty.
- MIMD – víceprocesorový počítač, ve kterém každý procesor má svůj program a zpracovává svá data.
- MISD – je značně nepravděpodobné, že by tato kategorie vznikla pro zvýšení výkonu, maximálně pro zvýšení spolehlivosti.

Z těchto 4 skupin se pro paralelní systémy vyčlenily 2 základní použitelné skupiny, a to SIMD a MIND, které se pro další roztřídění ještě modifikují:

- MSIMD – systém ve kterém pracuje několik navzájem nezávislých systémů provádějících každý svůj vlastní program
- SPMD - všechny procesory provádí stejný program nezávisle na sobě bez jakékoliv synchronizace.

1.5.2 KLASIFIKACE POČÍTAČŮ

Historie i současnost počítačů je bohatá a různorodá. Vznikala celá škála technických prostředků. Můžeme ale dosledovat jisté zákonitosti jak ve vývoji, tak v rozdělení a charakteristice technických prostředků. V zásadě můžeme počítače rozdělit podle několika kritérií. V této kapitole si budeme všimnout rozdělení počítačů především:

- podle generací,
- podle velikosti,
- podle konstrukce,
- podle standardů,
- podle výkonu,
- podle operačního systému,
- podle vzhledu,
- podle technologie atd.

Zcela jistě bychom našli i další způsoby rozdělení, tento výčet vám však stačí pro dostatečné seznámení s problematikou. Jak uvidíte, dělení je silně účelové a v některých případech se může jeden způsob klasifikace překrývat s jiným.

ROZDĚLENÍ PODLE GENERACÍ POČÍTAČŮ:

V literatuře se můžeme setkat s různým členěním. V podstatě jsou počítače rozděleny do čtyř generací, někde se uvádí i nultá generace, pátá generace resp. tři a půltá generace.

První rozdělení bylo na základě součástkové základny:

1. generace: elektronky,
2. generace: (samostatné) tranzistory,
3. generace: integrované obvody,
4. generace: mikroprocesory.

Postupně se zavedla nultá generace – elektromechanické prvky, zejména relé a členění bylo rozšířeno o mezistupně. Dva a půltá generace, tři a půltá generace atd.

Další rozdělení na generace vychází ze schopností řízení a ovládání procesu zpracování dat:

1. generace: programování ve strojovém kódu nebo v jednoduchých strojově závislých jazycích
2. generace: vyšší programovací jazyky (Algol 60, Fortran, Cobol)
3. generace: operační systémy a modernější programovací jazyky
4. generace: operační systémy a jazyky přizpůsobené uživatelům
5. generace: prvky umělé inteligence, ovládání v přirozeném jazyce

Ve dnešní době se počítače rozdělují do generací spíše v závislosti na konfiguraci, rychlosti a základním stavebním prvku.

KATEGORIE POČÍTAČŮ PODLE VELIKOSTI

Toto členění se částečně překrývá s členěním podle konstrukce.

Historické členění:

- Superpočítače
- Sálové počítače (mainframe)
- Minipočítače
- Mikropočítače

Soudobé členění

- Superpočítače
- Sálové počítače (datová skladiště)
- Servery (souborové, databázové, síťové – FTP, WWW aj.)
- Pracovní stanice
- Osobní počítače (desktop, notebook, tablet, PC mini PC apod.)
- Kapesní počítače (Personal digital assistant, Mobile digital assistant, tablet, phablet, smartphone apod.)

Dále počítače řídicí, jednoúčelové, vestavěné do jiných zařízení aj.

KATEGORIE PODLE KONSTRUKCE

Podle konstrukce v současné době dělíme počítače spíše na:

- Počítače otevřené architektury – PC
- Počítače uzavřené architektury – počítače pro řízení technologických procesů, „mikročipy“, speciální jednoúčelové konstrukce počítačů.

My se dále budeme zabývat počítači architektury PC. Ty můžeme podle konstrukce rozdělit na:

- Počítače do racku – servery,
- Počítače klasické koncepce PC,
- Přenosné počítače – notebooky,
- Tablet PC,
- Barbone systémy, Mini PC,
- Tenký klient.



Obrázek 5: Server umístěný do racku

1.5.3 SUPERPOČÍTAČE

Superpočítač je počítač s vysokou úrovní výkonu ve srovnání s klasickým počítačem. Výkon superpočítače se běžně měří v operacích s plovoucí desetinnou čárkou za sekundu (FLOPS) namísto milionů instrukcí za sekundu (MIPS). Od roku 2017 existují superpočítače, které lze provádět přes 10¹⁷ propadne (sto kvadrilion propadne, 100 nebo 100 petaflops PFLOPS). Od listopadu 2017 provozuje všech 500 nejrychlejších superpočítačů na světě operační systémy založené na Linuxu. Podle některých zdrojů (<https://cs.qaz.wiki/wiki/X86-64>) jsou superpočítače založené na procesorech CISC na vzestupu (obrázek 6). Podle tohoto zdroje, který sleduje nejvýkonnějších 500 superpočítačů.



Obrázek 6 Tianhe-2 – OS Linux, procesor Intel Xeon

Mezi základní architektury superpočítačů patří koncepty Cray (podle Seymoura Craye) vycházející z kompaktního designu a lokálního paralelismu. Nárůst výpočetního výkonu byl dále způsoben zavedením tzv. masivně paralelních systémů.

Rostoucí náklady na provoz superpočítačů byly hnacím faktorem v trendu sdružování zdrojů prostřednictvím distribuované infrastruktury superpočítačů. Nejprve se v USA objevila národní superpočítačová centra, následovala Německo a Japonsko. Evropská unie zahájila Partnerství pro moderní výpočetní techniku v Evropě (PRACE) s cílem vytvořit trvalou celoevropskou infrastrukturu superpočítačů se službami na podporu vědců v celé Evropské unii.

UPŘESNĚNÍ POJMŮ



Masivně paralelní. Ve výpočtech se jako masivně paralelní označuje použití velkého počtu procesorů (nebo samostatných počítačů) k provádění sady koordinovaných výpočtů paralelně (současně).

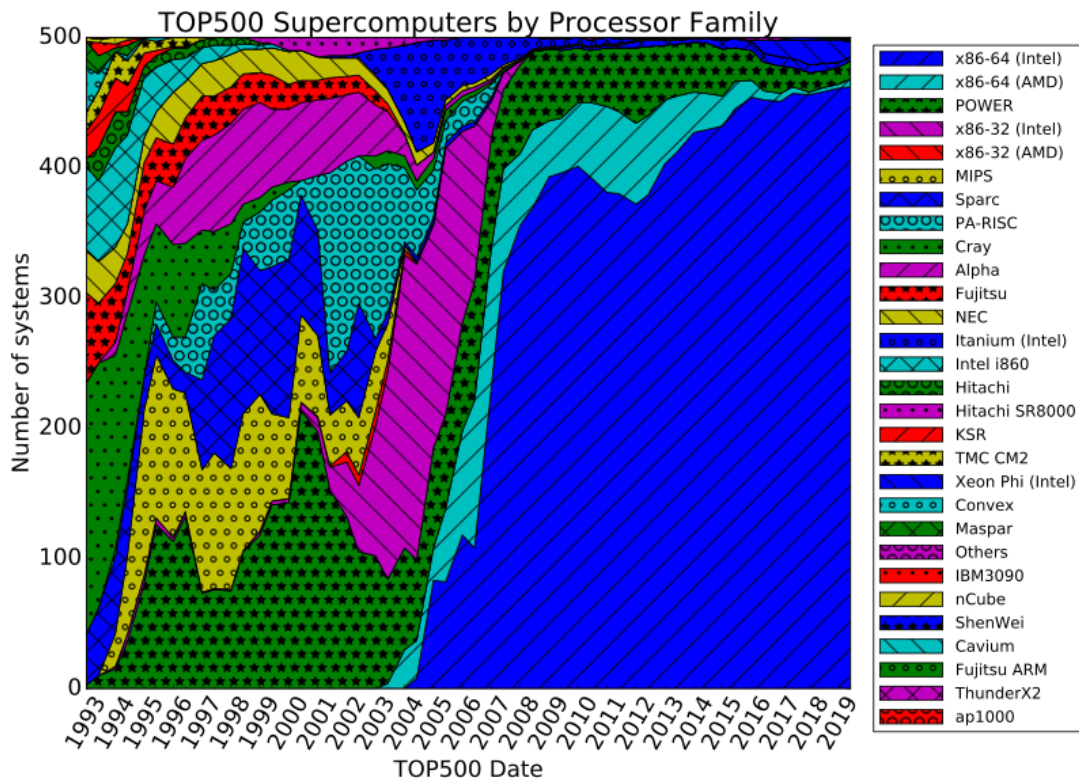
Masivně paralelní procesorová pole (MPPA). Je integrovaný elektronický prvek, který má masivně paralelní řadu stovek nebo tisíců CPU a RAM pamětí. Tyto procesory si navzájem předávají práci prostřednictvím rekonfigurovatelného propojení kanálů.

Distribuovaný systém. Je systém, jehož součásti jsou umístěny na různých počítačích v síti, které komunikují a koordinovat své kroky prostřednictvím předávání zpráv k sobě navzájem z jakéhokoliv systému. Komponenty vzájemně spolupracují za účelem dosažení společného cíle.

Grid computing. Je použití široce distribuovaných počítačových zdrojů k dosažení společného cíle.

Cluster computing. Je sada volně nebo pevně připojených počítačů, které spolupracují tak, že, v mnoha ohledech, které mohou být považovány za jeden systém. Na rozdíl od gridových počítačů mají počítačové klastry každý uzel nastavený na provádění stejného úkolu, který je řízen a naplánován softwarem.

Na obrázku 7 lze porovnat vývoj používání operačních systémů v segmentu superpočítačů.



Obrázek 7: Superpočítače podle architektur procesorů (převzato https://en.wikipedia.org/wiki/File:Processor_families_in_TOP500_supercomputers.svg)



SHRNUTÍ KAPITOLY

V této kapitole byly vysvětleny základní pojmy jako hardware a software, dále se studenti seznámili se základními koncepty počítačů, základními architekturami a s tím souvisejícími platformami a způsoby zpracování informací.

2 VÝVOJOVÉ TRENDY A MOBILNÍ IT

RYCHLÝ NÁHLED KAPITOLY



Současné trendy v oblasti hardware plynou z krátkodobých a dlouhodobých koncepcí předních výrobců hardware, potřeby trhu (požadavků zákazníků) a technologických inovací. Trendy lze shrnout do tří oblastí: vývoj a modifikace existujícího hardware, vývoj a inovace zařízení založených na existujících technologiích a vývoj nových technologií. Současné tendence nelze zúžit na vývoj hardware PC, protože technologie ICT se prolínají do různých platform. Zejména v době Internetu věcí (Internet of Things, IoT) a Internetu všeho (Internet of Everything, IoE), je standardem nejen komunikace zařízení mezi sebou, ale i jejich řízení a komunikace s člověkem prostřednictvím internetu, tedy komunikace věcí, lidí a procesů. Ta vyžaduje sofistikované nástroje a technologie, ve své podstatě to znamená distribuci procesorů, paměti a bezdrátových technologií do komunikujících zařízení. Předpokládané vývojové trendy v oblasti IT budou tedy směřovány především do roviny komunikací, mobility a virtuální reality, do roviny virtuálního, propojeného a komunikujícího světa. S oblastí komunikací vyvstává i otázka, jaké technologie budou ovlivňovat funkcionalitu Internetu. Stále více se hovoří o bezpečnosti a nezávislosti. S tím souvisí i nové komunikační infrastruktury a konkurence v oblasti poskytovatelů přenosových cest. V neposlední řadě se zvyšují požadavky na výpočetní výkon, rychlost zpracování a objem dat. V tomto smyslu dochází k vyčerpání možností stávajících technologií a zavedených principů. Dvoustavová jednotka informace (bit) se do budoucna ukazuje jako nedostatečný a možným trendem je náhrada bitu Qubitem, tedy kvantovým bitem. V této souvislosti lze hovořit o nástupu kvantových počítačů. Nelze pominout ani tendence vyvíjet technologie založené na jiné než křemíkové bázi, například uhlíkové (nebo dokonce organické), nanotechnologie, membránové počítače apod.

CÍLE KAPITOLY



Cílem kapitoly je prohloubit znalostí studentů v oblasti inovací hardware a vývojových trendů. Text studentům objasní souvislosti mezi stávajícími technologiemi, společenskými potřebami, vizemi a předpokládaným vývojem podpořeným dostupnými technologiemi. Studenti si doplní znalosti získané zejména v předmětu základy informačních technologií, ujasní si aktuální tržní nabídku v segmentu procesorů a získají přehled o rozdílech v parametrech a výkonu. Po nastudování doporučených zdrojů se budou schopni orientovat v současné nabídce a klasifikaci procesorů. Dále se budou schopni kvalifikovaně orientovat v alternativních technologiích.



KLÍČOVÁ SLOVA KAPITOLY

Hardware, procesory, WiFi, Qubit, kvantový počítač, organický počítač.



ČAS POTŘEBNÝ KE STUDIU

Vzhledem k obsahové náplni kapitoly, tj. novým trendům, tendencím a vizím, je probíraná problematika v tomto výukovém materiálu předkládána ve stručné formě, protože dochází k trvalému vývoji. Doporučuji proto jednotlivé okruhy dále doplňovat o aktuální informace, dostupné zejména prostřednictvím elektronických informačních zdrojů v prostředí Internetu a v prostředí Moodle. Předpokládaný čas pro zvládnutí základů kapitoly je 5 hodin, doporučený čas pro doplnění informací prostřednictvím elektronických zdrojů a Moodle je rovněž 5 hodin. Dále je vhodné věnovat 2 hodiny samostatným úkolům v prostředí Moodle.

2.1 Základní komponenty PC

Jak bylo uvedeno dříve, klasický počítač vychází z modulárního konceptu, kdy se počítač skládá z komponentů s přesně definovanými funkcemi. To umožňuje vyrábět jednotlivé komponenty různými výrobci, při zachování kompatibility jednotlivých částí.

Mezi základní části počítače patří především:

- Procesor (CPU)
- Grafická karta (Graphic Card)
- Operační paměť (Memory)
- Pevné a optické disky (HDD)
- Základní deska (mainboard či motherboard)
- Rozšiřující sloty

Dále počítač obsahuje periferie pro přímou komunikaci s uživatelem:

- vstupní – Klávesnice; Myš; Trackball; Tablet; Joystick; Gamepad; Scanner; Web kamera
- výstupní - Monitor; Tiskárna; Reproductor; Plotter apod.

Při výběru počítačových komponentů doporučuji vycházet z aktuálních nabídek prodejců. Na obrázku 4 je přehled komponentů nabízených společností Alza (www.alza.cz).



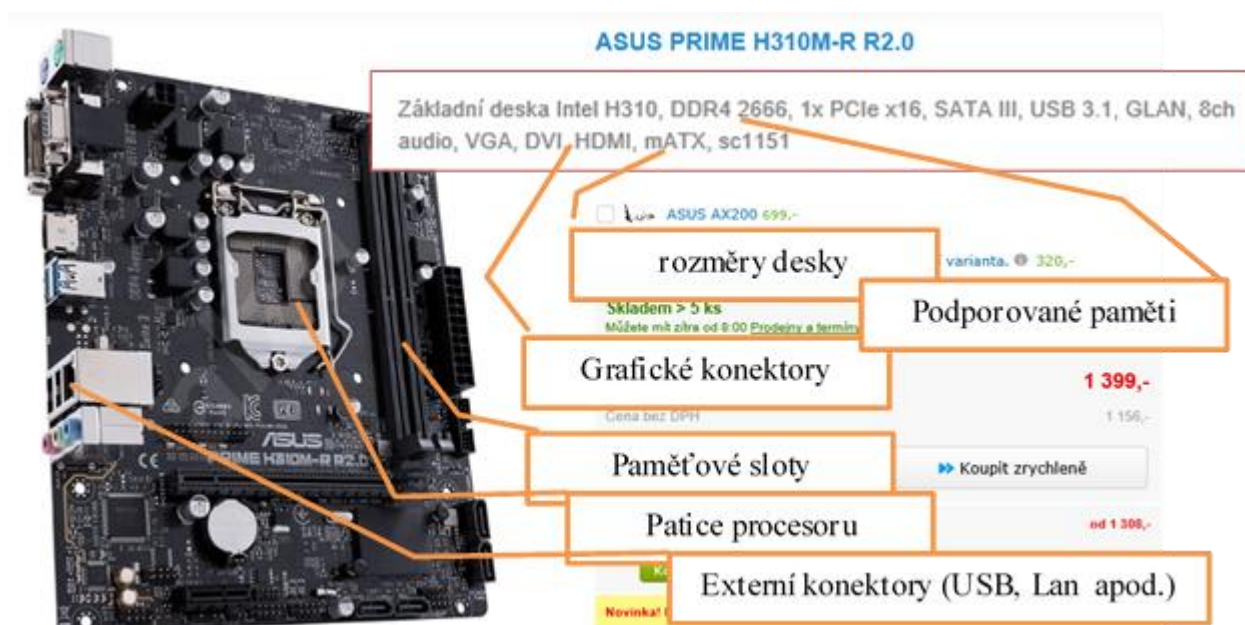
Obrázek 8: Počítačové komponenty

V těchto skriptech se nebudeme zabývat všemi komponenty, ukážeme si vlastnosti a parametry základních desek a procesorů.

2.1.1 ZÁKLADNÍ DESKA

Základní deska (v angličtině hojně označována jako motherboard či mainboard) je základním prvkem každého počítače. Hraje zásadní roli v propojení jednotlivých komponentů a má řadu parametrů, které jsou rozhodující pro správný chod celého systému. Určuje typ procesoru a dalších použitelných komponent. Její parametry jsou určovány čipovou sadou (čipsetem). Základní deska se nenachází pouze ve skříních počítačů, ale také u televízí, routerů, dvd přehrávačů apod. Jsou do ní zapojeny veškeré periférie, pro které funguje jako centrální propojení. Dnešní základní desky v sobě běžně obsahují prakticky celý počítač, kromě operační paměti a procesoru. Na základní desce bývá integrována zvuková karta, grafická karta, síťová karta a řadiče, vzhledem k modularitě lze však zpravidla tyto komponenty použít i jako rozšiřující komponenty prostřednictvím příslušných rozhraní (sběrnic). Například grafickou kartu zpravidla dokupujeme např. u herních počítačů kvůli výkonu. Integrované řešení na základní desce (nebo čím dál častěji přímo v procesoru) je myšlené pouze na kancelářské aplikace.

Parametry desky jsou určeny tzv. čipsety, v minulosti existovaly 2 čipy - Northbridge a Southbridge (česky někdy překládané jako severní a jižní můstek, můstky proto, že čipy zajišťovaly komunikaci mezi jednotlivými komponentami základní desky), kterým se dohromady říkalo chipset, ten určuje, které funkce jsou pro daný hardware dostupné. Dnes jsou na trhu výhradně čipové sady od společností AMD a Intel. Pokud se podíváme na aktuální nabídku čipových sad společnosti Intel, tak postupně jde o H410, H470, B460 a Z490. Posledně jmenovaná je zároveň číselně nejvyšší což znamená, že nabízí nejvíce funkcí a podporuje i přetaktování procesoru. Severní a jižní je odvozeno podle jejich pozice na desce. Northbridge zajišťoval komunikaci mezi procesorem (CPU), operační pamětí (RAM) a grafickou kartou. Pro stále vyšší výkon a objem přenesených dat však začal představovat úzké hrdlo a proto se začal integrovat přímo do CPU. Southbridge se stará o I/O operace (vstup/výstup). Zprostředkovává komunikaci mezi disky, USB, LAN, audio... Na desce ho nalezneme jako čip s pasivním chladičem.



Obrázek 9: "Základní deska, obrázek upravený z www.alza.cz

Do základní desky usazujeme především procesor, existují dvě základní platformy – Intel a AMD, podle výrobce volíme příslušnou procesorovou patici (socket). U AMD je k dispozici AM4 pro procesory Ryzen, případně TR4 pro 32jádrové procesory Threadripper. U Intelu je situace obdobná a patice LGA 1151 je určena pro procesory Intel Coffee Lake, kdežto LGA2066 pak podporuje profesionální a poměrně drahé Skylake-X a Kaby Lake-X. Podle účelu volíme rozměr základní desky, čím menší deska, tím méně dostupných konektorů. Mezi standardní formáty patří formát ATX, má rozměry 305 x 244 mm, ovšem to je spíše jen jakýsi doporučený rozměr, který dodržují spíše jen dobře vybavené základní desky. Formát Micro ATX má rozměry 244 x 244 mm, jde o základní desky, které jsou nabízejí méně rozšiřujících slotů, konektorů rozhraní SATA a méně rozšiřujících slotů PCI a PCI Express sběrnice. Mini ITX je formát určený pro minipočítače, na deskách s rozměry 170 x 170 mm jsou integrovány externí zadní I/O porty, 1 slot pro karty a 2 sloty pro paměťové moduly DIMM. Speciální kategorií jsou tzv. MiniPC, které mají své specifické kategorie a rozměry desek.

Na základní desce je oscilátor, generující základní taktovací frekvenci BCLK (zkratka anglického výrazu Base Clock), což je frekvence, která je výchozí pro taktování (pracovní „rychlost“) procesoru či paměti.



PRO ZÁJEMCE

V současnosti se zejména v souvislosti s mobilními ICT setkáváme s tzv. SoC (system on a chip) – „systém na čipu“. Jedná se o integrovaný obvod, který odstraňuje hranici mezi mikroprocesorem a mikropočítačem, obsahuje kromě vlastního procesoru i další subsys-

témy pro zpracování grafiky, zvuku nebo připojení periférií, nejde tedy o rozdělení do čipsetů nebo samostatných karet, jak to známe z osobních počítačů. Jader procesoru je v SoC v dnešní době i více.

V souvislosti se základní deskou je nutné si uvědomit, že počítač si musí pamatovat některá nastavení, například počet a typy paměti, počet a typy disků, místo, odkud se bude nahrávat operační systém apod. K tomu slouží BIOS – Base Input Output System, software nahraný do speciálního paměťového čipu základní desky. Najdete ho na každém PC označeném jako IBM kompatibilní (na Macbooku Air od společnosti Apple BIOS není, na tabletech je součástí firmware dostupného pomocí BootLoaderu). V současnosti je Bios nahrazen tzv. UEFI (Unified Extensible Firmware Interface).

2.1.2 PROCESOR

Procesor neboli CPU je hlavní čip zodpovědný za provedení všech úkonů počítače. Přijímá pokyny od softwaru (programů spuštěných na daném počítači) a zpracovává je dál k jednotlivým komponentům počítače. Zásadně se od něj odvíjí rychlost samotného PC či notebooku. Procesor je hlavní elektronická součástka v počítači, která vykonává instrukce počítačového programu tak, že provádí základní aritmetické, logické, kontrolní a vstupní/výstupní operace specifikované instrukcemi. Procesor, označovaný jako srdce, či motor počítače, patří mezi hlavní výpočetní jednotky nejen osobních počítačů, ale také například chytrých mobilních telefonů. Často je označovaný také zkratkou z anglických slov central processing unit – CPU. Procesory nejsou vyrobeny za účelem jediné funkcionality, nýbrž pracují tak, jak to aplikace potřebují. Zároveň je procesor jakýsi mozek počítače, řídí jeho ostatní části a jejich podíl na společném výpočetním díle celého systému. Hlavními výrobci procesorů do počítačů jsou společnosti Intel a AMD. Procesor provádí výpočetní operace, jednotlivé operace dohromady skládají instrukce, jejichž svazky se nazývají programy. Program je zpracováván sekvenčně, což znamená, že jeho instrukce procesor provádí jednu po druhé v takzvaných instrukčních cyklech. Množství těchto cyklů za jednu sekundu se nazývá frekvence procesoru. Jednotkou je Hertz (Hz), frekvenci zpravidla udáváme v giga jednotkách (GHz), případně mega (MHz, milion hertzů). Frekvence se také často označuje jako kmitočet či takt procesoru. Celkový výkon procesoru však ovlivňují i jiné faktory, jako například jeho architektura, počet jader aj., pouze vyšší frekvence neznamená výkonnější procesor.

Velká část procesorů Intel i AMD dokáže svou frekvenci přizpůsobovat automaticky. Zajišťují to technologie Turbo Boost (Intel) a Turbo Core (AMD), přičemž obě mají obdobný efekt. Když je procesor intenzivně vytěžován, zvýší dočasně frekvenci až po mezní „boost“ limit, čímž si zajistí dostatek výkonu. Pokud takový výkon není potřeba, frekvence zůstane nižší, a ve výsledku se ušetří energie.

Základním stavebním kamenem procesoru jsou jádra, postavená na příslušné architektuře. Jedná o samostatné výpočetní jednotky, více jádrový procesor lze považovat za vysoce integrované zapojení více jedno jádrových procesorů. Jádra jsou na sobě výpočetně nezávislá, díky čemuž je procesor schopen v jednu chvíli zpracovávat několik různých instrukcí a obsluhovat tak několik programů najednou. Někdy bývá počet jader zaměňován s počtem zpracovávaných vláken. Z pohledu operačního systému se instrukce dělí na vlákna, která mohou být zpracovávána paralelně (souběžně). Z toho důvodu se o těchto procesorových vláknech mluví také jako o virtuálních nebo logických jádrech. Důležité je poznamenat, že procesorová vlákna výkonnostně nezastupují plnohodnotná fyzická jádra, protože s nimi sdílí hardwarové prostředky, čímž je jejich výkon omezen. Tato technologie firmy Intel se nazývá Hyper-Threading, procesory AMD využívají technologii SMT (Simultaneous multithreading). Přestože se taková technologie považuje za efektivní, nemůžeme tvrdit, že dvoujádrový procesor se čtyřmi vlákny může být roven plnohodnotnému čtyřjádrovému procesoru.



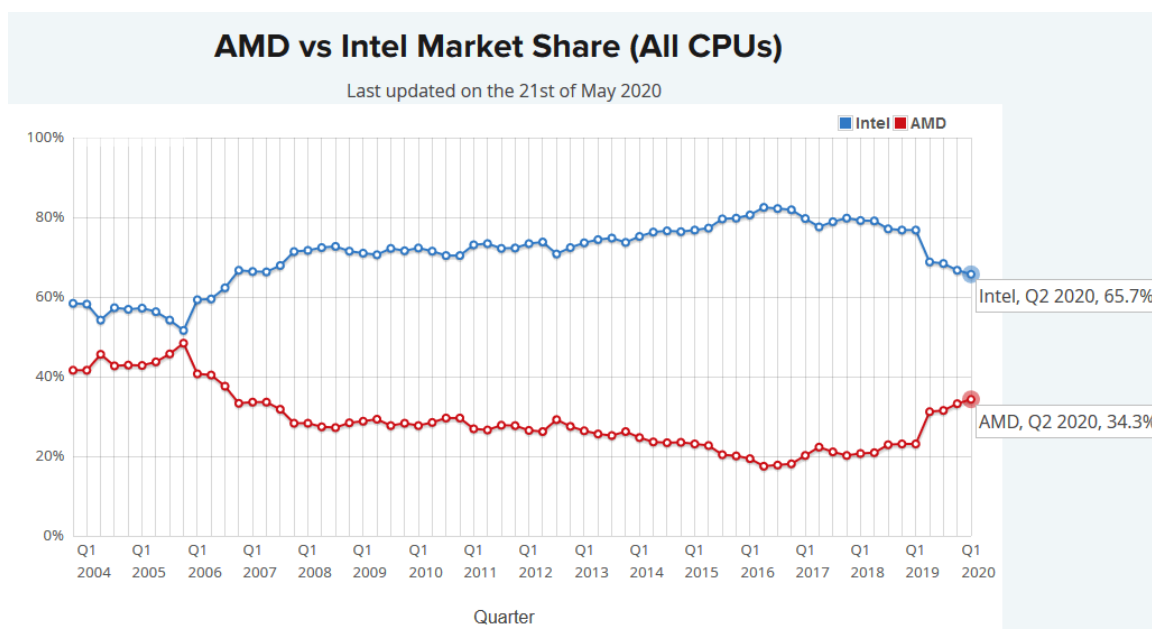
OTÁZKY 2-1

Pokuste se vyjmenovat základní parametry procesoru, (především na platformě pro PC)

Označení	Vysvětlení
Intel Celeron	Rodina procesorů Intel Celeron zastupuje nejnižší výkonnostní a cenovou kategorii. Jedná se o dvoujádra bez HyperThreadingu a s pevnou frekvencí, která je zpravidla nižší než 3 GHz.
Intel Pentium	Dnes se na trhu pod tímto jménem vyskytují čipy nižší třídy se dvěma jádry a až na výjimky bez technologií HyperThreading nebo Turbo Boost. Frekvence současné generace se pohybuje kolem 3,5 GHz.
Intel Core i3	Řada obsahuje čipy nižší střední třídy, které poslouží při každodenní kancelářské práci, ale také při hraní nenáročných her. Dvě fyzická jádra s HyperThreadingem a frekvence drží se kolem úrovně 4 GHz bez technologie Turbo Boost.
Intel Core i5	Zlatá střední cesta s dostatkem výkonu i na náročné aplikace. Čtyři fyzická jádra bez HyperThreadingu a frekvence přes 4 GHz s technologií Turbo Boost. Standardně je v každé generaci alespoň jeden model s odemčeným násobičem.
Intel Core i7	To nejlepší, co si může běžný smrtelník dopřát. Čtyři fyzická jádra s HyperThreadingem, technologií Turbo Boost a frekvencí přes 4 GHz. Stejně jako u i5, i zde se vyskytuje model s otevřeným násobičem.
Intel Core i9	Novinka nastupující do segmentu HEDT (high-end desktop), který dosud ovládaly extrémní verze řady Core i7. Nabídne až 18 jader s HyperThreadingem a vysoké frekvence s Turbo Boost. Otevřený násobič je samozřejmostí u všech modelů této řady.
Intel Xeon	Serverová řešení společnosti Intel. Specifikace se mezi jednotlivými modely v nabídce často ztelně liší, ale za pravidlo lze považovat vyšší počet jader, nižší frekvence a mnohonásobně větší vyrovnávací paměť v porovnání s desktopovými procesory. Standardem jsou také technologie HyperThreading a Turbo Boost.

Obrázek 10: "Procesory Intel, obrázek upravený z www.alza.cz

V současnosti existují dva hlavní výrobci desktopových (PC) procesorů, kteří si dělí ta-
křka sto procent tohoto trhu. Prvním je Intel, druhým AMD. Oba dva své produkty dělí na
řady a konkrétní modelová označení. Řady slouží k hrubšímu rozdělení celé nabídky podle
účelu a výkonu procesoru.



Obrázek 11: "Procesory podíl na trhu, zdroj /www.cnews.cz

Intel v současnosti uvedl 10. generaci desktopových procesorů, kterou nazývá Comet Lake s následujícími parametry: Intel Core i3 bude mít 4 jádra a 8 vláken, Core i5 6 jader a 12 vláken, Core i7 získá 8 výpočetních jader a k tomu 16 vláken a konečně to nejvý-
konnější v podobě Core i9 se posouvá na 10 jader a 20 vláken. Připravena je jedenáctá
generace.

PRO ZÁJEMCE



Další informace lze čerpat například z následujících zdrojů, odkud je převzata i část
uváděných informací:

<https://www.alza.cz/co-je-procesor>

testy <https://www.arecenze.cz/procesory/>

<https://www.alza.cz/procesory-intel/18843284.htm>

<https://www.svethardware.cz/prehled-desktopovych-procesoru/22566>

<https://www.svethardware.cz/procesory/>

značení procesorů - <https://www.levnapc.cz/znaceni-procesoru-intel.html>

<https://www.cnews.cz/intel-procesory-core-11-generace-tiger-lake-pro-notebooky-uvadeni-modely-parametry-vykon-platforma-intel-evo/galerie-202682-14/>

<https://www.pocitarna.cz/blog/typy-procesoru-aneb-notebook-se-vybira-hlavne-i-podle-procesoru/>

<https://www.alza.cz/intel-generace-procesoru>

10 gen <https://www.czc.cz/geek/procesory-intel-core-10-generace-jsou-tu-proc-by-vas-mely-zajimat/clanek>

11 gen, <https://www.cnews.cz/intel-procesory-core-11-generace-tiger-lake-pro-notebooky-uvadeni-modely-parametry-vykon-platforma-intel-evo/galerie-202682-14/>

<https://itigic.com/cs/intel-rocket-lake-s-architecture-specifications-and-features/>

Inovace procesorů probíhá změnou mikroarchitektury procesoru, která je pak udávána jako generace procesorů. Každá generace je značena pořadovým číslem a kódovým označením. Například Intel v dnešních dnech nabízí devátou generaci procesorů Core, pojmenovanou Coffee Lake Refresh, a AMD druhou generaci procesorů Ryzen 2, kterou značí jako Zen +.

Zejména u tabletů a notebooků je důležitým parametrem TPD. Vzhledem k tomu, že výkonnost procesoru je vykoupena energetickou náročností, je u nejvýkonnějších procesorů jako doprovodný efekt spotřebované energie poměrně značné zahřívání, které je možné kompenzovat výkonnými ventilátory. Tablety a notebooky jsou však určeny pro mobilitu a jeden z limitujících faktorů je hmotnost. Proto se u této třídy používá zpravidla pasivní chlazení (bezventilátorové). Proto výrobci udávají i TDP, což je zkratka anglického výrazu Thermal Design Power, který referuje o maximálním tepelném výkonu procesoru čili o množství tepla, které při svém maximálním zatížení může produkovat. TDP je parametr používaný především ve spojení s dimenzováním chlazení, nižší TDP znamená nižší spotřebu procesoru. Spotřeba úzce souvisí s výrobní technologií, která se udává v nanometrech a její hodnota informuje o rozměrech tranzistorů v procesoru. Čím menší tranzistory jsou, tím více jich je možné na čip umístit a zároveň zmenšit jeho prostorovou a energetickou náročnost. V současnosti Intel používá 14 nm technologii a AMD dovede vyrobit procesory s technologií 7nm (AMD EPYC 7542 má parametry 32 jádrový, 64 vláken, 2,9GHz, TDP 225W) Boost 3,4 GHz, 128MB L3 cache, socket AMD SP3, 7nm, cena se pohybuje přes 100000 Kč).

Jak bylo uvedeno dříve, pokud existují komponenty nebo dílčí části s různou rychlostí práce, je vhodné vyrovnávat jejich rychlost pomocí dílčích (vyrovnávacích) pamětí. V procesoru existují paměti označované jako L1, L2 a L3 Cache, (L podle layer – vrstva). Čím je úroveň paměti nižší, tím menší je její objem, vyšší její rychlost a zároveň má blíže k samotnému procesoru. L3 cache je v procesoru nejpomalejší, nejobjemnější a sdílí ji všechna jádra. Paměti L2 a L1 jsou implementovány přímo v jádře. Zpravidla platí, že čím větší paměť L3 cache, tím lépe. Nemusí to však být zcela pravda, například Intel u svých

posledních HEDT procesorů architektury Skylake-X zrevidoval návrh vyrovnávacích pamětí a redukoval velikost L3 ve prospěch L2 cache. (www.alza.cz)

Označení	Vysvětlení
AMD Sempron	Procesory AMD Sempron jsou tím nejlevnějším, co můžete v nabídce AMD najít. K dispozici je dvoujádrová a čtyřjádrová verze obsahující integrované grafické jádro. Obě pracují na velmi nízkých takttech a jsou založeny na staré výrobní technologii 28 nm.
AMD Athlon X4	Tyto modely mají základ taktáž v zastaralé 28nm výrobní technologii, jedná se o čtyřjádra s frekvencemi až kolem 4 GHz a technologii Turbo Core. Nedisponují grafickým jádrem.
AMD FX	Tvář temné doby pro AMD: procesory, jež si nechaly technologicky utéct řadu Core od Intelu. Produkovány jsou zastaralou technologií 32 nm ve verzích se čtyřmi, šesti a osmi jádry. Otázka jader je v tomto případě kvůli sdílení některých zdrojů kontroverzní téma, protože vnitřní topologie procesorů AMD FX odpovídá spíše polovičnímu počtu jader se schopností zpracovávat více vláken.
AMD Ryzen 3	Novinka od společnosti AMD, jejíž výroba, stejně jako všech procesorů Ryzen, probíhá moderním 14nm procesem. Jedná se o čtyřjádrové procesory bez SMT, s Turbo Core a frekvencemi přes 3 GHz.
AMD Ryzen 5	Odpověď společnosti AMD na populární procesory Intel Core i5. V této cenové kategorii se však jedná o premiérové modely se šesti jádry, které společně s těmi čtyřjádrovými sdílí technologie Turbo Core a SMT, díky čemuž se dostalo celých 12 výpočetních vláken do kategorie procesorů střední třídy.
AMD Ryzen 7	Procesory Ryzen vyšší třídy. Osm jader s SMT, tedy celkem 16 vláken. Samozřejmostí je technologie Turbo Core pro automatické přetaktování a absence grafického jádra.
AMD Ryzen 9	S nástupem třetí generace procesorů AMD Ryzen se nabídka rozrůstá o kategorii Ryzen 9. Společnost AMD totiž na trh pošle dokonce až 16jádrový procesor AMD Ryzen 9 3950X. Jeho předběžné výkonnostní testy jsou opravdu nádherná a s naprostým přehledem se stane nejvýkonnějším dostupným CPU, pokud samozřejmě nebudeme počítat procesory postavené spíše pro serverovou platformu.
AMD Threadripper	HEDT řešení v podání AMD. Dnes jsou dostupné dva modely AMD Ryzen Threadripper. Oba samozřejmě sdílí moderní technologie, včetně té výrobní, a frekvence kolem 4 GHz po aplikaci Turbo Core. Jejich doménou v portfoliu AMD je 12 a 16 fyzických jader s SMT, takže nižší z těchto modelů má celkem 24 výpočetních vláken, zatímco vyšší jich má 32.

Obrázek 12: "Procesory AMD, obrázek upravený z www.alza.cz

Spojícím prvkem mezi procesorem a základní deskou je patice procesoru, tzv. Socket. Jedná se o konektor, do kterého se pouzdro procesoru vloží a zamkne. V současnosti existují dva hlavní typy socketů. Rozšířenější je LGA, který spojuje desku s procesorem kontaktními plochami. Druhý typ je ustupující PGA, jehož princip spočívá v uzamykání procesoru pomocí pinů přesně pasujících do otvorů v patici. V současnosti jsou standardně nabízeny Sockety Intel 1150, Intel 1151, Intel 1200, Coffee Lake, Intel 3647 a AMD AM4.

ODPOVĚDI 2-1



Nejpodstatnější parametry procesorů (především na platformě pro PC, podle www.alza.cz)

Socket – socket neboli patice je hardwarový konektor procesoru, do kterého se samotný čip usazuje. Procesor a základní deska musí mít socket označený stejně, je to první krok k funkční sestavě.

Počet jader – každé z jader zastupuje prakticky samostatnou výpočetní jednotku. Čím více má procesor jader, tím více operací dokáže zpracovávat najednou neboli paralelně. Vyšší počet jader je lepší, stejnou pozornost si však zaslouží i další parametry.

Frekvence – frekvence je reálný parametr definující základní rychlost procesoru, pokud není ovlivněn vnějším zásahem. Hlavní jednotkou je Hertz a v kontextu procesorů používáme jednotky gigahertzů čili miliardy Hertzů.

Max. frekvence – maximální frekvence je hodnota, které dokáže procesor dosáhnout po automatickém přetaktování.

Automatické přetaktování – díky této technologii dokážou procesory zvýšit frekvenci v zátěži až na výrobcem dané maximum. Společnost Intel tuto technologii nazývá Turbo Boost, AMD potom Turbo Core. Ne všechny procesory jí však disponují.

Integrované GPU – integrované GPU znamená, že základní grafický čip je integrován do pouzdra procesoru. To má jednu velkou výhodu – pro běžnou kancelářskou práci nemusíte kupovat dedikovanou grafickou kartu, zároveň však nemůžete očekávat vysoký výkon v náročných grafických aplikacích.

HyperThreading – HyperThreading je technologie společnosti Intel, díky které dokáže jedno jádro procesoru zpracovávat dvě softwarové operace najednou. Do jisté míry je tedy procesor schopen pracovat, jako by měl dvojnásobný počet jader.

Multi-Threading – celý název je simultaneous multithreading (SMT) a jedná se o AMD procesory využívaný ekvivalent technologie HyperThreading.

Otevřený násobič – otevřený násobič znamená, že je procesor snadno přetaktovatelný pouhým navýšením hodnoty tohoto parametru.

Virtualizace – virtualizace umožňuje na počítači spouštět virtuální stroje. Jedná se o takové počítače v počítači, které mají přístup k hardware, aniž by ovlivňovali hostitelský operační systém. Virtualizace je dnes již téměř samozřejmostí u valné většiny procesorů.

Pro servery – jedná se o procesory specializované na nepřetržitý provoz v serverech. Specifikem jim je například velký počet jader nebo většinová absence grafického jádra.

TDP – je to zkratka Thermal Design Power, kterou se značí množství tepla produkované procesorem. Hlavní jednotkou je Watt. TDP přímo nevypovídá o energetické spotřebě procesoru, je jí ale přímo úměrné.

Snížená spotřeba – procesory s tímto parametrem jsou optimalizované pro nízký odběr elektrické energie. Jejich společným znakem je snížená frekvence v porovnání se standardními modely.

L2 cache – cache je vyrovnávací paměť procesoru. Stojí mezi operační pamětí a výpočetními jednotkami procesoru, aby zamezila snižování rychlosti procesoru rychlostí paměti. Každé jádro má samostatnou L2 cache.

L3 cache – L3 cache má procesor jen jednu a sdílí ji všechna jádra. Obecně platí, že čím větší cache, tím lépe.

2.2 Technologické trendy v existujícím hardware

V této podkapitole bude provedeno srovnání na základě technologických inovací prezentovaných především na veletrhu Consumer Electronics Show (CES) v Los Angeles a CES Asia 2017, tak jak byly prezentovány na elektronických portálech, především na portálu svethardware.cz. Tyto informace jsou doplněny o další technologické novinky na základě internetových obchodů, zejména pak portálu alza.cz.

Podle Gartneru¹ došlo v roce 2017 ke změně prioritní technologické oblasti vývoje z roviny hardwaru do roviny software a služeb. Analytici očekávali masivní zvýšení investic, do softwaru měly firmy po celém světě vložit o 7,2 procenta více financí, než v roce 2016, celková částka byla prognózována na 357 miliard dolarů. Investice do služeb v oblasti IT měly růst výrazněji, měly dosáhnout 943 miliard dolarů, tedy téměř o pět procent více než v roce 2016.

2.2.1 VÝVOJOVÉ TRENDY V OBLASTI PC

Platforma osobních počítačů zaznamenala značné změny, pokles podílu na trhu a značnou diferenciaci. V minulosti zahrnovala tato platforma klasické stolní počítače a notebooky. Okrajově pak byly do této platformy zahrnovány serverové počítače. V současnosti je trh diferencován do široké škály produktů, počínaje chytrými telefony, přes tablety, tablet PC, počítače 2v1, netbooky, notebooky, ultrabooky, chromebooky, All in One PC, mini PC až po klasické PC. Definovat, kde končí a začíná která platforma je problematické, například tablety a chytré telefony shodně dosahují úhlopříčky displeje 7 palců. Tablety stejně jako chytré telefony mohou obsahovat operační systém Android, různé tablety však mohou obsahovat rozdílné OS, např. Android, Windows apod. Tablet nemá klávesnici, 2 v 1 PC má oddělovací klávesnici, mini PC se blíží spíše segmentu průmyslových počítačů. Je problematické jednoznačně říci „toto zařízení je počítač“ a „toto zařízení není počítač“. Jak bylo uvedeno, mobilní telefon má v některých případech shodné parametry jako tablet, přesto jej za počítač nepovažujeme. V některých případech nehovoříme ani v případě některých tabletů o počítači, přestože je zřejmé, že nelze rozhodnout o tom, zda je zařízení počítačem podle toho, jaký má instalovaný operační systém či zda má klávesnici. Z tohoto pohledu je tak spíše než o počítačích PC nutno hovořit o informačních a komunikačních technologiích.

Široké spektrum ICT, které vycházejí z platformy PC, vedlo k přesunu činností dříve specifických pro PC na nová zařízení. Z tohoto důvodu byl zaznamenán pokles odbytu klasických PC. Herní systémy se přesunuly do specializovaných herních konzol a zařízení, Internet a komunikace do tabletů a mobilů, klasické PC v současnosti zůstávají spíše do-

¹ Citováno z https://www.tyden.cz/rubriky/veda/technologie/technologicke-trendy-pro-rok-2017_406901.html

ménou kanceláří a z domácností jsou vytlačeny tablety a notebooky. Přese všechny předpovědi však platforma PC nekončí. Trendem je mobilita, prodej směřuje do relativně nového segmentu mini PC.

Tablet PC

Přetrvává trend tabletů, nově se objevuje pojem tablet PC pro zařízení, které např. Alza.cz specifikuje následovně: „Tablet PC, to je snadno přenosná kombinace notebooku s Windows a tabletu. Jsou menší, ale výkonnostně srovnatelné se standardními notebooky. Můžete na nich psát a pracovat jako na notebooku, ale i hrát hry jako na tabletu. Lze je ovládat myší a touchpadem nebo dotyky na displeji. Tablet PC jsou zařízení vhodná na cesty a častý přesun mezi kancelářemi či klienty. Vyznačují se dotykovou obrazovkou a dlouhou výdrží na baterii. Jako multimediální centrum umí komunikovat s další chytrou elektronikou. A v případě vypojení nebo překlopení klávesnice se změní v zařízení vhodné pro zábavu například na gauči“².

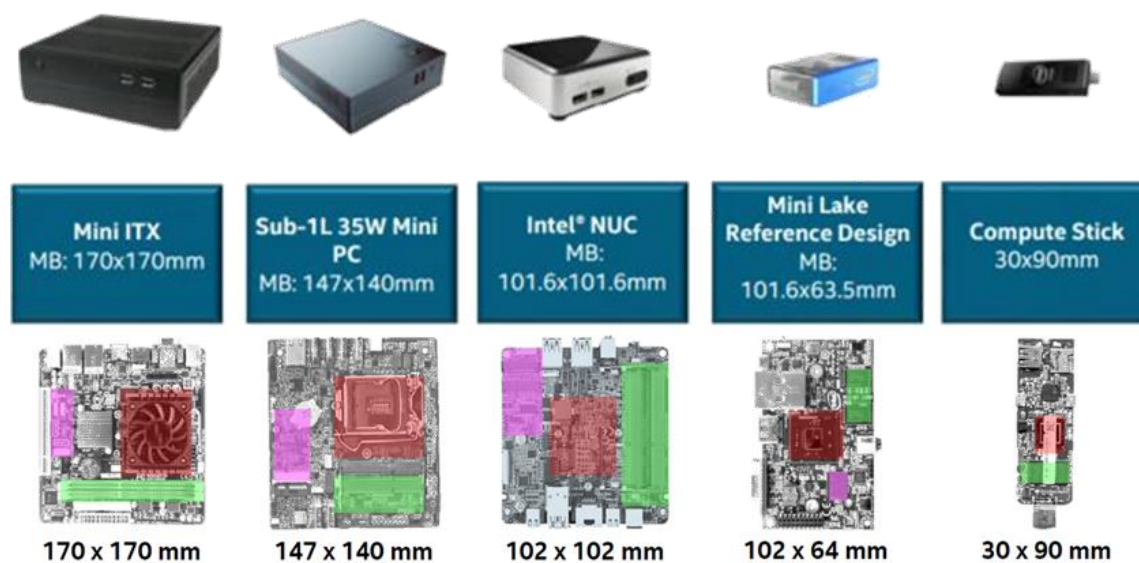
Mini PC

Na CES 2016 se projevil trend nástupu mini PC založených na procesorech Intel, do té doby dominantní mini PC s OS Android jsou postupně vytlačovány mini PC s OS Windows 10. Cenové relace zařízení osazených procesory ARM se pohybují od hranice 1000 Kč, při osazení procesory Intel Atom se pohybují v cenové relaci od 3000 Kč.



V současnosti (od roku 2015) je prosazována koncepce Intelu stanovující 5 základních rozměrů počítačů mini PC.

² Citováno z <https://www.alza.cz/notebooky/dle-vyuziti/tablet-pc/18848493.htm>



Obrázek 13 Rozměry a základní desky pro mini PC




PRO ZÁJEMCE



Přehled rozměrů základních desek počítačů řady mini PC a technických parametrů je dostupný např. na <http://www.10stripe.com/featured/form/atx.php>, dále na <https://www.cnews.cz/intel-chce-pet-typu-mini-pc-predvedl-standard-sub-1l-pro-14-x-14-cm-desky/> nebo na <https://liliputing.com/2015/04/intel-mini-pcs-getting-smaller-cheaper.html>

Nepotvrdil se trend pro kombinaci mini PC s mikro dataprojekty (především z důvodu malé rozlišovací schopnosti), stávající zařízení jsou spíše navržena pro připojení na TV přijímač, resp. klasický displej. V budoucnosti lze předpokládat i zobrazování prostřednictvím skládacího, resp. pružného displeje. V blízké budoucnosti lze však pružné displeje očekávat spíše v segmentu tabletů a mobilních telefonů.

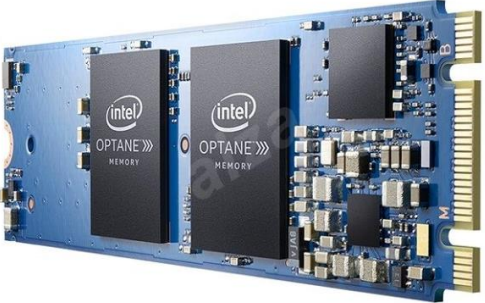
Nepotvrdili se ani tendence používání virtuálních klávesnic, není nadále zřejmý ani trend skládacích klávesnic, částečně je zřejmý návrat klávesnic Bluetooth, v minulosti vytlačených bezdrátovými klávesnicemi 2,4 Ghz.

<p>Mini Projector, iXunGo DLP Pico Video Projector</p>	
	<p>Native resolution 854*480 (support 1920*1080), DLP technology brighter than HD LCD projector with 2,000 Lumens. Support 1080P HDMI IN & WiFi Wireless Connectivity with 120" Display in Portable Size for Home Entertainment & Business Presentation</p>
<p>Dostupné z: https://www.amazon.com/dp/B076GKG265/ref=sspa_dk_detail_0?psc=1</p>	
<p>Lenovo ukázalo budoucnost: ohebný telefon a skládací tablet</p>	
	<p>Ohebný telefon Lenovo Cplus (vlevo). První vizí je ohebný smartphone. V „normálním“ stavu působí jako kterýkoliv jiný telefon, snad kromě toho, že je užší a protáhlejší. To má však své opodstatnění. Zařízení lze totiž ohnout přes zápěstí a nosit jako náramek Vpravo je ukázka ohybného displeje</p>
<p>Dostupné z https://mobilizujeme.cz/clanky/lenovo-ukazalo-budoucnost-ohebny-telefon-a-skladaci-tablet</p>	
<p>Ukázka skládací klávesnice</p>	
	<p>Skládací Bluetooth klávesnice Mini, dobíjecí přenosná, BT bezdrátová skládací s dotykovým panelem pro tablety Samsung, IOS, PC</p>
<p>Dostupné z https://www.lightinthebox.com/cz/prenosny-dvakrat-skladaci-bluetooth-klavesnice-bt-bezdratova-skladaci-klavesnice-touchpad-pro-ios-android-windows-tablet_p6063278.html?prm=1.3.5.0</p>	

Projekční klávesnice	
	<p>Laserová klávesnice EPIC Laser Keyboard (vlevo), s integrovanou myší. Laser této klávesnice promítá na stole a obraz je zcela plochý.</p> <p>Laserová klávesnice LK200S (vpravo), hmotnost 150 gramů, rozměry zařízení jsou 7 x 4,7 x 11,8 cm.</p>
Dostupné z http://www.kancelarske-sluzby.cz/laserova-klavesnice-k-tabletum-epic-laser-keyboard-eng a https://dotekomanie.cz/2016/02/laserova-klavesnice-recenze/	

V souvislosti s nástupem mini PC je nutné se zmínit o Intel Optane. Jedná se o SSD ve formátu přídatné karty do slotu PCI Express 3.0 ×4 nebo ve formátu U.2, resp. M.2. (blíže např. na <https://www.svethardware.cz/intel-uvadi-optane-pro-akceleraci-beznych-pc/44168>)

Intel Optane

Intel Optane Memory 16GB M.2 80MM	
	<p>SSD disk rozhraní M.2, PCIe NVMe 3.0 x2, čtení až 900MB/s, zápis až 145MB/s, 22x80mm, kompatibilní s čipsety B250/H270/Z270 + Kaby Lake Core CPU 7. generace</p>
Dostupné na https://www.alza.cz/intel-optane-ram-16gb-d4845364.htm	

Tato paměť není SSD diskem, jedná se spíše o cacheovací paměť která určitým způsobem kapacitně nahrazuje operační paměť. Nejedná se však o klasickou RAM, která je volatelná, (po vypnutí či restartu PC ztratí veškerá data). Intel Optane Memory není volatelná, uložená data v ní zůstávají i po ztrátě napětí. Je osazena paměťovými čipy s technologií 3D Xpoint. Tato paměť je určena především pro kombinaci s klasickým HDD diskem. Je určena především pro systémy s procesory Kaby Lake a novějšími řad Core i3, i5 a i7.

PRO ZÁJEMCE



Recenze, srovnání výkonu a testy lze nalézt např. na stránkách Alza.cz, <https://www.alza.cz/intel-optane-revolucni-ssd>.

Blíže o technologii a NAND pamětech a technologii 3D Xpoint lze nalézt např., na <https://pctuning.tyden.cz/hardware/disky-cd-dvd-br/47047-intel-optane-32-gb-v-testu-zrychlete-stary-disk-az-petkrat>

2.2.2 TRENDY A INOVACE V SEGMENTU PROCESORŮ

Jak bylo zmíněno, důležitým trendem ve vývoji IT je mobilita, s ní souvisí i nízká energetická náročnost. Zvyšování výkonu procesorů směřuje do segmentu nízkoenergetického hardware, zejména v kontextu IoE pak také do výkonných serverových aplikací umělé inteligence a cloudového prostředí.

Procesory ARM t

Předními výrobci procesorů ARM jsou Broadcom a Qualcomm, které v současnosti jednají o sloučení. Většina mobilních zařízení, tabletů či mobilů, je vybavena těmito procesory. V současnosti jsou však vyvíjeny i vysoce výkonně procesory ARM určeny právě pro cloudové služby nebo pro servery. Procesory ARM dokonce směřují i do segmentu superpočítačů. Společnost Cray, přední výrobce superpočítačů, plánuje tvořit nové superpočítače na architektuře ARMv8 (64bit), na speciálně upravených jádrech na 14nm výrobní technologii, využívající procesory ThunderX2. Měly by obsahovat 54 procesorových jader pracujících na 3 GHz, šesti kanálový řadič pro paměti DDR4.

V současnosti uvedl na trh Qualcomm procesor Centriq 2400, vyráběný 10nm technologií. Je osazen 18 miliardami tranzistorů na ploše menší než 400 mm². Je tvořen celkem 48 vysoce výkonnými 64bitovými jedno vláknovými jádry ARM (Armv8-A), která budou pracovat na 2,3 až 2,6 GHz. Qualcomm uvádí, že tato jádra jsou propojena obousměrnou segmentovanou kruhovou sběrnicí, která pracuje s celkovou propustností 250 GB/s. K dispozici bude 512 kB L2 cache, která bude sdílená páry jader a 60 MB L3 cache pro všechna jádra. Systémová paměť může být až 768 GB. K dispozici je šesti kanálový paměťový řadič pro DDR4 a 32 linek PCIe 3.0.

Zajímavým trendem může být i kombinace různých jader na jednom čipu. Firma MediaTek uvedla mobilní procesor Helio X30 vyrobený 10nm technologií, který má dvě jádra Cortex-A73 pracující až na frekvenci 2,8 GHz, čtyři jádra Cortex-A53 na 2,2 GHz a čtyři jádra Cortex-A35 s maximem na 2 GHz. Dohromady 10 jader dokáže podle potřeby regulovat výkon a spotřebu použitím vhodné kombinace jader.

Posun k procesorům ARM oznámil v roce 2020 rovněž Microsoft, který údajně pracuje na vlastním návrhu procesorů ARM, které by používal ve svých datových centrech místo řešení do Intelu. Kromě toho zvažuje také vlastní procesory pro počítače Surface.

Intel Atom

Procesory Intel Atom byly v minulosti levné, nízkoenergetické procesory pro mobilní zařízení s malými nároky na výkon. V současnosti přichází na trh řada C3000, která je tvo-

řena v současnosti 20 typy, určenými pro IoE, pro síťová úložiště a sítě obecně a pro servery a cloudová úložiště, obdobně jako nové modely ARM. Hlavním důvodem uvedení nové řady Atomů je zejména nutnost konkurenčního produktu k procesorům ARM ve výpočetně náročnějším prostředí s nízkými nároky na spotřebu.

Všechny procesory jsou vyráběné 14nm technologií (architektura Goldmont), mohou obsahovat 16 procesorových jader a podporují ECC paměti. Je podporováno až 256 GB operační paměti DDR4 v dvoukanálovém zapojení. SoC podle verze disponuje podporou ethernetu s rychlostí 4 x 2,5 GbE až 4 x 10 GbE od Intelu, integrována je podpora virtualizace Intel VT, podpora SATA 6 Gb/s, USB 3.0 a eMMC. Čipy pro IoT mají teplotní toleranci v rozmezí -40 až +85 stupňů Celsia.

Modely určené pro síťové nasazení a IoT mají integrovaný systém hardwarového šifrování pomocí Intel QuickAssist s rychlostí až 20 Gb/s.

Nejnižší model Intel Atom C3308 má 2 jádra na frekvenci 1,6 GHz a je určený pro IoT, čemuž odpovídá TDP 9,5 W. Cenová relace se pohybuje do 1000 Kč.

Nejvýkonnější modely C3958 a C3955 pro serverové a síťové nasazení disponují 16 jádry, mají frekvenci 2 GHz a 2,1 GHz, TDP je 31 W, respektive 32 W. Cenová relace modelů je do 15000 Kč.

PRO ZÁJEMCE



Bližší informace lze dohledat např. na <https://www.svethardware.cz/cray-si-vybral-arm-pro-utok-na-trhu-superpocitacu/45518> nebo na <https://www.svethardware.cz/qualcomm-avedl-na-trh-10nm-centriq-2400-s-18-miliardami-tranzistoru/45484>.

Informace k procesorům Atom jsou dostupné na <https://www.zive.cz/clanky/intel-predstavil-20-novych-procesoru-rady-atom-maji-az-16-jader/sc-3-a-189104/default.aspx>

Aktuální přehled desktopových procesorů lze dohledat na <https://www.mapujevirtuálníobjektydoreálnéhoprostředísvethardware.cz/prehled-desktopovych-procesoru/22566>.

Informace o 9. generaci Intel Core a osmijádrovém Core i7-9700K lze dohledat např. na <https://www.svethardware.cz/rysuje-se-9-generace-core-s-osmijadrovym-core-i7-9700k/45598>

Bližší informace vztahující se ke značení procesorů jsou např. na <https://www.zive.cz/clanky/nevyznate-se-ve-znaceni-procesoru-intelu-tady-mate-tahak/sc-3-a-186152/default.aspx>

2.3 Vývoj a inovace zařízení založených na existujících technologiích

Podle prezentovaných aplikací a hardware na CES 2016 lze vývoj a inovaci nových zařízení, vycházejících z existujících technologií vztáhnout k IoT a posunout do roviny, kdy existuje teoretická možnost udělat chytré zařízení z čehokoli, na CES bylo předváděno například chytré zrcadlo, které zobrazí zprávy, počasí či maily nebo chytré lžičky, počítající kalorie. IoT se posouvá k praktické platformě chytrých domovů a měst.

Vývoj IoT a IoE podle americké výzkumné firmy Gartner³, má v roce 2017 jednoho společného jmenovatele, vytváření takzvané inteligentní digitální sítě (intelligent digital mesh). „Jde v podstatě o podobný systém, jaký vidíme v dnešních populárních "chytrých domácnostech", jen ještě povýšený na vyšší úroveň – přístroje všeho druhu budou ještě více než dnes propojeny v obřích síťových systémech, v nichž spolu budou samostatně komunikovat a vzájemně se řídit. To samozřejmě klade velké nároky jak na umělou inteligenci a prostupnost sítí, tak třeba na platební či komunikační systémy. A v neposlední řadě též na zabezpečení“.

V rámci posunu IoT k IoE nebude v budoucnu docházet pouze ke k propojení věcí, procesů a lidí, předpokládá se implementace umělé inteligence a s tím spojená schopnost přístrojů se učit, a to i v oblasti softwarové kooperace, např. Google i Microsoft vybavují pokročilými systémy umělé inteligence své cloudové služby. Dále by v rámci IoE měly větší roli začít hrát virtuální asistenční služby a virtuální umělá inteligence (prozatím např. ve formě hlasových asistentů jako Siri a Cortana).

Softwarové klony

Samostatnou vizí související nejen s IoE je vytváření digitálních kopií, v praxi formou softwarových klonů skutečných přístrojů, které mohou analyzovat a napodobit chování techniky v reálném světě. Tyto technologie předurčují (např. v souvislosti s projektem Microsoft HoloLens) další formu IoE, a to virtuálního IoE, který nejen propojuje věci, procesy a osoby, ale rovněž umělou inteligenci a virtuální objekty.

2.3.1 KOMUNIKACE PRO PODPORU IOT, IOE

V souvislosti s IoT a IoE tedy vyvstává nutnost kvalitnějších komunikačních kanálů, zejména zvyšování rychlosti bezdrátových technologií a vytváření alternativních páteřních cest.

³ Převzato z https://www.tyden.cz/rubriky/veda/technologie/technologicke-trendy-pro-rok-2017_406901.html

V segmentu bezdrátových komunikací došlo k rozšíření standardu IEEE 802.11, což je standard pro Wi-Fi s dalšími doplňky pro lokální bezdrátové sítě (Wireless LAN, WLAN) vyvíjený 11. pracovní skupinou IEEE LAN/MAN standardizační komise (IEEE 802). Standard 802.11ac zavedený v roce 2013 a uvedený do praxe v roce 2014 využívá pásma 2,4 GHz a 5 GHz. V současnosti dochází k implementaci standardu 802.11ad, který dále pásma rozšiřuje o pásmo 60 GHz. Zejména IoE však bude vyžadovat další, nové standardy.


802.11 ac

802.11 ad

Standard	Rok vydání	Pásmo [GHz]	Maximální rychlost [Mbit/s]
původní IEEE 802.11	1997	2,4	2
IEEE 802.11a	1999	5	54
IEEE 802.11b	1999	2,4	11
IEEE 802.11g	2003	2,4	54
IEEE 802.11n	2009	2,4 nebo 5	600
IEEE 802.11y	2008	3,7	54
IEEE 802.11ac	2013	2,4 a 5	1000
IEEE 802.11ad	2012	2,4, 5 a 60	7000

Obrázek 14 Standardy 802.11x

Na trhu se v současnosti již objevují zařízení, umožňující komunikaci standardem 802.11ad, například Alza.cz nabízí router TP-LINK Talon AD7200, jehož teoretická datová propustnost je 7,133 Gb/s, především díky využití pásma 60 GHz, v němž přenáší až 4,6 Gb/s.

Router TP-LINK Talon AD7200	
	<p>WiFi router 802.11a/b/g/n/ac/ad až 7200Mbps, Tri-Band (2.4GHz 800Mbps + 5 GHz 1733Mbps + 60GHz 4600Mbps), 8x pevná anténa + 1x interní anténa, 3x USB 3.0, 1x GWAN, 4x GLAN</p> <p>Router TP-LINK Talon AD7200 umožňuje poskytovat silné a stabilní 5GHz a 2.4GHz WiFi připojení. Disponuje 32 anténami zformovaných do speciálního anténního pole pásma 60GHz, které zajišťují koncentraci WiFi signálů</p>
Dostupné z https://www.alza.cz/tp-link-talon-ad7200-d4691965.htm?o=5	

Ještě většího význam mají technologie související s pátevní infrastrukturou. Zejména je nutné vytvářet alternativní a nezávislé pátevní sítě.

Společnost SpaceX si zaregistrovala značku Starlink, bude se jednat o vytvoření celosvětové internetové sítě, zajišťované prostřednictvím satelitů. Celosvětová internetová satelitní síť má být realizována prostřednictvím 4425 satelitů na nízké orbitě, a to celkem v 83 výškových vrstvách mezi 1110 a 1325 km nad zemí, systém má být dokončen v roce 2024. Satelity budou spojeny se tzv. základnami, budou tvořit páteř pro komunikaci, zatímco koncoví zákazníci se budou připojovat právě prostřednictvím základen. Na rozdíl od jiných satelitních sítí pro přístup k Internetu má mít Starlink zpoždění pouze okolo 25 ms a propustnost do jednoho gigabitu.



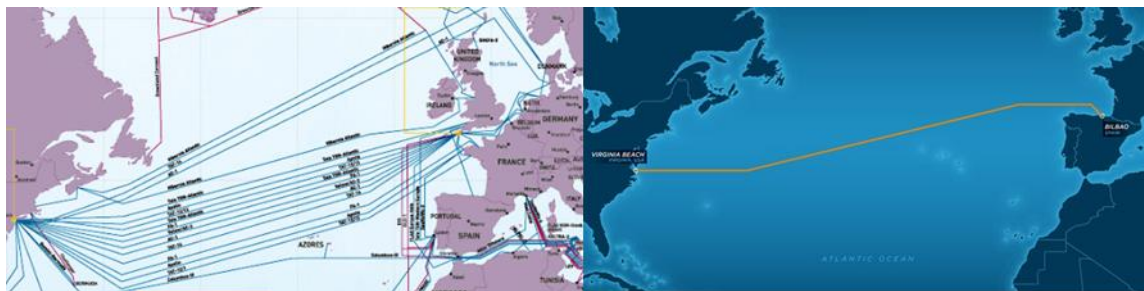
PRO ZÁJEMCE

Oblast komunikací lze nalézt např. na <https://www.svethardware.cz/spacex-za-dva-roky-vypusti-prvni-z-4425-internetovych-satelitu/44380>, dále pak na <https://www.svethardware.cz/celosvetova-satelitni-sit-spacex-bude-mozna-znama-jako-starlink/45209>.

Informace o podmořských kabelech jsou dostupné např. na <https://www.svethardware.cz/microsoft-a-facebook-uz-maji-nejrychlejsi-podmorsky-kabel/45237> a https://technet.idnes.cz/google-podmorsky-kabel-internet-dah-/kratke-zpravy.aspx?c=A160630_150254_tec-kratke-zpravy_dvz

S alternativní infrastrukturou souvisí i zvýšení propustnosti stávající infrastruktury, zejména pak nutnost posílení páteřních podmořských kabelů. Otázka propustnosti úzce souvisí i s bezpečností, např. v roce 2012 způsobil značné škody na infrastruktuře na východním pobřeží USA hurikán Sandy, především v New Jersey a New Yorku, kde ústí převážná část transatlantických kabelů. Došlo k narušení komunikace mezi celou Severní Amerikou a Evropou a značnému omezení přenosu dat.

Společnosti Microsoft, Facebook a Telxius dokončily propojení mezi španělským Bilbaem a Virginia Beach kabelem Marea, 6600 kilometrů dlouhým, v hloubce až 5,2 kilometrů. V současnosti se jedná o kabel s nejvyšší datovou propustností propojující Spojené státy a Evropou. Marea dokáže přenést 160 Tb dat za sekundu, nevede obvyklými místy jako jiné kabely propojující Evropu s USA a plní současně funkci zálohy pro internetové spojení s USA. Marea posiluje stávající komunikaci a propustnost mezi USA, Evropou i Afrikou. (V současnosti mezi USA a Evropou proteče o 55 % více dat než prostřednictvím pacifických linek a o 40 % více dat než prostřednictvím latinskoamerických).



Obrázek 15 Podmořský kabel Marea (vpravo) a část zbývajících kabeláže (vlevo)

Vzhledem ke společnostem podílejícím se na projektu Marea lze soudit, že tyto společnosti mají v úmyslu posilovat služby související s cloudovými službami a sociálními sítěmi, u Microsoftu lze předpokládat očekávání nárůstu počtu uživatelů Bing, Office 365, Skype, Xbox Live či Microsoft Azure a s tím související vyšší nároky uživatelů na přenosová data spojená s cloudovými službami, kabel Marea má tak pomoci uspokojit budoucí poptávku. Kabel bude provozovat firma Telxius, která patří do španělské telekomunikační skupiny Telefónica,

Obdobně firma Google společně s pěti asijskými firmami provozuje od roku 2016 kabelovou síť mezi USA a Japonskem o kapacitě 60 Tb dat za sekundu prostřednictvím šesti párů optických vláken. Výstavba byla ohlášena v srpnu 2014, vývoj a financování zajišťovalo konsorcium FASTER, složené z šesti společností včetně Googlu. Cena projektu se pohybovala okolo 300 milionů dolarů, délka z amerického Oregonu k Japonsku je 9000 km.

Kabel Google

Z těchto informací je rozeznatelné riziko monopolizace Internetu a rozdělení majoritní komunikace mezi malý počet významných komunikačních hráčů, což by vedlo ke zvýšení rizika a možnosti „vypnutí“ Internetu, resp. možnosti cíleného snížení propustnosti Internetu.

2.3.2 ROZŠÍŘENÁ A VIRTUÁLNÍ REALITA

Jedním z fenoménů současnosti je virtuální realita (VR), která nahrazuje vjemy z okolí novými. Vidíme jiný svět kolem sebe, slyšíme jeho zvuky a jsme vytrženi z opravdové reality. Ve většině případů se nejedná o nové technologie, ale spíše o využití možností výpočetního výkonu stávajících počítačů, bezdrátových technologií a Internetových technologií. VR bývá realizována pomocí headsetu a hardwarově výkonného PC nebo herní konzole PlayStation 4.

Virtuální realita (VR)

Cílem VR je prezentovat fiktivní digitální svět v pokročilé simulaci včetně snímání vašeho pohybu. Základem VR jsou tedy speciální brýle (headset s displeji, které pro každé oko zvlášť promítají obraz), které detekují polohu hlavy a zabezpečují prezentaci stereoskopického obrazu (pomocí dvou malých obrazovek, resp. dvou výřezů obrazu). Obraz bývá doplněn prostorovým zvukem, v současnosti jsou ve vývoji prostředky pro realizaci dalších vjemů, např. oblek s elektromagnetickými impulsy. Iluzi přenesení do virtuálního

světa umocňují další senzory, které snímají pohyb hlavy a údaje v reálném čase přenášejí do aplikace nebo hry. Interakci s prostředím ve virtuálním světě pak zajišťují další ovladače v ruce uživatele. Virtuální brýle v současnosti vyrábí několik velkých společností jako je Oculus, HTC, Samsung nebo Google.

Současné trendy virtuální reality směřují k využití mobilních telefonů a příslušných aplikací prostřednictvím speciálních brýlí. Tato forma VR je nepoměrně levnější, podle Alza.cz je důležité mít alespoň Full HD rozlišení, čtyř jádrový procesor, pohybové senzory akcelerometr a gyroskop, což je v drtivé většině dnešní technologický standard, dále je nutná speciální aplikace, která prezentuje VR ve dvou samostatných oblastech displeje.



TextGoogle Glass

Za jednu z prvních, prakticky využitelných technologií souvisejících s virtuální realitou, lze považovat projekt Google Glass. Google Glass patří do kategorie nositelných technologií, tzv. Wearables. V tomto případě se však nejedná o simulaci virtuální reality ale o jakési informační okno zobrazované před očima pomocí projekce na speciální projekční plochu, realizované nositelným počítačem s náhlavním (head-up) displejem. Google označuje tuto technologii spíše za rozšířenou realitu.

Rozšířená realita (AR)

AR oproti VR pracuje s okolím a mapuje virtuální objekty do reálného prostředí. Informace poskytované uživateli jsou kombinací informací hands-free formátu smartphonů, a komunikace s internetem prostřednictvím hlasových příkazů. Reálný svět dotvářely virtuální prvky, brýle mohly skrytě nahrávat videa, což kolidovalo s ochranou osobnosti. Tento problém je však obecným problémem mobilních záznamových prostředků (kamery v autě apod.).

Technologie Google Glass	
	<p>Zobrazení dat odpovídá displeji s úhlopříčkou 63,5 cm (1 280 × 720 bodů) ze vzdálenosti 2,5 metru. Fotoaparát má rozlišení 5 megapixelů, video 720p. 16 GB vnitřní paměť (uživatelská 12 GB), wi-fi standardu b/g, Bluetooth 4. Aplikace, pro konfiguraci pro Android 4.0.3 a vyšší.</p>
<p>Dostupné z https://mobil.idnes.cz/google-glass-c91-/mob_tech.aspx?c=A130416_151154_mob_tech_ram</p>	
<p>Google Cardboard s aplikací</p>	

	<p>ColorCross CardBoard (alza.cz)</p> <p>Brýle pro virtuální realitu včetně magnetického tlačítka, NFS, pro telefony s úhlopříčkou 4-5.5"</p> <p>Cena na alza.cz v listopadu 2017 139 Kč</p>
<p>Dostupné na https://www.alza.cz/gaming/colorcross-cardboard-d4020191.htm#prislusenstvi</p>	
<p>Lenovo Explorer</p>	
	<p>Brýle pro Windows Mixed Reality, 2x LCD 2.89" rozlišení až 2880x1440, 90Hz, HDMI 2.0, USB 3.1 Gen 1, Bluetooth 4.0, 4x AA, kompatibilita Windows 10</p> <p>Cena na Alza.cz v listopadu 2017 11999 Kč</p>
<p>Dostupné na https://www.alza.cz/gaming/lenovo-explorer-d5104470.htm</p>	

Google uvedl své brýle v testovacím provozu v roce 2013, vzhledem k malé prodejnosti Google tento projekt v roce 2015 ukončil.

O vysokém potenciálu této technologie svědčí fakt, že Google ohlásil aktualizaci, která přináší podporu Bluetooth, takže k brýlím lze nyní připárovat například ovládací zařízení, jako jsou myš nebo klávesnice. Nová verze má označení XE23. Rovněž Apple oznámil na rok 2018 uvedení vlastních brýlí pro VR a AR.

Google
XE23

PRO ZÁJEMCE



Bližší informace o projektu Google CardBoard a ukázky aplikací pro VR na mobilním telefonu najdete např. na <https://www.svetandroida.cz/google-cardboard-prvni-pohled-201409/> nebo na <https://vr.google.com/cardboard/>

O VR lze nalézt na Internetu spoustu informací, můžete si prostudovat základy VR např. na <https://doupe.zive.cz/clanek/vzhuru-do-jinych-svetu-aneb-jak-funguje-virtualni-realita>.

Kombinací VR a AR, založené na integraci VR do AR, přináší Microsoft HoloLens. Technologie je některými odborníky označována jako holografická (zařízení, které zobrazuje interaktivní trojrozměrné objekty v běžné realitě). Microsoft popisuje technologii jako mixovanou realitu, která přináší možnosti virtuální teleportace označené jako holoportace. Projekt vychází z mapování objektů do reality, kdy největším problémem je prostorová stabilizace namapovaného prostředí s reálným prostředím při pohybu obrazu.

Microsoft tento produkt představil v San Franciscu na konferenci BUILD 2015, označil je jako brýle budoucnosti, které mají do reálného světa kolem nás vložit ten virtuální s hologramy.



Obrázek 16 Microsoft HoloLens - brýle pro mixovanou realitu

**Inteli-
gentní ma-
pování**

Microsoft HoloLens obsahují jednoduchý průhledový displej (HUD). Brýle inteligentně mapují okolní prostor a chápou gesta, pohyby, pohled očí i hlas uživatele, což umožňuje pracovat s obsahem a informacemi přirozenými gesty a pohyby, tedy způsobem, jaký je přirozený v klasické realitě. Digitální svět je namapovaný do skutečného a stává se jeho součástí, stejně jako fyzické objekty v okolí. Průhlednost displeje umožňuje trvalou pozornost a kontrolu nad okolním světem, čímž překonává zejména displeje mobilní elektroniky i klasické displeje VR, které způsobují ztrátu pozornosti a kontroly nad realitou, uživatel neví, co se děje v okolí.

Potřebný výpočetní výkon a hardware je implementován do helmy. Helma má hmotnost 579 gramů, uchycení zabezpečuje polstrovaný prstenec, který přilehne k hlavě a hmotnost rovnoměrně rozloží. Náhlavní prstenec se následně přitáhne zadním kolečkem, obdobně, jako např. na cyklo přilbě. Brýle musí být fixovány poměrně těsně, protože pro správné zobrazení musí být fixována vzdálenost očí.

Brýle jsou vybaveny 32bitovým procesorem Intel Atom 14 nm technologie, nejsou známé další parametry procesoru, vzhledem k vývoji procesorů Atom se bude jednat pravděpodobně o řadu architektury Goldmont s podporou ECC paměti. Výkon procesorů Atom

je spíše průměrný, v některých případech nejsou schopny konkurovat výkonným ARM procesorům, proto je mimo vestavěného grafického procesoru (GPU) součástí hardware ještě další, 24 jádrový procesor HPU (Holographic Processing Unit), který dokáže zpracovat až bilion operací za vteřinu. HPU obsahuje SRAM cache o velikosti 8MB a 1 GB DDR3 RAM. Zpracovává vstupy ze senzorů a kamer a zajišťuje, aby všechny hologramy vypadaly přirozeně a byly fixovány na svém místě v reálném světě tak, jako skutečné objekty, tj. aby při pohybu nevznikaly žádné prodlevy nebo zpoždění.

Brýle mají dále k dispozici 2 GB RAM a 64 GB paměti pro uživatelská data, jsou osazeny operačním systémem Windows 10. Napájení je zajištěno akumulátorem zabezpečujícím chod cca 3 hodiny, který lze klasicky nabíjet přes mikro USB. S okolím se brýle propojují přes Wi-Fi nebo Bluetooth.

Brýle disponují prostorovým zvukem zabezpečovaným několika reproduktory zabudovanými do prstence přiléhajícího k hlavě, čímž nedochází k přenosu zvuku do okolí.

Pro zachycení hlasových pokynů slouží soustava mikrofonů, dále lze hardware rozšířit o senzory pro snímání rukou pro hmatové vjemy s hologramem (umožní uživateli „doslova“ sáhnout na hologram) a ovládání přirozenými gesty. Ke snímání pohybu slouží rovnou čtveřice kamer, vždy dvě na každé straně, přičemž každá z nich pod jiným úhlem. Další kamera slouží na měření hloubky ostrosti, přesněji řečeno, identifikuje, jak daleko od předmětů jste. Pro video hovory, interakci a sdílení reality slouží 2Mpx kamera, natáčení videa zvládá v HD rozlišení.

Aby byl dojem z hologramu přirozený, snímají brýle neustále uživatelskou aktivitu pomocí akcelerometru, gyroskopu a magnetometru, což Microsoft označuje jako IMU (inerciální jednotka měření), která má zajistit to, že brýle pochopí to, kam se zrovna díváte, nebo kam se pohybujete. Ostrost a kvalitu hologramů podle okolního osvětlení zabezpečuje senzor okolního osvětlení.

Nejvýraznější částí brýlí je vizor, na který se prostřednictvím dvou projektorů s HD rozlišením promítají hologramy. Vizor je téměř zcela průhledný a zatmavení je minimální. Okolní realita je viditelná bez omezení. To je hlavní rozdíl oproti brýlím virtuální reality, které reálný svět eliminují a zobrazují pouze virtuální. Koncepce Microsoftu umožňuje, aby byly hologramy přirozeně vloženy do reálného světa a oba světa byly propojené.

PRO ZÁJEMCE



Další informace o HoloLens lze získat např. na <https://mobilenet.cz/clanky/vyzkouseli-jme-microsoft-hololens-s-hologramy-vsude-kolem-19969>, dále např. na <https://mobilenet.cz/clanky/vr-uz-je-nuda-vyrobci-se-nyni-snazi-dohnat-microsoft-hololens-33054> nebo na <https://mobilenet.cz/clanky/znovu-jme-vyzkouseli-microsoft-hololens-a-opet-to-byl-zazitek-31037>.

Obdobně Google s Lenovem oznámili práci na projektu Tango 3D, který má nastartovat novou generaci cenově přístupných přístrojů, jež bude možné používat k mapování interiérů, rozšířené realitě a mnoha dalším účelům. V rámci projektu Tango se vývojáři Google pokusili navrhnout řešení, které dokáže vnímat svou polohu v prostoru a v reálném čase ji identifikovat a zpracovávat. Telefon či tablet s dostatečným výpočetním výkonem specializovaným na zpracování vizuálních dat, dokáže následně vytvářet 3D modelování v reálném prostředí. Pro VR vyvinul Google aplikaci Daydream.

2.4 Vývoj nových technologií

Stávající vývoj je limitován technologickými a materiálovými možnostmi. Předpokládá se, že možnosti výroby počítačů na křemíkové bázi se posunuly k limitu lidských schopností a stávající technologie má fyzikálně dané hranice: pod jisté rozměry se obvody zmenšit nemohou. Rovněž stávající principy zpracování informací založených na binární podobě informace využívající jako jednotku informace Bit. Problém se dá obejít několika způsoby, ale tím nejdůležitějším je snaha o vývoj tzv. kvantových počítačů.

Současně existují koncepty počítačů vyvíjených na jiné, než křemíkové bázi, např. uhlíkové počítače či organické počítače, membránové počítače nebo dokonce počítače na bázi DNA.

2.4.1 QUBIT

Kvantový bit (Qubit) je jednotka kvantové informace. U všech současných (klasických) počítačů je základní jednotka informace „bit“, jehož stav je definován hodnotou 1 nebo 0. Posloupnost několika qubitů nazýváme kvantový registr. Osm spojených qubitů tvoří qubyte.

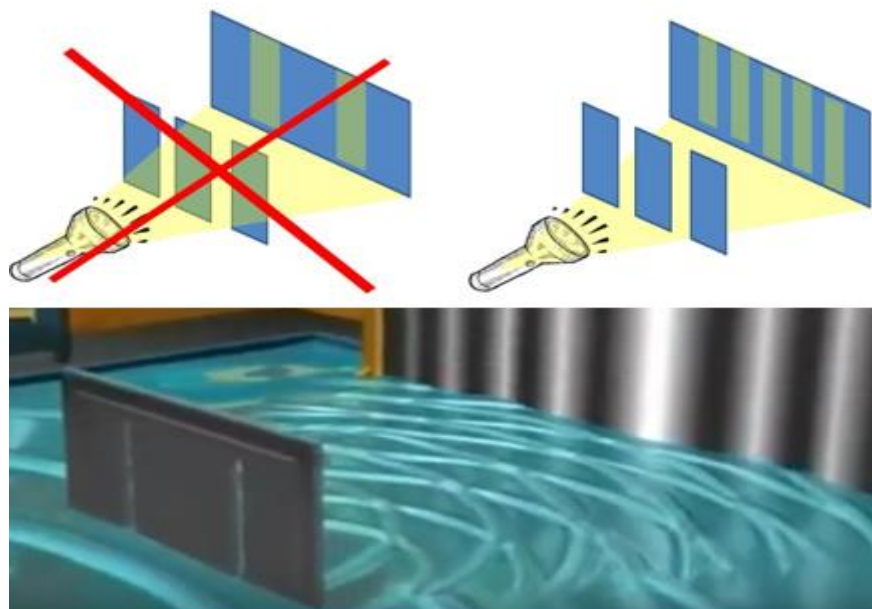
Kvantový bit

Qubit může nabývat hodnot 1 a 0, ale současně i libovolné hodnoty mezi tím, tedy stavy „ani 1 ani 0“. Logické stavy qubitu se proto označují symbolem $|0\rangle$ a $|1\rangle$. Qubit zahrnuje navíc všechny superpozice $a|0\rangle + b|1\rangle$. Superpozice znamená, že nelze jednoznačně rozhodnout, zda má hodnotu 0, nebo 1, uvažujeme pak, že má současně hodnotu 1 i 0, kvantový stav může nabývat zároveň všech svých hodnot. Konkrétní hodnotu $|0\rangle$, nebo $|1\rangle$ nabude teprve v okamžiku měření.

Někteří autoři demonstrují kvantový stav na imaginárních číslech, kdy, zjednodušeně řečeno, číslo má dvě složky, reálnou, udávající jeho reálnou hodnotu a imaginární, udávající jeho umístění v prostoru.

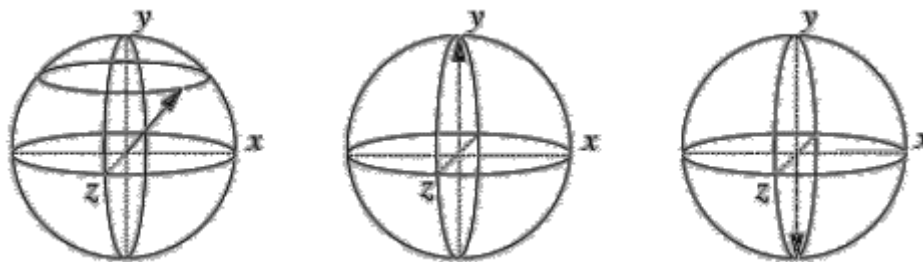
Základní práce s kvantovými informacemi je založena na neurčitosti informace a kvantových jevech plynoucích z vlnové povahy kvantových jevů. Pro pochopení kvantové povahy je uváděn tzv. dvouštěrbinový experiment, kdy je dokázáno, že světlo, které prochází dvěma štěrbinami, není na stínítku soustředěno do dvou míst za jednotlivými štěrbinami, ale je díky interferenci rozloženo do více míst (viz obrázek 5).

Klasický bit má dva stavy – 0 nebo 1. S nimi pracují tranzistory, které napětí propouští nebo nepropouští. Qubit funguje na stejném principu, jeho hodnota je ale uložena v kvantových vlastnostech částice. Vzhledem k tomu, že částice může mít 2^N kvantových stavů (superpozice), lze pro data využít obrovské množství prostoru. Oproti klasickému chápání informace je zde ale vlastnost kvantové mechaniky – až do okamžiku měření se přenáší oba stavy – tedy nula i jednička – zároveň.



Obrázek 17 Kvantová teorie, vlnová povaha informace, interference

Dvoustavový systém, který představuje kvantový bit, je možné názorně zobrazit jako vektor v tzv. Riemannově kouli s umístěním v počátku souřadnic (viz obrázek 6). V něm je $|1\rangle$ reprezentována jako vektor směřující k severnímu pólu, $|0\rangle$ k jižnímu. Z úhlu, který vektor svírá, se svislou osou je možné vyčíst poměrné zastoupení $|1\rangle$ a $|0\rangle$ ve stavovém vektoru. Úhel, o který je vektor otočen kolem svislé osy se nazývá fáze, která nemění poměr $|1\rangle$ a $|0\rangle$, ale je významná vzhledem k jevu kvantové interference.



Obrázek 18 Repräsentace dvoustavového systému: na obrázku vlevo je znázorněn doposud neměřený stav Qubitu, v němž jsou v určitém poměru zastoupeny stavy $|1\rangle$ a $|0\rangle$, na prostředním obrázku je pak vlastní stav $|1\rangle$ a vpravo stav $|0\rangle$

2.4.2 KVANTOVÝ POČÍTAČ

Klasický počítač pracuje tak, že algoritmus řeší jako konečnou sekvenci příkazů realizovaných jako posloupnost instrukcí, sekvenční přístup znamená řešení problému stylem krok-za-krokem. Kvantový počítač postupuje sekvenčně, krok je však realizován kvantovým výpočtem.

Klasický počítač musí při výpočtu či řešení algoritmu čekat na úplné informace na vstupu, aby mohl začít řešit úlohu, kvantový počítač má vstupy neurčité, proto má počáteční stav úlohy teoreticky již v určité fázi řešení, čímž je výpočet složitých algoritmů mnohonásobně rychlejší než u nejrychlejších klasických počítačů. Neurčitost však současně znamená, že nelze zjistit hodnoty průběhu výpočtu, protože bychom z neurčité hodnoty vytvořili určitý stav a tím bychom zneplatnili kvantový výpočet.

Mezi základní operace kvantových počítačů patří kvantová interference qubitů, což je vzájemné ovlivňování stavu qubitů, kdy dojde k vzájemnému vyrušení stavu qubitů, nebo naopak k zesílení pravděpodobnosti jejich stavu. Další operací je kvantové provázání (propletení) qubitů (entanglement), což je forma korelace mezi částicemi, kdy dvě provázané částice vzájemně přejímají svoje vlastnosti.

Praktická realizace je poměrně komplikovaná, existují dohady, zda realizované kvantové počítače jsou reálnými kvantovým počítači nebo pouze simulací a jsou to pouhé emulátory.

První 2 qubitový kvantový procesor na světě fungující na bázi elektroniky sestrojili v roce 2009 vědci, vedení profesorem Schoelkopfem z Yale University, jednalo se o konkrétní model, nikoli matematickou reprezentaci.

Mezi první počítače označované za kvantové patřil 5 qubitový počítač společností IBM z roku 2000 určený pro testovací účely této technologie. V roce 2007 existoval už 16 qubitový počítač a v roce 2011 byl představen první, 128 qubitový, komerční, kvantový počítač D-Wave One, jehož procesor je vyrobený ze supravodivých kovů. V roce 2013 došlo ke

komerčně průlomovému kroku, kdy americká vesmírná agentura NASA koupila ve spolupráci s Googlem počítač firmy D-Wave,

Stále však přetrvávaly pochybnosti o výkonu a využitelnosti, odborníci se neshodli, zda opravdu jde o první praktický příklad této technologie nebo zda se jedná o emulaci. Počítač stojí 15 milionů dolarů.

Při testování výkonnosti, kdy byl kvantový počítač D-Wave s 442 qubitovým čipem (V5, Vesuvius) výkonnostně srovnáván se sedmi pracovními stanicemi Lenovo D20 se čtyř jádrovými procesory Xeon X5550 na frekvenci 1,6 GHz a 16 GB operační paměti, byl klasický počítač složený z Xeonů a běžného testovacího softwaru s operačním systémem Linux i pětikrát rychlejší, kvantový zase našel více tzv. „dobrých řešení“.

**Kvantový
počítač D-
Wave**

Při specializovaných úlohách optimalizace byl kvantový počítač 3 600x rychlejší, než klasická počítačová sestava. Nutno však dodat, že kvantový počítač byl složený z tzv. optimalizátoru, který disponoval dvěma čtyř jádrovými Xeony, starajícími se o propojení a zpracování dat z kvantového čipu.

Následný model D-Wave Two obsahoval kvantový čip s 512 Qubity. V roce 2015 generace D-Wave 2X disponovala kvantovým čipem s více než tisícovkou Qubitů. Díky tomu dokázal počítač v jednu chvíli analyzovat až 2^{1000} možností, což je více možností, než je počet částic ve známém vesmíru.

Podle informací z roku 2017 je připravený vyrobit reálný Qubitový počítač (nikoli emulaci) i Microsoft.⁴ Ze stejného zdroje pochází i informace, že IBM sestrojilo 50 qubitový procesor, který běží při teplotě $-273,135\text{ }^{\circ}\text{C}$.

⁴ Dostupné z <https://vtm.zive.cz/clanky/microsoft-jsme-pripraveni-vyrobir-prvni-quantovy-pocitac/sc-870-a-185007/default.aspx>



Obrázek 19 Kvantový počítač



PRO ZÁJEMCE

Pro doplnění informací o kvantových počítačích a Qubitu doporučuji <https://vyvoj.hw.cz/teorie-a-praxe/trendy/kvantovy-pocitac-princip-funkce-mozne-pouziti.html>, dále <http://vtm.e15.cz/kvantovy-pocitac-se-utkal-s-dnesnimi-procesory-zvitezil> nebo např. https://technet.idnes.cz/kvantovy-pocitac-za-ctvrt-miliardy-d6e-/veda.aspx?c=A130517_115931_veda_mla

Informace o druhé řadě kvantového počítače Dave jsou dostupné např. na <https://www.zive.cz/clanky/d-wave-2x-nova-generace-quantoveho-pocitace/sc-3-a-179417/default.aspx>

Dvou štěrbinový experiment (Youngův experiment) lze nalézt např. na <https://www.youtube.com/watch?v=ODLRz-0KBBU>, odkud je rovněž použita část demonstrativního obrázku.



K ZAPAMATOVÁNÍ

Kvantová koherence – zabývá se myšlenkou, že všechny objekty mají vlastnosti podobné vlnám. V případě, že je vlnová povaha objektu rozdělena do dvou vln, mohou obě

vlny vzájemně (koherentně) interagovat tak, že vytvoří jediný stav, který bude superpozicí jejich dvou stavů.

Kvantová superpozice – tvoří základní princip kvantové mechaniky. Uvádí, že libovolné dva či více kvantových stavů lze sčítat, čímž vznikne nový, platný kvantový stav.

Kvantová provázanost – jedná se o fyzikální jev, ke kterému dochází, když jsou skupiny částic generovány, interagují nebo sdílejí prostorovou blízkost tak, že kvantový stav každé jedné částice nemůže být popsán nezávisle na stavu ostatních částic.

Kvantový počítač vs. klasický počítač: základní rozdíly

- Klasické počítače používají bity ve formě tranzistorů, které nabývají hodnot 1 nebo 0. Kvantové počítače používají Qubity (proton, elektron), které mohou nabývat hodnot 1, 0 nebo současně obou hodnot v takzvaném stavu superpozice.
- Kvantové počítače jsou díky vlastnostem kvantové mechaniky schopny některé problémy počítat mnohem rychleji než klasické počítače (např. kryptografické problémy).
- Klasické počítače mohou na rozdíl od kvantových libovolně kopírovat data.
- Klasické počítače pracují ve své podstatě bez chyb. Kvantové počítače mohou při výpočtech nebo provádění operací chybovat, a je proto nutné neustále kontrolovat správnost prováděných operací.

2.4.3 DALŠÍ POČÍTAČOVÉ TECHNOLOGIE

Optické počítače a nanotranzistory

Současné křemíkové procesory pracují s elektrickou energií. Jejich efektivita ale není dokonalá, takže vznikají velké ztráty v podobě přeměny na teplo. Vzhledem k tomu, že stávající technologie pracují na základě pohybu elektronů, je jedním z důvodů zmenšování čipů minimalizace drah, které po kterých se elektrony pohybují. Stávající technologie pro výrobu tranzistorů se pohybují na úrovni 10 nanometrů. I přes technologická omezení lze předpokládat zvládnutí výroby extrémně malých tranzistorů, které pracují s jedním nebo dvěma elektrony, v současnosti existuje experimentální model s tranzistorem 1,5 nm, který pracuje pouze s jediným elektronem. Jedním z velkých omezení je využívání metalického spojení počítačových komponent, které začíná být vzhledem k propustnosti nedostatečné. Logicky tedy vyvstala otázka, zda místo zmenšování vzdáleností nelze zvýšit rychlost pohybu, tedy zvýšit rychlosti přenosu signálu. Protože nejrychleji pohybující se částicí je foton (u fotonů dosahuje 300 tisíc kilometrů za sekundu, u elektrického proudu ve vodiči se

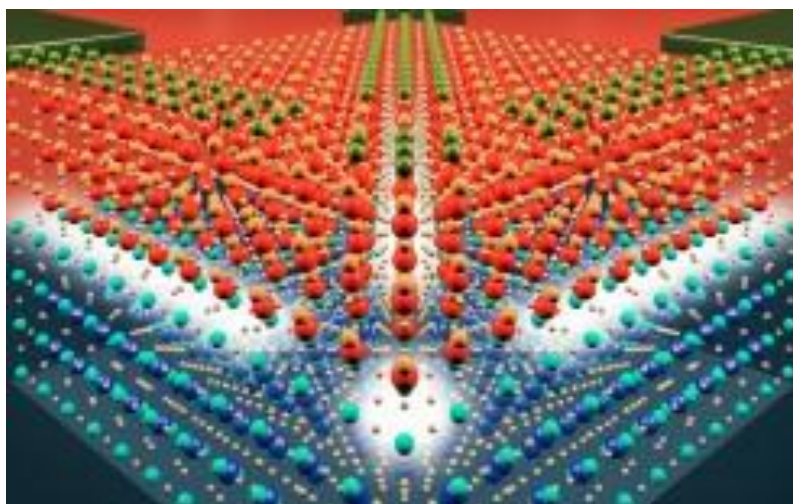
tomuto číslu blíží), nabízí se konstrukce fotonových, resp. optických počítačů. Princip optických počítačů je obdobný principu elektronických počítačů, pouze se pohybují částice světla, fotony. Další výhodou optických systémů je oproti elektronickým dále lepší propustnost cest, menší energetické ztráty a tepelné zatížení, které může dále elektrony zpomalovat. U optických, respektive optoelektronických systémů je největším problémem, že na rozdíl od elektronů fotony vzájemně nereagují. Z toho důvodu je obtížnější se světlem pracovat. Řešení, které by bylo založeno čistě na optické bázi, by využívalo světelné čipy pracující pouze s fotony; včetně přenosu dat mezi dalšími částmi počítače. To je však v současnosti komplikované, např. z důvodu praktického omezení daného velikostí součástek, které světlo směřují, i když nanolasery a nanotranzistory jsou známy od počátku století.

Vývoj v oblasti optoelektroniky se v současnosti proto orientuje na napodobení klíčových součástek elektronických počítačů. Společnost Fujitsu se podařilo úspěšně vyrobit hybridní řešení⁵, kdy na křemíkový čip navazují optické cesty. Křemíková fotonika tedy funguje tak, že v elektronických obvodech nepřenáší informaci elektrony, ale fotony, zpracování je však prováděné prostřednictvím elektronů. Fotonické mikročipy představují, podle zástupců firmy Fujitsu, obrovský skok ve výpočetní technice.

Tvorba plně fotonových počítačů pokročila, základní technologie a komponenty se podařilo laboratorně napodobit a vyrobit. Protože základem stávající elektroniky je tranzistor, proto se pozornost soustřeďuje hlavně na něj. Princip tranzistoru spočívá v tom, že dva proudy elektronů spolupracují na vytvoření jednoho výsledného proudu na základě pravidel logiky.

*Tranzistor
na úrovni
práce s
elektronem*

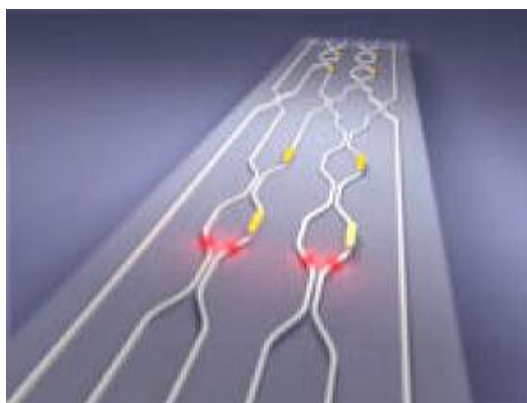
Na následujícím obrázku je schéma tranzistoru pracujícího na úrovni elektronu, zjevné jsou tři vodiče (zeleně), které směřují do prostřední zelené části. Podle stavu vodičů dojde k přechodu nebo naopak k izolaci elektronu přes středový blok z jednoho vodiče na druhý.



Obrázek 20 Průřez tranzistorem 1,5 nm. Zdroj: University of Pittsburg

⁵ Čerpáno z <http://vtm.e15.cz/fujitsu-vyrobilo-opticke-spojenci-pro-kremikove-cpu>

Jak bylo zmíněno, fotony nedokážou vytvářet obdobné interakce. Pokrok v oblasti křemíkové fotoniky, umožnil vývoj polarizačního děliče paprsků (rozdělí tok paprsků světla do dvou samostatných informačních kanálů) o rozměrech 2,4 x 2,4 mikronu. Současně se v roce 2007 na Harvardské univerzitě podařilo nasimulovat optický tranzistor („optistor“) pomocí polovodičových nanovláken, kdy jsou stavy měněny fotonem. K přepnutí stavu stačí podle vědců jediný foton. To by mohlo umožnit stavbu velmi výkonných počítačů, pamětí, elektroniky i materiálů. Dalším problémem optických členů je obousměrný pohyb světla, kdy nelze vyloučit zpětné ovlivnění čipu na výstupu. Řešením jsou usměrňovače a izolátory, které jsou již laboratorně testovány⁶.



Obrázek 21 Grafické znázornění optického obvodu a kvantové provázání procházejících fotonů

Zdroj: University of Bristol, převzato z <http://vtm.e15.cz/kvantovy-svetelny-cip-ktery-lze-preprogramovat>

V současnosti je vyvíjen i kvantový optický čip.

PRO ZÁJEMCE



Informace o nanotranzistorech lze čerpat z <http://vtm.e15.cz/nanotranzistor-zpusobi-prevrat-v-elektronice>, informace o optických a dalších technologiích si můžete dále rozšířit na <https://www.cnews.cz/budoucnost-pocitacu-x-laserove-quantove-a-dna/>

Dále doporučuji zájemcům prostudovat např. <https://cdr.cz/clanek/prulom-v-oblasti-kremikove-fotoniky-slibuje-az-milionkrat-rychlejsi-pocitace> nebo <http://vtm.e15.cz/dulezity-milnik-pro-tvorbu-svetelnych-cipu-prekonan>

O kvantovém optickém čipu lze zjistit informace např. na <http://vtm.e15.cz/kvantovy-svetelny-cip-ktery-lze-preprogramovat>

Chemické a biologické počítače, DNA počítače

⁶ Blíže např. v <http://vtm.e15.cz/dulezity-milnik-pro-tvorbu-svetelnych-cipu-prekonan>

Specifické místo mezi výpočetními technologiemi zaujímají chemické a biologické počítače. U nich většinou není primární výkon, ale dokonalé přizpůsobení prostředí, ve kterém mají fungovat. Neřeší zpravidla konkrétní algoritmy ale spíše je podstatná odezva na vstupy, obdobně, jako tomu bylo u analogových počítačů. Chemické počítače fungují principu, kdy molekuly polymerů, mění svou vnější podobu na základě chemických reakcí, biologické procesy jsou určené například pro výpočty postavené na buněčných metabolických procesech. Chemické sloučeniny simulují změny a reakce prostřednictvím složek směsí různých koncentrací, výpočty a výsledky jsou dány koncentrací po smíchání různých poměrů látek. Tímto způsobem nemusí výpočty probíhat jedno vláknově (jako v elektronickém počítači), ale v několika směrech a vrstvách najednou. Fungování takového počítače pak připomíná procesy, probíhající v mozku. Příkladem je chování obarvené kapaliny nalité do jiné, kdy prostupování obou složek připomíná tvorbu mraků nebo růst houby. Klasické počítače pracují se součástkami, uspořádanými do vyrobených obvodů. Chemický počítač v ideálním případě nic takového potřebovat nebude. Rozlitím chemického počítače do dvou nádob vzniknou dva samostatně fungující systémy s menší výpočetní silou. Mnohem blíže realitě jsou biologické počítače, označované také jako molekulární nebo DNA technologie. S nadsázkou se dá říct, že DNA počítače existují miliardy let uvnitř všech živočichů a rostlin. Molekuly DNA jsou v prostředí pro výpočty propojeny do komplexu, který funguje jako řídicí a paměťová složka. V DNA počítačích je jako paměťové médium pro uchování informací i jako procesor sloužící k provádění výpočtů využito molekul nukleových kyselin. V roce 2003 se podařilo izraelským vědcům vyvinout DNA komplex, tedy biologický počítač vytvořeným člověkem. Další počítače DNA v řadě byly označeny MAYA a MAYA-II prozatím zvládají pouze jednoduché algoritmy. Předpokládá se, že systémy založené na DNA by mohly vést ke vzniku nedeterministického chování Turingova stroje a tím by byly dány základy zcela nového typu počítače. DNA počítač lze v současnosti definovat jako systém pro řešení příslušně formalizovaných úloh, zejména takových, které vedou k mnoha řešením a alternativám, z nichž hledáme řešení vyhovující zadaným podmínkám. Jedná se o úlohy obtížně vyčíslitelné, kdy se namísto sofistikovaného algoritmu používá metoda „hrubé síly“. Např. prolamování šifrovacích algoritmů v současnosti znamená testování všech možností. Na DNA počítači budou všechny alternativy reprezentovány různými molekulami DNA, řešením bude mezi nimi výběr jedné, splňující podmínky. Existují vědecké publikace a studie informující o představení komunikačního systému, který dokázal přenášet genetické informace mezi jednotlivými buňkami, který byl předveden týmem pod vedením Jerome Bonneta, současně existují informace o záznamu až 700 TB dat do jednoho gramu DNA.

Biologický tranzistor

V segmentu biologických počítačů lze předpokládat, že do jisté míry může zůstat zachována základní koncepce založená na polovodičových tranzistorech, obdobně jako u optických počítačů by potom zůstal základním stavebním kamenem tranzistor na biologické bázi. Vědcům se již v minulosti podařilo vyvinout klíčové součásti, které jsou nutné pro

stavbu organického počítače, zejména biologický tranzistor (Transcriptor⁷), který dokáže provádět logické operace a zesilovat signál. Pro stavbu počítače v moderním pojetí je nutné mít k dispozici tři hlavní funkce, respektive části. Informace je nutné ukládat (a číst), přenášet je a zpracovat prostřednictvím logických operací. Biologické počítače se vzhledem ke své teoretické velikosti a biologické povaze budou využívat především v medicíně, kdy v kombinaci s nanoroboty mohou významně přispět k novým formám léčby.

Uhlík místo křemíku, nanotrubičky

Již po desetiletí probíhá vývoj počítačových procesorů používáním čím dál menších a hustěji osázených křemíkových tranzistorů. Nespolehlivost těchto součástek při velikostech menších než 10 nanometrů nutí vědce hledat jiné řešení. V současnosti existuje koncept, který namísto křemíkových tranzistorů uhlíkové nanotrubičky o průměru 1-2 nanometrů, tedy jediného uhlíkového atomu. Prototyp obsahoval pouhých 178 tranzistorů (oproti miliardám v současných IT) a zvládal jen velmi jednoduché úkoly. Výhodou uhlíkového počítače jsou malé rozměry dané využitím nanotechnologií a s tím související nízká spotřeba. Samozřejmě, že technologií je daleko více, existují i další možnosti. Každá struktura, která umí pracovat s informacemi, se může stát počítačem svého druhu. Existují studie a projekty s genetickými tranzistor pracující uvnitř buněk či studie membránových počítačů využívajících povrchové napětí kapalin. Souhrnně lze říci, že schopnosti křemíkových počítačů mají své limity, vývoj směřuje do segmentu jiných anorganických materiálů, zejména uhlíku, je snaha využívat, respektive kopírovat živé organismy a využívat duální povahu světla, ať ve formě fotonu nebo ve formě kvantové energie. Vývoj nových materiálů vede jednoznačně k nanotechnologiím.

PRO ZÁJEMCE



Opět si zájemci mohou nastudovat více informací o zmiňovaných technologiích, např. na <http://vtm.e15.cz/biologicky-tranzistor-bude-zaklad-organickeho-pocitace>,

dále na <http://www.stoplusjednicka.cz/uhlik-misto-kremiku-jak-budou-vypadat-pocitace-budoucnosti>,

na <http://vtm.e15.cz/biologicky-tranzistor-bude-zaklad-organickeho-pocitace>,

na <http://www.tyinternety.cz/ostatni/uhlik-ma-brzy-zmensit-a-zrychlit-pocitace-i-internet/>,

na http://technet.idnes.cz/pocitac-z-uhlikovych-nanotrubicek-dtc-veda.aspx?c=A130930_140103_veda_mla,

na <http://zoom.iprima.cz/clanky/pocitac-z-uhlikovych-nanotrubic-ma-byt-rychlejsi-levnejsi-i-chladnejsi>

nebo na <http://zoom.iprima.cz/clanky/pocitac-z-uhlikovych-nanovlaken-je-tady-uz-ted-se-vyrovna-kremikovym-kolegum>

⁷ Blíže je technologie popsána na <http://vtm.e15.cz/biologicky-tranzistor-bude-zaklad-organickeho-pocitace>



SHRNUTÍ KAPITOLY

Kapitola měla představit základní počítačové komponenty a nové trendy a technologie. Je nutné si uvědomit, že nové technologie nejsou jen inovací stávajících, že problém nelze zúžit do nabídky novinek na trhu, například nových řad počítačů či telefonů. Jak bylo zmíněno, nové trendy vycházejí ze tří rovin, Ty nejběžnější inovace jsou dány vývojovými koncepcemi jednotlivých firem, není však jisté, zda se plánované nové modely a technologie osvědčí a budou úspěšné. O tom nás přesvědčila historie firmy Nokia, kdy technologický vývoj a inovace vlastního operačního systému nebyly dostatečné a vedly k zániku firmy. Další rovinou je poptávka zákazníků. Jako příklad inovace lze uvést vznik nového typu IT zařízení, tabletu. Úspěch byl vyvolán vyhodnocením trhu, kdy byla poptávka po mobilních zařízeních. Ty však již na trhu existovaly ve formě PDA a MDA (personální nebo mobilní digitální asistent), byly však příliš drahé a určeny pro firemní klientelu. Apple vyhodnotil poptávku a vyvinul zařízení, které se zaměřilo na poptávku na trhu – zařízení především pro mladou generaci s požadavky na komunikaci, internet, sociální sítě a hry, nikoli tedy na plnohodnotnou kancelářskou či profesionální činnost. Poslední rovinu inovací tvoří produkty a technologie, které jsou buď vyvinuté na základě nových objevů, nebo jsou vyvolány vyčerpáním možností existujících technologií. Jako příklad lze uvést léta používaný princip tvorby světla pomocí plamene, realizovaný žárovkou, který byl téměř zcela vytlačen LED diodami, tedy jinou technologií. V podstatě lze konstatovat, že většina takovýchto inovací vychází z obecně známých principů, nově pojatých a implementovaných. Příkladem je DNA koncept počítače, kdy pravidla uchovávání genetických informací jsou známé poměrně dlouho.



KONTROLNÍ TEST

	Otázka	Odpověď A	Odpověď B	Odpověď C
1	Qubit je	Jednotka kvantové informace	Jednotka biologického počítače	Limitní hodnota bitu
2	Vývoj osobních počítačů směřuje k platformě	Počítačů bez operačního systému	Mini PC	Herních počítačů
3	Intel Optane Memory	Není volatilní	Je volatilní	Je permanentní
4	Mezi rozměry základní desky Mini PC nepatří	170x170 mm	120x102 mm	102x102 mm
5	Procesor Centriq 2400	Je ARM procesor	Je vyráběný firmou Intel	Je grafický procesor
6	Procesor Centriq 2400	Je osazen 2 miliardami tranzistorů	Je osazen 8 miliardami tranzistorů	Je osazen 18 miliardami tranzistorů

7	Intel QuickAssist	Je systém snímko- vání paměti	Je podpora vývoje procesorů Atom	Je integrovaný sys- tém hardwarového šifrování
8	Nejvýkonnějším stan- dardem 802.11 je	802.11 ah	802.11 dh	802.11 ad
9	Starlink je	Cloudová služba pro práci s multi- mediálním obsa- hem	Celosvětová multi- mediální síť pro správu audio vizu- álních aplikací	Celosvětová inter- netová síť, zajišťo- vaná prostřednic- tvím satelitů
10	Rozšířená realita	Mapuje virtuální objekty do reál- ného prostředí	Rozšiřuje virtuální realitu o vjemy a podněty	Je počítačově upra- vená realita pro tvorbu virtuální re- ality

ODPOVĚDI



1-A, 2-B, 3-A, 4-B, 5-A, 6-C, 7-C, 8-C, 9-C, 10-A

3 KOMUNIKACE, SÍŤ



RYCHLÝ NÁHLED KAPITOLY

Tato kapitola obsahuje seznámení se základními službami, dále objasní principy internetu a s tím související základy komunikace v síti včetně protokolů a základních příkazů (ipconfig, tracert, ping, arp) sloužících k zjištění dostupnosti zdroje na Internetu. Budou objasněny pojmy související s DNS serverem, maskou, bránou. Tyto znalosti jsou aplikovatelné pro vzdálenou správu, komunikační služby a v rámci služeb pro přenos a publikaci hypertextů založené na http protokolu (www).



CÍLE KAPITOLY

Cílem kapitoly je seznámit studenty se základními pojmy, se kterými se setkají při práci s Internetem, jako rozsáhlou sítí počítačů. Studenti po prostudování textu budou schopni identifikovat síťové prostředí a základní problémy v komunikaci v rámci Internetu



ČAS POTŘEBNÝ KE STUDIU

V této kapitole je vhodné si uváděné příkazy prakticky odzkoušet. Z tohoto důvodu je čas potřebný ke studiu v rozsahu 2-5 hodin, podle míry praktické práce.



KLÍČOVÁ SLOVA KAPITOLY

Počítačová síť, TCP/IP protokol, ISO/OSI model, maska sítě, brána, IP adresa, služby Internetu, komunikace

3.1 Principy komunikace v sítích

Pro komunikaci (jakoukoliv, ne pouze počítačovou), je nutné zajistit, aby systémy byly kompatibilní, aby komunikovaly podle jednotných pravidel. Pro funkční komunikaci, musí být stanovena jednoznačná pravidla. Tato pravidla jsou dána komunikačním protokolem

(můžete srovnat například s „protokolem společenským“, „hradním protokolem“ apod., definujícími chování ve společnosti či v prezidentské kanceláři).

Komunikační protokol musí jednoznačně ošetřit všechny stavy, ke kterým může v počítačové síti dojít a musí být schopen zabezpečit komunikaci. Komunikační protokol je množina pravidel, která určují syntaxi a význam jednotlivých zpráv při komunikaci. Je zřejmé, že musí být univerzální, zejména s ohledem na rozmanitost počítačových systémů komunikujících v prostředí Internetu, počínaje různými operačními systémy (Windows, Linux, IOS, Android apod.), různých hardwarových platform (PC, Apple, tablety, Smartphone apod.), či nepřeberným množstvím komunikujícího software atd.

3.1.1 KOMUNIKAČNÍ PROTOKOLY

Je tedy zřejmé, že nastavení komunikačního protokolu je zásadní a podstatná věc, která zabezpečuje komunikaci a funkčnost služeb v síti, potažmo v Internetu. Porovnejme nejdříve prostředí, se kterým se setkáte na OPF. Příklady jsou opět zjednodušené, slouží pro pochopení principů.

IPX/SPX

Každý student OPF má účet Novell, prostřednictvím kterého se přihlašuje do fakultní sítě. V této síti má k dispozici přidělené prostředky (disk L, disk K apod. tiskárny) a přidělená práva (na disku L může zapisovat, mazat, vytvářet adresáře, na disku K může prohlížet obsah, na vybraných tiskárnách může tisknout apod.). Do sítě se student přihlašuje svým jménem a heslem, aby uživatel mohl v síti pracovat, musí mít vytvořený účet s právy.

Jedná se tedy o síť centrálně spravovanou, autentizace je na úrovni uživatelů a prostředků vytvářených správcem. Tato síť funguje na protokolu IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange) což je sada protokolů používaná síťovým operačním systémem Novell NetWare a používá se zejména pro komunikaci typu klient/server⁸.

Výhodou takovéto sítě je bezpečnost a organizovanost, nicméně použití této sítě jako základu Internetu by bylo nereálné, protože by každý uživatel musel mít přihlašovací údaje a práva k přiděleným prostředkům.

NetBIOS

Jiný typ sítě, se kterým se studenti na OPF setkávají, je síť Microsoft. Tato síť je založena na protokolu NetBIOS (resp. NetBEUI - NetBIOS Extended User Interface). Protokol je určen především pro komunikaci v malých lokálních sítích, způsob adresování nepodporuje routování. Není tedy možné spojit více sítí bez nutnosti použití brány. Zjednodušeně

⁸ Architektura klient/server je blíže popsána v kapitole 4

se dá říct, že každý uživatel rozhoduje o prostředcích, které nabídne ostatním a přiděluje jejich práva. Počítač je v síti identifikovaný jménem, může být vázaný na skupinu počítačů. Základním problémem je nekoordinovanost při přidělování jména počítače, kdy jméno je přidělováno při instalaci operačního systému. V síti potom může docházet k nejednoznačnosti v názvech počítačů, kdy více počítačů má shodný název („doma“, „mamka“, „Jirka“ apod.). V dnešní době tento protokol nemá větší využití, nicméně slouží většinou ke sdílení tiskáren a souborů v síti menšího rozsahu. Protokol je charakteristický vysokou přenosovou rychlostí a jednoduchostí konfigurace. NetBEUI v současnosti umožňuje pracovat nad protokolem TCP/IP, čímž dochází k rozložení komunikace mezi tyto protokoly a zefektivnění síťového provozu.

Tento protokol by byl z pohledu správy sítě pro Internet vhodný, nevyžaduje centrální správu, uživatelé rozhodují o tom, co nabídnou ostatním. Je však zcela nepoužitelný z důvodu nekoordinovaného přidělování jmen počítačů⁹ a malé bezpečnosti.

TCP/IP

V dnešní době použití IPX/SPX a NetBIOS značně kleslo, díky univerzálnosti TCP/IP protokolu, (Transmission Control Protocol / Internet Protocol – „primární přenosový protokol/protokol síťové vrstvy“), který obsahuje sadu protokolů pro komunikaci v počítačové síti. Síťová komunikace je rozdělena do tzv. vrstev. Výměna informací mezi vrstvami je přesně definována. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší.

V rámci tohoto protokolu je adresace prováděna na úrovni síťového rozhraní prostřednictvím IP adresy. Přidělování této adresy je vázané pravidly, část adres je přidělována, část může být distribuována libovolně (obdobně jako jméno počítače v síti Microsoft) za dodržení určitých pravidel. V současnosti existuje TCP/IP ve verzi 4 a 6. V těchto skriptech se z důvodu snazšího pochopení zaměříme na IP verze 4.

V současnosti je tedy nutné, aby každý počítač, který chce přistupovat do sítě internet, komunikoval pomocí protokolu TCP/IP.

3.2 Konfigurace připojení

Volbu protokolu, služeb a klientských aplikací v prostředí Windows¹⁰ provádíme v Centru síťových připojení, v nastavení adaptéru vybereme příslušné síťové rozhraní a ve vlastnostech můžeme nastavit parametry připojení.

Základní parametry¹¹, pro které musíme při použití TCP/IP protokolu upřesnit nastavení jsou:

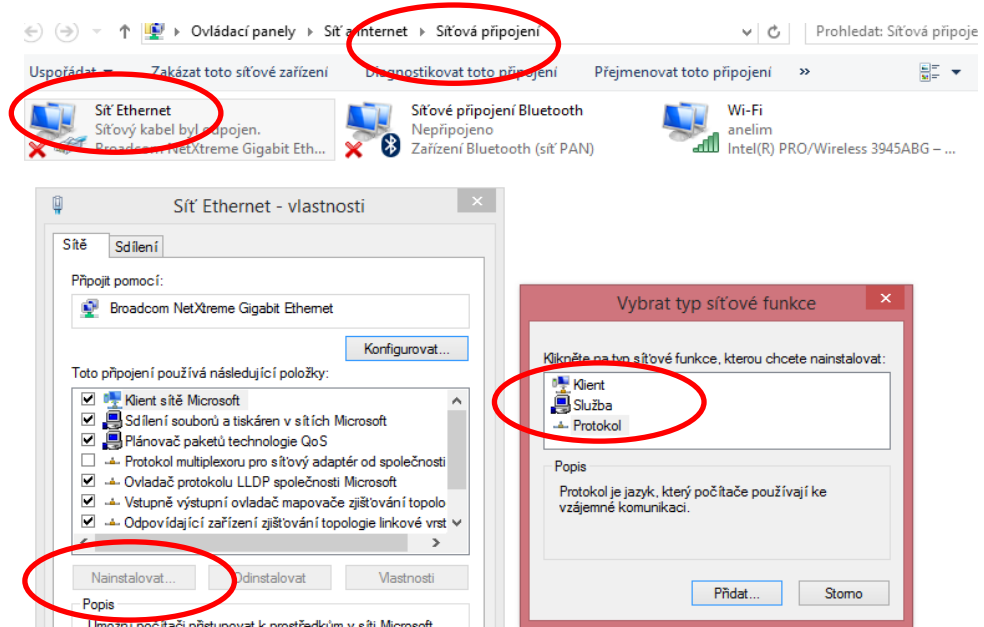
- IP adresa počítače
- Síťová maska (Net Mask)

⁹ Účelově zjednodušeno na základě peer to peer sítě Microsoft

¹⁰ Vzhledem k softwarovému vybavení učeben OPF a dominantnímu postavení tohoto operačního systému v segmentu PC uvádíme příklady nastavení v OS Windows

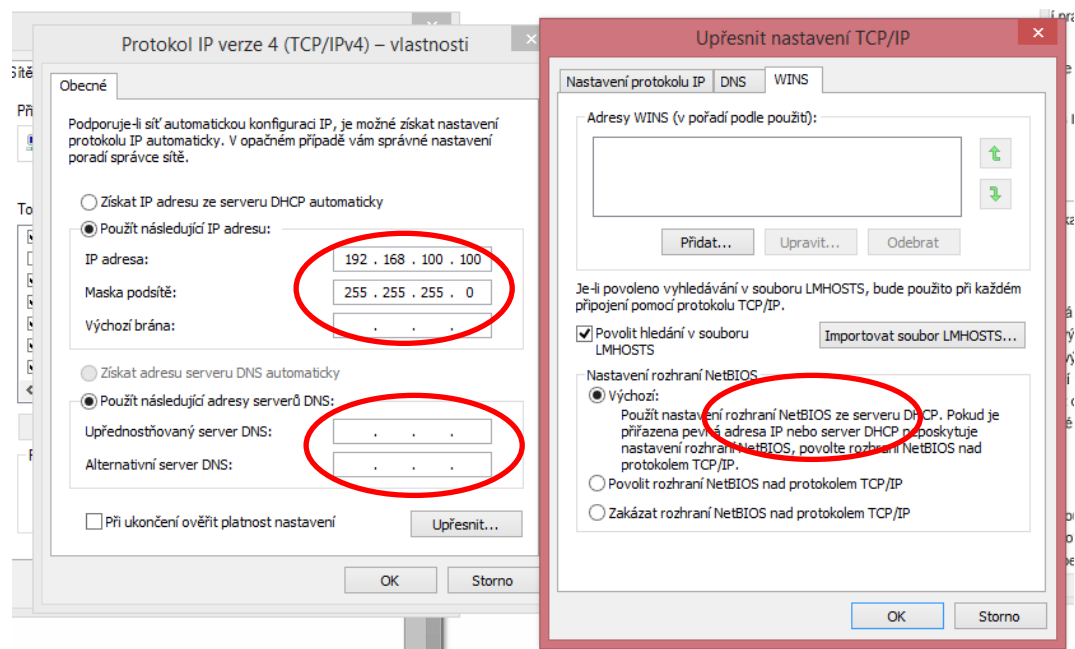
¹¹ Převzato s úpravami z <http://www.gvp.cz/local/new/ucebnice/vyptech/internet/tcpip.htm>

- Výchozí brána (Default Gateway)
- IP adresa Domain Name Serveru (DNS)
- IP adresa Domain Name Serveru (DNS)



Obrázek 22: Nastavení síťových připojení v MS Windows

Zdroj: Vlastní zpracování



Obrázek 23: nastavení IP adresy, masky, brány a DNS, povolení NetBIOS

Zdroj: Vlastní zpracování

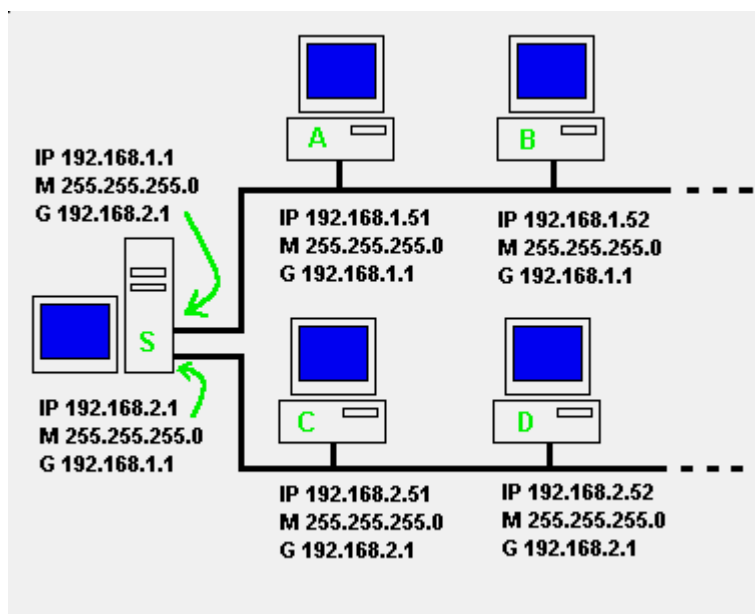
Nastavení může provádět buď uživatel, je-li povolena DHCP služba, lze použít vzdálenou konfiguraci prostřednictvím DHCP serveru.

IP adresa verze 4 je čtyřbajtové číslo, které identifikuje síťovou kartu počítače v rámci protokolu TCP/IP. Zapisuje se ve formátu čtyř čísel oddělených tečkou např.: 192.168.100.1. Rozsah je od 0.0.0.0 do 255.255.255.255. Tato čísla přiděluje správce systému podle pokynů organizace IANA (Internet Assigned Numbers Authority).

Internet je spojením mnoha menších místních a regionálních sítí. Tyto sítě jsou propojeny pomocí bran (Gateway). Brána je hardwarový prostředek (např. počítač), který obsahuje více síťových rozhraní, propojuje více sítí a předává data mezi těmito sítěmi.

Net Mask (síťová maska), určuje rozmezí adres, které patří do místní sítě. Opět zjednodušeně, maska obsahuje hodnoty 255 nebo nula, IP adresy, které se shodují na pozicích, kde je v síťové masce nula (přičemž nuly se doplňují postupně zprava), patří do stejné místní sítě¹². Má-li počítač nastavené IP=194.168.1.1 a Net Mask=255.255.255.0, všechny adresy začínající 194.168.1. patří do stejné lokální sítě. Na posledním místě mohou mít cokoliv. Pokud má PC v lokální síti komunikovat s PC mimo tuto síť, komunikuje prostřednictvím brány.

Na obrázku je znázorněna situace, kdy počítač označený S propojuje síť 192.168.1.* se sítí 192.168.2.*



Obrázek 24: Brána počítačové sítě

Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://www.gvp.cz/local/new/ucebnice/vyptech/inter-net/tcpip.htm>

¹² Velmi zjednodušený příklad, ve skutečnosti maska může mít i jiné hodnoty, které definují rozsah a počet povolených adres, toto je však obsahem jiných předmětů.

Síť na obrázku je složena ze dvou podsítí (místních sítí). Pokud chce počítač A komunikovat s počítačem B, na základě znalosti obou IP adres a své síťové masky zjistí, že počítač B je ve stejné podsíti a proto mu může poslat zprávu přímo.

Když chce počítač A komunikovat s počítačem C, na základě znalosti IP adres a síťové masky zjistí, že počítač C není ve stejné podsíti a proto směřuje svou komunikaci na bránu (tedy na komunikaci směrem z podsítě ven). Brána zabezpečí komunikaci s druhou podsítí, tedy s počítačem C.

DNS - Domain Name Server (Domain Name System, Domain Name Services) zajišťuje překlad číselných IP adres na doménové adresy. Např. `www.opf.slu.cz = 193.84.209.5`. Každý počítač musí znát DNS, jinak by nemohl používat adresaci pomocí doménových adres. Pokud nezná PC adresu DNS serveru, může komunikovat pomocí IP adres, nemůže komunikovat pomocí doménových adres. Fungují tedy základní služby Internetu. IP adresa DNS serveru i IP adresa počítače může být přidělována vzdáleně prostřednictvím DHCP.

3.2.1 PŘÍKAZOVÝ ŘÁDEK A ZÁKLADNÍ PŘÍKAZY

Pro základní práci s protokolem TCP/IP a kontrolu funkčnosti komunikace můžete použít příkazů zadávaných v tzv. příkazovém řádku (podle verze OS se jedná o příkaz `CMD`, `command` apod.). Příkazy se zadávají ve tvaru **PŘÍKAZ**, *mezera*, *oddělovač*, *parametry*. Oddělovačem je znak „-“, nebo „/“. Parametrů (atributů) může být více. Parametr „?“ zobrazí nápovědu k příkazu. V prostředí Windows nezáleží na velikosti znaků (malá, velká písmena).

Například příkaz ***ping -4 www.volny.cz*** má parametr „4“, který udává, že se má pracovat s IP adresami verze 4. Příkaz zkouší dostupnost adresy `www.volny.cz`.

- ***Ipconfig*** - vypíše aktuální konfiguraci TCP/IP počítače, nastavení všech adaptérů dostaneme použitím příkazu `ipconfig /all`.
- ***Ping*** – příkaz ***ping adresa*** (IP nebo doménová) vyšle k zadanému počítači požadavek na odpověď a vypíše, zda a za jak dlouho je tento počítač dostupný a kolikrát přešla komunikace přes aktivní síťový prvek (router).
- ***Tracert*** - funguje podobně jako Ping, ale navíc vypisuje i všechny brány, kudy zpráva prošla.
- ***Nslookup*** – příkaz zjišťující IP adresu na základě DNS

Následující obrázek ukazuje výpis příkazu ***ipconfig /all***.

Na výpisu je vidět používané síťové adaptéry (Intel Wireless), fyzická adresa adaptéru a přidělené IP adresy verze 6 a 4.dále je zobrazena maska a adresa brány a DNS serveru.

Rovněž je zřejmé, že kooperuje TCP/IP protokol a NetBIOS.

```

C:\Windows\system32\cmd.exe
Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) PRO/Wireless 3945ABG - síťové připojení
Physical Address. . . . . : 00-18-DE-BF-DA-45
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv6 Address. . . . . : 2a00:1028:83ac:a02:b87c:12cf:6449:3d1b(Preferred)
Temporary IPv6 Address. . . . . : 2a00:1028:83ac:a02:85a5:62a:352f:a93(Preferred)
Link-local IPv6 Address . . . . . : fe80::b87c:12cf:6449:3d1b%4(Preferred)
IPv4 Address. . . . . : 192.168.10.3(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 28. září 2014 7:47:55
Lease Expires . . . . . : 1. října 2014 20:59:03
Default Gateway . . . . . : fe80::1%4
DHCP Server . . . . . : 192.168.10.1
DHCPv6 IAID . . . . . : 192.168.10.1
DHCPv6 Client DUID. . . . . : 67115230
DNS Servers . . . . . : fe80::1%4
Primary WINS Server . . . . . : 192.168.10.1
Secondary WINS Server . . . . . : 192.168.10.1
NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter Síť Ethernet:
Media State . . . . . : Media disconnected
  
```

Obrázek 25: Výpis nastavení síťového adaptéru příkazem ipconfig

Zdroj: Vlastní zpracování

Parametry (atributy) jednotlivých příkazů můžeme vypsát pomocí parametru „-?“. Pokud je parametr vyžadován, vypíše se seznam parametrů u některých příkazů i při zadání příkazu bez uvedení požadovaných parametrů.

```

C:\Windows\system32\cmd.exe
C:\Users\mobil\000>ping -?
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v IOS]
           [-r count] [-s count] [[-j host-list] | [-k host-list]]
           [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
           [-4] [-6] target_name

Options:
-t          Ping the specified host until stopped.
           To see statistics and continue - type Control-Break;
           To stop - type Control-C.
-a          Resolve addresses to hostnames.
-n count   Number of echo requests to send.
-l size     Send buffer size.
-f          Set Don't Fragment flag in packet (IPv4-only).
-i TTL     Time To Live.
-v IOS     Type Of Service (IPv4-only. This setting has been deprecated
           and has no effect on the type of service field in the IP
           Header).
-r count   Record route for count hops (IPv4-only).
-s count   Timestamp for count hops (IPv4-only).
-j host-list Loose source route along host-list (IPv4-only).
-k host-list Strict source route along host-list (IPv4-only).
-w timeout Timeout in milliseconds to wait for each reply.
-R          Use routing header to test reverse route also (IPv6-only).
           Per RFC 5095 the use of this routing header has been
           deprecated. Some systems may drop echo requests if
           this header is used.
-S srcaddr Source address to use.
-c compartment Routing compartment identifier.
-p          Ping a Hyper-V Network Virtualization provider address.
-4         Force using IPv4.
-6         Force using IPv6.
  
```

Obrázek 26: Parametry příkazu ping

Zdroj: Vlastní zpracování

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. Ušechna práva vyhrazena.

C:\Users\mobil_000>ping www.volny.cz

Pinging www.volny.cz [2a00:da80:0:500::56] with 32 bytes of data:
Reply from 2a00:da80:0:500::56: time=15ms
Reply from 2a00:da80:0:500::56: time=15ms
Reply from 2a00:da80:0:500::56: time=16ms
Reply from 2a00:da80:0:500::56: time=15ms

Ping statistics for 2a00:da80:0:500::56:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 15ms, Maximum = 16ms, Average = 15ms

C:\Windows\system32\cmd.exe
C:\Users\mobil_000>ping -4 www.seznam.cz

Pinging www.seznam.cz [77.75.72.3] with 32 bytes of data:
Reply from 77.75.72.3: bytes=32 time=15ms TTL=245
Reply from 77.75.72.3: bytes=32 time=16ms TTL=245
Reply from 77.75.72.3: bytes=32 time=15ms TTL=245
Reply from 77.75.72.3: bytes=32 time=14ms TTL=245

Ping statistics for 77.75.72.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 14ms, Maximum = 16ms, Average = 15ms

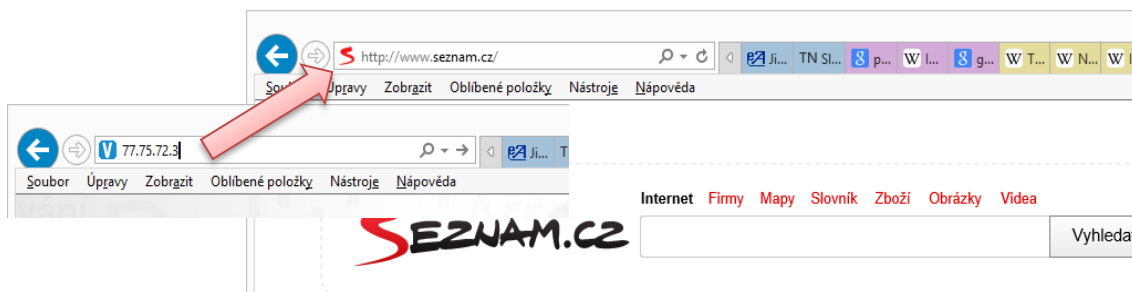
C:\Users\mobil_000>

```

Obrázek 27: Příkaz ping

Zdroj: Vlastní zpracování

Předchozí obrázek ukazuje výpis příkazu ping, v prvním případě bez parametru. Je zjišťována dostupnost počítače s doménovou adresou www.volny.cz a www.seznam.cz. Pokud zadáme příkaz s parametrem „4“ vypíše se IPv4, jinak IPv6 (pokud je dostupná). Protože je dostupný DNS server, dojde k překladi na IP adresu (u verze IPv4 77.75.72.3). Současně se vypíše čas dostupnosti a životnost dat (datagramu) která je dána hodnotou TTL (zjednodušeně, čím vyšší hodnota TTL tím delší čas má příkaz na nalezení hledaného počítače (Hodnota udává počet povolených přechodů přes aktivní síťové prvky. Že je doménová adresa v podstatě „převlečenou“ IP adresou si můžeme ověřit tak, že zjištěnou IP adresu zadáme jako adresu stránky do prohlížeče.



Obrázek 28: IP adresa zadaná do prohlížeče

Zdroj: Vlastní zpracování

Jak bylo zmíněno, počítač při směrování požadavků nemůže znát doménové adresy, ty se evidují na tzv. DNS serverech. Počítač, zjednodušeně řečeno netuší, jakou IP adresu ve skutečnosti má například počítač, na kterém je provozován www server s doménou www.opf.slu.cz. Obrátí se proto na DNS server, který obsahuje rozsáhlou databázi IP adres na internetu, a ten mu ji sdělí, pokud ji DNS server nezná, zjistí si ji od dalšího DNS serveru na Internetu. Windows v příkazovém řádku pomocí příkazu nslookup a adresy serveru (například nslookup www.seznam.cz) zobrazí síťovou komunikaci, která nejprve vedla přes váš DNS server a posléze onen dotazovaný www server.

```
C:\Users\mobil>nslookup www.opf.slu.cz
Server: UnKnown
Address: 192.168.10.1

Non-authoritative answer:
Name: joanes.opf.slu.cz
Addresses: 2001:718:2201:209::7
          193.84.209.7
Aliases: www.opf.slu.cz
```

Obrázek 29: Zjištění IP adresy zadané domény

Zdroj: Vlastní zpracování

Pokud potřebujeme znát „vzdálenost“ počítače (počet přechodů přes aktivní prvky a čas), lokalizaci v příslušné podsíti, resp. podsítě, přes které je komunikace realizována, lze použít příkaz *tracert*.

```
C:\Windows\system32\cmd.exe
C:\Users\mobil_000>tracert -4 www.seznam.cz
Tracing route to www.seznam.cz [77.75.72.3]
over a maximum of 30 hops:
  0  <1 ms    <1 ms    <1 ms    unknown-1-215.opf.slu.cz [193.84.215.11]
  1  <1 ms    <1 ms    <1 ms    193.84.223.30
  2  <4 ms    <3 ms    <3 ms    rka-gw.opf.slu.cz [193.84.223.31]
  3  <2 ms    <2 ms    <1 ms    195.113.175.24
  4  <11 ms   <11 ms   <10 ms   195.113.157.186
  5  <11 ms   <10 ms   <10 ms   r105-r05.cesnet.cz [195.113.156.157]
  6  <11 ms   <10 ms   <11 ms   r45-r94-112.cesnet.cz [195.113.157.1451]
  7  <8 ms    <11 ms   <11 ms   195.113.235.55
  8  <11 ms   <11 ms   <11 ms   r21-pos0-0-stm16.cesnet.cz [195.113.156.114]
  9  <8 ms    <8 ms    <8 ms    nix5.seznam.cz [91.210.16.195]
 10  <8 ms    <8 ms    <8 ms    77.75.75.222
 11  <8 ms    <8 ms    <8 ms    www.seznam.cz [77.75.72.3]
 12  <8 ms    <8 ms    <8 ms

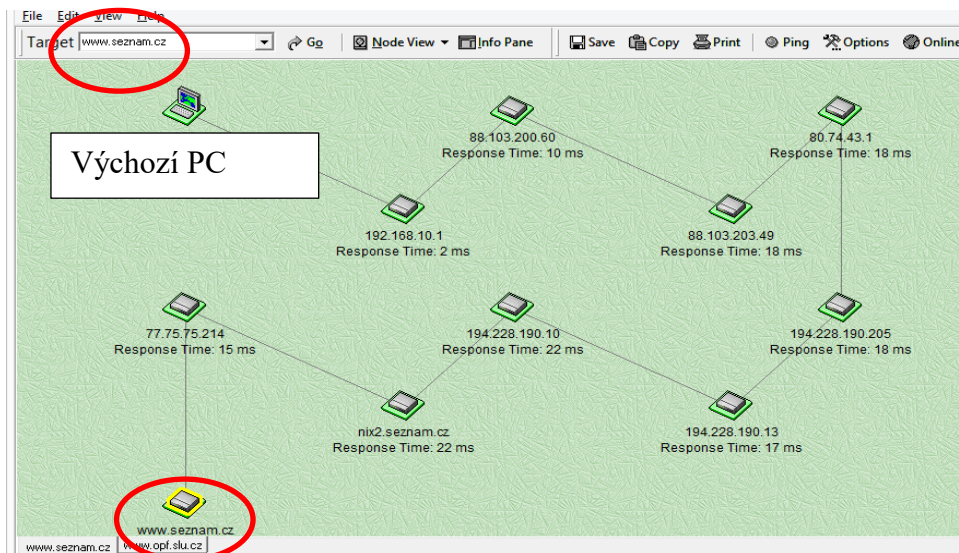
Trace complete.
```

Obrázek 30: Příkaz tracert pro adresu www.seznam.cz

Zdroj: Vlastní zpracování

V praxi existuje nespočet aplikací, které využívají příkaz tracert, dovedou „vytrasovat“ cestu k cíli a zobrazit ji pomocí grafické nástavby. Následující obrázek ukazuje výpis programu NeoTrace Pro. Program dovede zobrazit cestu mezi počítači, časovou dostupnost i informace o síti ve které se příslušný aktivní síťový prvek nachází. Na následujícím obrázku

je výpis aktivních prvků při sledování dostupnosti serveru www.seznam.cz. Dále je ukázán výpis informací o počítači www.opf.slu.cz (z výpisu je zřejmé, že se jedná o server joanes.opf.slu.cz).



Obrázek 31: Trasování pomocí programu NeoTrace Pro

Zdroj: Vlastní zpracování

Node Info Window

joanes.opf.slu.cz

Previous | Node 16 of 16 | Next

To receive output for a database update, use the `-B` flag.

Information related to '193.84.192.0 - 193.84.223.255'

Abuse contact for '193.84.192.0 - 193.84.223.255' is 'abuse@slu.cz'

```

inetnum: 193.84.192.0 - 193.84.223.255
netname: OPF_SLU-T34CZ
descr: Silesian University in Opava
descr: Slezska univerzita v Opave
descr: Opava, Karvina
country: CZ
org: ORG-SUIO1-RIPE
admin-c: SUIO2-RIPE
tech-c: SUIO2-RIPE
status: ASSIGNED PI
mnt-by: TENCZ-MNT
mnt-lower: RIPE-NCC-END-MNT
mnt-by: RIPE-NCC-END-MNT
remarks: Please report network abuse -> abuse@slu.cz
source: RIPE Filtered

organisation: ORG-SUIO1-RIPE
org-name: Silesian University in Opava
org-type: OTHER
address: Slezska univerzita v Opave
address: Na Rybnicku 1
address: Opava
address: 746 01
address: The Czech Republic
phone: +420 553684600
            
```

Obrázek 32: Informace o počítači z databáze RIPE

Zdroj: Vlastní zpracování

Údaje o dostupných okolních počítačích se ukládají v ARP tabulce. Zjištění evidovaných počítačů je možné příkazem *arp - a* pomocí protokolu arp (viz protokol TCP/IP), současně je evidováno, zda je adresa přidělená (dynamická) nebo pevně zadaná (statická).

```
C:\Users\mobil_000>arp -a
Interface: 192.168.10.4 --- 0x4
Internet Address      Physical Address      Type
192.168.10.1          50-67-f0-ec-90-60    dynamic
192.168.10.3          00-18-de-bf-da-45    dynamic
192.168.10.8          bc-ee-7b-49-d9-10    dynamic
192.168.10.10         bc-ee-7b-49-d8-ca    dynamic
192.168.10.255       ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250      01-00-5e-7f-ff-fa    static
255.255.255.255      ff-ff-ff-ff-ff-ff    static
```

Obrázek 33: Příkaz arp

Zdroj: Vlastní zpracování

3.3 Typy sítí dle jejich dosahu

Budeme-li se pohybovat v oblasti sítí, měli bychom zde zmínit alespoň okrajově základní členění sítí dle jejich dosahu. Některé z níže uvedených pojmů se již běžně nepoužívají, s jinými se však stále ještě setkáme:

3.3.1 WAN (WIDE AREA NETWORK)

Tato zkratka reprezentuje počítačovou síť, která je z hlediska dosahu geograficky nejrozsáhlejší. V obecné rovině je za WAN požadována síť Internet. Síť WAN jsou používány pro propojení lokálních sítí nebo jiných sítí menšího rozsahu, tato zkratka je stále hojně používána. Budování takovýchto sítí je velmi nákladné a stalo se záležitostí těch největších hráčů na trhu. Můžeme se zde setkat s technologiemi jako ATM, X.25, MobileFI apod.

3.3.2 MAN (METROPOLITAN AREA NETWORK)

Označení, které bylo hojně používáno v USA v počátcích vznikající sítě internet. Jak vyplývá z jeho názvu, jednalo se o síť geograficky příslušné svým rozsahem městu (často velkým městům). V Evropě se toto označení neuchytilo, snad z důvodu rychlého růstu dílčích sítí přesahujícího hranice měst.

3.3.3 LAN (LOCAL AREA NETWORK)

Stále hojně používané označení pro lokální síť. Často se jedná o síť pokrývající svým rozsahem domácnost či firmu. Přenosové rychlosti v rámci těchto sítí bývají relativně vysoké, v současnosti nejrozšířenějšími technologiemi v LAN sítích jsou Ethernet či Wi-Fi.

3.3.4 PAN (PERSONAL AREA NETWORK)

V současné době často používaný pojem. Jedná se o malou „personální“ síť, tvořenou komunikujícími zařízeními, jako jsou např. mobilní telefony, PDA zařízení, či notebooky. Jedná se o malou síť, jejíž dosah bývá pouze několik metrů. Používá se pouze pro komunikaci samotných zařízení, popř. pro připojení do okolních sítí či k internetu. Rychlost připojení v rámci těchto sítí bývá pomalejší. Na této úrovni se můžeme setkat s technologiemi jako FireWire, USB, BlueTooth, IrDA apod.

3.4 Nejčastější síťové architektury v prostředí internetu

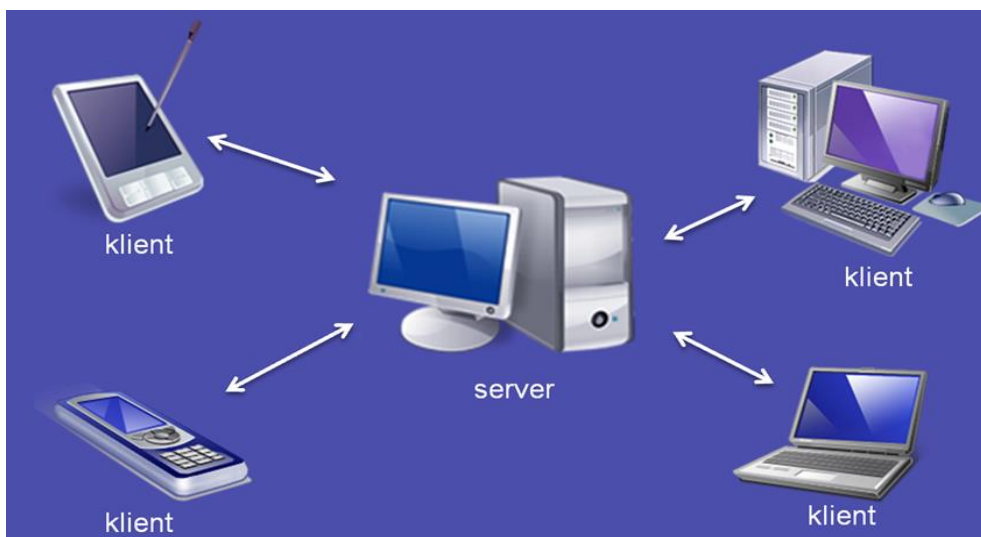
Budeme-li dále hovořit o internetových aplikacích, či o komunikaci mezi počítači v rámci internetu, je nutná alespoň základní představa o nejčastěji používaných síťových modelech v tomto prostředí.

3.4.1 ARCHITEKTURA KLIENT/SERVER

Architektura klient/server je jedním z nejčastěji používaných modelů v prostředí internetu.

Tento model, můžeme nalézt jak na hardwarové (fyzické propojení), tak na softwarové (komunikační) úrovni. Za server je v tomto případě považován počítač či aplikace poskytující určité služby ostatním. Jako příklad můžeme uvést tiskové servery, souborové servery či webové servery). V pozici klienta pak vystupují ostatní zařízení či klientské aplikace, které využívají služeb poskytovaných serverem. Z logiky věci je samozřejmě klientů větší počet.

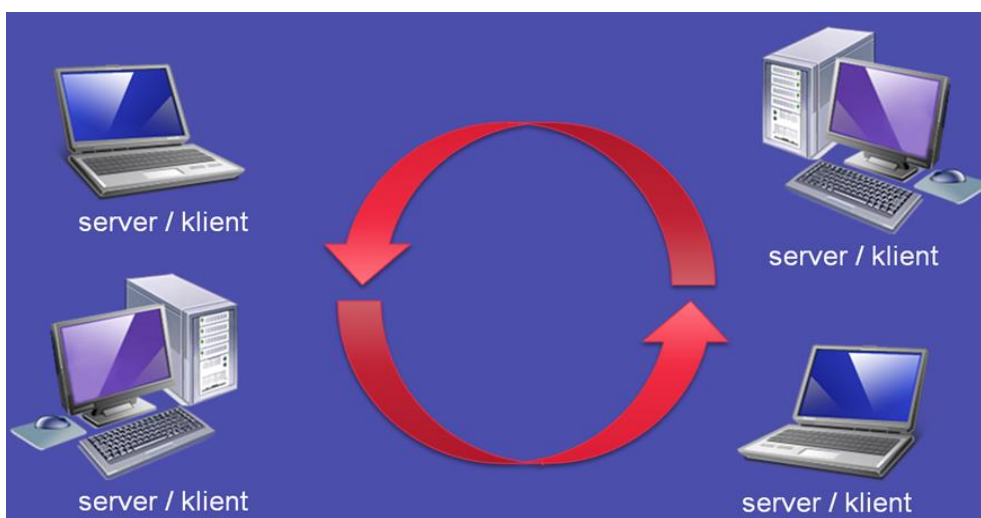
Oproti tomu architektura peer to peer (P2P), se vyznačuje tím, že všichni účastníci sítě vystupují jako server (poskytovatel služby či dat), a zároveň jako klient (využívají služeb). Tento model v poslední době nachází velké uplatnění v oblasti sdílení obsahu, např. v sociálních sítích.



Obrázek 34: Architektura klient/server

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy*, výukový materiál OPF v Karviné

3.4.2 ARCHITEKTURA P2P



Obrázek 35: Architektura P2P



SHRNUTÍ KAPITOLY

Kapitola měla představit pracovní nástroje pro práci se sítí, zejména pro identifikaci problémů souvisejících s dostupností počítačů v síti Internet. Byly představeny principy práce v počítačové síti, na konkrétních příkladech byly ukázány standardní nástroje a práce s TCP/IP protokolem. Byly ukázány základní síťové architektury.

4 ZÁKLADY INTERNETOVÝCH TECHNOLOGIÍ, SLUŽBY INTERNETU

RYCHLÝ NÁHLED KAPITOLY



V této kapitole se studenti seznámí se základními službami Internetu, provozovanými pomocí aplikačních protokolů TCP/IP protokolu. S tímto protokolem se seznámí důkladněji, včetně jeho srovnání s ISO/OSI modelem.

CÍLE KAPITOLY



Cílem kapitoly je seznámit studenty se základními pojmy, se kterými se setkají v kontextu fungování Internetu. Studenti po prostudování textu budou mít ujasněny vztahy mezi IP adresami a doménovými jmény a jejich přidělováním a správou. Na pojem internet se můžeme dívat z několika pohledů. Budeme-li se dívat na internet jako na medium, můžeme jej považovat za soubor technických prostředků umožňující šíření a sdílení dat (informací) v elektronické podobě po celém světě, a to bez omezení týkajících se typu a obsahu.

Na internet můžeme nahlížet také jako na celosvětovou počítačovou informační síť, která vznikla propojením samostatných dílčích sítí, ať už státních, univerzitních, regionálních či na federální úrovni (ve Spojených státech).

Federal Networking Council v roce 1995 definoval pojem internet jako globální informační systém, který:

- Je logicky propojen prostřednictvím globálně unikátního adresového prostoru založeného na protokolu IP, popř. na jeho rozšířeních a následnících,
- Je schopen zajistit vzájemnou komunikaci na základě rodiny protokolů TCP/IP, popř. na jeho rozšířeních a následnících (nebo pomocí protokolů kompatibilních s protokolem IP),
- Zajišťuje, používá a zpřístupňuje (veřejně či soukromě) služby vyšší úrovně na výše uvedené infrastruktuře.

Pojem internet se v odborné literatuře vyskytuje jako technický pojem (psáno s malým počátečním písmenem), pak jej chápeme jako technologicky propojenou počítačovou síť, která funguje na bázi protokolů (např. TCP, IP, UDP). V této rovině jsme se pak v dřívějších dobách ještě mohli setkat s dalšími pojmy, jako jsou intranet (vnitřní, interní síť organizace) či extranet (síť geograficky přesahující rozsah internetu).

Setkáme-li se s pojmem Internet (psáno s velkým počátečním písmenem), chápeme tento pojem jako název mezinárodní sítě (International Network), jako vlastní jméno celosvětové informační a komunikační sítě. Velmi často jsou laickou veřejností zaměňovány pojmy internet a WWW. Internet bychom měli chápat jako komunikační prostor pro výměnu, získávání a publikování informací, bez ohledu na jejich původ, formu, či jazyk. Kdežto WWW (World Wide Web) je zkratka služby poskytované samotným internetem. Jedná se o množinu propojených dokumentů a dalších zdrojů pomocí sítě odkazů.



ČAS POTŘEBNÝ KE STUDIU

Čas potřebný ke studiu v rozsahu 3-4 hodin, podle míry praktické práce.



KLÍČOVÁ SLOVA KAPITOLY

DNS, doména, IANA, NIC, infrastruktura, Cesnet, služby Internetu

4.1 Služby Internetu

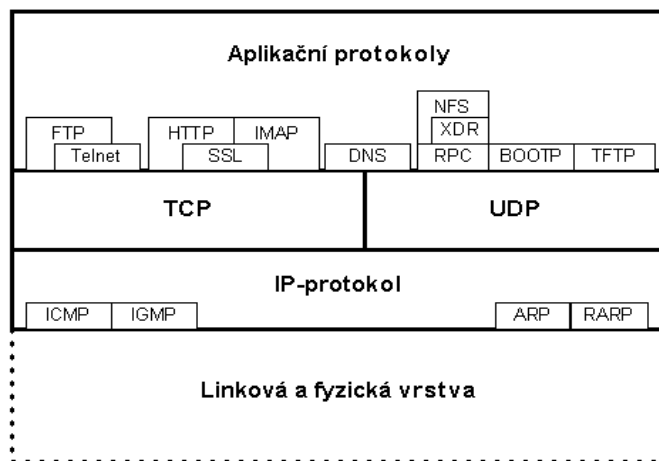
Pod pojmem „služby Internetu“ si většina laické veřejnosti představí chatování, stahování filmů a písniček nebo komunikaci a sdílení dat v sociálních sítích, popřípadě internetovou telefonii. V některých případech taky nákupy přes Internet elektronické bankovníctví či podobné služby. Tento výukový materiál se zabývá Internetovými službami založenými na aplikačních protokolech TCP/IP modelu, především službami pro přenos souborů, přenos hypertextů, správou elektronické pošty a vzdálenou správou. Je nutné si předem uvědomit, že vzhledem k povaze Internetu, který je v podstatě velkou počítačovou sítí, je nutné pro pochopení těchto služeb a jejich správu znát i elementární základy práce v počítačové síti.

4.2 Protokol TCP/IP

Vznik sady protokolů TCP/IP datujeme do sedmdesátých let 20. století¹³. Počátkem osmdesátých let byl protokol akceptován ministerstvem obrany USA jako perspektivní, v roce 1982: ministerstvo obrany USA přikazuje použití protokolů TCP/IP u všech sítí nově připojovaných k síti internet a od roku 1983 na protokoly TCP/IP přechází celý Internet.

¹³ Text a obrázky převzato a částečně upraveno z Kabelová, Dostálek, 2010

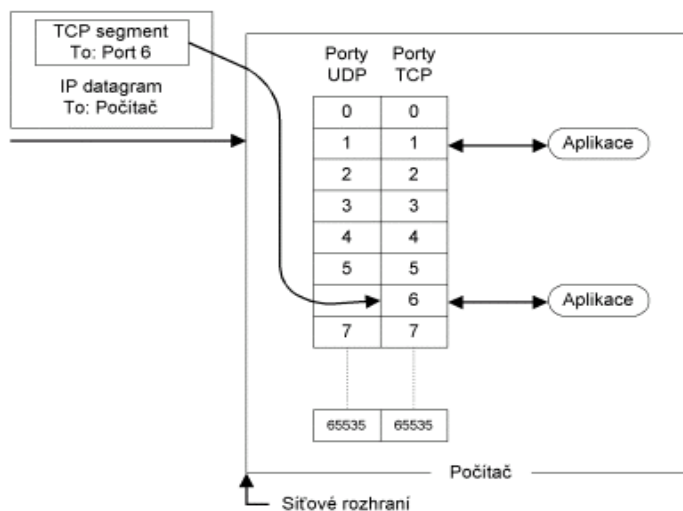
Rodina protokolů TCP/IP se nezabývá (až na výjimky) fyzickou a linkovou vrstvou. V praxi se i v Internetu používají pro fyzickou a linkovou vrstvu často protokoly vyhovující normám ISO OSI, které standardizoval ITU.



Obrázek 36: Protokoly a vrstvy TCP/IP modelu

Zdroj: Kabelová, Dostálek, 2010

Cílová aplikace je v Internetu adresována (jednoznačně určena) IP-adresou, číslem portu a použitým protokolem (TCP nebo UDP). Protokol IP dopraví IP-datagram na konkrétní počítač. Na tomto počítači běží jednotlivé aplikace. Podle čísla cílového portu operační systém pozná, které aplikaci má TCP-segment doručit.



Obrázek 37: TCP/IP adresace

Zdroj: Kabelová, Dostálek, 2010

4.2.1 TCP A UDP PROTOKOLY

Protokoly TCP a UDP odpovídají transportní vrstvě. TCP (resp. UDP) dopravuje data pomocí TCP segmentů (resp. UDP datagramů), které jsou adresovány jednotlivým aplikacím. Jinak řečeno, protokol TCP (resp. UDP) zajišťuje spojení mezi aplikacemi běžícími na vzdálených počítačích. Adresace je realizována prostřednictvím portů. Port si lze představit jako přechodový bod na rozhraní mezi transportní a aplikační vrstvou, který je jednoznačně identifikovatelný (svým číslem, tedy tzv. číslem portu). Samotný port však není ani odesílatelem, ani příjemcem – tím je entita aplikační vrstvy (například systémový proces, aplikace úloha apod.), která se "přidruží" k příslušnému portu a skrz něj komunikuje s entitami transportní vrstvy (které implementují protokoly TCP či UDP). Protokol TCP je spojovanou službou (connection oriented), tj. službu která mezi dvěma aplikacemi naváže spojení – vytvoří na dobu spojení virtuální okruh. Tento okruh je plně duplexní (data se přenášejí současně na sobě nezávisle oběma směry). Přenášená data jsou číslována. Ztracená nebo poškozená data jsou znovu vyžádána. Konce spojení ("odesílatel" a "adresát") jsou určeny číslem portu z rozsahu hodnot 0 až 65535. U čísel portů se rozlišuje, zde se jedná o porty protokolu TCP nebo UDP (za číslo napíše lomítko a TCP nebo UDP, port 53/tcp nemá nic společného s portem 53/udp). Základní jednotkou přenosu v protokolu TCP je TCP segment. TCP segment se vkládá do IP-datagramu. IP-datagram se vkládá do linkového rámce. Protokol UDP je jednoduchou alternativou k protokolu TCP. Protokol UDP je nespojovaná služba (na rozdíl od protokolu TCP), tj. nenavazuje spojení. Odesílatel odešle UDP datagram příjemci a nepožaduje potvrzení o přijetí. UDP datagramy jsou baleny do IP-datagramu.

Číslo	TCP	UDP	Služba
20	tcp	udp	FTP (data)
21	tcp	udp	FTP (příkazy)
22	tcp	udp	SSH
23	tcp	udp	Telnet
25	tcp	udp	SMTP
53	tcp	udp	DNS
66	tcp	udp	Oracle SQL*NET
80	tcp	udp	HTTP
88	tcp	udp	Kerberos
110	tcp	udp	Post Office Protocol - Version 3, POP3
143	tcp	udp	IMAP
366	tcp	udp	SMTP, Simple Mail Transfer Protocol
443	tcp	udp	HTTPS
647	tcp		DHCP Failover Protocol
993	tcp		IMAPS, SSL
995	tcp		POP3S, SSL
5800	tcp		VNC - Java klient
5900	tcp		VNC

Tabulka 1 Přehled nejčastějších portů

Adresátem UDP datagramu nemusí být pouze jednoznačná IP-adresa, tj. síťové rozhraní konkrétního počítače. Adresátem může být skupina stanice – adresovat lze tzv. oběžník. Adresovat lze všeobecné oběžníky (broadcast), nebo adresné oběžníky (multicast).

4.2.2 IP PROTOKOL

IP protokol přenáší tzv. IP-datagramy mezi vzdálenými počítači. Každý IP-datagram ve svém záhlaví nese adresu příjemce, což je úplná směrovací informace pro dopravu IP-datagramu k adresátovi. Adresace je realizována pomocí IP adresy. Síť může přenášet každý IP-datagram samostatně. IP-datagramy tak mohou k adresátovi dorazit v jiném pořadí, než byly odeslány. Každé síťové rozhraní v síti Internet má svou celosvětově jednoznačnou IP-adresu (jedno síťové rozhraní může mít více IP-adres, avšak jednu IP-adresa nesmí používat více síťových rozhraní).

Některé linkové protokoly jsou určeny pro dopravu dat v rámci lokální sítě, jiné linkové protokoly dopravují data mezi sousedními směrovači rozsáhlé sítě. IP-protokol na rozdíl od linkových protokolů dopravuje data mezi dvěma libovolnými počítači v Internetu, tj. přes více LAN.

Data jsou od odesílatele k příjemci dopravována (směrována) přes směrovače (router). Na cestě od odesílatele k příjemci se může vyskytnout celá řada směrovačů. Každý směrovač řeší samostatně směrování k následujícímu směrovači. Data jsou tak předávána od směrovače k směrovači. Z angličtiny se ustálil výraz následující hop (next hop), jako následující uzel, kam se data předávají. Hopem se rozumí buď následující směrovač, nebo cílový stroj.

IP-protokol je tvořen vlastním protokolem IP a několika dílčími protokoly.

Protokol ICMP je služební protokol, který je součástí IP-protokolu. Protokol ICMP slouží k signalizaci mimořádných událostí v sítích postavených na IP-protokolu. Protokol ICMP svoje datové pakety balí do IP-protokolu, tj. pokud budeme prohlížet přenášené datagramy, pak v nich najdeme za linkovým záhlavím záhlaví IP-protokolu následovány záhlavím ICMP paketu. Základními funkcemi a indikovanými hodnotami jsou echo, nedoručitelný datagram, volba rychlosti odesílání, žádost o směrování změna směrování, maska podsítě, synchronizace dat, vypršení doby životnosti apod.

Protokol IGMP je podobně jako protokol ICMP služebním protokolem (podmnožinou) protokolu IP. Pakety IGMP-protokolu jsou baleny do IP-datagramů. Protokol IGMP slouží k šíření adresných oběžníků (multicasts)

Protokol ARP (Address Resolution Protocol) řeší problém zjištění linkové adresy pro tější stanice ze znalosti její IP-adresy vysláním linkového oběžníku (linková adresa

FF:FF:FF:FF:FF:FF) s prosbou: “Já stanice o linkové adrese HW1, IP-adrese IP1, chci komunikovat se stanicí o IP-adrese IP2, kdo mi pomůže s nalezením linkové adresy stanice o IP-adrese IP2? Stanice IP2 takovou žádost uslyší a odpoví. V odpovědi uvede svou linkovou adresu HW2.

Protokol ARP určuje jednoznačný vztah mezi jednoznačnou IP-adresou příjemce (unicast) a linkovou adresou příjemce. To je možné tehdy, když mezi IP-adresami a linkovými adresami existuje jednoznačný vztah. Tento vztah se anglicky nazývá mapping, tj. mapování IP adres na linkové adresy.

4.2.3 APLIKAČNÍ PROTOKOLY

Aplikační protokoly v TCP/IP modelu odpovídají několika vrstvám ISO OSI. (Relační, prezentační a aplikační vrstva).

Prostřednictvím aplikačních protokolů se zabezpečují služby aplikací. Nečastěji používané protokoly jsou patrné z obr. 27. Mezi nejpoužívanější protokoly patří telnet (i když z důvodu bezpečnosti je na ústupu), FTP, HTTP, POP3, IMAP, SMTP, DNS, DHCP, SSH a další.

Telnet

Protokol Telnet slouží pro tzv. vzdálené přihlašování, neboli k tomu, aby uživatel jednoho počítače získal přístup ke vzdálenému počítači prostřednictvím vzdáleného terminálu. Na displeji lokálního počítače je zobrazen displej vzdáleného počítače, klávesnice se chová jako připojená ke vzdálenému počítači. V dnešní době existuje poměrně velká skupina kvalitnějších aplikací jako VNC, TeamViewer apod. Přesto má telnet své místo, především pro vzdálenou správu na úrovni příkazového řádku a využívání výpočetní kapacity vzdáleného počítače (aplikace, soubory, periferie apod.). Protokol Telnet je koncipován tak, aby umožňoval "spolupráci" různých platform – není závislý na prostředí klientského počítače (ze které se uživatel přihlašuje ke vzdálenému počítači) ani serveru (vzdálený počítač). Díky tomu je například možné, aby se prostřednictvím protokolu Telnet uživatel počítače PC s MS Windows přihlásil na dálku k Unixovému počítači, a pracoval s Unixovými aplikacemi, které běží na tomto vzdáleném počítači a na jeho počítači PC by provozovány být nemohly.

HTTP

(HyperText Transfer Protocol) slouží k přenosu hypertextových souborů vytvořených jazykem HTML. Na vzdáleném počítači musí být spuštěn HTTP server a na lokálním HTTP klient (prohlížeč HTML, např. Internet Explorer). HTTP má vyhrazen TCP port 80. Protokol umožňuje prohlížeči vyžádat si na serveru konkrétní WWW stránku, a následně ji zobrazit na klientském počítači. Tento protokol je spolu s elektronickou poštou nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech. Samotný protokol HTTP neumožňuje šifrování ani zabezpečení integrity dat. Pro zabezpečení HTTP se často používá TLS spojení nad TCP. Toto použití je označováno jako HTTPS (Security

Hyper Text Transfer Protocol) slouží k přenosu zašifrovaných hypertextových souborů typu HTML. Na vzdáleném počítači přitom musí být spuštěn HTTP server s podporou šifrování (tzv. SSL protokolem).

FTP

(File Transfer Protocol) je protokol pro přenos souborů mezi uzlovými počítači sítě, může být používán nezávisle na použitém operačním systému.

SMTP

(Simple Mail Transfer Protocol) je používán pro přenos (odesílání) elektronické pošty. SMTP server má vyhrazen port 25. Klient s ním komunikuje prostřednictvím klientské aplikace, nebo zasíláním příkazů z příkazového řádku (například příkazy služby telnet).

POP3

(Post Office Protocol, v současnosti verze 3) zajišťuje komunikaci mezi poštovním serverem a osobním počítačem uživatele, je určen ke stahování došlé pošty z poštovního serveru. Služba běží na TCP portu 110. Klient s ním komunikuje prostřednictvím klientské aplikace, nebo zasíláním příkazů

IMAP

(Internet Message Access Protocol) je protokol pro vzdálený přístup k e-mailové schránce prostřednictvím e-mailového klienta. IMAP nabízí oproti jednodušší alternativě POP3 pokročilé možnosti vzdálené správy (práce se složkami a přesouvání zpráv mezi nimi, prohledávání na straně serveru a podobně) a práci v tzv. on-line i off-line režimu. V současné době se používá protokol IMAP4.

4.3 Služby založené na aplikačních protokolech

Komunikace v prostředí sítě a od toho se odvíjející služby mohou být v podstatě dvojího typu. V případě, že každý počítač (uživatel, aplikace) definuje, které prostředky a v jakém rozsahu poskytne ostatním a následně komunikaci neřídí, ale ponechává na přistupujícím počítači, jedná se o síť „rovný s rovným“ označovanou jako peer to peer. V této síti není „řídící“ počítač, každý má stejná práva a schopnosti, které si sám určuje.

Základní služby Internetu jsou založeny na architektuře (principech, formě komunikace), kdy jeden počítač poskytuje prostředky a služby (server) a ostatní počítače k těmto službám přistupují (klienti). Tato architektura je označována jako server/klient (klient/server). Jako server si můžeme představit hardwarový prostředek, na kterém se příslušná

služba provozuje (počítače označené jako servery, dislokované zpravidla v tzv. „serverovnách“. Prakticky je ale server zabezpečován aplikací, která na příslušném počítači běží (serverem tedy může být libovolný počítač, na kterém je spuštěna příslušná aplikace).

Serverová aplikace zabezpečuje práva uživatelů (klientů), disponibilní prostředky, pravidla, formy komunikace a správu dat. Data jsou umístěna centrálně na datovém prostředku (disk, datové pole) obsluhovaném serverem. Pro přístup k datům a jejich organizaci mohou být využívány další softwarové produkty (databáze, programy apod.).

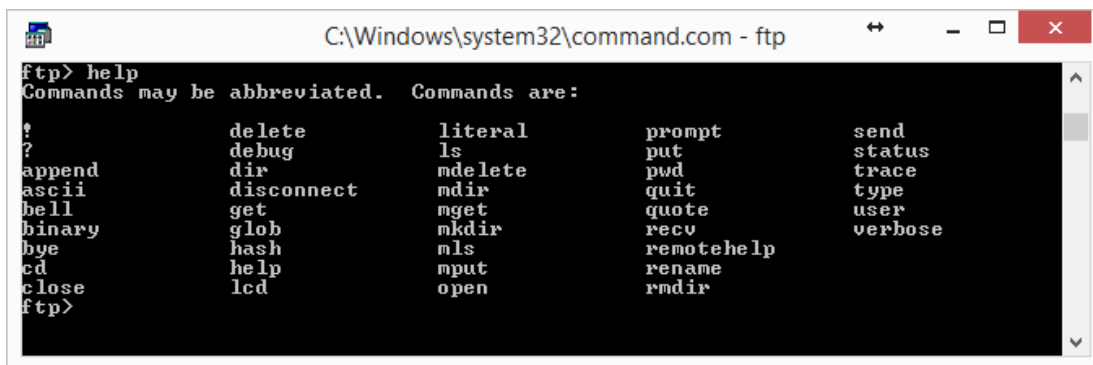
Klient (uživatel, počítač, aplikace) prostřednictvím specifického software přistupuje k aplikacím zabezpečovaným serverem.

(Ve firemním prostředí může být typickým příkladem software pro účetnictví. Software je nainstalovaný na jednom počítači, na kterém jsou současně uložena data. K programu přistupují klienti (účetní) z jiných PC prostřednictvím klientské části účetního software, mají nadefinované role a práva a přidělenou část prostředků, které mohou spravovat, např. faktury., sklad apod.). Blíže byly tato principy popsány v kapitole 1.6.

4.3.1 FTP

FTP je jeden z nejstarších protokolů, využívá porty TCP/21 a TCP/20. Pakety na portu 21 slouží k řízení komunikace – řídicí kanál. Port 20 slouží k vlastnímu přenosu dat – datový kanál. Porty klienta jsou dynamické a přiděluje je OS. Protokol je interaktivní a umožňuje řízení přístupu (přihlašování login/heslo), specifikaci formátu přenášeného souboru (znakově – binárně), výpis vzdáleného adresáře atd. V současné době není považován za bezpečný a z tohoto důvodu pro něj byla definována některá rozšíření (RFC 2228). Hesla a soubory jsou ve standardním protokolu zasílána jako běžný text (nejsou šifrovaná) což silně snižuje bezpečnost (ohrožuje jméno, heslo, ale i přenášená data). Protokol FTP stanoví pravidla komunikace mezi klientem a serverem. Jedná se o spojovanou spolupráci. Klient naváže spojení se serverem, předá uživatelské jméno a heslo. Po navázání spojení může probíhat práce s adresáři a soubory a přenos dat mezi klientem a serverem. Spojení se obvykle ukončí z podnětu klienta. Úsek komunikace mezi otevřením a uzavřením spojení se nazývá relace. Řídicí kanál se po dobu relace otevře jen jednou a zůstává otevřen, zatímco datový se po přenosu dat (souboru) uzavře a pro každý datový přenos je nutné kanál znovu otevřít. Podle toho, kdo otevírá datový kanál, se rozlišuje aktivní a pasivní režim spojení. Přenos souborů probíhá v textovém nebo binárním režimu v závislosti na typu souboru nebo potřebách uživatele. Chování serveru nebo klientského programu je určeno konfigurací. O komunikaci lze vést na obou stranách spojení protokol. Aplikační protokol FTP využívá vrstvy TCP k zabezpečenému přenosu datových segmentů. Prostřednictvím klientského programu telnet lze zabezpečit přenos datových segmentů, i když komunikujeme se serverem na jiném portu než 23, tj. např. na portu 21; příkazy pak musejí být přímo příkazy protokolu FTP. V protokolu je použit model klient-server. FTP server poskytuje data pro ostatní počítače. Klient se k serveru připojí a může provádět různé operace (výpis adresáře, změna adresáře, přenos dat atd.). Operace jsou řízeny sadou příkazů, které

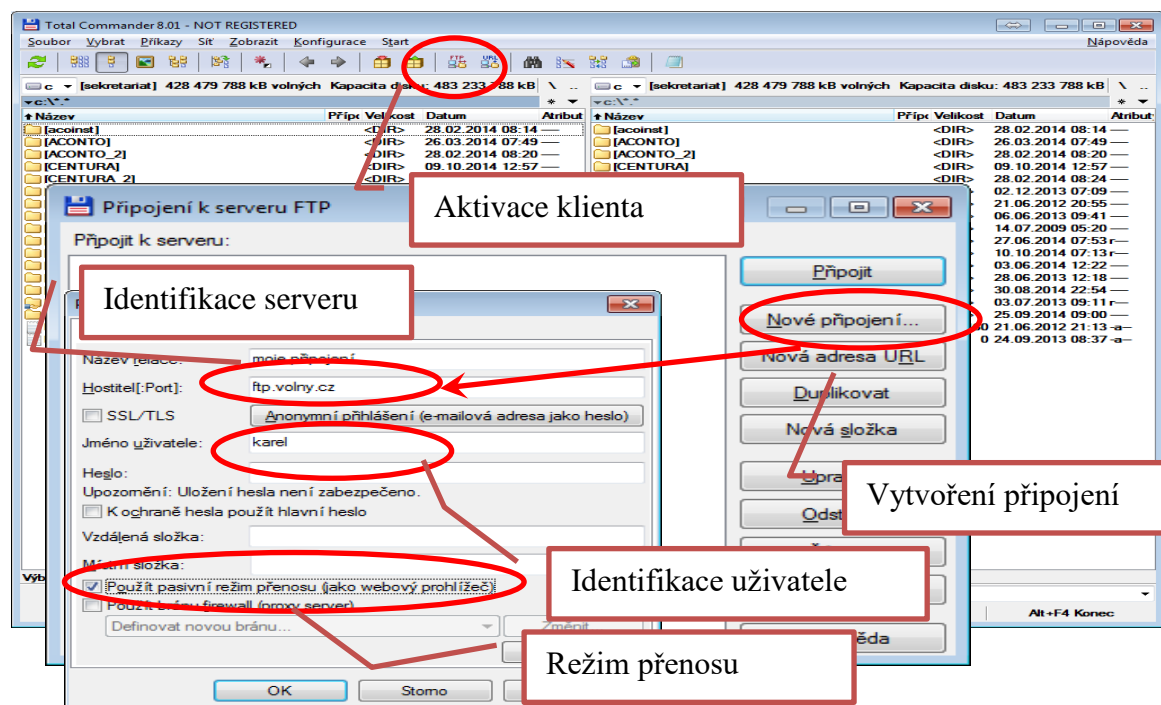
jsou definovány v rámci FTP protokolu, proto kdokoliv může vytvořit klienta pro jakékoliv prostředí nebo operační systém. Existuje mnoho programů pro FTP servery i klienty a mnoho je jich volně dostupných. Operační systém MS Windows má implementovaného klienta FTP, spustitelného pomocí příkazové řádky. Seznam dostupných příkazů je k dispozici příkazem *help*.



Obrázek 38: Příkazy FTP dostupné v příkazovém řádku

Zdroj: vlastní zpracování

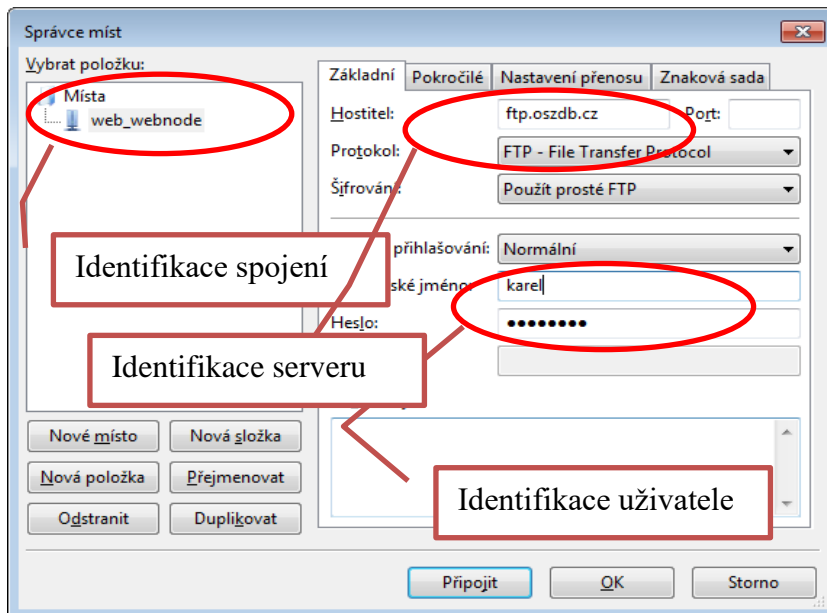
Mezi nejrozšířenější FTP klienty patří Windows Commander (Total Commander). Program umožňuje nadefinovat připojení a ve dvou oddělených oknech zobrazovat soubory lokálního a vzdáleného počítače, jako by se jednalo o soubory lokální.



Obrázek 39: Total Commander jako FTP klient

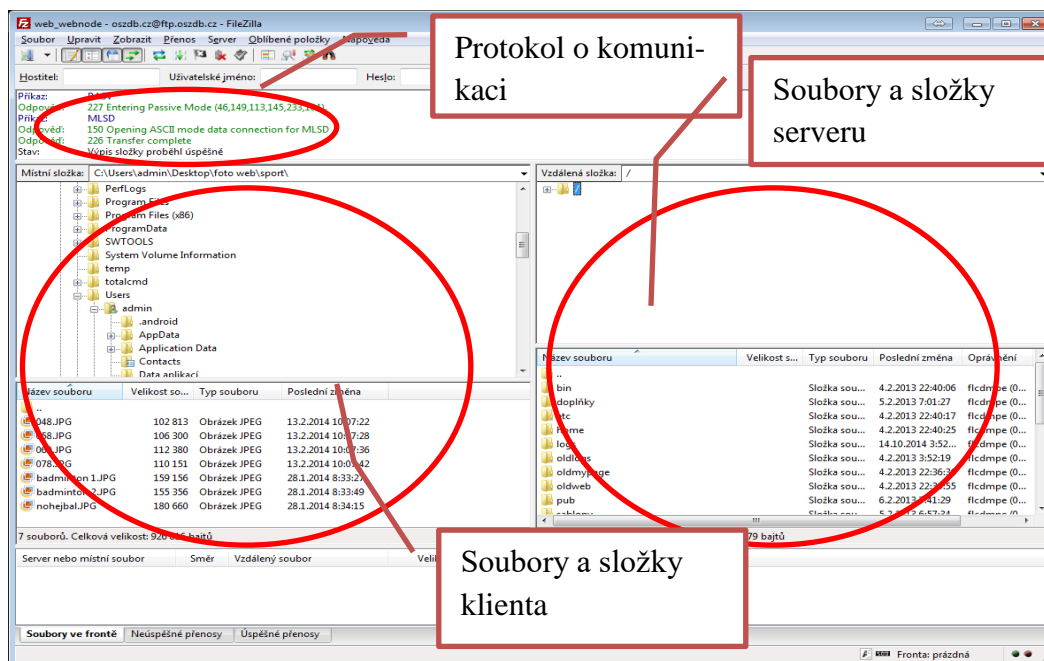
Zdroj: vlastní zpracování

Dalším, poměrně výkonným klientem je software FileZilla, který je k dispozici rovněž jako serverová aplikace. Software umožňuje definování množiny připojení, správu souborů včetně práce s atributy apod. Základní nastavení spojení a správu souborů ukazují následující obrázky.



Obrázek 40: FTP klient FileZilla

Zdroj: vlastní zpracování



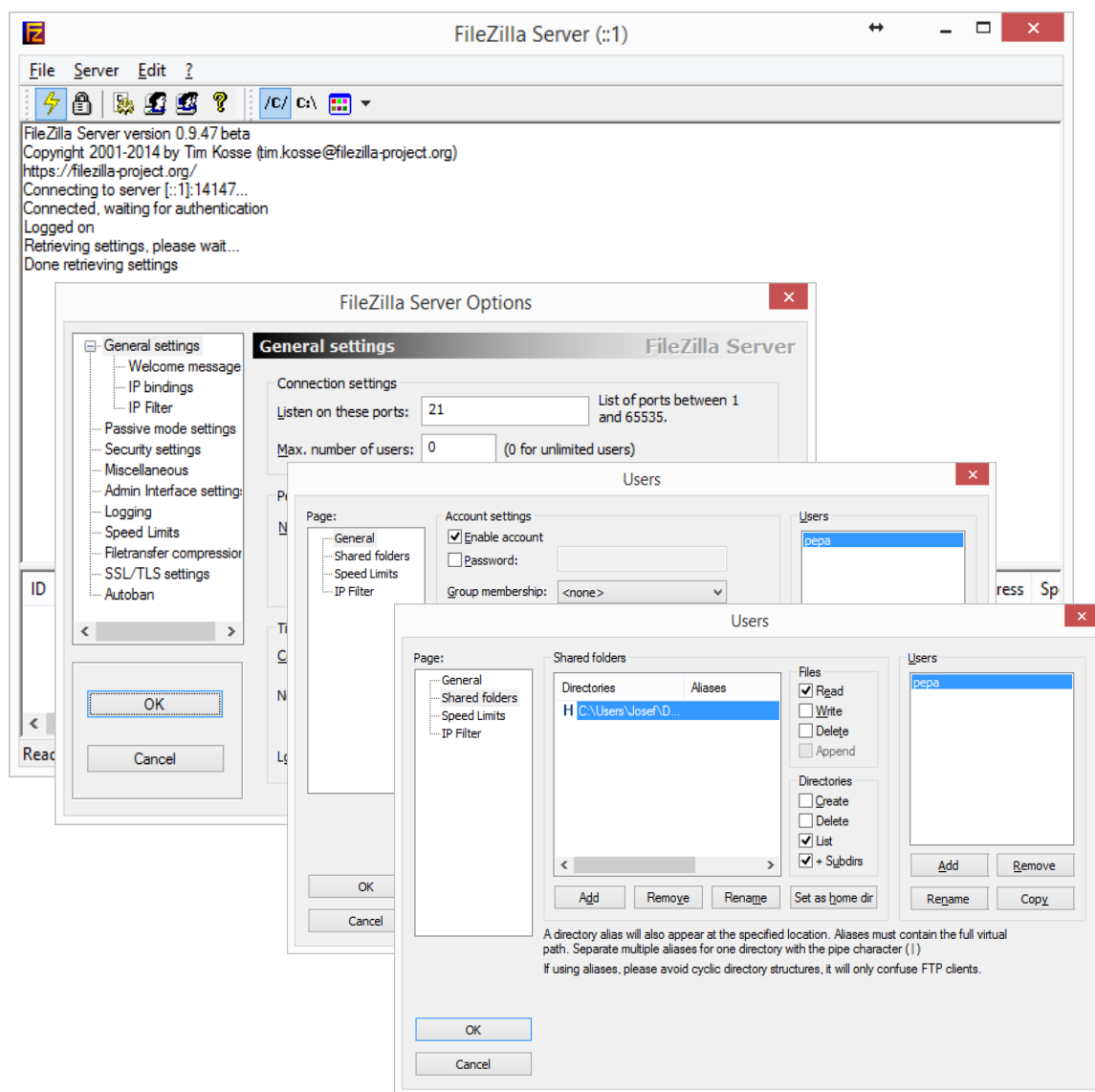
Obrázek 41: Pracovní oblast FileZilla klienta

Zdroj: vlastní zpracování

Jako klient FTP v dnešní době funguje i většina www prohlížečů.

Pokud chceme aktivně využívat FTP službu a provozovat vlastní FTP server musíme mít příslušný software. V segmentu Free aplikací je velmi výkonnou serverovou aplikací např. Caesar nebo FileZilla server.

základní nastavení FTP serveru spočívá ve vytvoření skupin uživatelů, nastavení pracovních prostorů, uživatelských jmen a hesel a uživatelských práv. K serveru můžeme v lokální síti přistupovat prostřednictvím IP adresy počítače, na kterém je příslušný software provozován.



Obrázek 42: Základní nastavení serveru FileZilla

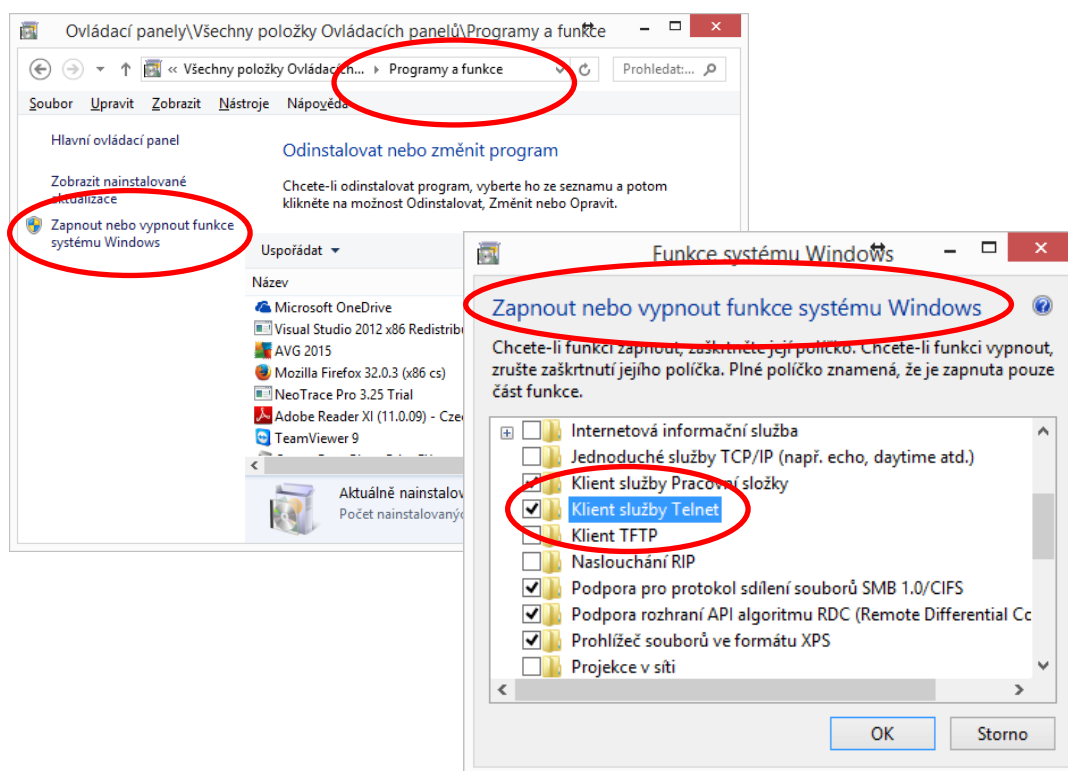
Zdroj::vlastní zpracování

4.3.2 TELNET

Telnet je aplikační protokol TCP/IP, program, pro komunikaci mezi dvěma počítači pomocí telnet protokolu a služba pro vzdálený přístup. Program je dodnes součástí Microsoft Windows a unixových systémů. Dříve se používal (spolu s protokolem telnet) pro připojení ke vzdálenému počítači prostřednictvím počítačové sítě jako emulace terminálu, která poskytovala možnost práce uživatele na vzdáleném počítači v příkazovém řádku. Telnet tak byl nástupcem terminálů, ze kterých se uživatelé připojovali ke vzdálenému počítači pomocí sériové linky. Hlavní nevýhodou telnetu je absence šifrování přenášených dat, a proto dnes uživatelé místo telnetu používají protokol SSH. V současné době se program telnet používá pro komunikaci mezi počítačovými programy (například simulace připojení webového prohlížeče k webovému serveru, při simulaci SMTP protokolu pro přepravu elektronické pošty a podobně).

Standardně není u novějších verzí příkaz telnet k dispozici. Přesto je telnet klient i telnet server součástí instalace, není však z bezpečnostních důvodů aktivní.

Aktivaci klienta i serveru lze provést v nabídce ovládací panely – programy a funkce (přidat ubrat programy) – zapnout nebo vypnout funkce systému windows.



Obrázek 43: aktivace klienta telnet v MS Windows

Zdroj: vlastní zpracování

Po aktivaci lze klienta spustit jako příkaz v příkazové řádce. Parametrem „?“ lze opět vypsat možnosti příkazu a parametry příkazu.

Příkazy Telnet se mohou nepatrně lišit podle verze OS, respektive podle použitého klienta. Seznam příkazů pro klienta MS Windows je dostupný např. na <http://windows.microsoft.com/cs-cz/windows/telnet-commands#1TC=windows-7>.

Vytvořit připojení programu Telnet k hostitelskému počítači nebo vzdálenému serveru lze příkazem *open* nebo *o*. Standardně se pro komunikaci používá port 23, lze jej však změnit, chceme-li přistupovat ke službě zabezpečené jiným protokolem.

Příkaz *open ftp.opf.slu.cz 100* by otevřel komunikaci se serverem ftp.opf.slu.cz (pokud by existoval) na portu 100.

Ukončit existující připojení programu Telnet lze příkazem *close* (*c*). Může být použit v kombinaci s názvem hostitele a číslem portu.

```

C:\Windows\system32\command.com
C:\USERS\MOBIL_~1>telnet /?
telnet [-a[-e řídicí znak][-f soubor protokolu][-l uživatel]
      [-t terminál][hostitel [port]]
-a      Pokus o automatické přihlášení Stejně jako parametr -l.
        Bude použito jméno aktuálně přihlášeného uživatele.
-e      Řídicí znak pro vstup do příkazového řádku klienta
        protokolu Telnet.
-f      Název souboru pro protokolování na straně klienta.
-l      Uživatelské jméno pro přihlášení. Vyžaduje u vzdáleného
        systému podporu možnosti TELNET ENVIRON.
-t      Určuje typ terminálu.
        Jsou podporovány pouze terminály typu vt100, vt52,
        ansi a vtnt.
hostitel  Určuje název hostitele nebo adresu IP vzdáleného počítače.
port     Číslo portu nebo název služby.
    
```

Obrázek 44: atributy příkazu telnet

Zdroj: vlastní zpracování

Příkaz *display* zobrazí aktuální nastavení klienta služby Telnet (seznam aktuálních pracovních parametrů). Úpravu parametrů je nutné provádět mimo relaci.

Ukončit program Telnet lze příkazem *quit* (*q*). Příkaz *set* nastaví typ terminálu pro připojení, zapne místní odezvu, nastaví ověřování NTLM, znak escape a protokolování, vypnutí místní odezvy nebo nastavení ověřování pro zadávání uživatelského jména a hesla při přihlašování lze provést příkazem *unset*. Příkaz *status* zjistí, zda je klient služby Telnet připojen.

4.3.3 POŠTOVNÍ SLUŽBY

E-mailové zprávy jsou obecně posílány e-mailovému serveru, který ukládá příchozí zprávy v příjemcově mailboxu. Uživatel později znovu získává tyto zprávy buď přes we-

bový prohlížeč, nebo přes e-mailového klienta, který používá jeden z e-mailových protokolů. Zatímco někteří klienti a servery upřednostňují používání vlastních protokolů, zároveň podporují i standardní protokoly (SMTP pro odesílání, pro přijímání se používá POP3 a IMAP), což jim dovoluje komunikovat s ostatními klienty

Poštovní klient je program, který zajišťuje odesílání zpráv a vybírání schránek. Příkladem je např. Microsoft Outlook, Mozilla Thunderbird, Opera a další. Je to v podstatě specializovaný editor, který umí kromě vytvoření zprávy také manipulovat se schránkami, odeslat zprávu nejbližšímu serveru a převzít zprávu ze serveru prostřednictvím POP3 nebo IMAP. Vlastním doručováním zprávy po síti až k adresátovi se klient nezabývá. Součástí klienta bývá adresář, který pomáhá uživateli udržet přehled o adresách.

Poštovní server (MTA) běží obvykle jako démon či Služba Windows a naslouchá na portu TCP/25. K tomuto portu se může připojit (navázat TCP spojení) buď poštovní klient, nebo jiný server, který předá zprávu k doručení. MTA zkontroluje, zda je zpráva určena pro systém, na kterém běží. Pokud ano, předá ji programu MDA (lokální doručení). Pokud je zpráva určena jinému počítači, naváže spojení s příslušným serverem a zprávu mu předá.

Při vyhledávání vzdáleného serveru, kterému má předat zprávu, musí MTA spolupracovat se systémem DNS. Od serveru DNS si vyžádá záznam pro cílovou doménu, který obsahuje IP adresu počítače, který se stará o doručení pošty v této doméně. Pokud DNS tento záznam neobsahuje, pokusí se poštovní server doručit zprávu přímo na počítač uvedený v adrese za zavináčem.

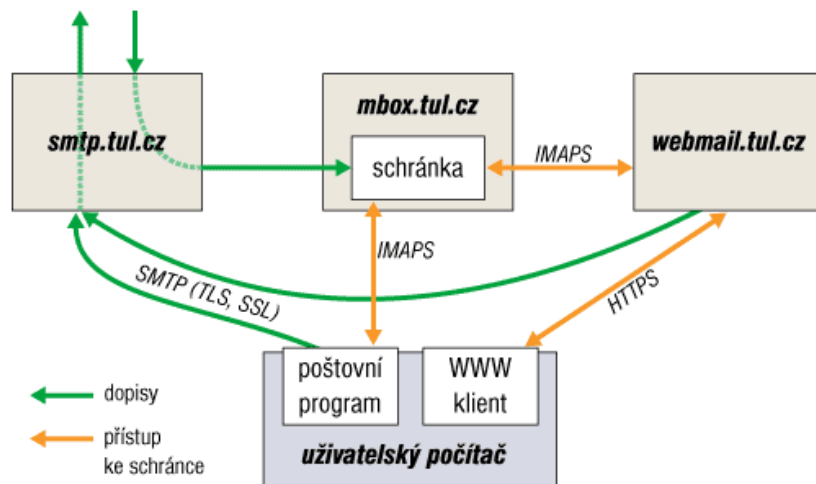
Poštovní server obsahuje v konfiguraci řadu parametrů, pomocí kterých můžeme mimo jiné nastavit, pro které domény MTA přijímá zprávy. Stejně tak je možné určit, od koho bude nebo nebude zprávy přijímat, což je velmi důležité z hlediska bezpečnosti a ochrany proti spamu.

Jako příklad fungování elektronické pošty a příslušných protokolů si ukážeme poštovní servery a komunikaci na technické univerzitě v Liberci¹⁴.

Jádro systému elektronické pošty tvoří tři servery:

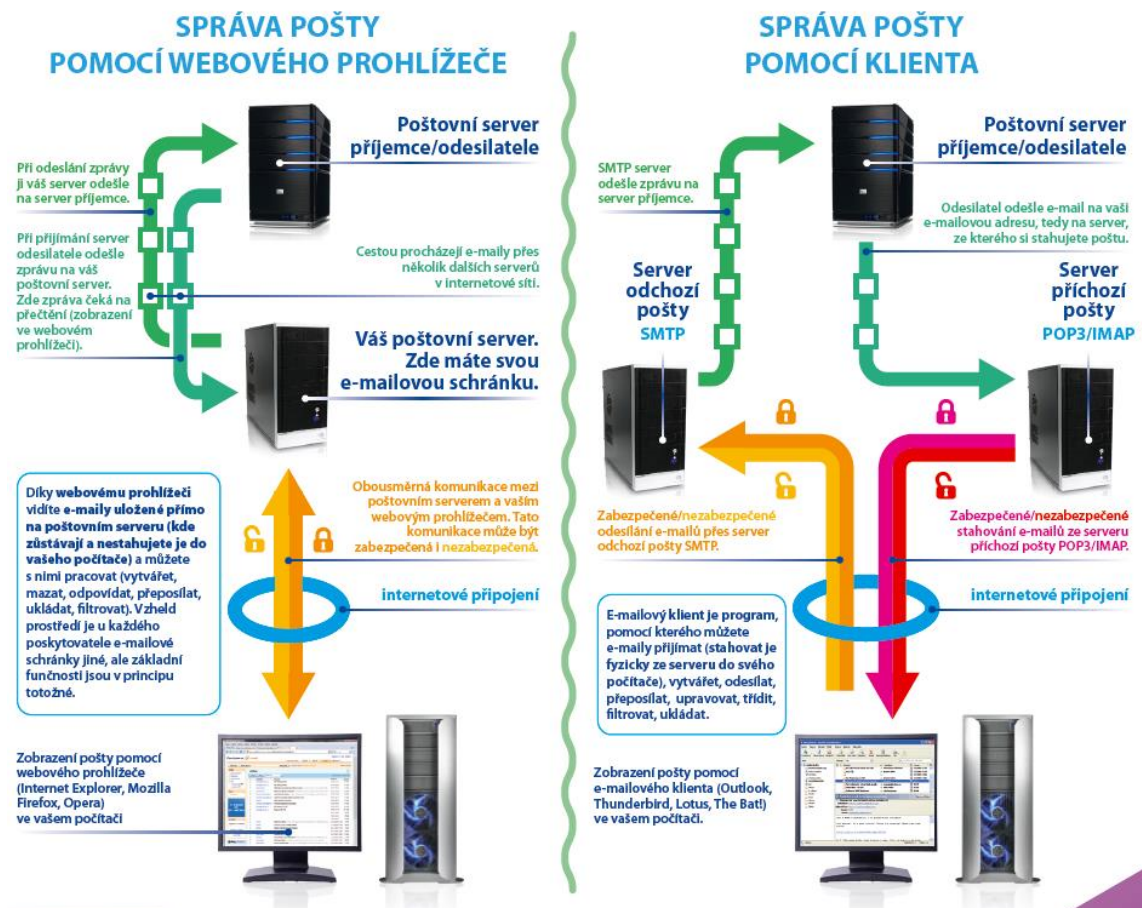
- mbox.tul.cz – zajišťuje příjem pošty a vede poštovní schránky uživatelů
- webmail.tul.cz – poskytuje přístup k elektronické poště prostřednictvím WWW rozhraní
- smtp.tul.cz – slouží k odesílání elektronických dopisů; jeho prostřednictvím také přichází elektronická pošta zvenčí

¹⁴ Převzato [online]. [vid. 1.9 2013]. Dostupné z: http://liane.tul.cz/cz/Elektronick%C3%A11_po%C5%A1ta



Obrázek 45: poštovní servery a přístup k poště - příklad TUL

Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: http://liane.tul.cz/cz/Elektronick%C3%A1_po%C5%A1ta



Obrázek 46: schéma správy pošty pomocí www klienta a poštovního klienta¹⁵

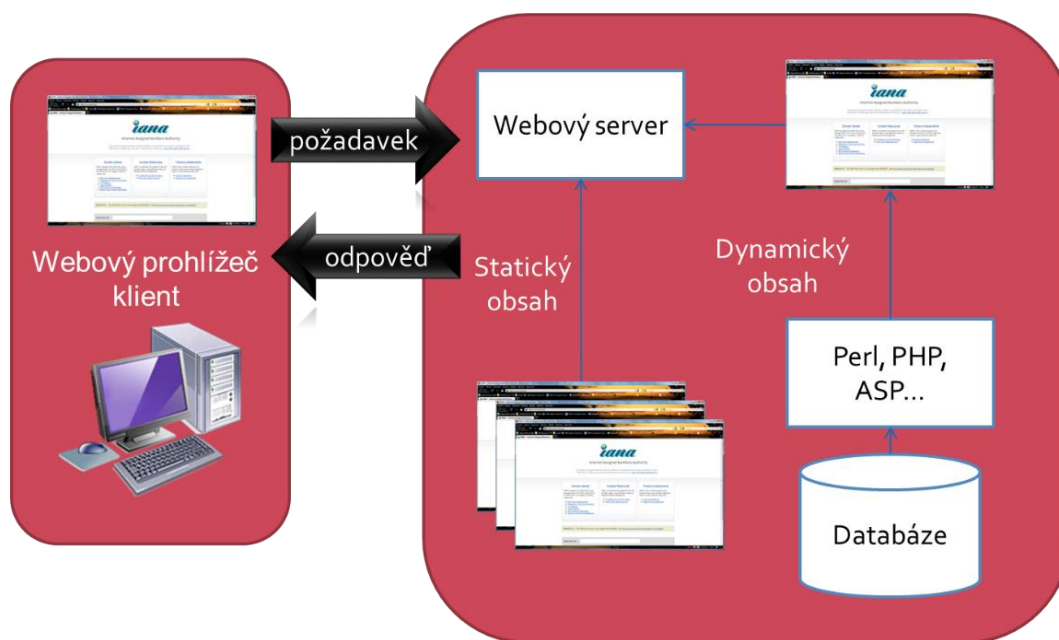
Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://www.urazydeti.cz/kit/postery/d03.pdf>

¹⁵ Převzato [online]. [vid. 1.9 2013]. Dostupné z: <http://www.urazydeti.cz/kit/postery/d03.pdf>

4.3.4 HTTP

Prostřednictvím aplikačního protokolu http (HyperText Transport Protokol, někdy též HyperText Transfer Protokol) je zabezpečována služba www stránek, WWW stránky jsou dokumenty psané ořevážně v html (HyperText Markup Language). Používá obvykle port TCP/80. Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz, obsahujícího označení požadovaného dokumentu, informace o prohlížeči apod. Server poté odpoví, zda se dokument podařilo najít, jakého typu dokument je atd., a zašle data samotného požadovaného dokumentu.

Při poskytování této služby je tedy využívána architektura klient/server, viz například Kapitola 3.4. Pod pojmem server si zde můžeme představit WWW server, který poskytuje informace v podobě WWW stránek, případně může poskytovat i další obsah. Pojem klient pak představuje prohlížeč (browser), který je nainstalovaný na zařízení uživatele (počítač, notebook, mobilní zařízení). Jeho úlohou je pak zobrazení webových stránek a případně přehrávání dalšího obsahu. Princip komunikace je znázorňuje obrázek.



Obrázek 47: Služba www - princip komunikace

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy*, výukový materiál OPF v Karviné

4.4 Organizace v prostředí internetu

I když se zdá být vývoj internetu a internetových technologií poněkud překotný, až chaotický, existují organizace, které se snaží svým způsobem tento vývoj usměrňovat. Následující text uvádí nejdůležitější z nich.

ISOC¹⁶ (INTERNET SOCIETY)

Tato nezisková organizace, založená v roce 1992 působí jako zastřešující orgán všech dalších organizací působících v oblasti internetu. Hlavním cílem organizace je podpora otevřeného rozvoje a využívání Internetu po celém světě, snaží se o koordinaci skupin odpovědných za vývoj standardů (podrobněji viz dále).

IETF¹⁷ (INTERNET ENGINEERING TASK FORCE)

IETF je vedoucí organizace v oblasti internetových standardů, která úzce spolupracuje s ISOC. Jedná se o sdružení odborníků z počítačového průmyslu, provozovatelů sítí a telekomunikací. Jejich hlavním cílem je rozvoj technických řešení, nových technologií, jejich výběr a příprava pro schvalování. Působnost této organizace je velmi široká, zasahuje do různých tematických oblastí, jako aplikace, protokol IP, Internet, správa sítí, bezpečnost apod.

IAB¹⁸ (INTERNET ARCHITECTURE BOARD)

IAB je organizace zabývající se otázkami celkové architektury internetu. Drží dozor nad organizací IETF. Má poradní funkci, provádí školení a řešení případných sporů. Zabývá se formálním vydáváním standardů, působí jako RFC editor. Zastřešuje činnost dalších organizací (IETF, IRTF, ICANN atd.).

IESG¹⁹ (INTERNET ENGINEERING STEERING GROUP)

Jedná se o organizaci zodpovědnou za technický management IETF a za schvalovací proces internetových standardů. Schvalovací proces provádí dle pravidel ISOC (podrobněji viz. dále).

IRTF²⁰ (INTERNET RESEARCH TASK FORCE)

IRTF je organizace zaměřená na vědu a výzkum. Jejím hlavním zaměřením je orientace na vývoj a budoucnost Internetu, protokolů, aplikací, architektury a internetových technologií.

¹⁶ ISOC. *Internet Society* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.internetsociety.org/>

¹⁷ IETF. *Internet Engineering Task Force* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.ietf.org/>

¹⁸ IAB. *Internet Architecture Board* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.iab.org/>

¹⁹ IESG. *Internet Engineering Steering Group* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.ietf.org/iesg/>

²⁰ IRTF. *Internet Research Task Force* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://irtf.org/>

IANA²¹ (INTERNET ASSIGNED NUMBERS AUTHORITY)

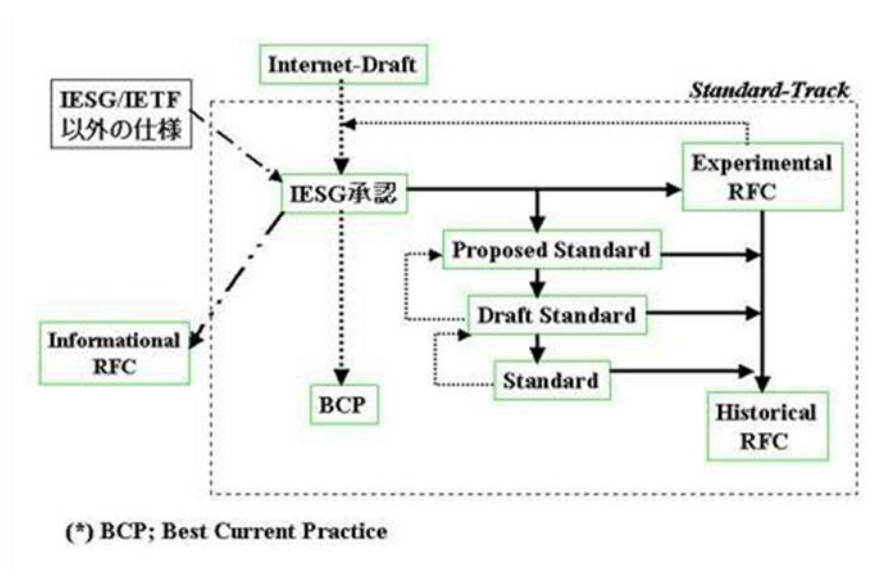
Organizace zabývající se správou doménových jmen a IP adres. V této oblasti má také na starosti vývoj a koordinaci celého procesu.

ICANN²² (INTERNET CORPORATION FOR ASSIGNED NAMES AND NUMBERS)

Organizace, která zastřešuje činnost regionálních registrátorů doménových jmen. Spolupracuje s organizací IANA.

4.5 Internetové standardy

Internetové standardy jsou dokumenty, které mají za cíl přispívat k procesu koordinace vývoje internetu a internetových technologií jako takových. Jejich význam je především ve sjednocování procesu komunikace mezi jednotlivými, často velmi odlišnými technologiemi, ať už v rovině hardwarové či softwarové. Standardy vznikají z tematicky sdružených RFC dokumentů, které prošly mnohdy náročným schvalovacím procesem (standard track).



Obrázek 48: RFC Standard Track

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy, výukový materiál OPF v Karviné*

²¹ IANA. *Internet Assigned Numbers Authority* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.iana.org/>

²² ICANN. *Internet Corporation For Assigned Names And Numbers* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.icann.org/>

4.5.1 RFC (REQUEST FOR COMMENT) DOKUMENTY

RFC dokumenty jsou dokumenty vydávané organizací IETF, zaměřené na metody, chování, výzkum a inovace aplikovatelné v prostředí internetu. Dokumenty jsou jednoznačně číslovány, jsou volně šiřitelné a dostupné pro širokou veřejnost na internetu.

Existují tři základní kategorie RFC dokumentů. První a nejdůležitější kategorií jsou **Standardy** – nejdůležitější kategorie. Dělí se dále na:

- *Proposed Standard* - počáteční úroveň, kdy ještě neexistuje dostatek zkušeností z praxe. Dokument sice může být kompletní z hlediska specifikace, jsou ale možné změny po stránce implementační či aplikační, např. RFC2535 (první definice DNSSEC),
- *Draft Standard* – dokumenty, na které jsou kladeny již větší nároky jak po stránce implementační, tak po stránce dostatečných zkušeností s provozem, např. specifikace směrovacího protokolu BGPv4,
- *Standard* – finální úroveň standardizace. RFC, které se dostane schvalovacím procesem až do kategorie Internet Standard, je již propracovaným a plně vyzrálým dokumentem. Jako příklad můžeme uvést poštovní protokoly SMTP a POP3, specifikaci kódování UTF-8 a mnoho dalších.

Dále existují tzv. ne-standardy, což jsou dokumenty, které z různých důvodů neprošly řádně schvalovacím procesem. Jedná se o dokumenty **Experimentální**, **Informativní** a **Historické**.

Poslední kategorií jsou dokumenty **BCP** (Best Current Practice), což jsou RFC dokumenty obsahující popis nejlepší současné praxe. Nedefinují tedy žádný konkrétní standard, ale působí jako jakýsi návod.

Podíváme-li se např. na oblast tvorby webu, nemůžeme zde nezmínit organizaci W3C²³ (World Wide Web Consortium), která se v této oblasti angažuje nejvíce. Zakladatelem této organizace, a v současné době také výkonným ředitelem je Tim Berners-Lee, zakladatel WWW. V této oblasti se setkáme hned s několika standardy. Již samotné zobrazování stránek a World Wide Web jako takový funguje na principu protokolu HTTP, který je také standardem, nemluvě o protokolech dalších, jako jsou TCP/IP využívaný v internetové síti, POP3 a SMTP pro zasílání pošty, FTP pro nahrávání (upload) a stahování (download) souborů apod. Při vytváření samotných webových stránek se rovněž vychází ze standardů, většinou příslušných zvolenému jazyku pro tvorbu webu. Mezi jinými zde můžeme zmínit např. HTML a XHTML jako základní jazyky pro tvorbu webu, XML jako formát pro přenos dat, CSS pro formátování a používání stylů, JavaScript jako zástupce skriptovacích jazyků na straně klienta, různé formáty pro multimedia (PNG, SVG, SMIL), MathML pro

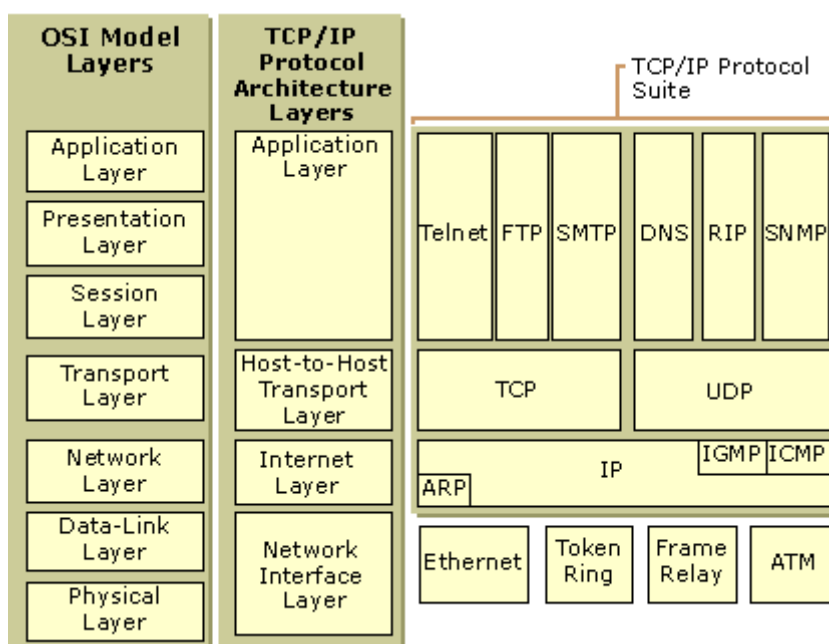
²³ W3C. *World Wide Web Consortium* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.w3.org/>

vkládání matematiky na web a další. Existuje další řada standardů, používaných v této oblasti, např. standardy týkající se přístupnosti webu (WCAG), standardy pro mobilní aplikace, pro různá zařízení přistupující k webu atd. Nebudeme zde rozebírat podrobně jednotlivé standardy, pouze je v následujících kapitolách zmíníme, pokud to bude nutné.

4.6 Síťové modely

Princip Internetu je založený protokolech, které zabezpečují komunikaci na několika „vrstvách“. Zjednodušeně řečeno, aplikace se mezi sebou snaží domluvit. Komunikace probíhá na vrstvě „aplikační komunikace“. Nepodaří-li se jim z nějakého důvodu zabezpečit vzájemnou komunikaci (např. nedostupnost počítače nebo neúplná znalost pravidel), obrátí se na další vrstvu, svá data „zabalí“ do balíčku, který předají této vrstvě a požádají tuto vrstvu o přenos. tohoto je zřejmé, že v prostředí internetu je několik komunikačních vrstev a na těchto vrstvách probíhají různé formy komunikace a přenosu dat. Ve většině služeb fungujících na Internetu pak komunikace probíhá tak, že jeden počítač poskytuje prostředky a určuje pravidla a jiný prostředky využívá, tj. komunikace probíhá na úrovni server – klient, viz Kapitola 3.4. Tento princip umožňuje pak komunikaci v heterogenním prostředí různých sítí, operačních systémů či hardware.

Pro vysvětlení, jakým způsobem komunikace v jednotlivých vrstvách sítě funguje, se velmi často používají tzv. síťové modely. Existují dva základní modely: model založený na rodině protokolů TCP/IP a referenční ISO/OSI model. Jelikož je daná problematika velmi obsáhlá a detaily jsou náplní jiného předmětu, vysvětlíme v této části publikace jen stručně základní principy. Srovnání obou těchto modelů je uvádí **Chyba! Nenalezen zdroj odkazů.**

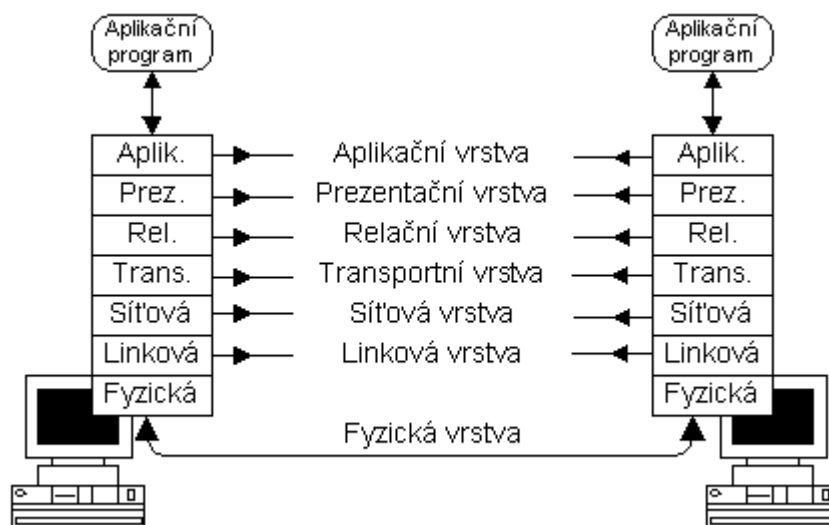


Obrázek 49: Srovnání síťových modelů

Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://technet.microsoft.com/en-us/library/cc958821.aspx>

4.6.1 ISO/OSI MODEL

Referenční model ISO/OSI vzniknul na základě snahy organizace ISO o standardizaci počítačových sítí. Tento model byl pak v roce 1984 přijat jako mezinárodní norma ISO 7498. Cílem tohoto modelu bylo poskytnout základ pro další normy, které slouží pro propojování systémů. Není zde tedy přímo specifikována samotná implementace systémů, ale jsou zde uvedeny principy pro základní komunikaci v síti pomocí sedmi vrstev. Na rozdíl od TCP/IP modelu nezahrnuje žádné protokoly a další detaily. V praxi se tento model využívá při programování jednotlivých součástí síťových rozhraní.



Obrázek 50: ISO/OSI model

Zdroj: Kabelová, Dostálek, 2010

Komunikace mezi vrstvami jednoho systému se pak řídí pravidly, která jsou nazývána jako rozhraní (interface). Komunikace mezi stejnými vrstvami různých systémů (zařízení) se pak nazývají podobně jako u TCP/IP modelu protokoly.

Jak již bylo zmíněno, model obsahuje sedm vrstev: fyzickou (*physical layer*), linkovou (*data link layer*), síťovou (*network layer*), transportní (*transport layer*), relační (*session layer*), prezentační (*presentation layer*) a aplikační (*application layer*).

Fyzická vrstva

Nejnižší, fyzická vrstva je zaměřena na fyzickou komunikaci v síti. Je zaměřena na fyzikální a elektrické vlastnosti zařízení, vlastnosti kabelů, stanovuje způsob přenosu. Hlavními funkcemi této vrstvy je navazování a ukončování spojení, modulace digitálních dat na signál používaný přenosovým médiem a efektivní rozložení zdrojů mezi uživatele v síti. Na této vrstvě pracují např. HUBy, opakovače a síťové adaptéry.

Linková vrstva

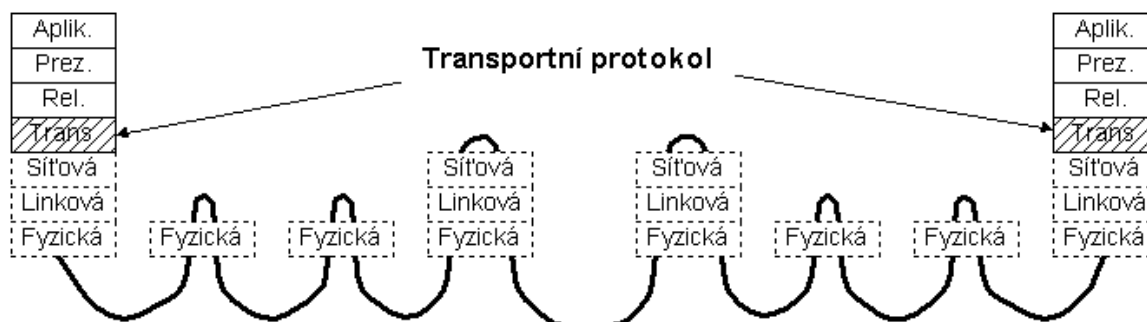
Linková vrstva má na starosti spojení mezi dvěma sousedními systémy. Definuje nastavení parametrů přenosu linek, oznamuje chyby. Na této vrstvě pracují např. mosty a přepínače.

Síťová vrstva

Tato vrstva má na starosti adresaci v síti. Zajišťuje tak spojení mezi systémy, které spolu bezprostředně nesousedí. Poskytuje směrovací funkce a stará se o doručení dat. Na této vrstvě tedy pracují např. směrovače. Můžeme zde zmínit také protokol IP, který pracuje na této vrstvě. Na síťové vrstvě je jednoznačně v celé WAN adresováno síťové rozhraní. Síťovým rozhraním může být např. karta pro Ethernet.

Transportní vrstva

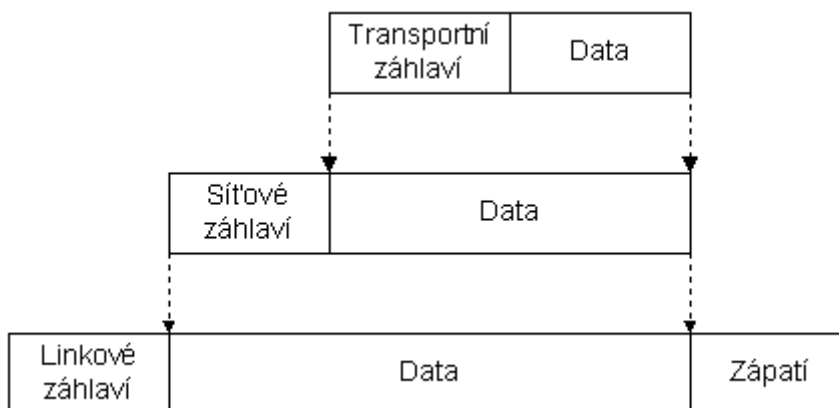
Transportní vrstva pak zajišťuje přenos dat mezi koncovými uzly. Její hlavní funkcí je starat se o kvalitu přenosu na takové úrovni, kterou vyžadují vyšší vrstvy. Můžeme se zde setkat se dvěma základními protokoly, spojově orientovaným TCP a nespojově orientovaným UDP.



Obrázek 51: Spojení na transportní vrstvě

Zdroj: Kabelová, Dostálek, 2010

Mezi dvěma počítači může být několik transportních spojení současně, jedno např. pro virtuální terminál a druhé pro elektronickou poštu. Aplikace jsou jednoznačně adresovány v rámci jednoho počítače. Jednotkou přenosu je transportní paket, který se opět skládá ze záhlaví a datové části. Transportní paket se přenáší v datové části síťového paketu.



Obrázek 52: Spojení na transportní vrstvě

Zdroj: Kabelová, Dostálek, 2010

Relační vrstva

Relační vrstva se stará o organizaci a synchronizaci mezi relačními vrstvami dvou systémů a řídí výměnu dat mezi nimi. Jedná se tedy o vytvoření a ukončení relačního spojení, synchronizaci a obnovení spojení apod. Na této vrstvě se můžeme setkat s protokoly jako NetBIOS, AppleTalk, SSL, RPC a další.

Prezentační vrstva

Na této vrstvě jsou pak data prezentována do tvaru, který používají aplikace. Jedná se např. o šifrování, konvertování nebo komprimaci. Vrstva je definována z důvodu možného rozdílu formátu dat při komunikaci různých aplikací na obou stranách. Vrstva se zaměřuje na strukturu dat, jejich význam pak přenechává vrstvě aplikační.

Aplikační vrstva

Na vrstvě aplikační se pak setkáme s protokoly umožňující komunikaci mezi aplikacemi a komunikačním systémem. Protokolů fungujících na této vrstvě je celá řada, můžeme zde uvést např. FTP, DNS, POP3, SMTP, DHCP, SSH a další.

4.6.2 TCP/IP MODEL

TCP/IP model je založen na rodině protokolů TCP/IP. Jeho vznik můžeme datovat do počátků sedmdesátých let devatenáctého století. Bývá také často nazýván jako *internet model*. Definuje čtyři kategorie funkcí, které musí být provedeny, aby byla komunikace na internetu úspěšná. Jedná se o otevřený standard, není tedy závislý na jedné instituci či firmě. Protokoly kolekce TCP/IP jsou veřejně posuzovány a diskutovány (byly především při jejich vzniku a průběžně vznikají další) prostřednictvím RFC dokumentů, viz Kapitola 4.5.

RFC dokumenty zahrnují také technické a organizační dokumenty týkající se internetu jako takového, včetně technických specifikací.

V TCP/IP modelu jsou definovány čtyři základní vrstvy, které znázorňují hierarchii činností. Mezi těmito vrstvami je přesně definována výměna informací, přičemž každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší. Jedná se o tyto vrstvy: aplikační (*application layer*), transportní (*transport layer*), síťová (*internet layer*) a vrstva síťového rozhraní (*network interface*). Komunikace mezi stejnými vrstvami je pak řízena tzv. komunikačními protokoly.

VRSTVA SÍŤOVÉHO ROZHRANÍ

Tato nejnižší vrstva umožňuje přístup k fyzickým médiím umožňujícím komunikaci v síti. Je samozřejmě specifická pro každou síť, podle její implementace. Jako příklad zde můžeme uvést např. Ethernet, Token ring, FDDI, X.25, SMDS a další.

SÍŤOVÁ VRSTVA

Síťová vrstva má na starosti síťovou adresaci, předávání tzv. datagramů a jejich směrování. Na této vrstvě se můžeme setkat s protokoly IP, ARP, ICMP a dalšími.

TRANSPORTNÍ VRSTVA

Transportní vrstva je pak implementována až v koncových zařízeních přistupujících k internetu. Poskytuje transportní služby prostřednictvím protokolu TCP (*transmission control protocol*) nebo méně spolehlivým protokolem UDP (*user datagram protocol*).

APLIKAČNÍ VRSTVA

Z hlediska služeb internetu nás bude zajímat nejvíce poslední vrstva, vrstva aplikační. Jedná se o vrstvu, která má na starosti procesy aplikací využívajících přenosu dat po síti pro poskytování konkrétních služeb. Jako příklad zde můžeme uvést FTP, HTTP, DHCP, Telnet, DNS, POP3, SMTP a další. Některé ze zmiňovaných služeb budou podrobněji rozebrány v následujících kapitolách.



SHRNUTÍ KAPITOLY

V této kapitole jste se seznámili se základy fungování Internetu, zejména s protokolem TCP/IP a jeho jednotlivými vrstvami. Dále jste se seznámili s protokoly aplikační vrstvy a službami provozovanými na této vrstvě.

5 PILÍŘE A ORGANIZACE INTERNETU, WWW

RYCHLÝ NÁHLED KAPITOLY



V této kapitole bude představen základní koncept Internetu, tři stěžejní pilíře na kterých je internet postaven. Studenti se seznámí s organizací a přidělováním IP adres a doménových jmen a s fyzickou infrastrukturou.

CÍLE KAPITOLY



Cílem kapitoly je seznámit studenty se základními pojmy, se kterými se setkají při práci s Internetem, zejména s hlavními pilíři. Studenti se blíže seznámí s fyzickou strukturo, vztahem mezi doménami a IP adresami a přidělováním doménových jmen a IP adres

ČAS POTŘEBNÝ KE STUDIU



Čas potřebný ke studiu v rozsahu 3-5 hodin, podle míry praktické práce.

KLÍČOVÁ SLOVA KAPITOLY



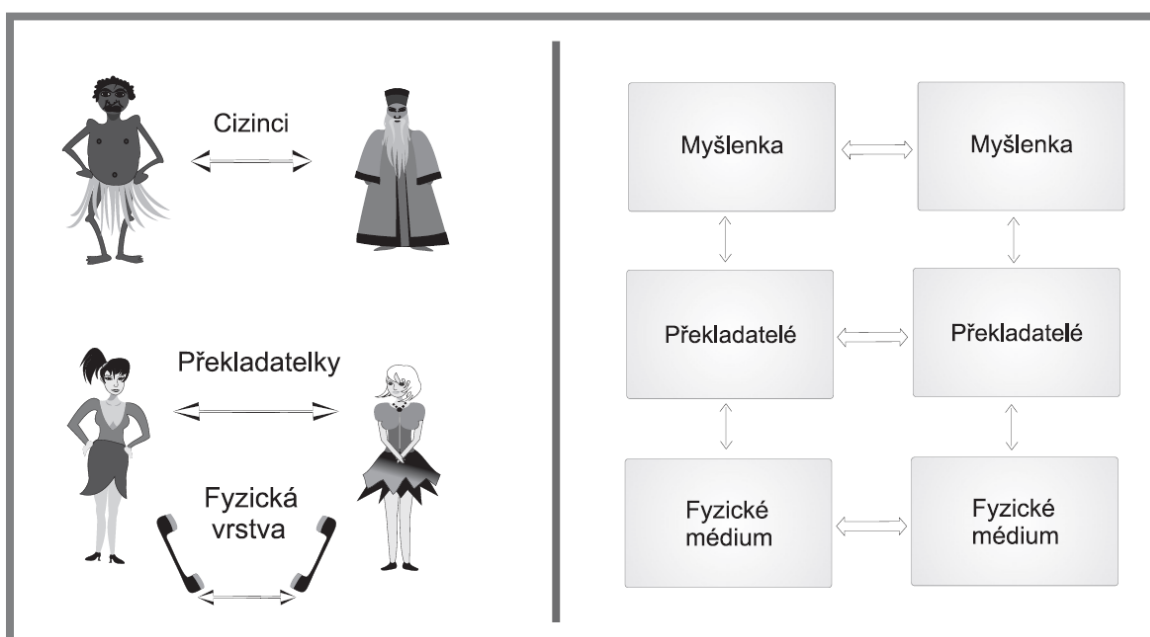
ISP, URL, DNS, doména, infrastruktura, Cesnet

5.1 Tři pilíře Internetu

Dá se jednoduše shrnout, že Internet má 3 základní pilíře, na kterých je postaven. V první řadě je to komunikační protokol TCP/IP, dále systém doménových adres a v neposlední řadě fyzická infrastruktura.

5.1.1 KOMUNIKAČNÍ PROTOKOLY

Komunikační protokoly tvoří jeden ze základních stavebních kamenů Internetu. Vzhledem k multiplatformnímu prostředí Internetu je nutné zabezpečit taková komunikační pravidla, která konečným počtem standardizovaných kroků zabezpečí komunikaci. Dostálek a Kabelová (2012) přirovnávají komunikaci k rozhovoru dvou vzdálených cizinců. Každý umí mluvit, ovládá proto komunikační protokol. Každý má však tento protokol odlišný (např. čeština, francouzština). Komunikace tedy teoreticky probíhá mezi cizinci, prakticky však cizinci musí předat komunikaci jiné komunikační vrstvě, která „má stejný protokol“ – tlumočnicím.



Obrázek 53: Třívrstvá komunikační architektura

zdroj: Kabelová, Dostálek (2012)

Prakticky tedy cizinci nehovoří spolu ale s tlumočnicemi. Pokud nejsou tlumočnice „v kontaktu“, nemohou se tedy domluvit, sáhnou opět po další komunikační vrstvě. Teoreticky tedy hovoří spolu tlumočnice, prakticky však každá tlumočnice hovoří „s telefonem“. Takto může být komunikace definovaná na více vrstvách, musí však být zabezpečené, že se na některé vrstvě komunikace uskuteční (definovaný konečný počet vrstev).

Komunikace mezi počítači (obecněji mezi aktivními síťovými prvky) v prostředí Internetu je založena na protokolu TCP/IP.

5.1.2 FYZICKÁ INFRASTRUKTURA

Aby mohla komunikace mezi počítači v Internetu probíhat, musí být k dispozici komunikační kanály, musí být vytvořena komunikační (síťová infrastruktura).

Internet využívá množinu takovýchto infrastruktur, jejichž provozovatelé (provideři) umožňují připojení uživatelů do sítě. Propojení sítí se realizuje prostřednictvím peeringu. Peering je pojmenování pro vzájemné propojení počítačové sítě dvou různých (např. telekomunikačních) společností za účelem výměny datového provozu na centrálních místech (exchange points) individuálně nebo hromadně. Výsledkem vzájemného propojení všech takových sítí po celém světě je tedy Internet (celosvětová síť). Peeringové uzly jsou páteří celého Internetu. Peering může být privátní - přímé propojení sítí dvou ISP nebo veřejný (public) - hromadné propojení více sítí v jednom místě.

V praxi se pro označení Peeringových uzlů používají zkratky IXP – Internet Exchange Point²⁴, NAP – Network Access Point, MAE – název společnosti Metropolitan Area Exchange, který se v USA vžil pro generické označování NAPs, aj. V rámci evropského prostoru se nejčastěji používá označení IXP. IXP je fyzickým místem, ve kterém si ISPs (Internet Service Providers) a CPs (Content Providers) vyměňují své datové toky. IXP snižuje společnostem náklady na zahraniční konektivitu. Analogií může být letiště-dopravní křižovatka (IXP), kde pasažéři (datové balíčky), přestupují, vystupují a nastupují do a z letadel různých dopravních společností (ISP, CP). Z pohledu provozovatele lze rozdělit IXP na 2 kategorie:

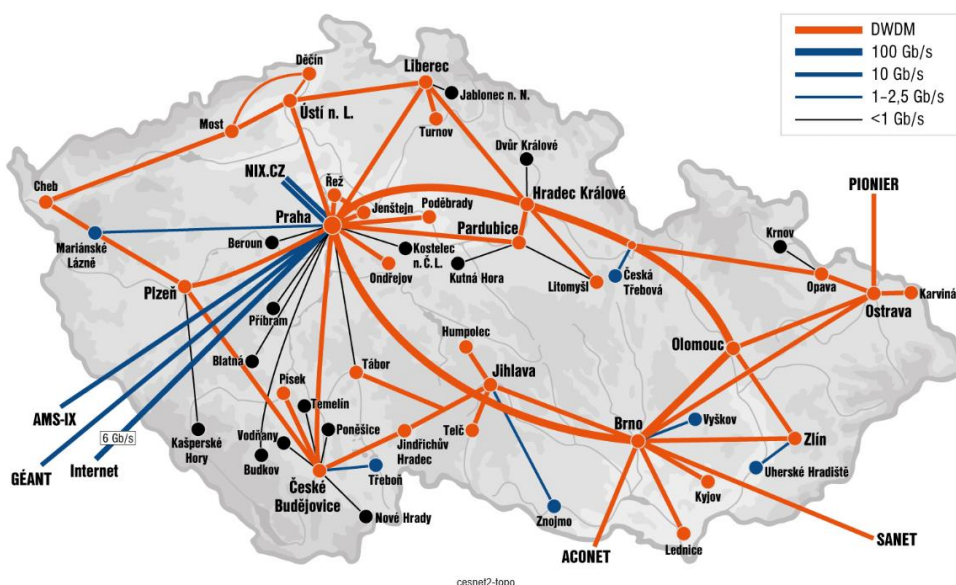
- Nekomerční IXP – asociace, neutrální organizace, kde vloženými penězi (ve formě poplatků za využívání IXP) sdružené společnosti financují provoz svého IXP.
- Komerční IXP – provozovatelem je soukromá firma, ISP jsou její zákazníci a poplatky platí provozovateli.

V Evropě jsou rozšířeny spíše nekomerční IXP, v USA převládá kategorie komerčních IXP. Obě kategorie mají své klady a zápory. V nekomerčním IXP sdružené společnosti samy rozhodují o vývoji, strategii, cenách a investicích svého IXP, naproti tomu v komerčním IXP platí zainteresované společnosti za služby poskytovateli služeb, který může být přímým konkurentem, a mají pouze dvě možnosti v rozhodování, a to sice buď služby odebírat, nebo neodebírat. Na druhé straně v nekomerčním IXP mají sdružené společnosti i příslušný díl zodpovědnosti za svá rozhodnutí a konsensus společností musí projít diskuzí a zpravidla v některých IXP hlasováním, zatímco komerční IXP dělá rozhodnutí dle svého uvážení, což umožňuje pružnější reakci na podněty z vnějšího okolí.

Mezi významná veřejná peeringová centra patří NIX.CZ. V současné době je sdružení největším neutrálním IXP v České republice a řadí se mezi deset největších IXP v Evropě. Síť připojené k platformě NIX.CZ mají možnost propojení s dalšími připojenými sítěmi ISP (Internet Service Providers) nebo IAP (Internet Access Provider). NIX.CZ patří mezi nekomerční IXP kde členové hlasují o ceníku, rozpočtu, vzniku/zániku nových uzlů, a volí statutární orgány.

²⁴ Zdroj: <http://www.lupa.cz/clanky/nix-cz-minulost-soucasnost-a-budoucnost/>

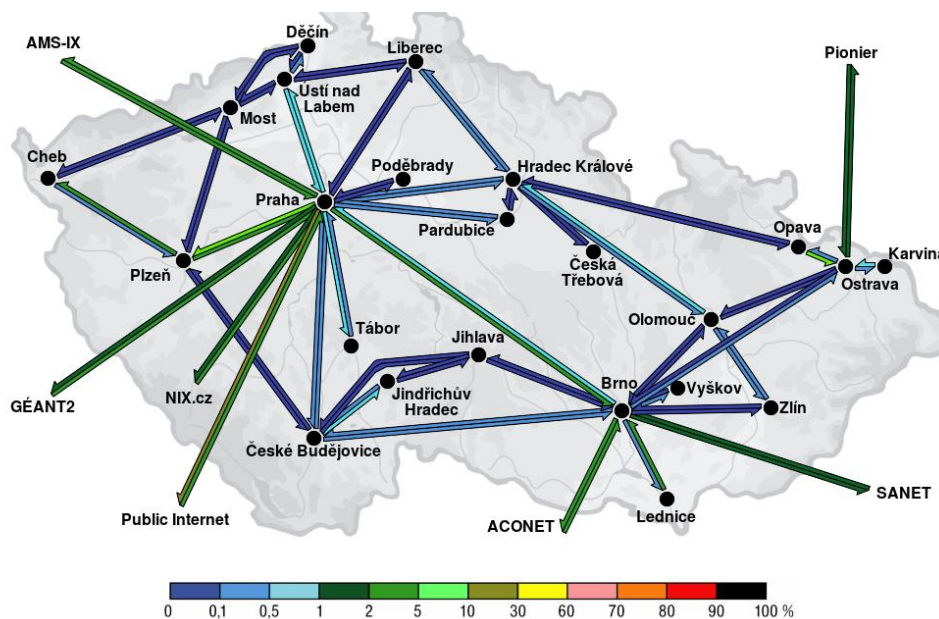
Slezská univerzita využívá síť TEN společnosti Cesnet. Síť Cesnet je součástí struktury Geant.



Obrázek 54: topologie sítě Cesnet

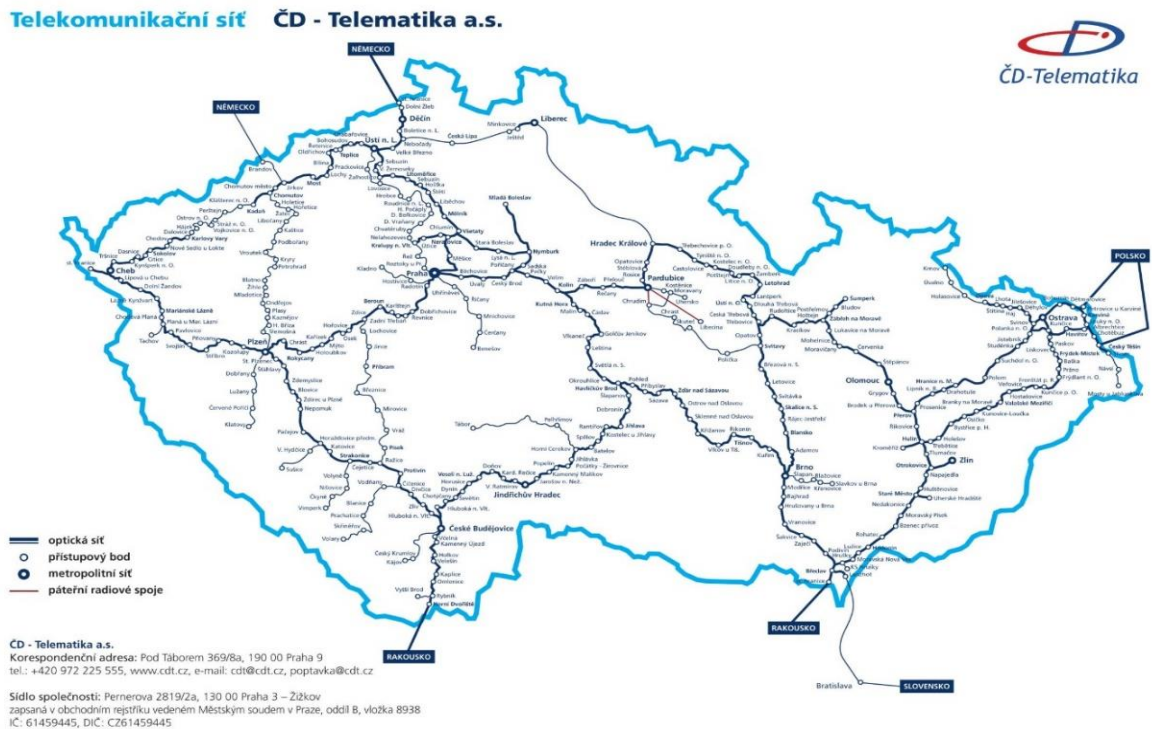
Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://www.cesnet.cz/sluzby/pripojeni/topologie/>

Na stránkách cesnet.cz jsou veškeré záležitosti týkající se této infrastruktura, včetně realizovaných propojení, projektů nebo např. aktuálního zatížení sítě.



Obrázek 55: Ukázka zatížení jednotlivých tras sítě Cesnet

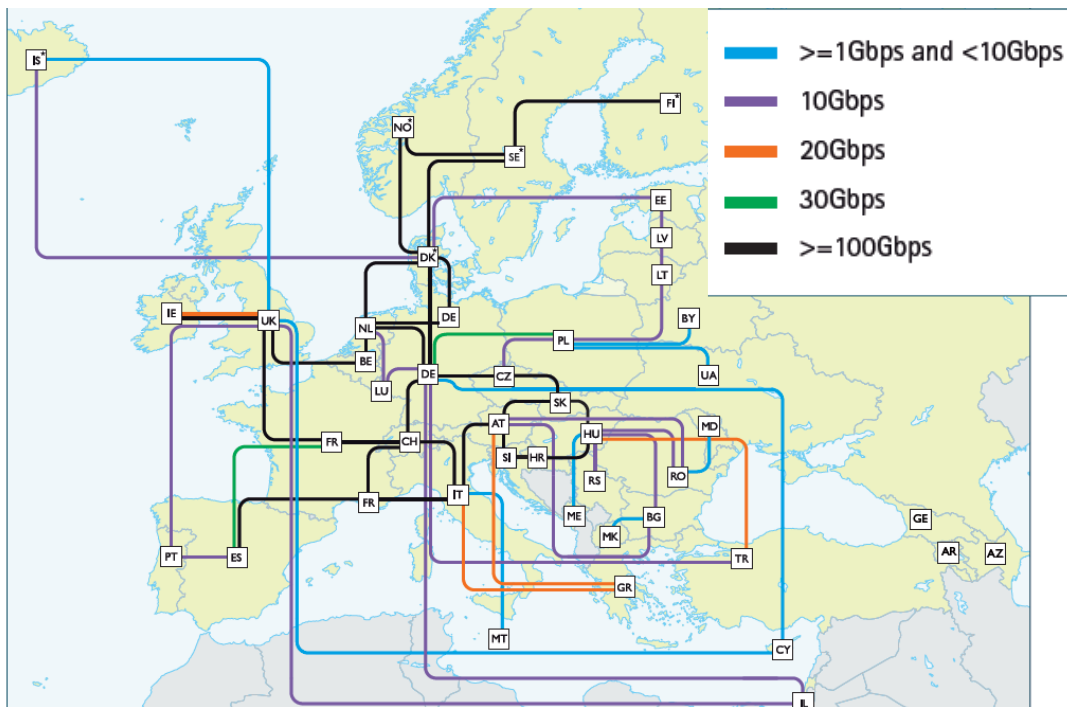
Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://netreport.cesnet.cz/netreport/>



Obrázek 56: ukázka topologie sítě ČD-Telematika

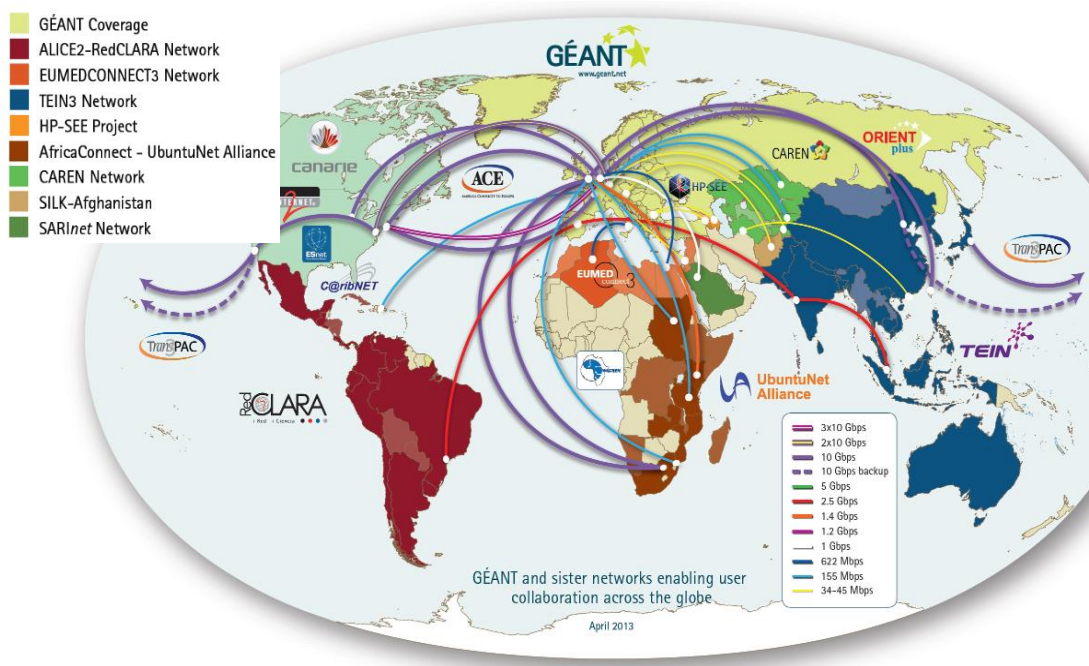
Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: <http://www.cdt.cz/cz/infrastruktura-97/>

V rámci konektivity se sítě propojují i v rámci států, resp. vytvářejí se nadnárodní struktury. Příkladem může být Geant – Pan evropská síť pro vědu a vzdělávání.



Obrázek 57: Pan evropská síť pro vědu a vzdělávání

Zdroj: Cesnet, [online]. [vid. 1.9 2013]. Dostupné z: www.geant.net



Obrázek 58: Světová konektivita - Geant

Zdroj: [online]. [vid. 1.9 2013]. Dostupné z: http://www.geant.net/Resources/Media_Library/Pages/Maps.aspx

5.1.3 DOMÉNY

Třetím pilířem Internetu je systém doménových adres. Na základě přidělování IP adres a doménových adres je vytvořena struktura doménových jmen, kdy každá doménová adresa je ve své podstatě „jinak zapsaná“ IP adresa příslušného počítače. Identifikaci (překlad adres) zabezpečují DNS servery na základě protokolu DNS. Blíže se doménám věnují kapitoly později.

5.2 Fungování internetu

Internet je v podstatě rozsáhlá počítačová síť, která vzniká propojením několika stovek sítí. Internet je tedy spousta spojených počítačů, pokud se však propojí několik počítačů, ještě to neznamená, že máte dílčí mini internet.

K tomu, aby informace byly vidět je potřeba mít na minimálně jednom počítači internetový server. Internetový server je tedy opět počítač, který umožní sdílet určité informace okolnímu světu. Opakem je klient. Klientem je každý počítač, co je připojen k internetu a má nainstalovaný prohlížeč²⁵. Zjednodušeně řečeno tedy pokud spustíme prohlížeč, tak ten se nám připojí k nějakému serveru a zobrazí nám informace, které na tom serveru jsou. To, o který server se jedná, je určeno jeho adresou. Každá tato adresa je jedinečná a v internetu nemohou být dvě stejné.

Pojem Internet (složenina předpony inter vyjadřující vztah mezi, a anglického net označujícího síť) se v tomto případě vysvětluje jako celosvětový systém navzájem propojených počítačových sítí, které propojují tzv. síťové uzly. Uzlem pak může být počítač nebo zařízení se speciální síťovou funkcí, například router. Klíčovou pro fungování Internetu, jak jej známe dnes, je také rodina protokolů TCP/IP²⁶.

Tyto dílčí sítě jsou provozovány subjekty (organizacemi), často nazývanými ISP (Internet Service Provider), tedy poskytovateli připojení. Tito poskytovatelé připojení vytvářejí fyzické propojení počítačů v internetu, přičemž často nabízejí také další služby s touto oblastí spojené. Jak je patrné z následujícího obrázku, tyto subjekty poskytují připojení také

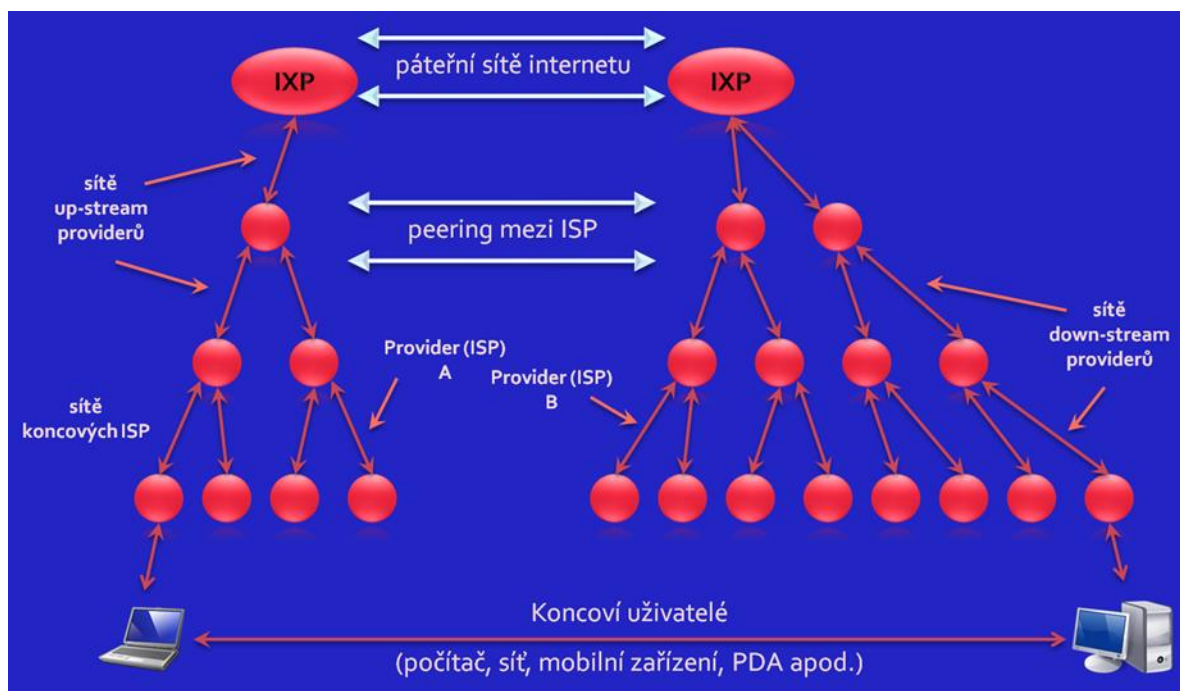
²⁵ <http://www.zasedalsiblog.cz/zjednodusena-struktura-internetu>

²⁶ Co Internet není? (podle <http://www.jaknainternet.cz/page/1795/struktura-internetu/>)

Obvykle bývá internet zaměňován za internetové stránky, prohlížeč, součást vybavení počítače či telefonu nebo určitou on-line službu. Někdy z neznalosti, častěji ale spíše ze snahy zobecnit význam pojmu Internet, aby lépe odpovídal tomu, jak jej známe z běžného života. Lze tedy rozlišovat mezi Internetem v užším a širším slova smyslu.

WWW stránky, e-mail atd. (Internet podle laika), jsou jen některé z tzv. služeb, které je možné na Internetu provozovat. Tyto služby zajišťují počítačové programy, které mezi sebou komunikují pomocí protokolů (protokoly jsou definovány jako seznam doporučení, jejichž dodržování vede k bezproblémovému fungování služeb). Další známé internetové služby jsou například instant messaging (protokoly ICQ, Jabber...), VoIP - internetová telefonie (protokol SIP, proprietární, tedy uzavřený protokol Skype...), přenos souborů (FTP) atd.

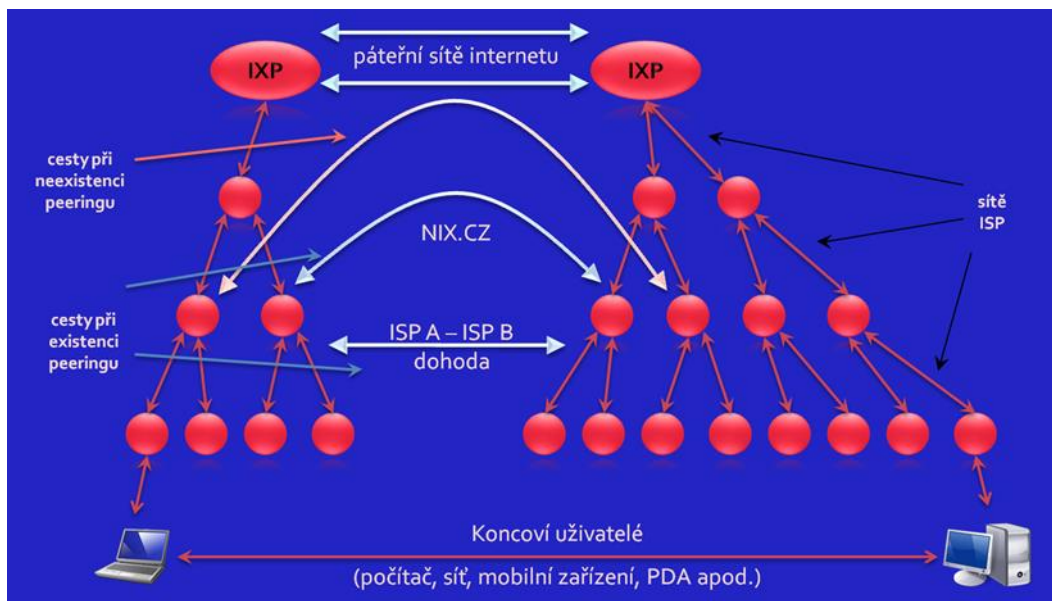
dalším subjektům. Pokud bychom se na toto jednoduché schéma dívali z pohledu hierarchie, budeme o větších poskytovatelích, resp. o jejich sítích, hovořit jako o sítích up-stream providerů. Tyto subjekty pak připojují menší subjekty, resp. jejich sítě (down-stream sítě providerů). Na obrázku jsou znázorněny tyto sítě obousměrnými šipkami. Na nejnížší úrovni této pomyslné hierarchie pak nalezneme sítě koncových providerů, tedy sítě poskytovatelů připojení, kteří do internetu fyzicky připojují koncové uživatele.



Obrázek 59: Zjednodušené schéma internetu

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy*, výukový materiál OPF v Karviné

Rychlost připojení v rámci jednotlivých sítí jednotlivých poskytovatelů připojení se liší v závislosti na použité technologii a v závislosti na jejich propojení se sítěmi ostatních poskytovatelů připojení. Ve skutečnosti internetová síť nevypadá takto jednoduše hierarchicky, byť zřejmě k nejrychlejšímu propojení dochází stále přes páteřní síť internetu, ke které jsou prostřednictvím přípojných bodů nazývaných IXP (Internet Exchange Point) připojeni největší poskytovatelé připojení. Konektivita, tedy rychlost spojení mezi dvěma body v internetu (někdy také rychlost připojení dané sítě vzhledem k určenému cíli), je dána způsobem propojení jednotlivých sítí mezi sebou.



Obrázek 60: Zjednodušené schéma internetu – peering

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy, výukový materiál OPF v Karviné*

Obrázek ukazuje, že v případě nepropojených sítí dvou poskytovatelů připojení, např. A a B, by data od koncového uživatele musely cestovat nejprve v rámci hierarchie sítě poskytovatele A až k páteřní síti internetu, a pak opět shora dolů v rámci hierarchie až k poskytovateli B. Konektivita je pak dána nejslabším článkem v rámci této cesty, tedy nejpomalejším spojením (opět dáno použitými technologiemi, často nejpomalejší spojení je mezi koncovým uživatelem a koncovým ISP, např. bluetooth, wi-fi spojení, či modem). Z důvodu zkrácení těchto cest a samozřejmě z důvodu urychlení přenosu informací dochází k propojování sítí jednotlivých ISP na nižší úrovni. Toto propojování je prováděno na základě bilaterálních dohod, popř. je zastřešováno organizacemi v rámci jednotlivých IXP (Internet Exchange Point). Tyto výměnné body pak poskytují infrastrukturu pro propojení jednotlivých sítí autonomních systémů jednotlivých ISP. V České republice působí zájmové sdružení ISP právě s cílem propojení sítí jednotlivých poskytovatelů připojení s názvem NIX.CZ²⁷ (Neutral Internet eXchange).

5.2.1 SLUŽBY ISP

Poskytovatelé připojení v současné době kromě samotného fyzického připojení nabízejí také další služby, jako např. web hosting, server hosting, registraci doménových jmen, tvorbu webových stránek a vývoj internetových aplikací, správu databází či další služby (VoIP, vyhledávací služby, mail servery apod.).

²⁷ NIX.cz, z.s.p.o. *Neutral Internet eXchange* [online]. [vid. 15. Prosince 2013]. Dostupné z: <http://www.nix.cz>

Také dochází s nástupem digitalizace k prolínání dvou, dříve samostatných, světů počítačových sítí a telekomunikací. Dochází jak ke konvergenci sítí, tak poskytovaných služeb. Není dnes výjimkou, že poskytovatel připojení, počítačová firma, poskytuje služby v oblasti IP telefonie (VoIP), že se setkáváme s televizním a rozhlasovým vysíláním po internetu (IPTV), že internetové připojení nabízí kabelové televizní společnosti apod.

5.2.2 TIER

Obecně se velké sítě dělí podle své velikosti do tříd zvaných „Tier“. „Tier 1 jsou společnosti, které stály u zrodu internetu nebo byly velkými operátory na důležitých trzích. Tyto sítě se mezi sebou propojují zdarma a ostatní od nich musí ‚nakupovat internet‘,“²⁸. Společností s označením Tier 1 je jen okolo patnácti. „Typicky jsou globální, mají tedy přípojně body na všech kontinentech.“

Dalším zajímavým prvkem internetu jsou takzvané Content Delivery Network (CDN), což jsou sítě pro rozložení zátěže pro distribuci software a jiných dat. Zlepšují uživatelský komfort, protože umožňují stahovat obsah z nejbližšího úložiště. Mezi nejznámější sítě patří Akamai, Amazon nebo Limelight. „Pokud by takové sítě neexistovaly, docházelo by k přetížení nejen u samotných serverů, ale i třeba na podmořských linkách.“¹⁰

Všichni poskytovatelé kromě Tier 1 musí nakupovat „tranzit“, tedy připojení do celého internetu. Tranzit se nakupuje buďto za fixní cenu nebo běžnější metodou podle přenesených dat.

5.3 Identifikace souborů a počítačů

Aby byla v prostředí internetu možná komunikace, je nutné vědět, s kým daná komunikace probíhá. To je zajištěno jednoznačnou identifikací všech zařízení a objektů v internetovém (sít'ovém) prostoru. Jedná se nejen o servery (od webových serverů přes file servery, záložní servery až po koncová zařízení jednotlivých uživatelů. Jako příklad zde můžeme uvést server, na kterém jsou umístěny soubory s www stránkami. WWW stránka je v podstatě soubor nebo množina souborů, které jsou umístěny na disku nějakého počítače (zpravidla serveru, který je přístupný 24 hodin denně, 7 dní v týdnu, 365 dní v roce). Ten, kdo chce zobrazit www stránku, by tedy měl zadat název souboru, který chce zobrazit, jeho umístění na příslušném disku a umístění příslušného počítače. Tyto údaje musí být současně jednoznačné a světově jedinečné, aby se příslušný soubor opravdu našel. Jednou z podmínek pro distribuci www stránek je tedy zřízení jednoznačné adresy, za kterou „schováme“ naše soubory na příslušných discích v příslušných počítačích.

²⁸ Golecký, A., Lehký úvod do peeringu aneb jak funguje NIX.CZ, přednáška, dostupné na: <https://www.root.cz/clanky/lehky-uvod-do-peeringu-aneb-jak-funguje-nix-cz/>

5.3.1 URI, URL, URN

Jak bylo zmíněno v předchozím příkladu, potřebujeme tedy pro zobrazení www stránky nějakou adresu, na kterou naše dokumenty uložíme a kde si je uživatelé Internetu najdou. Tato adresa musí, obdobně jako „klasická domovní“ adresa splňovat určitá pravidla. V této souvislosti se zjednodušeně někdy místo „adresa“ používá pojem „odkaz“ na soubor, či „URL odkaz“ na soubor. V praxi se pak můžeme setkat s pojmy URI odkaz, URL odkaz a URN odkaz. URI je množina obsahující podmnožiny URL a URN. URI adresa je obecné označení, zatímco URL adresa a URN adresa jsou konkretizující pojmy.

URI

URI (*Uniform Resource Identifier* - jednotný identifikátor zdrojů) představuje způsob, jak identifikovat jednotlivé zdroje v Internetu. Zkratka URI není tolik známá a rozšířená jako URL, jedná se spíše o obecný pojem, který řeší umístění zdroje i jeho pojmenování.

URN

URN (*Uniform Resource Name* - jednotný identifikátor jména) popisuje především název internetového zdroje, nestará se už o jeho dostupnost, resp. cestu k němu.

URL

URL (*Uniform Resource Locator* - jednotný lokátor zdroje) je standardně nejrozšířenější způsob uvádění adresy, často je zaměňován s URI. Běžně se jako URL adresa označuje řetězec znaků, které popisují konkrétní umístění dat. URL adresa popisuje tedy především místo, kde je daný dokument (obecněji data) umístěný. URL adresu může tvořit několik částí, které dohromady vytváří konkrétní adresu umístění dokumentu. URL adresa zpravidla obsahuje doménové jméno a schéma (resp. protokol), dále pak může (ale není to povinné) obsahovat tzv. port (identifikuje službu Internetu: Web, pošta, stahování souborů apod.), název souboru a další parametry. Například „http://www.opf.slu.cz/html/vyuka.htm“ je URL odkaz ukazující na dokument „vyuka.htm“, ten leží v adresáři „html“ na počítači s názvem (s doménovým jménem) „www.opf.slu.cz“. Tento dokument je dostupný pomocí protokolu http, viz Kapitola 4.6.

5.3.2 DOMÉNA, IP ADRESA

Níže uvedený výklad je opět účelově zjednodušený a slouží pro pochopení principu. V předchozí podkapitole je uvedeno, že se v daném URL jedná o název počítače „www.opf.slu.cz“. Přesněji, jedná se o počítač, který je v nějakém prostoru nazvaný www.opf.slu.cz. Tomuto názvu se říká *doména*, resp. *doménové jméno*. Doménové jméno můžeme přirovnat opět ke „klasické“ adrese, v níž jsou jednotlivé údaje odděleny tečkou (příklad: „karel.novak.kosmonautu.karvina.ceska_republika“). Jednotlivé části pak nazýváme domény *k-tého řádu*, přičemž *k* udává pozici zprava a směrem doleva se vždy údaj

upřesňuje a konkretizuje. U `www.opf.slu.cz` je tedy „cz“ doména prvního řádu, „slu.cz“ je doména druhého řádu (existuje množina domén druhého řádu „XXX.cz“, které jsou podmnožinou domény „cz“), „opf.slu.cz“ je doména třetího řádu (opět existuje množina domén 3. řádu „YYY.slu.cz“, které jsou podmnožinou domény „slu.cz“) atd. Doménu musíme mít, abychom mohli být identifikovatelní. Je to jakýsi „prostor“, ke kterému se návštěvníci našich stránek dostanou a ve kterém máme naše soubory (texty, fotky, videa apod.) potřebné pro naše stránky. Je zřejmé, že někde musíme tento „prostor“ umístit a musíme tento „prostor“ zpřístupnit ostatním. Pokud bychom tento „prostor“ umístili na svém počítači, dobře by se nám stránky vytvářely, protože bychom mohli pracovat na svém počítači. Těžko bychom ale zabezpečili dostupnost našeho počítače pro všechny návštěvníky našich stránek. Přesto je zřejmé, že naše soubory jsou na nějakém počítači. Tento počítač musí být jednoznačně identifikovatelný v prostředí internetu.

Prostředkem pro jednoznačnou identifikaci je tzv. *IP adresa*. Adresa (ve verzi IPv4) má tvar „XXX.XXX.XXX.XXX“ (čtyři tříčíselné části oddělené tečkou), kde XXX je v rozsahu 0 až 255 (např. 193.160.100.100). Tuto adresu má každý počítač v internetu. Aby byla jednoznačně zabezpečená identifikace, jsou tyto adresy přidělovány podle stanovených pravidel. Jednoznačnost IP adresy musí být zabezpečena v rámci uzavřené skupiny, pokud existuje více zařízení se stejnou IP adresou, jsou identifikované skupinou, ve které se nachází (opět je výklad účelově zjednodušený).

Adresa IP se skládá ze dvou částí. Net - ID (adresa sítě) a Host - ID (adresa počítače)²⁹. Podle toho jak jsou jednotlivé sítě rozlehlé (kolik mají hostů) rozlišujeme tři hlavní třídy IP adres - A, B a C.

Třída A: dovoluje adresování jen 126 sítí, ale v každé z nich může být až 16 miliónů počítačů. Rozsah hodnot IP adres je: 0.0.0.0 až 127.255.255.255.

Třída B: umožňuje adresovat už 16 tisíc sítí a 65 tisíc počítačů v každé síti. První dva byte je adresa sítě a další dva adresa počítače. V České republice ji mají významné organizace. Rozsah hodnot ve třídě B je: 128.0.0.0 až do 191.255.255.255.

Třída C: umožňuje adresovat až 2 milióny sítí. V každé síti může být 254 počítačů. IP adresa třídy C je v ČR nejpoužívanější. První tři byte jsou adresou sítě a jeden byte adresou počítače. Rozsah je: 192.0.0.0. až 223.255.255.255.

Speciální IP adresy: některé IP adresy jsou vyhrazeny pro speciální účely. Rozsah adres 224.0.0.0 - 239.255.255.255 je zařazen do třídy D a je využívána pro multicasting. Adresy 240.0.0.0 - 247.255.255.255 patří do třídy E, jsou rezervovány. Adresy 127.0.0.0 a 127.0.0.1 jsou tzv. loopback adresy. Pošleme-li data na tuto adresu, nebudou vysílána přes

²⁹ [online]. [vid. 1.9 2013]. Dostupné z:<http://site.the.cz/?id=2>, část textu převzata pro výukové účely

žádný ze síťových adaptérů počítače do sítě. Pouze zjistíme, zda je funkční software, nezávisle na tom, funguje-li síťový hardware. Adresu 127.0.0.1 má v podstatě každý počítač, identifikuje lokální počítač.

Síťové adresy jsou adresy, jejichž host část obsahuje samé nuly. Tyto adresy jsou využívány IP protokolem ke správnému směřování paketů mezi sítěmi.

Broadcast adresa, 255.255.255.255 je určena všem hostům v dané síti. Používá se k hromadnému rozesílání paketů.

Pokud je síť izolovaná (Intranet), lze použít libovolné IP adresy. Při připojení vnitřní sítě k Internetu by ale mohla nastat situace, že bude existovat více shodných IP adres. V tomto případě je počítač vnitřní sítě identifikovatelný IP adresou brány. Z důvodů minimalizace konfliktů jsou pro vnitřní síť rezervované IP adresy 10.0.0.0 až 10.255.255.255 (třída A), 172.16.0.0 až 172.31.0.0 (třída B) a 192.168.0.0 až 192.168.255.0 (třída C).

Jak bylo dříve uvedeno, na zdroje se odkazujeme pomocí doménové adresy. Domény prvního řádu spravuje organizace IANA, viz Kapitola 4.4. Opět velmi zjednodušeně, IANA přidělila České republice doménu 1. řádu „cz“. Současně s touto doménou přidělila k této doméně jistou množinu IP adres např. adresy 193.160.100.0 až 193.160.200.0. Organizace IANA musí současně technicky zabezpečit evidenci přidělených adres k dané doméně a nepřetržitou dostupnost této evidence (tzv. DNS záznamy na DNS serveru). V České republice se o správu domén 2. řádu pod doménou „cz“ a o správu přidělených IP adres stará organizace NIC.cz. Například Slezská univerzita požádala tuto organizaci prostřednictvím registrátora o přidělení domény 2. řádu „SLU.CZ“. Současně s touto doménou dostala SU přidělenou množinu jednoznačných IP adres z množiny přidělených k doméně „cz“ (např. 193.160.200.0 až 193.160.200.200). Organizace NIC opět musí zabezpečit evidenci přidělených adres a domén. Na Slezské univerzitě je OPF, FPF, FVP a Matematický ústav. Tyto součásti dostali od správce domény „slu.cz“ přiděleny domény 3. řádu „opf.slu.cz“, „fpf.slu.cz“, „fvp.slu.cz“ a „mu.slu.cz“. Současně byly opět přiděleny skupiny IP adres a vede se evidence těchto domén a přidělených adres (např. OPF dostane adresy 193.160.200.0 až 193.160.200.20). Správci na Slezské univerzitě disponují tedy těmito adresami a zřídí webový server se jménem „www.opf.slu.cz“ a adresou „193.160.200.1“ (uvedené adresy jsou jen jako příklad).

Napíše-li uživatel do internetového prohlížeče odkaz „www.opf.slu.cz/index.htm“, přeloží se na příslušném DNS serveru toto doménové jméno na příslušnou IP adresu, najde se příslušný počítač o dané IP adrese a v příslušném počítači soubor „index.htm“.

5.3.3 WWW KLIENT

Klient (browser, prohlížeč) je v podstatě prostředníkem mezi uživatelem a WWW serverem. Má na starosti tři **základní činnosti**:

1. Komunikace s uživatelem – jedná se o činnost na základě uživatele, např. vyžádání webových stránek (případně dalších souborů) od WWW serveru, přehrávání multimediálních souborů. Pro rozšíření funkčnosti obsahuje další doplňkové moduly (pluginy).
2. Komunikace s WWW serverem – jedná se o vysílání požadavků na WWW server. Dále pak po přijetí dat ze serveru provádí dekodování dat a instrukcí (interpreter), sestavuje webovou stránku (rozložení prvků) a nakonec zobrazuje webovou stránku v okně (popř. přehrává další multimediální obsah).
3. Spolupráce s dalšími programy – spolupráce s dalšími programy pak závisí na dalších aplikacích nainstalovaných na zařízení, kde se nachází také prohlížeč. Jedná se např. o poštovní programy, editory atd.

Mezi nejpoužívanější WWW klienty v současné době patří Google Chrome (Google), Firefox (Mozilla Foundation), Internet Explorer (Microsoft), Safari (Apple Inc.), Opera, Opera Mobile (Opera Software ASA) a další. Jejich zastoupení na trhu se stále mění, často v závislosti na oblibě zařízení, která uživatelé pro využívání WWW služby používají. Např. v poslední době roste obliba mobilních zařízení (chytré telefony, tablety).

5.3.4 WWW SERVER

Webový server můžeme vnímat jak v podobě HW, tak v podobě SW. Má na starosti zpracovávání požadavků od WWW klientů a zasílání informací na základě těchto požadavků (např. zaslání webové stránky nebo souboru). Dále zaslá klientovi stavový kód odpovědi o tom, zda došla odpověď v pořádku nebo zda nastala chyba. Chybové odpovědi jsou pak číslovány dle typu chyby. Přijímané požadavky od klientů jsou protokolovány a zpravidla ukládány do log souborů. Tyto log soubory jsou pak využívány při řešení problémů, v tom lepším případě pak bývají často zdrojem pro analýzu návštěvnosti webových stránek. Jako příklad webových serverů zde můžeme uvést např. Apache http server (Apache Software Foundation), IIS (Microsoft), GWS (Google Web Server), nginx (NGINX, Inc.), a další. Mezi poměrně jednoduché servery patří například server Xitami. Mezi základní nastavení, které musíme u serverů provést, je definování uživatelů a skupin uživatelů, definování prostoru (disků, adresářů) pro umístění souborů a defaultní názvy souborů (stránek), které se zobrazí v případě, že se klient připojí k serveru a neuvede soubor pro zobrazení. Součástí www serverů bývá i propojení s databází (např. MySQL) a programovacím jazykem (např. PHP, Perl apod.). Většina www serverů současně funguje jako Ftp server.



SHRNUTÍ KAPITOLY

V této kapitole jste se seznámili se základy, principy a pravidly fungování Internetu

6 ZÁKLADNÍ TECHNOLOGIE PRO TVORBU WWW



RYCHLÝ NÁHLED KAPITOLY

V této kapitole se seznámíte s hypertextem, což je základní stavební kámen pro tvorbu www stránek. Seznámíte se rovněž s jazykem HTML pomocí kterého se www stránky vytvářejí. Protože dnešní webové stránky jsou dynamické, seznámíte se i s JavaScriptem, jako základním nástrojem pro tvorbu dynamických stránek. Naučíte se rovněž základy tvorby www stránek a seznámíte se se základními nástroji pro jejich vytváření.



CÍLE KAPITOLY

Cílem kapitoly je naučit studenty tvorbě jednoduchých www stránek a možnostem jejich publikování.



ČAS POTŘEBNÝ KE STUDIU

V této kapitole je vhodné si uváděné příkazy prakticky odzkoušet. Z tohoto důvodu je čas potřebný ke studiu v rozsahu 2-5 hodin, podle míry praktické práce.



KLÍČOVÁ SLOVA KAPITOLY

Počítačová síť, TCP/IP protokol, ISO/OSI model, maska sítě, brána, IP adresa, služby Internetu, komunikace

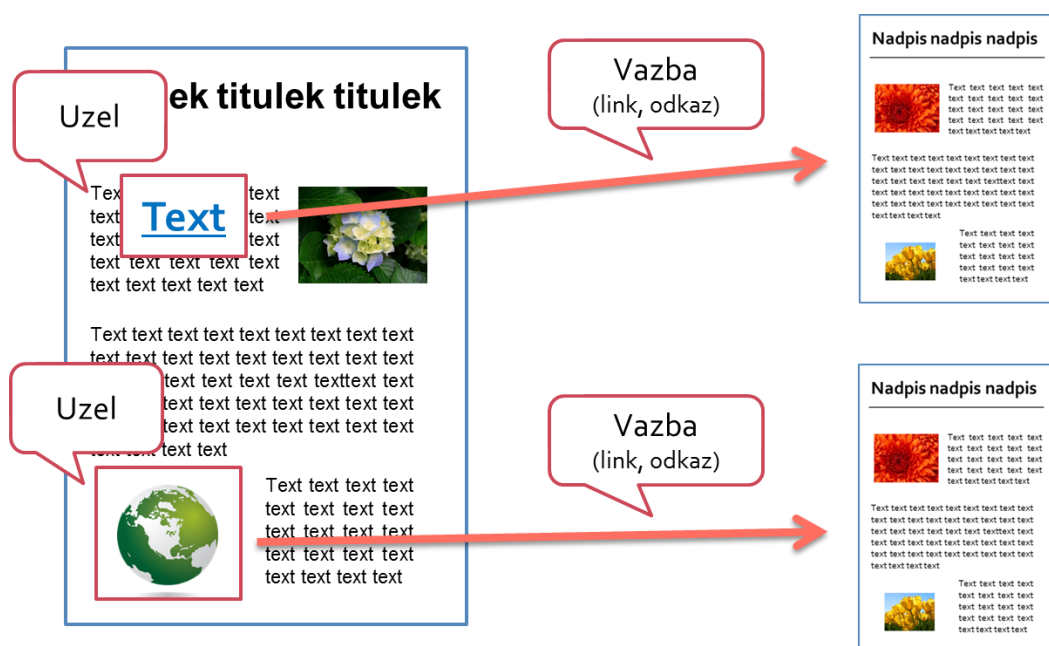
6.1 Princip Hypertextu

Hypertext je v podstatě text obsahující odkazy na další informace. Na rozdíl od lineárního textu, který známe z klasických dokumentů, umožňuje tzv. nesequenční přístup k informacím. Zobrazení dalších informací je pak umožněno prostřednictvím odkazů.

Jak znázorňuje následující obrázek, *základní jednotkou informace* je dokument (www stránka). Informace jsou uloženy v síti uzlů propojených asociativními vazbami. Uživatel se pak může libovolně mezi nimi pohybovat pomocí vazeb.

Vazby (link, hyperlink) jsou v podstatě odkazy na další objekty. Mohou směřovat na jakékoliv místo lokálně v počítači nebo v internetu (www stránka, obrázek, multimediální soubor, nezobrazitelné soubory všech typů formátů).

Uzly jsou pak vyznačená místa v dokumentu, které reprezentují dané odkazy. Jedná se často o zvýrazněný text, netextové informace (obrázky, animace, další multimediální soubory), v HTML 5 to mohou být také blokové elementy.



Obrázek 61 Technologie hypertextu

Zdroj: Botlík, Slaninová, 2014, *Služby Internetu a internetové systémy, výukový materiál OPF v Karviné*

6.2 Služba WWW

Zřejmě nejpoužívanější službou internetu je služba WWW (World Wide Web). Jedná se o službu využívající internetový protokol http, popř. https. Tato služba využívá technologie hypertextu, která byla popsána v předchozí kapitole.

World Wide Web, v doslovném překladu „světová rozsáhlá síť“, „celosvětová síť“ je označení pro systém prohlížení, ukládání a odkazování dokumentů nacházejících se v Internetu. Dokumenty (webové stránky) si prohlídíme pomocí webového prohlížeče, jsou uloženy na webových serverech a jsou navzájem propojeny pomocí hypertextových odkazů

zapisovaných ve formě URL (například <http://www.seznam.cz> nebo <http://www.google.com>). Webové stránky jsou popsány pomocí HTML jazyka a pro jejich přenos mezi počítači je používán HTTP protokol.

Jak vyplývá ze samotného principu hypertextu, umožňuje tato služba integrovaný pohled na data a informace z různých zdrojů (multimediální přístup).

Jejími charakteristickými znaky jsou:

- Nezávislost na platformě, operačním systému či jediné firmě
- Pružnost
- Neustálý vývoj standardů a doporučení

6.2.1 ZÁKLADY TVORBY WEBU

Tvorba webových prezentací je obvykle v kompetenci tzv. webmastera. Webmaster zastává v rámci procesu tvorby webu několik funkcí současně:

- **Tvůrce obsahu** – vytváření, úprava HTML dokumentů, vkládání dat (internetové aplikace, údržba integrity obsahů)
- **Návrhář** (webdesigner) – tvorba layoutu www stránek, tvorba grafiky, rozhoduje o celkovém vzhledu a navigační struktuře webu
- **Programátor** – tvorba www stránek pomocí samotných jazyků, ať už skriptovacích nebo jiných (PHP, ASP, CGI, Java, JavaScript, Flash, .Net)
- **Administrátor** (správce) – stará se o bezpečný provoz www serveru, o bezpečnost provozovaných skriptů, o zálohování dat, o bezpečnost důvěrných informací
- **Public relations** – člověk, který se stará o marketing a styk se zákazníkem, případně s ostatními uživateli webu

U menších webů většinou všechny tyto funkce zastává jeden člověk sám, u rozsáhlejších websitů mohou jednotlivé funkce zastávat různí lidé.

Než přistoupíme k vlastní tvorbě webu, musíme zde zmínit několik základních pojmů.

Internetová stránka

Pod pojmem internetová stránka rozumíme jakoukoliv stránku, která je součástí webu, bez ohledu na to, jakým způsobem byla vytvořena.

Web (website)

Web, popř. website je v podstatě kolekce internetových stránek, která tvoří jeden celek. Vyznačuje se určitou strukturou, danou propojením stránek pomocí odkazů. Struktura webu bývá často zobrazována v podobě mapy webu.

Domovská stránka (homepage)

Výchozí stránka, která se zobrazí jako první, zadáme-li do prohlížeče název domény bez udání názvu souboru. Většinou se jedná o soubor s názvem index.html nebo default.htm (podle operačního systému, na kterém běží web server).

6.2.2 INTERNETOVÁ STRÁNKA

Někdy také HTML stránka (podle základního jazyka používaného pro tvorbu www stránek). Jedná se o ucelený soubor informací, předávaný uživateli. Může být reprezentován ve dvou tvarech:

- **zdrojový kód** – tvar, ve kterém je stránka uložena v počítači (na webovém serveru), je psán v některém jazyce pro popis dokumentu (HTML, CSS atd.)
- **vizuální tvar** – tvar, ve kterém je stránka prezentována uživateli prostřednictvím prohlížeče.

Zobrazování internetové stránky prostřednictvím prohlížeče probíhá následujícím způsobem: nejprve dojde ke stažení a otevření souboru se zdrojovým kódem stránky (soubory s příponami .htm, .asp či .php). Dále dojde k zobrazení částí obsažených ve zdrojovém kódu (např. text). Nakonec dochází k načítání objektů, na které jsou v kódu uvedeny odkazy (např. grafika).

6.2.3 TYPY WEBOVÝCH PREZENTACÍ

Webové prezentace můžeme dělit na dvě základní skupiny: statické a dynamické. Statické webové prezentace jsou typické prezentace elektronických dokumentů. Využívají všech výhod hypertextu, nereagují však kromě zobrazení zdroje odkazu na požadavky uživatele. Naopak dynamické webové prezentace jsou typické svou uživatelskou interaktivitou, často zobrazují informace proměnlivé v čase, bývají automaticky generovány. Jako příklad můžeme uvést stránky vyhledávačů, blogy, e-shopy, webové aplikace apod.

6.3 Technologie pro vytváření webových stránek

V současné době existuje nepřehledné množství technologií, které lze použít pro tvorbu webových stránek či aplikací. Podíváme-li se na ně z hlediska tvorby statických či dynamických stránek, pro statické webové stránky je typické použití kombinace HTML, XHTML a CSS. Aktuálně je standardem HTML verze 5 a CSS3.

Dynamické webové stránky se pak liší v tom, zda jsou použité technologie zpracovávány na straně klienta nebo na straně serveru. Mezi technologie zpracovávány na straně klienta můžeme zařadit JavaScript a všechny jeho další mutace (Jscript, VBScript), ECMA Script, dále také XML, Flash či Java-applety nebo ActiveX prvky.

Na straně serveru se pak setkáme s jazyky jako je Perl či CGI skripty, skriptovací jazyky PHP, ASP, Server Side JavaScript či ASP.Net. Z dalších můžeme ještě zmínit např. Python, ColdFusion, Java Server Pages či kombinaci několika technologií, v poslední době velmi oblíbený AJAX.

6.3.1 NÁSTROJE PRO TVORBU WEBOVÝCH PREZENTACÍ

Pro tvorbu webových prezentací budeme potřebovat několik málo nástrojů. Pro samotné vytváření webových stránek postačí některý z editorů. Můžeme se setkat se dvěma základními kategoriemi editorů určených pro tvorbu webových stránek.

WYSIWYG editory (What You See Is What You Get) jsou editory, které umožňují vizuálně zobrazovat webovou stránku přímo v procesu jejího vytváření. Webmaster může vkládat na webovou stránku jednotlivé prvky pomocí sofistikovaných nástrojů, často poskytují nepřeberné množství již připravených prvků a šablon. Kdybychom šli do důsledků, při tvorbě pomocí WYSIWYG editorů není nutné znát podrobnosti o jednotlivých příkazech zvoleného jazyka. Editor totiž daný kód vytváří automaticky sám. V mnoha případech bohužel velmi nepřehlednou formou. Z čehož vyplývá, že pokud se chcete profesionálně věnovat tvorbě webových stránek, je vhodné se také dozvědět o používaných jazycích co nejvíce. Jako příklad WYSIWYG editorů můžeme zmínit FronPage Express či Adobe Dreamweaver z řady komerčního softwaru, Nvu pak jako zástupce softwaru nekomerčního. WYSIWYG editorům a tvorbě www stránek jejich pomocí bude věnována samostatná kapitola.

NonWYSIWYG editory jsou ve velké míře používány profesionálními webdesignery. Hlavním důvodem, proč používat nevizuální editory, tedy editory pracující pouze s kódem, je absolutní kontrola nad napsaným kódem. Existuje mnoho editorů tohoto typu, často používaných také programátory pro svou schopnost vizuálně zobrazovat psaný kód, některé editory jsou schopny také syntaktické kontroly napsaného kódu. Po vizuální stránce je pak možné webovou stránku zobrazit přímo v prohlížeči. Jako příklad můžeme uvést PSPad Editor, WebPage Maker, HTML Kit, Ace Html, či starší HomeSite. V neposlední řadě si můžeme vystačit také s poznámkovým blokem.

Kromě editorů pro samotnou tvorbu webových stránek potřebuje tvůrce webu také software pro práci s grafikou (pro úpravu obrázků, vytváření ikon, pozadí, animací, tlačítek apod.). Zde můžeme zmínit opět několik zástupců jak komerčního softwaru (Adobe Photoshop, Adobe Illustrator, Macromedia Fireworks), tak softwaru nekomerčního (Gimp, Inkspace, IrfanView).

Pro umístění webových stránek na webovém serveru je pak vhodné mít nainstalovaného FTP klienta pro snazší a rychlejší přenos souborů z lokálního počítače na webový server. Existují sice poskytovatelé webhostingu, kteří nabízejí přímé nahrání souborů přes webové rozhraní, tato možnost se však stává v případě většího množství souborů uživatelsky velmi nepřívětivou. FTP klientů existuje nepřeberné množství, od samostatných aplikací (CuteFTP, AceFTP, WS FTP), až po FTP klienty, kteří jsou součástí jiného softwaru (Total Commander).

6.4 Práce s hypertextem

Hypertexty bývají realizovány pomocí značkovacích jazyků. Základní normou je GML (Generalized Markup Language). Jazyk SGML (Standard Generalized Markup Language), který je definován v ISO normě 8879 z roku 1986 byl příliš obecný, umožňoval definici vlastních značkovacích jazyků (sad značek a jejich vzájemných vztahů) pomocí tzv. definic typu dokumentu (DTD) a má obsáhlou množinu volitelných parametrů, počínaje maximální délkou názvů značek a konče určením znaků použitelných jako oddělovače značek od textu. Komplexnost a přílišná volnost standardu SGML zbrzdila jeho praktické využití. To přišlo s definováním HTML.

6.4.1 HTML

Každá stránka je na disku nebo na serveru uložena ve formě zdrojového kódu. Tento kód je psaný v jazyce HTML. Pomocí jazyka HTML se tvoří webové stránky. HTML znamená HyperText Markup Language - hypertextový (hypertext = odkaz) značkovací jazyk. Stránky jsou tedy zapsány pomocí zdrojového kódu. Webové stránky vytváříte pomocí HTML značek (značka, nebo-li také tag). Značek je mnoho a vždy se zapisují do špičatých závorek

HTML je nejznámější aplikací SGML (Standard Generalized Markup Language) Jedná se tedy o značkovací jazyky.

Podstata a principy značkovacích jazyků vychází z faktu, že v každém dokumentu existuje vlastní obsah a formáty, prostřednictvím kterých je obsah zobrazen. Dále existuje v dokumentech množina znaků, které jsou charakteristické pro prostředí, ve kterém byl dokument vytvořen. Například dokument napsaný ve Wordu obsahuje mimo textové informace taky údaje o znakové sadě, o parametrech stránky, o autorovi apod. tyto údaje jsou zaznamenány pomocí speciálních znaků programu Word. Pokud by dokument byl otevřen v jiném programu, tento by nevěděl, jak interpretovat řídicí znaky a formáty Wordu.

Z toho plyne, že pro Internet, jako pro multiplatformní prostředí, je nutné zabezpečit, aby formátovací a řídicí znaky měly jednotně interpretovatelnou podobu. Toho se dosáhlo tak, že došlo k oddělení obsahu, který je psán prostým, neformátovaným textem. Řídicí a

formátovací znaky jsou rovněž zapsány prostým textem ve formě značek, které obsahují informace pro program, kterým je dokument zobrazován. Značky jsou většinou „párové“, tj. mají část, která se uvádí před textem, kterého se týká a část, která se umísťuje za text a určuje konec platnosti značky. Aby se rozlišilo, že se jedná o počáteční nebo koncovou značku, má koncová značka jiný tvar.

Například zápis ` text ` obsahuje značku ``, jejíž koncový tvar je `` a říká, že text mezi značkami se má zobrazit tučně. Množinu značek, kterou HTML používá pro označení formátů a informací příslušné značky, určuje příslušné DTD (definice typu dokumentu), které definuje verzi HTML. WWW stránky zapsaná v HTML jazyce tvoří tzv. zdrojový kód. Ten lze upravit prostřednictvím libovolného textového editoru, např. poznámkového bloku a prohlédnout v libovolném prohlížeči tak, že z menu prohlížeče vyvoláme příkaz Zobrazit > Zdrojový kód (případně View > Source, apod.).

Pro psaní HTML dokumentu platí několik zásad. Každý dokument je uzavřený (začíná a končí) do značky `<html> </html>`.

Dokument má dvě základní části, hlavičku `<head>` a tělo `<body>`. V hlavičce jsou obsaženy informace o dokumentu, vlastní text je v části `<body>`.

Dvojici značek a textu, který je ve značkách uzavřený se říká obecně element, u HTML se mu říká Tag. Tagy jsou párové (obsahují dvojici značek, počáteční a konečnou, které označují místo pro realizaci značky) a mohou se do sebe vnořovat. Obecně platí, že se tagy nesmí křížit (tag nemůže skončit, pokud neskončily všechny tagy v něm vnořené). Při psaní značek se rozlišuje velikost písma. Součástí značek jsou atributy, které dále značku upřesňují. HTML toleruje nepárovost (neuvedení konečné značky u párového tagu), křížení i nedodržení velikosti písma, obecně je to však ve značkovacích jazycích hrubá chyba. HTML definuje i množinu nepárových značek.

Základy HTML se dají shrnout následovně, dokument v jazyku HTML má předepsanou strukturu. Ta obsahuje:

Deklaraci typu dokumentu – značka `<!DOCTYPE html>` (sděluje prohlížeči, že otevřel HTML dokument).

Kořenový element – prvek html (značky `<html>` a `</html>`) (reprezentuje HTML dokument).

Hlavičku dokumentu – prvek head (značky `<head>` a `</head>`) (obsahuje metadata, která se vztahují k celému dokumentu. Definuje kódování, název dokumentu, autora, popis, klíčová slova, titulek dokumentu nebo kaskádové styly).

Tělo dokumentu – prvek body (značky `<body>` a `</body>`) (zahrnuje vlastní obsah dokumentu).

V konfrontaci s XML lze snadno vyvodit, že HTML je specifický XML dokument.

6.4.2 VERZE JAZYKA HTML

Verze 0.9 – 1.2 (1991–1993), nepodporují grafické rozhraní. Verze 2.0 byla vydána v listopadu 1995, odpovídá syntaxi SGML. Přidává k původní specifikaci interaktivní formuláře a podporu grafiky. Verze 3.2 (1997, komunita W3C) přidává tabulky, zarovnávání textu a stylové prvky pro ovlivňování vzhledu. Původně připravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá. Verze 4.0 (1997, W3C). Do specifikace jazyka přibyly nové prvky pro tvorbu tabulek, formulářů a byly standardizovány rámy (frames).

V této verzi lze charakterizovat význam (sémantiku) jednotlivých částí dokumentu, a vzhled (styly). Verze byla revidovaná, verze 4.01 (1999, W3C) opravuje některé chyby, mělo se jednat o poslední verzi HTML, další specifikace měla být XHTML (verze 1.0, 1.1, 2.0 resp. 5.0), následník HTML, využívající univerzální jazyk XML. HTML 5.0 ukončuje závislost HTML na SGML opravuje chyby předešlé verze, vyřazuje zastaralé prvky, přidává nové sémantické prvky, přidává podporu nových a moderních technologií, zavádí nový systém vývoje jazyka (nové verze vydávány zhruba ve dvouletých cyklech).

HTML je samo o sobě pouze značkovací jazyk k vytváření struktury dokumentů. Ovšem s nástupem jeho poslední verze se zažitě používá označení HTML5 pro celý balíček, do kterého spadají jak kaskádové styly CSS, což je samostatný jazyk na stylování dokumentů, tak JavaScript (správný název je ECMAScript – označení JS je pouze lidové), který je také samostatným skriptovacím jazykem na straně uživatele. Ve skutečnosti, bychom měli správně označovat novou verzi technologií jako HTML5, CSS3 a ES5. Zkráceně se ale tato trojice označuje jako technologický balíček HTML5.

Kompletní dokumentace o těchto technologiích je k dispozici na stránkách konsorcia World Wide Web Consortium W3C

Odkazy na základní značky jsou dostupné na mnoha místech a webových stránkách, odkud je čerpán i následující obsah textu³⁰.

³⁰ Další informace naleznete především na stránkách:

<http://w3c.github.io/html-reference/spec.html#hgroup>

dále v českém jazyce: <http://www.html5.cz/>, www.jakpsatweb.cz, <http://programujte.com/clanek/2010082200-html5-nove-vlastnosti/>, <https://www.zdrojak.cz/serialy/webdesigneruv-pruvodce-po-html5/>, <http://www.itnetwork.cz/html-css/html-manual>, http://www.fce.vutbr.cz/studium/materialy/0u2/reference_html.pdf

v anglickém jazyce: <http://html5doctor.com/the-hgroup-element/>, <http://www.w3schools.com/>

Základní značky HTML a jejich atributy

Struktura dokumentu

```
<HTML>
<HEAD> <TITLE>...</TITLE>...</HEAD>
<BODY> ... </BODY>
</HTML>
```

Záhlaví dokumentu

```
<TITLE>...</TITLE>   titulek (název) dokumentu
<LINK>               definování vazeb na další dokumenty
<STYLE>...</STYLE>   definice stylů
<SCRIPT>...</SCRIPT> vložení skriptů (může být i kdekoliv v těle)
```

Atributy elementu <BODY>

```
bgcolor   barva pozadí ("#rrggbb" nebo název barvy)
text      barvy textu ("#rrggbb" nebo název barvy)
background URL obrázku, který se použije na pozadí
```

Seznamy

```
<UL>...</UL>   seznam s odrážkami
                type   tvar odrážek (disc, square, circle)
<OL>...</OL>   číselný seznam
                type   způsob číslování (1 – arabské číslice, a – malá
                        písmena, A – velká písmena, i – malé římské
                        číslice, I – velké římské číslice)
                start   počáteční hodnota odrážky
<LI>          položka seznamu
                type   stejné jako u <UL> nebo <OL>
                value   hodnota číslice v odrážce pro položku
                        číselného seznamu
<DL>...</DL>   seznam definic
<DT>          definovaný pojem
<DD>          text definice
compact      zhuštěný tvar seznamu (pro všechny druhy)
```

Obrázky

```
<IMG src="URL"> zařazení obrázku do textu
                align  způsob zarovnání (left, right,
                        middle, bottom)
top,          alt   text pro znakové prohlížeče
                width šířka
                height výška
                border šířka rámečku
                hspace velikost horizontálního odsazení od textu
                vspace velikost vertikálního odsazení od textu
```

Tabulky

```
<TABLE>...</TABLE> vymezení tabulky
                align  způsob obtékání tabulky textem (left,
                        right) nebo umístění tabulky doprostřed
                        (center)
                cellspacing mezera mezi buňkami
                cellpadding odsazení obsahu buňky od rámečku
                width   šířka tabulky v pixelech nebo procentech
                border   šířka rámečku
                rules    zobrazení vnitřních čar (all, cols,
                        rows, none)
                frame    způsob zobrazení vnějších obrysů tabulky
                        (void, above, below, hside,
                        vside, lhs, rhs, border)
<TR>...</TR>      vymezení řádku tabulky
<TH>...</TH>      buňka záhlaví tabulky
<TD>...</TD>      buňka tabulky
<COLGROUP> </COLGROUP> vymezení skupiny sloupců
<COL>           specifikace atributů pro sloupce
                bgcolor  barva pozadí buňky
                align    vodorovné zarovnání (left, center,
                        right, justify)
                valign   svislé zarovnání (top, middle, bottom)
                width    šířka buňky v pixelech nebo v procentech
                        z celkové šířky tabulky
                nowrap   text v buňce se nebude zalamovat
                rowspan  počet sloučených řádků (pro TD, TH)
                colspan  počet sloučených sloupců (pro TD, TH)
                span     počet sloupců se společným nastavením
                        (pro COL resp. COLGROUP)
```

Vybrané entity

```
&lt;   <
&gt;   >
&amp; &
&quot; "
&nbsp;  nezlomitelná mezera
&deg; °
&ndash; - (krátká pomlčka)
&mdash; — (dlouhá pomlčka)
&ldquo; "
&rdquo; "
&times; ×
&frac; 1/2
```

Odkazy

```
<A href="URL">...</A> odkaz na určitě URL
                title   název odkazu
<A name="kotva">...</A> definice kotvy
```

Logické styly písma

```
<EM>...</EM>   zvýraznění
<STRONG>...</STRONG> silně zvýraznění
<BIG>...</BIG>   velké písmo
<SMALL>...</SMALL> malé písmo
<SUB>...</SUB>   dolní index
<SUP>...</SUP>   horní index
```

Fyzické styly písma

```
<TT>...</TT>   neproporcionální písmo (psací stroj)
<I>...</I>     kurziva
<B>...</B>     tučné
<U>...</U>     podtržené
<FONT>...</FONT> definice velikosti, barvy, fonu písma
                color  barva
                size    velikost 1-7
```

Formátování textu a strukturování dokumentu

```
<P>...</P>     vymezení odstavce
                align  způsob zarovnání (left, right, center)
<BR>          nový řádek
                clear  vynechání místa až skončí obrázky (left,
                        right, all, none)
<HR>          horizontální čára
                align  způsob zarovnání (left, right, center)
                size   výška v pixelech
                width   délka v pixelech nebo v procentech
<Hn>...</Hn>   nadpis úrovně n (1-6)
                align  způsob zarovnání (left, right, center)
<PRE>...</PRE> předformátovaný text
<SPAN>...</SPAN> vymezení části textu se společným
                  formátováním písma
<DIV>...</DIV> Označení související části dokumentu
                align  způsob zarovnání (left, right, center)
```

Formuláře

```
<FORM>...</FORM> vymezení formuláře
                name   jméno formuláře (v rámci dokumentu
                        jedinečné)
                <INPUT type="text"> definice vstupního resp. výstupního prvku
                        druh vstupního pole
                        text   jednořádkové testové pole
                        checkbox zaškrtnávací políčko
                        radio   přepínač
                        submit  tlačítko k odeslání dat na server
                        reset    tlačítko pro obnovení hodnot ve formuláři
                        button   obecné tlačítko (pro navázání skriptu)
                motné vlastnosti INPUT prvků (smyslované podle typu):
                name   jméno (potřebné pro zpracování skripty)
                value   hodnota
                checked nastavuje implicitní zaškrtnutí
                size    rozměr pole ve znacích
                maxlength maximální počet znaků
<TEXTAREA>...</TEXTAREA> víceřádkové vstupní pole
                name   jméno (potřebné pro zpracování skripty)
                rows   počet řádků
                cols   počet sloupců
<SELECT>...</SELECT> výběrové pole (rozbalovací nabídka)
                size   počet viditelných voleb
                multiple možnost výběru více položek současně
<OPTION>...</OPTION> identifikace volby ve výběrovém poli
                value   odeslaná hodnota
```

Komentáře (poznámky)

```
<!-- Toto je komentář ... -->
```

Barvy

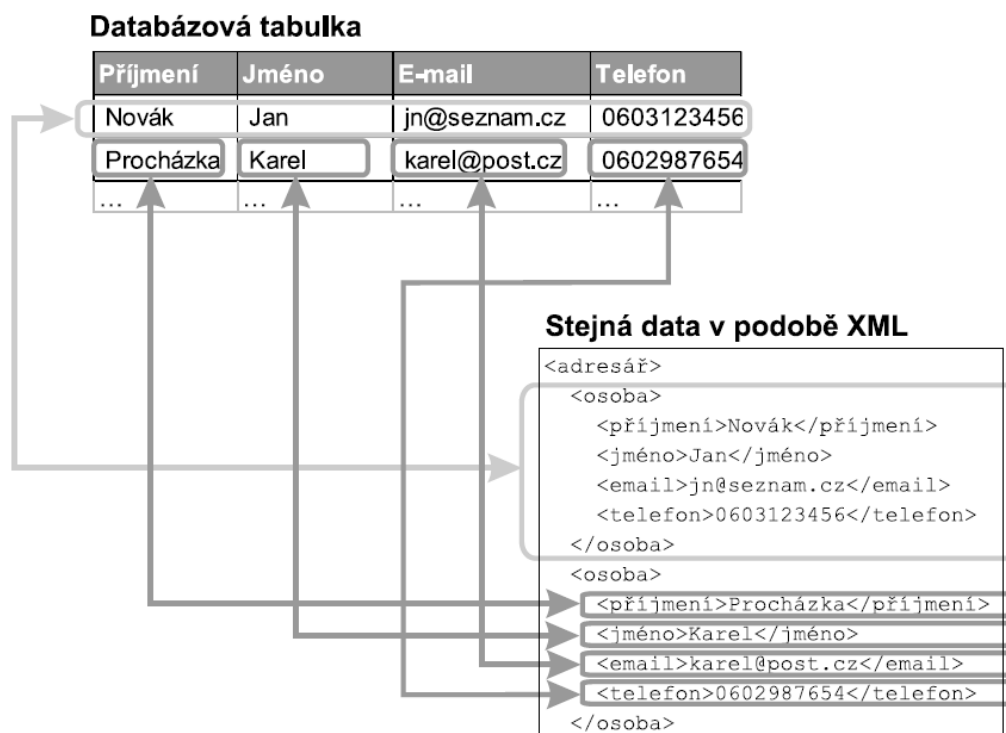
```
"#rrggbb" v barevném modelu RGB
black  černá
purple fialová
navy   modrá
olive  hnědá
green  zelená
teal   tyrkysová
maroon červená
silver stříbrná (bílá)
gray   šedá
fuchsia světle fialová
blue   jasně modrá
yellow žlutá
lime   světle zelená
aqua   světle tyrkysová
red    světle červená
white  jasně bílá
```

6.4.3 XML

XML (eXtensible Markup Language) je jazyk, který umožňuje označit význam jednotlivých částí textu, a ne jejich vzhled. Největší přínos XML spočívá v tom, že v dokumentech můžeme používat vlastní značky (tagy). Jedná se opět o podmnožinu SGML, oproti HTML si zachovává možnost definování vlastních DTD, a tedy i vlastních značek, pro jednotlivé skupiny dokumentů. Narozdíl od SGML je mnoho parametrů předem určeno a nelze je měnit, je dána např. maximální délka názvů značek, použité oddělovače a speciální znaky atd. XML už počítá s podporou různých jazyků, takže není tak úzce svázáno s angličtinou jako většina předchozích počítačových technologií.

Syntaxe zápisu dokumentů v XML je oproti SGML poměrně přísná, což umožňuje mnohem snazší a levnější vývoj aplikací, které umožňují s tímto jazykem pracovat.

Základní funkcí značek v XML je přidělení obsahu. Lze pomocí značky určit, že se jedná o výrobek, barvu, u dokumentu o kapitolu, odstavec, v matematice o proměnnou, vzorec apod. Poměrně obsáhle popsal práci s XML Kosek (2000).



Obrázek 62: přepis dat pomocí XML

Zdroj: XML pro každého, Kosek (2000)

6.4.4 FORMÁTOVÁNÍ V HTML, KASKÁDOVÉ STYLY

Formátováním HTML dokumentů se zabývá nepřeberné množství publikací. V prvopočátcích bylo formátování omezené na tagy HTML, později došlo k rozšíření možností pomocí samostatného jazyka CSS.

Protože se jazyk HTML vyvíjel, vznikaly časem různé způsoby, jak formátovat text. Proto dnes existují dva odlišné způsoby, jak v HTML třeba obarvit písmo nebo ztučnit text.

- Starší způsob používá přímo HTML tagy. (Například kurzíva se dělá pomocí tagů `<i>` a `</i>`: `<i>kurzíva</i>`). Pomocí tagů se některé věci nedají udělat.
- Novější způsob prostřednictvím tzv. CSS stylů, používá se tag `<style>` a obecný atribut "style", kterému se přiřazuje nějaká definice.

Důvod implementace CSS do procesu tvorby www stránek vyplynul zejména z malých možností HTML. Formátování HTML³¹ bylo potřebné přizpůsobit faktu, že každý text má obsah a formu. Když mluvíme o formátu (formě) webových stránek, myslíme tím třeba barvu a velikost písma, pozadí, zarovnání atd., tj. vše, co nepatří do obsahu. Hlavním smyslem CSS je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

Obecně řečeno, CSS je nějaký zápis, který určuje vzhled (barvy, dekorační obrázky, rozmístění prvků) HTML dokumentu. V praxi se pak častěji používají tyto názvy: Kaskádové styly, styly, CSS soubory. Soubor s kaskádovými styly má příponu `css`. Sám o sobě nemá smysl. Jeho funkčnost se projeví až po propojení s HTML souborem.

CSS je kolekce metod pro grafickou úpravu webových stránek. Ta zkratka znamená Cascading Style Sheets, česky "kaskádové styly". Kaskádové, protože se na sebe mohou vrstvit definice stylu, ale platí jenom ta poslední.

Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány CSS1, CSS2 a CSS3.

V současnosti se dá říct, že už se celý web formátuje pomocí CSS. Ze starého HTML formátování zůstalo minimum. Pomocí CSS lze např.:

- Nastavit libovolnou a přesnou velikost písma, prokládání, kapitálky.
- Udělat odsazení prvního řádku odstavce, zvětšit řádkování.
- Zrušit nebo zvětšit prázdný prostor po odstavci.
- Automaticky formátovat nadpisy (například je všechny udělat zelené).

³¹ převzato s úpravami z www.jakpsatweb.cz

- Zvýrazňovat odkazy po přejetí myší.
- Udělat automaticky grafické odrážky.
- Určité části textu zneviditelnit, zprůhlednit nebo nezobrazit.
- Předefinovat grafický význam běžných tagů (například všechno, co je kurzívou, udělat i tučně).
- Nastavit pozadí čehokoliv, stránky, tabulky ale třeba i odstavce; pozadí se nemusí opakovat a může mít přesnou pozici.
- Umístit nějaký objekt (třeba kus textu) kamkoliv do stránky, může se to i překrývat.
- Přidat k čemukoli rolovací lišty, oříznout to, orámovat, nastavit okraje.
- V kombinaci se skripty je dnes CSS nejmocnější zbraň pro "rozhybání" stránek.
- Hlavní význam CSS spočívá v tom, že fungují hodně automaticky, přičemž se vzhled celého webu deklaruje jedním souborem.

Styl se může deklarovat třemi způsoby:

4. Přímo v textu zdroje u formátovaného elementu pomocí atributu `style="..."`.
5. Pomocí "stylopisu" (angl. "stylesheet") v hlavičce stránky. Stylopis je jakýsi seznam stylů. Je v něm obecně napsáno, co má být jak zformátováno, například že nadpisy mají být zelené. Do stránky se stylopis píše mezi tagy `<style>` a `</style>`.
6. Použitím externího stylopisu, tj. souboru *.css, na který se stránka odkazuje tagem `<link>`. V souboru je umístěný stylopis. Hlavní výhoda je v tom, že na jeden takový soubor se dá nalinkovat mnoho stránek, takže pak všechny vypadají podobně.

Kaskádové styly mají jinou syntaxi než HTML. První na řadě je selektor³² (ten určuje, co má být formátováno). V složených závorkách ({}) se nachází vlastnosti a hodnoty (jinak také blok deklarácí). Zapiše se vlastnost, dvojtečka, mezera, hodnota a výčet hodnot se ukončí středníkem (;). Mezery je možné vynechávat nebo jich napsat více za sebou, funkčnost to neovlivní. Pro přehlednost je lepší mezery psát. Selektor udává, jaký element bude blok deklarácí formátovat. Selektorem může být třída (class), identifikátor (id), tag, nebo různé kombinace. Identifikátoru se v CSS předřazuje křížek (#, Ctrl+Alt+X), třídám tečka a tagu se nepředřazuje nic, napíše se tak jak je (bez špičatých závorek). Je možné formátovat více elementů najednou, selektory oddělíme čárkami. Další možností je formátování vnořených elementů, pro takové formátování se selektory oddělují mezerami. Jako selektory by se neměla používat slova s diakritikou.

³² Blíže např. na <http://www.klikzone.cz/CSS-navod/pravidla-CSS.php>,
http://dum.hajduch.net/VY_32_INOVACE_1ICT9roc_09_B ap.

6.4.5 JAVASCRIPTY

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládané různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Script v JavaScriptu

Script v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele. Důležité je si uvědomit, že JavaScript běží na straně klienta, všechny ty aplikace jsou tedy spouštěny v prohlížeči u uživatele. To je obrovský rozdíl oproti serverovým jazykům, jako je např. PHP. Pomocí JavaScriptu tedy můžeme měnit obsah webové stránky u uživatele, což nabízí např. tvorbu dynamických menu, různých roletek a dalších kontejnerů, které umožňují ušetřit místo na stránce když jsou zavřené a po najetí myši se otevrou. To je jistě šikovné a přehledné. JavaScript je skvělý k formátování textu, pomocí něj si můžete do napsané zprávy vkládat smajlíky nebo dokonce formátovat text jako ve Wordu. Když je na webové stránce nějaký editor, je to na 90% JavaScript. Další využití nalezneme u ukazatelů času a data a dalších efektů na webových stránkách (např. padající sníh na vánoce).

Protože odeslání stránky na server a čekání na následnou odpověď serveru je pomalé, používá se JavaScript také na validaci webových formulářů. Když například napíšete špatně email, webová stránka vás na to ještě před odesláním upozorní a není třeba stránku znovu načítat. Musíme si však uvědomit, že protože JavaScript běží na klientovi, může si ho uživatel vypnout nebo přepsat, proto nesmíme na takovouto validaci spoléhat a email musíme podruhé zkontrolovat i na serveru. Výhodou JavaScriptu je však pohodlnost a efektivnost bez zbytečných načítání stránky a prodlev. To samé platí u dříve zmíněných JavaScriptových menu, které by se i s vypnutým JavaScriptem měly zobrazit. Je jedno, že už to nebude tak hezké, protože JavaScript má 99% lidí zapnutý a některé prohlížeče ho dokonce ani neumožňují vypnout, problém je hlavně například u indexovacích robotů Googlu, který je potom nevidí, protože JavaScript nepoužívá. To by mohlo pak znamenat, že se přes menu nedostane na zbytek webu. JavaScriptové frameworky (například nejznámější JQuery, kterou používá i gigant Google, nebo Mootools), obsahují tzv. widgety, předpřipravené miniaplikace, například menu, kde si jen nastavíte, co v nich má být zbytek za vás řeší framework a to i v případě, kdyby někdo JavaScript vypnul³³.

³³ Převzato s úpravami z <http://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>

Syntaxe jazyka JavaScript je volně založena na syntaxi jazyka Java. Java je objektově orientovaný programovací jazyk a JavaScript může být považován jako velmi podobný jazyk.

JavaScript a Java jsou dva ovšem odlišné programovací jazyky.

Z pohledu programovacího jazyka i u JavaScriptu existují proměnné, funkce, podmínky, operátory atd. Jak plyne z názvu, programový kód je zapsán v tzv. skriptu. Celý skript je ohraničen párovým tagem `script`. To znamená, že tagy `<script>` a `</script>` udávají prohlížeči informaci o tom, že má mezi těmito tagy předpokládat skript. Pro vypsání textu pomocí JavaScriptu se používá funkce `document.write('text')`.

Skripty můžete umístit:

- Do hlavičky HTML dokumentu (tj. mezi tagy `<head>` a `</head>`).
- Do těla HTML dokumentu (tj. mezi tagy `<body>` a `</body>`).
- Do externího souboru (V HTML dokumentu bude odkazováno na tento soubor).

Jestliže umístíme tento kód do HTML dokumentu, spustí se pokaždé, jakmile je dokument načten. To znamená, že se po každé aktualizaci stránky spustí znovu.

Další možností je spouštění scriptu při událostech, jako je například kliknutí na vybraný prvek. Události (Events) umožňují vložit JavaScript do HTML prvku. Jedná se tedy o speciální atributy, které se přidávají přímo do HTML prvku. Například tag `h1` naformátuje vybraný text jako nadpis, vypíše daný text, atribut `onClick` je událostí JavaScriptu, která při kliknutí na text vyvolá příkaz „alert“ (hláška).

```
<h1 onClick="alert('Toto text hlášky');">Klikni na tento text!!!</h1>
```

Základy syntaxe JavaScriptu

Pro psaní scriptů platí několik zásad.

- Příkazy se oddělují středníkem nebo koncem řádku. V praxi se doporučuje středníky používat, protože přidávají přehlednost.
- Oproti HTML tagů záleží na velikosti písmen, ať už v názvech proměnných či v názvech objektů.
- Řetězce se uzavírají do uvozovek. Přípustné jsou i apostrofy (na české klávesnici Alt + 39). Je to jedno, zda se použije apostrofy nebo uvozovky. Někdy jsou apostrofy nutné, když potřebují zanořovat různé druhy uvozovek.

- Pokud potřebujeme napsat nějaký speciální znak, který JavaScript interpretuje (například zmíněné uvozovky) do stránky, musí se využít tzv. escape sekvence. Před tento znak se napíše zpětné lomítko.

JavaScript zná speciální hodnotu pro pravdu: true a pro nepravdu: false. Objekty a jejich metody a vlastnosti se oddělují tečkami: objekt.podobjekt.vlastnost. Programové sekvence se uzavírají do složených závorek {} (zejména při větvení a deklaraci funkcí).

6.5 HTML

Jak bylo zmíněno dříve, v současnosti je pro tvorbu www stránek nejpoužívanější jazyk HTML5. Podle Root.cz³⁴ lze shrnout nové vlastnosti a možnosti následovně:

FILE API – umožňuje nahrávat neomezené množství souborů najednou, segmentaci velkého souboru na straně klienta a postupný upload těchto částí na server, kde se serverový script postará opět o jeho složení. Tím je možné naprosto obejít omezení maximálního uploadu.

FULLSCREEN API – Toto rozhraní odstraňuje nutnost používat Flash při nutnosti přepnout dokument přes celou obrazovku. To se využívá například při přehrávání videí nebo při prohlížení fotografií na sociálních sítích.

GEOLOCATION API – Pomocí tohoto rozhraní lze bez nutnosti jakýchkoliv doplňků identifikovat globální pozici. Ta je vypočítávána ze všech možných známých parametrů, na nichž závisí přesnost. Čím více údajů, tím je API přesnější. Prohlížeč se vždy nejprve zeptá, jestli chcete dané stránce sdělit své údaje o poloze.

CSS3 – je některými zdroji označován jako součást HTML5, slouží ke stylování dokumentů. Prakticky se jedná o samostatný jazyk, sloužící obecně pro aplikaci stylů ve značkovacích jazycích, tedy i v XML apod.

@FONT-FACE – Pomocí tohoto nástroje lze načíst ze serveru libovolný font a použít ho na webu. Font se načítá při načítání stránky a funguje jako standardní webové fonty Arial, Times new Roman, Verdana a další.

MEDIA API – Umožňuje přehrávat video a hudbu přímo v prohlížeči. API umožňuje naprogramovat celý přehrávač opět bez použití plug-inu Flash, bez kterého bylo přehrávání na internetu před vydáním tohoto API nemožné. Zahrnuje metody jako play(), pause(), load() a canPlayType(). Rozhraní umožňuje vytvořit vlastní prostředí přehrávače a v kombinaci s Fullscreen API i nahradit doposud využívaný Flash.

³⁴ HTML5: co přináší a proč se o něj zajímat, dostupné na <https://www.root.cz/clanky/html5-co-prinasi-a-proc-se-o-nej-zajimat/>, online 1.9.2016

TEXT TRACK API – Toto API umožňuje připojovat k přehrávaným multimédiím titulky, popisky nebo metadata.

SVG – HTML5 podporuje využití vektorové grafiky v podobě souboru .svg. Tento formát zápisu lze vytvořit ve většině vektorových editorů a lze jej snadno implementovat do HTML kódu nebo jako externí soubor svg.

CANVAS – Asi největší podporu ze strany prohlížečů má nový atribut Canvas, umožňující vykreslovat grafy pomocí svého javascriptového rozhraní. Jedná se o prvek pro generování jakékoliv grafiky pomocí JS. Jde o tzv. plátno umožňující dynamické vykreslování bitmap a jednoduché grafiky jako je tomu například v jazyce C++.

GAMEPAD API – Toto rozhraní je schopno propojit gamepad s internetovou aplikací a ovládat ji. Jde o využití především při tvorbě internetových her.

OFFLINE WEB APPLICATIONS – Do nástupu HTML5 bylo pro funkčnost internetové aplikace zapotřebí neustálé připojení k internetu. Součástí nové specifikace balíku Offline web applications je rozhraní umožňující omezenou funkčnost webové aplikace i při tzv. offline režimu, tedy v situaci, kdy nejsme připojeni k internetu. Pomocí tohoto rozhraní lze ukládat data nejprve k uživateli a až následně na server. Při přechodu do offline režimu (výpadek připojení), bychom o tato data nepřišli, na server by se odeslala až po obnově spojení.

SÉMANTIKA – HTML5 zavedlo nové tagy, které slouží pro zpřehlednění a optimalizaci celého projektu. Místo strukturování částí stránky do tzv. divů umožňuje HTML používat tagy přesně podle toho, jaká část stránky je jimi strukturována. Například máme samostatně stojící nezávislý prvek stránky, kterým může být článek, položka v e-shopu atd. Tag `<div></div>` lze nahradit tagem `<article></article>`, obdobně obsah hlavičky tagem `<header></header>` nebo obsah patičky realizovat tagem `<footer></footer>`.

Přehled tagů a atributů je souhrnně definován na stránkách konsorcia W3C. Uvádíme převzatý seznam značek. Český rejstřík HTML 5 specifikace obsahuje popis všech HTML 5 tagů včetně ukázek jejich použití a vysvětlení atributů. Rejstřík, včetně odkazů, byl převzat z e-www stránek „Abecední rejstřík tagů - Český HTML 5 manuál“³⁵.

6.5.1 HTML STRUKTURA

<u>base</u>	Tag base v HTML 5 umožňuje nastavit kořenovou složku pro relativní odkazy v dokumentu.
<u>body</u>	Tag body v HTML 5 označuje tělo celého HTML 5 dokumentu, ve kterém nalezneme obsah celých webových stránek.

³⁵ Dostupné na <http://www.itnetwork.cz/html-css/html-manual/html-5-abecedni-rejstrik-tagu>

<u>DOCTYPE</u>	Tag DOCTYPE v HTML 5 sděluje prohlížeči, že kód zobrazuje HTML 5 dokument.
<u>head</u>	Tag head v HTML 5 označuje hlavičku dokumentu, kde můžeme specifikovat např. kódování dokumentu.
<u>Hlavička HTML dokumentu</u>	Popis hlavičky head HTML dokumentu a elementů do ní patřících, tedy title, style, link, base, meta, script a noscript.
<u>html</u>	Tag html v HTML 5 obaluje celý HTML dokument, který obsahuje hlavičku a tělo.
<u>link</u>	Tag link v HTML 5 se používá k provázání dokumentu s externím souborem, nejčastěji k CSS stylům.
<u>meta</u>	Tag meta v HTML 5 poskytuje tzv. metadata. Jedná se o informace vložené do HTML dokumentu.
<u>Struktura HTML dokumentu</u>	Popis struktury HTML dokumentu, tagů doctype, html, head a body.
<u>style</u>	Tag style v HTML 5 slouží k vložení stylování přímo do HTML dokumentu.
<u>title</u>	Tag title v HTML 5 obsahuje titulek stránky. Každá HTML stránka ho musí v hlavičce obsahovat.

6.5.2 TEXTOVÉ TAGY

<u>abbr</u>	Tag abbr v HTML 5 označuje zkratku v textu. Může se jednat o zkratku i o zkratkové slovo.
<u>b</u>	Tag b v HTML 5 označuje text, který se stylisticky odlišuje od ostatního textu, ale zároveň není důležitý.
<u>cite</u>	Tag cite se v HTML 5 používá k citování názvu díla nebo práce.
<u>code</u>	Tag code v HTML 5 označuje úryvek zdrojového kódu. Ve výchozím nastavení je vykreslen neproporcionálním písmem.
<u>del</u>	Tag del v HTML 5 slouží k označení odstraněného textu po editaci dokumentu. Element obsahuje atributy cite a datetime .
<u>dfn</u>	Tag dfn v HTML 5 označuje části textu, kde definujeme nějaký pojem.
<u>em</u>	Tag em v HTML 5 označuje část textu, která má větší význam než okolní text. Text je vykreslen kurzívou.
<u>Frázové tagy</u>	Mezi frázové tagy v HTML (pro označení textu) patří elementy abbr, em, strong, dfn, code, samp, kbd, var, b, i, u, s a mark.
<u>h1</u>	Tag h1 v HTML 5 označuje nadpis nejvyšší úrovně, který obvykle obsahuje název webových stránek.
<u>h2</u>	Tag h2 v HTML 5 označuje nadpis článku či podstránky. Leží vždy pod nadpisem h1.
<u>h3</u>	Tag h3 v HTML 5 označuje nějakou menší sekci než nadpis úrovně druhé.

<u>h4</u>	Tag h4 v HTML 5 označuje nějakou menší sekci než nadpis úrovně třetí.
<u>h5</u>	Tag h5 v HTML 5 označuje nějakou menší sekci než nadpis úrovně čtvrté. Jeho používání je však řídké.
<u>h6</u>	Tag h6 v HTML 5 označuje tu nejmenší možnou sekci. Téměř se nepoužívá.
<u>i</u>	Tag i v HTML 5 zvýrazňuje text, je vykreslován kurzívou.
<u>ins</u>	Tag ins v HTML 5 slouží k označení nově vloženého textu po editaci dokumentu. Obsahuje atributy cite a datetime.
<u>kbd</u>	Tag kdb v HTML 5 označuje text, který má být uživatelem vložen z klávesnice. Patří mezi frázové tagy.
<u>mark</u>	Tag mark v HTML 5 se používá při zvýraznění části citace, která je klíčová.
<u>Nadpisy</u>	Nadpisy slouží v HTML pro lepší orientaci a jsou klíčové pro SEO. Například Google extrahuje informace právě z nadpisů.
<u>p</u>	Tag p se v HTML 5 používá k rozdělování textu do odstavců. Dokument je poté přehlednější a lépe se styluje.
<u>q</u>	Tag q označuje krátkou řádkovou (inline) citaci. Obsahuje atribut cite.
<u>rp</u>	Tag rp v HTML 5 obsahuje většinou závorky, které se zobrazí v případě, že prohlížeč ruby anotace nepodporuje.
<u>rt</u>	Tag rt v HTML 5 označuje danou informaci o výslovnosti v tzv. ruby anotaci.
<u>ruby</u>	Tag ruby v HTML 5 definuje tzv. ruby anotace, které se používají v asijské typografii.
<u>ruby, rt a rp</u>	Tagy ruby, rt a rp se v HTML 5 používají k vložení tzv. ruby anotací do asijské typografie.
<u>s</u>	Tag s v HTML 5 vykresluje obsah přeškrtnutě (zpravidla označuje text, který již není aktuální nebo korektní).
<u>samp</u>	Tag samp v HTML 5 slouží k označení textu.
<u>strong</u>	Tag strong v HTML 5 označuje silnější zdůraznění než tag em. Text je vykreslen tučně.
<u>sub</u>	Tag sub se v HTML 5 používá k označení dolního indexu. Využijeme ho např. při psaní vzorců nebo indexaci proměnných.
<u>sup</u>	Tag sup se v HTML 5 používá k označení horního indexu. Využijeme ho při psaní mocnin nebo poznámek pod čarou.
<u>u</u>	Tag u v HTML 5 označuje podtržený text.
<u>var</u>	Tag var v HTML 5 slouží k označení proměnné v textu a patří mezi frázové tagy. Vykresluje se kurzívou.
<u>wbr</u>	Tag wbr v HTML 5 označuje místo v dlouhém slově, kde je vhodné ho zalomit.

6.5.3 LAYOUT

<u>article</u>	Tag article v HTML 5 označuje samotný článek. Ten může mít svoji hlavičku a i patičku.
<u>aside</u>	Tag aside v HTML 5 označuje postranní panel, který obvykle obsahuje doplňky k článku.
<u>details</u>	Tag details v HTML 5 označuje rozbalovací sekci s detaily. Pomocí tagu summary ji přiřadíme titulek.
<u>figcaption</u>	Tag figcaption v HTML 5 popisuje obsah v jeho rodičovském tagu figure.
<u>figure</u>	Tag figure v HTML 5 označuje samostatnou ilustraci, týkající se článku. Její nadpis vložíme pomocí tagu figcaption.
<u>footer</u>	Tag footer v HTML 5 označuje patičku stránky. Obvykle obsahuje copyright a informace o autorovi.
<u>header</u>	Tag header v HTML 5 označuje hlavičku stránky. Ta zpravidla obsahuje nadpis h1.
<u>Layout (rozložení stránky)</u>	Pro rozložení stránky do sekcí (layoutu). Se používají především tagy header, hgroup, nav, footer, section, article a aside.
<u>nav</u>	Tag nav v HTML 5 obsahuje navigační prvky. Může být součástí header nebo stát samostatně pod ním.
<u>section</u>	Tag section v HTML 5 se používá zejména k označení "těla" dokumentu mezi hlavičkou a patičkou.
<u>summary</u>	Tag summary v HTML 5 označuje titulek sekce s detaily, kdy po kliknutí na něj se ukážou detaily.

6.5.4 TABULKY

<u>caption</u>	Tag caption v HTML 5 označuje nadpis tabulky, který je ve výchozím nastavení vycentrován.
<u>col</u>	Tag col v HTML 5 definuje styl sloupců ve skupině. Nachází se v rodičovském tagu colgroup.
<u>colgroup</u>	Tag colgroup v HTML 5 umožňuje seskupovat sloupce do skupin a těm poté nastavovat různé CSS styly.
<u>Jednoduché tabulky</u>	Ke tvorbě jednoduchých HTML tabulek se využívají tagy table, tr, td, th, atributy border, colspan, rowspan, headers a scope.
<u>Pokročilé tabulky</u>	Popis tagů k tvorbě pokročilých HTML tabulek. Tagy thead, tfoot, tbody, caption, colgroup, col a atribut span.
<u>table</u>	Tag table v HTML 5 označuje tabulku a obsahuje jednotlivé řádky s buňkami.
<u>tbody</u>	Tag tbody v HTML 5 označuje tělo tabulky, které obsahuje její data.
<u>td</u>	Tag td v HTML 5 označuje buňku tabulky, která může obsahovat text, obrázky a další libovolné elementy.

<u>tfoot</u>	Tag tfoot v HTML 5 označuje patičku tabulky, ve které je obvykle její celé shrnutí.
<u>th</u>	Tag th v HTML 5 označuje hlavičkovou buňku tabulky. Ve výchozím nastavení je obsah vykreslen uprostřed a je tučný.
<u>thead</u>	Tag thead v HTML 5 označuje hlavičku tabulky, kde je obvykle popis jednotlivých sloupců.
<u>tr</u>	Tag tr v HTML 5 označuje řádek tabulky a obsahuje jednotlivé buňky.

6.5.5 SEZNAMY

<u>datalist</u>	Tag autocomplete v HTML 5 označuje seznam možností, které jsou následně nabízeny v inputu pomocí našeptávače.
<u>dd</u>	Tag dd v HTML 5 označuje popis vysvětlovaného pojmu v slovníčku pojmů.
<u>dl</u>	Tag dl v HTML 5 označuje slovníček pojmů. Také má své vlastní tagy pro položky.
<u>dt</u>	Tag dt v HTML 5 označuje ve slovníčku pojmů vysvětlovaný pojem.
<u>li</u>	Tag li v HTML 5 označuje jednu položku seznamu a nejčastěji obaluje text.
<u>ol</u>	Tag ol v HTML 5 označuje uspořádaný seznam, který je řazen dle nějakého klíče.
<u><i>Seznamy a slovníček pojmů</i></u>	Popis tagů k tvorbě HTML seznamů a slovníku pojmů - ul, ol, li, dl, dt, dd, atributy reversed, start, type a value.
<u>ul</u>	Tag ul v HTML 5 označuje neuspořádaný seznam. Standardně mají jeho položky odrážky.

6.5.6 MULTIMÉDIA

<u>area</u>	Tag area v HTML 5 označuje jednu oblast na mapě (obrázku). Typicky se jedná o různé tvary.
<u>audio</u>	HTML 5 tag audio se používá pro vložení zvuků nebo hudby do HTML stránky.
<u>canvas</u>	Tag canvas v HTML 5 označuje plátno, na které je možné vykreslovat např. JavaScriptem. Používá se pro animace a hry.
<u>embed</u>	Tag embed se používá k vložení externí aplikace nebo plug-inu. Často se používá k vložení Flash aplikace.
<u>img</u>	Tag img v HTML 5 označuje obrázek, který má v sobě několik atributů. Je nepárový.
<u>map</u>	Tag map v HTML 5 umožňuje definovat mapu na obrázku přímo na straně klienta.
<u>object</u>	Tag object se v HTML 5 používá k vložení multimediální aplikace do HTML dokumentu. Nejčastěji se jedná o Flash.

<u>Obrázky a obrázková mapa</u>	Tagu img pro vložení obrázku do stránky používá atributy src, alt, width, height. Umožňuje vytvářet obrázkové klikací mapy v kombinaci s elementy map a area.
<u>source</u>	HTML 5 tag source se používá pro vložení zdroje videa nebo hudby pro tagy audio a video.
<u>track</u>	HTML 5 tag track se používá pro vložení textové stopy pro tagy audio a video.
<u>video</u>	HTML 5 tag video se používá pro vložení videa do HTML stránky.

6.5.7 FORMULÁŘE

<u>button</u>	Tag button v HTML 5 označuje tlačítko, které nemusí být bezprostředně součástí formuláře a může obsahovat další tagy.
<u>fieldset</u>	Tag fieldset se v HTML 5 používá s tagem legend k seskupení formulářových polí a může být označen i jako GroupBox.
<u>Formuláře</u>	Tag form se používá ke tvorbě webových formulářů v HTML a může obsahovat atributy method, action, name, autocomplete, accept-charset, target a enctype.
<u>input</u>	Tag input se používá ve formulářích pro vložení různých vstupních polí.
<u>input typu file</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ file slouží k nahrání souborů na web.
<u>input typu hidden</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ hidden slouží k vložení skrytého pole.
<u>input typu checkbox</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ checkbox slouží k vložení zaškrtnávacího pole.
<u>input typu password</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ password slouží k zadání hesla.
<u>input typu radio</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ radio slouží k vložení přepínacího tlačítka.
<u>input typu reset</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ reset obnoví výchozí hodnoty ve formuláři.
<u>input typu submit</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ submit slouží k odeslání formuláře.
<u>input typu text</u>	Tag input se používá v HTML formulářích pro vložení různých vstupních polí. Typ text slouží k zadání krátkého textu.
<u>keygen</u>	Tag keygen se v HTML 5 používá k bezpečnému odeslání formuláře pomocí certifikátu.
<u>Label</u>	Popis tagu label, který slouží jako popis polí v HTML formulářích. Obsahuje atributy for a form.
<u>output</u>	Tag output v HTML 5 označuje výsledek nějaké početní operace. Může se jednat např. o výstup JavaScriptu.
<u>Select</u>	Popis tagu select, který umožňuje do HTML formuláře vložit vysouvací nabídku (combo box). Souvisí s tagy option a optgroup.

textarea Slouží ke vložení textového pole do HTML formuláře. Obsahuje atributy autofocus, cols, disabled, form, maxlength, name, placeholder, readonly.

6.5.8 OSTATNÍ TAGY

a Tag a v HTML 5 označuje hypertextový odkaz. Pomocí něj se pohybujeme v rámci našeho webu nebo odkazujeme na web cizí.

address Tag address v HTML 5 poskytuje kontaktní informace k dokumentu nebo jeho části. Bývá špatně chápán, neoznačuje adresu.

bdi Tag bdi v HTML 5 označuje text, který může být psán jiným směrem textu, než text okolní.

bdo Tag bdo v HTML 5 označuje přepsání aktuálního směru textu, např. na směr zprava doleva.

blockquote Tag blockquote označuje blokovou citaci.

br Tag br se používá k zalomení řádku. Můžeme tak např. zalamovat text v odstavci.

command Tag command se v HTML 5 používá k vložení příkazu do kontextového menu nebo k zachycení klávesové zkratky ve stránce.

div Tag div se v HTML 5 používá k seskupování logicky souvisejících blokových elementů a k jejich stylování.

hr Tag hr v HTML 5 označuje oddělovač, který je vykreslen jako horizontální čára. Používá se při změně tématu.

iframe Tag iframe v HTML 5 označuje inline rámeček, umožňuje tedy vložení další HTML stránky do HTML dokumentu.

komentáře Komentáře slouží v HTML pro autora stránky jako poznámky k lepší orientaci zejména v komplexním kódu.

menu Tag menu umožňuje v HTML 5 vložit toolbar s menu, podobně jako ho známe např. z Windows aplikací.

meter Tag meter v HTML 5 označuje měřič, podobný progressbaru, který graficky zobrazuje určitou část z celku.

progress Tag progress v HTML 5 označuje ukazatel pokroku v nějaké činnosti, známý také jako progressbar.

script Tagy script a noscript slouží ke vložení klientských skriptů (JavaScriptu) do HTML dokumentu. Obsahují atributy async, defer, type, charset a src.

span Nemá žádný sémantický význam. Používá se k seskupování logicky souvisejících řádkových (inline) elementů a funguje tedy podobně, jako element <div>.

time HTML 5 tag time se používá pro označení data nebo času v HTML stránce.

6.5.9 ZASTARALÉ TAGY (PODPOROVANÉ Z DŮVODŮ KOMPATIBILITY)

<u>big</u>	Tag big v HTML dříve označoval větší text, než byl ten normální. V současnosti se větší text realizuje pomocí CSS.
<u>center</u>	Tag center v HTML vycentroval obsah. Nově se centruje pomocí CSS vlastností.
<u>font</u>	Tag font v HTML umožňoval nastavit typ, velikost a barvu písma. Nyní písmo nastavujeme v CSS dokumentu.
<u>frame</u>	Tag frame v HTML označoval rámeček, ve kterém se zobrazovala samostatná HTML stránka.
<u>frameset</u>	Tag frameset v HTML označoval soubor rámečků. Z HTML byly však rámečky již odebrány.
<u>noframes</u>	Tag noframes v HTML obsahoval text, který se zobrazil, když prohlížeč nepodporoval rámečky.
<u>small</u>	Tag small v HTML označoval a označuje menší text než ten okolní. Má stejnou velikost jako nadpis 5. úrovně.
<u>strike</u>	Tag strike v HTML označoval přeškrtnutý text. Nově se to dělá přes CSS vlastnost.
<u>Stylování</u>	Pro snazší hromadnou modifikovatelnost do HTML stránky vkládáme pouze obsah. Úprava vzhledu elementů (stylování) se vyčlenila do samostatného souboru CSS.

Jazyku HTML5 je věnován poměrně široký prostor na www stránkách, doporučuji dále studium např. <https://www.zdrojak.cz/serialy/webdesigneruv-pruvodce-po-html5/>, www.jakpsatweb.cz nebo <https://www.zdrojak.cz/serialy/webdesigneruv-pruvodce-po-html5/>

6.6 Kaskádové styly

6.6.1 CHARAKTERISTIKA CSS3

CSS3³⁶ – Jsou velkým přínosem pro internetové aplikace a jejich vzhled, jsou chápány jako součást HTML5 označovaná jako CSS3, která slouží pro stylování internetových dokumentů. Verze CSS3 je v podstatě průlom grafického zpracování webů. Umožňuje totiž vytvořit téměř každý grafický prvek stránky pouze pomocí tohoto stylovacího jazyka a tím velmi urychlit načítání stránek. Dalo by se zjednodušeně říct, že je tím grafika definována jako vektorová, a tedy je možné ji zobrazovat v jakýchkoliv zařízeních bez změny kvality, což je u použití obrázkové grafiky nemožné. Je tedy velmi snadné dokument přizpůsobit

³⁶ Převzato z <https://www.root.cz/clanky/html5-co-prinasi-a-proc-se-o-nej-zajimat/>

velikosti uživatelské obrazovky. Možnosti lze shrnout pro aplikaci v následujících oblastech::

- Animace.
- Možnosti background-image (obrázkové pozadí).
- Border image (obrázkový rámeček).
- Border radius (zaoblené rohy).
- Box shadows (stíny).
- Rozměry.
- Position: Fixed (fixní pozice).
- Font settings (možnosti nastavení písma).
- Přechody.
- Hyphenation (dělení slov).
- Media queries.
- Multiple-column layout.
- Multiple backgrounds.
- Opacity (neprůhlednost).
- Pointer events.
- Text overflow.
- Text shadow (stín textu).
- Transforms 2D (2d transformace).
- Transform 3D (3d transformace).
- Transitions.

Opět, obdobně jako u specifikace HTML5 uvádím abecední rejstřík tak, jak je specifikován na [www](http://www.w3.org) v příspěvku „Abecední rejstřík - Český CSS 3 manuál“³⁷.

6.6.2 PŘÍKAZY A KLÍČOVÉ VÝTAZY

Text a písmo

<u>@font-face</u>	Pomocí CSS 3 pravidla @font-face vytváříme vlastní font s jeho vlastním názvem.
<u>color</u>	Pomocí CSS 3 vlastnosti color nastavujeme barvu písma/textu v HTML elementu.
<u>content</u>	Pomocí CSS 3 vlastnosti content nastavujeme, co bude před a za elementem.
<u>counter-increment</u>	Pomocí CSS 3 vlastnosti counter-increment můžeme přičítat k číslu vlastní hodnotu a tak tvořit seznamy.

³⁷ Převezto z <http://www.itnetwork.cz/html-css/css-manual/cesky-css-3-manual-rejstrik>

<u>counter-reset</u>	Pomocí CSS 3 vlastnosti counter-reset definujeme ID počítadla a zároveň ho resetujeme.
<u>direction</u>	Pomocí CSS 3 vlastnosti direction nastavujeme směr textu v HTML elementu, tedy zleva doprava nebo zprava doleva.
<u>font</u>	Pomocí CSS 3 vlastnosti font nastavíme písmo v HTML elementu, styl, variantu, tloušťku, velikost, řádkovaná a rodinu.
<u>font-family</u>	Pomocí CSS 3 vlastnosti font-family nastavujeme rodinu písma v HTML elementu. Rodinou písma se myslí font (např. Arial).
<u>font-size</u>	Pomocí CSS 3 vlastnosti font-size nastavujeme velikost písma/textu v HTML elementu.
<u>font-size-adjust</u>	Pomocí CSS 3 vlastnosti font-size-adjust nastavujeme velikost písma fontu. Podporuje pouze Mozilla Firefox.
<u>font-stretch</u>	Pomocí CSS 3 vlastnosti font-stretch nastavujeme, zda bude text užší nebo širší. Nepodporuje žádný webový prohlížeč.
<u>font-style</u>	Pomocí CSS 3 vlastnosti font-style nastavujeme styl písma v HTML elementu. Vlastnost se používá pro kurzívu.
<u>font-variant</u>	Pomocí CSS 3 vlastnosti font-variant nastavujeme variantu písma v HTML elementu. Vlastnost se používá pro kapitálky.
<u>font-weight</u>	Pomocí CSS 3 vlastnosti font-weight nastavujeme tučnost písma v HTML elementu. Vlastnost se používá pro tučné písmo.
<u>Fonts (písma)</u>	Praktická ukázka vytvoření a použití vlastního písma pomocí CSS3. Již žádná závislost na web-safe fonts.
<u>hanging-punctuation</u>	Pomocí CSS 3 vlastnosti hanging-punctuation nastavujeme, jak bude umístěné interpunkční znaménko.
<u>letter-spacing</u>	Pomocí CSS 3 vlastnosti letter-spacing zvyšujeme rozestoupení znaků v textu, tedy mezery mezi jednotlivými písmeny.
<u>line-height</u>	Pomocí CSS 3 vlastnosti line-height nastavujeme výšku řádku textu, tedy řádkování.
<u>Multiple Columns (sloupce)</u>	Členění textu do sloupců jako například v novinách za použití CSS3.
<u>quotes</u>	Pomocí CSS 3 vlastnosti quotes nastavujeme vlastní uvozovky pro citace.
<u>tab-size</u>	Pomocí CSS 3 vlastnosti tab-size nastavujeme velikost znaku tabulátoru. Funguje pouze pro elementy textarea a pre.
<u>Text Effects (efekty textu)</u>	CSS3 manuál. Ukázka práce s novými možnostmi textu a efekty, které nám CSS3 přináší. Stínování, zalamování, tvorba efektů.
<u>text-align</u>	Pomocí CSS 3 vlastnosti text-align stylujeme zarovnání textu v HTML elementu, např. vlevo, vpravo, na střed a do bloku.
<u>text-align-last</u>	Pomocí CSS 3 vlastnosti text-align-last nastavujeme zarovnání posledního řádku.
<u>text-decoration</u>	Pomocí CSS 3 vlastnosti text-decoration nastavujeme dekoraci textu v HTML elementu, tou se myslí např. podtržení.
<u>text-decoration-color</u>	Pomocí CSS 3 vlastnosti text-decoration-color nastavujeme barvu podtrhnutí.

<u>text-decoration-line</u>	Pomocí CSS 3 vlastnosti text-decoration-line nastavujeme textu čáru (např. podtržení).
<u>text-decoration-style</u>	Pomocí CSS 3 vlastnosti text-decoration-style nastavujeme styl čáry u textu.
<u>text-indent</u>	Pomocí CSS 3 vlastnosti text-indent nastavujeme odsazení prvního řádku odstavce, jako je to u českých odstavců.
<u>text-justify</u>	Pomocí CSS 3 vlastnosti text-justify nastavujeme mezery k hodnotě justify (vlastnost text-align).
<u>text-overflow</u>	Pomocí CSS 3 vlastnosti text-overflow nastavujeme chování vytékání textu z HTML elementu a jeho případné zkrácení.
<u>text-shadow</u>	Pomocí CSS 3 vlastnosti text-shadow přidáme textu stín. Docílíme tak hezkého efektu např. v nadpisech.
<u>text-transform</u>	Pomocí CSS 3 vlastnosti text-transform ostylujeme text tak, aby se zobrazoval pouze velkými nebo malými písmeny.
<u>unicode-bidi</u>	Pomocí CSS 3 vlastnosti unicode-bidi můžeme text vykreslit zrcadlově převrácený.
<u>white-space</u>	Pomocí CSS 3 vlastnosti white-space nastavujeme chování mezer v textu. Můžeme ponechat několik mezer vedle sebe.
<u>word-break</u>	Pomocí CSS 3 vlastnosti word-break nastavujeme rozdělování slov v CJK textu.
<u>word-spacing</u>	Pomocí CSS 3 vlastnosti *word-spacing* nastavujeme rozestupy (mezery) mezi jednotlivými slovy v textu.
<u>word-wrap</u>	Pomocí CSS 3 vlastnosti word-wrap nastavujeme zalamování (rozdělování) slov v textu.

Pozadí

<u>background</u>	CSS 3 vlastnost background slouží k ostylování pozadí HTML elementu. Umožňuje přidat vícenásobné pozadí.
<u>background-attachment</u>	CSS 3 vlastnost background-attachment slouží k přichycení pozadí HTML elementu jako fixní při rolování stránkou.
<u>background-clip</u>	CSS 3 vlastnost background-clip slouží k oříznutí pozadí HTML elementu.
<u>background-color</u>	CSS 3 vlastnost background-color slouží k nastavení barvy pozadí HTML elementu.
<u>background-image</u>	CSS 3 vlastnost background-image slouží k nastavení obrázku pozadí HTML elementu. Umožňuje přidat vícenásobné pozadí.
<u>background-origin</u>	CSS 3 vlastnost background-origin nastavuje, k čemu se má vztahovat pozice obrázku na pozadí HTML elementu.
<u>background-position</u>	CSS 3 vlastnost background-position slouží k nastavení pozice pozadí HTML elementu.
<u>background-repeat</u>	CSS 3 vlastnost background-repeat slouží k nastavení opakování pozadí HTML elementu.

background-size CSS 3 vlastnost background-size slouží ke změně velikosti pozadí HTML elementu.

Backgrounds (pozadí) CSS3 manuál. Vlastnost background. Ukázka práce s novými možnostmi, které nám CSS3 přináší. Více obrázků na pozadí, velikost, poloha zobrazení.

Pozicování a velikost

<u>bottom</u>	Pomocí CSS 3 vlastnosti bottom nastavujeme dolní pozici HTML elementu.
<u>box-sizing</u>	Pomocí CSS 3 vlastnosti box-sizing nastavujeme, co bude šířka a výška elementu zahrnovat.
<u>clear</u>	Pomocí CSS 3 vlastnosti clear nastavujeme, v které části elementu není povolen plovoucí obsah.
<u>clip</u>	Pomocí CSS 3 vlastnosti clip nastavujeme viditelnou část absolutně pozicovaných elementů.
<u>display</u>	Pomocí CSS 3 vlastnosti display nastavujeme jakým způsobem má být HTML element vykreslován.
<u>float</u>	Pomocí CSS 3 vlastnosti float nastavujeme, zda má být element plovoucí.
<u>height</u>	Pomocí CSS 3 vlastnosti height nastavujeme výšku HTML elementu, můžeme tak měnit jeho velikost/rozměry.
<u>left</u>	Pomocí CSS 3 vlastnosti left nastavujeme levou pozici HTML elementu.
<u>max-height</u>	Pomocí CSS 3 vlastnosti max-height nastavujeme maximální výšku HTML elementu. To je výhodné zejména u obrázků.
<u>max-width</u>	Pomocí CSS 3 vlastnosti max-width nastavujeme maximální šířku HTML elementu. To je výhodné zejména u obrázků.
<u>min-height</u>	Pomocí CSS 3 vlastnosti min-height nastavujeme minimální výšku HTML elementu.
<u>min-width</u>	Pomocí CSS 3 vlastnosti min-width nastavujeme minimální šířku HTML elementu.
<u>overflow</u>	Pomocí CSS 3 vlastnosti overflow nastavujeme chování HTML elementu v případě, že jeho obsah přeteče rozměry elementu.
<u>overflow-x</u>	Pomocí CSS 3 vlastnosti overflow-x nastavujeme chování elementu v případě, že jeho obsah přeteče šířku elementu.
<u>overflow-y</u>	Pomocí CSS 3 vlastnosti overflow-y nastavujeme chování elementu v případě, že jeho obsah přeteče výšku elementu.
<u>position</u>	Pomocí CSS 3 vlastnosti position nastavujeme typ pozice elementu, statickou, relativní, absolutní nebo fixní.
<u>resize</u>	Pomocí CSS 3 vlastnosti resize můžeme uživateli dovolit roztahovat element jako například u textarey.
<u>right</u>	Pomocí CSS 3 vlastnosti right nastavujeme pravou pozici HTML elementu.

<u>top</u>	Pomocí CSS 3 vlastnosti top nastavujeme horní pozici HTML elementu.
<u>vertical-align</u>	Pomocí CSS 3 vlastnosti vertical-align nastavujeme svislé zarovnání HTML elementu.
<u>visibility</u>	Pomocí CSS 3 vlastnosti visibility můžeme skrýt HTML element.
<u>width</u>	Pomocí CSS 3 vlastnosti width nastavujeme šířku HTML elementu, můžeme tak měnit jeho velikost/rozměry.
<u>z-index</u>	Pomocí CSS 3 vlastnosti z-index nastavujeme hloubku (z-souřadnici) HTML elementu na stránce.

Rámeček

<u>border</u>	Pomocí CSS 3 vlastnosti border nastavujeme rámeček okolo HTML elementu.
<u>border-bottom</u>	Pomocí CSS 3 vlastnosti border-bottom nastavujeme spodní hranu rámečku okolo HTML elementu.
<u>border-bottom-color</u>	Pomocí CSS 3 vlastnosti border-bottom-color nastavujeme barvu dolní hrany rámečku okolo HTML elementu.
<u>border-bottom-left-radius</u>	Pomocí CSS 3 vlastnosti border-bottom-left-radius nastavujeme poloměr zakulacení levého dolního rohu rámečku.
<u>border-bottom-right-radius</u>	Pomocí CSS 3 vlastnosti border-bottom-right-radius nastavujeme poloměr zakulacení pravého dolního rohu rámečku.
<u>border-bottom-style</u>	Pomocí CSS 3 vlastnosti border-bottom-style nastavujeme styl čáry dolní hrany rámečku okolo HTML elementu.
<u>border-bottom-width</u>	Pomocí CSS 3 vlastnosti border-bottom-width nastavujeme šířku (tloušťku) spodní hrany rámečku okolo HTML elementu.
<u>border-color</u>	Pomocí CSS 3 vlastnosti border-color nastavujeme barvu rámečku okolo HTML elementu.
<u>border-image</u>	Pomocí CSS 3 vlastnosti border-image nastavíme obrázek jako rámeček okolo HTML elementu.
<u>border-image-outset</u>	Pomocí CSS 3 vlastnosti border-image-outset nastavujeme, jak daleko od obsahu divu má být obrázkový rámeček.
<u>border-image-repeat</u>	Pomocí CSS 3 vlastnosti border-image-repeat nastavujeme, jestli se bude obrázkový rámeček opakovat.
<u>border-image-slice</u>	Pomocí CSS 3 vlastnosti border-image-slice nastavujeme, jak se obraz rámečku uřízne.
<u>border-image-source</u>	Pomocí CSS 3 vlastnosti border-image-source nastavujeme adresu k obrázku rámečku.
<u>border-image-width</u>	Pomocí CSS 3 vlastnosti border-image-width nastavujeme šířku rámečku s obrázkem.
<u>border-left</u>	Pomocí CSS 3 vlastnosti border-left nastavujeme levou hranu rámečku okolo HTML elementu.
<u>border-left-color</u>	Pomocí CSS 3 vlastnosti border-left-color nastavujeme barvu levé hrany rámečku okolo HTML elementu.

<u>border-left-style</u>	Pomocí CSS 3 vlastnosti border-left-style nastavujeme styl čáry levé hrany rámečku okolo HTML elementu.
<u>border-left-width</u>	Pomocí CSS 3 vlastnosti border-left-width nastavujeme šířku (tloušťku) levé hrany rámečku okolo HTML elementu.
<u>border-radius</u>	Pomocí CSS 3 vlastnosti border-radius nastavujeme poloměr zakulacení rohů rámečku okolo HTML elementu.
<u>border-right</u>	Pomocí CSS 3 vlastnosti border-right nastavujeme pravou hranu rámečku okolo HTML elementu.
<u>border-right-color</u>	Pomocí CSS 3 vlastnosti border-right-color nastavujeme barvu pravé hrany rámečku okolo HTML elementu.
<u>border-right-style</u>	Pomocí CSS 3 vlastnosti border-right-style nastavujeme styl čáry pravé hrany rámečku okolo HTML elementu.
<u>border-right-width</u>	Pomocí CSS 3 vlastnosti border-right-width nastavujeme šířku (tloušťku) pravé hrany rámečku okolo HTML elementu.
<u>border-style</u>	Pomocí CSS 3 vlastnosti border-style nastavujeme styl čáry rámečku okolo HTML elementu.
<u>border-top</u>	Pomocí CSS 3 vlastnosti border-top nastavujeme horní hranu rámečku okolo HTML elementu.
<u>border-top-color</u>	Pomocí CSS 3 vlastnosti border-top-color nastavujeme barvu horní hrany rámečku okolo HTML elementu.
<u>border-top-left-radius</u>	Pomocí CSS 3 vlastnosti border-top-left-radius nastavujeme poloměr zakulacení levého horního rohu rámečku.
<u>border-top-right-radius</u>	Pomocí CSS 3 vlastnosti border-top-right-radius nastavujeme poloměr zakulacení pravého horního rohu rámečku.
<u>border-top-style</u>	Pomocí CSS 3 vlastnosti border-top-style nastavujeme styl čáry horní hrany rámečku okolo HTML elementu.
<u>border-top-width</u>	Pomocí CSS 3 vlastnosti border-top-width nastavujeme šířku (tloušťku) horní hrany rámečku okolo HTML elementu.
<u>border-width</u>	Pomocí CSS 3 vlastnosti border-width nastavujeme šířku rámečku okolo HTML elementu.
<u>Borders (rámečky)</u>	CSS3 manuál. Vlastnost border. Použití zakulacených rámečků, přidání vlastního stínu, použití obrázku jako rámečku.
<u>outline</u>	Pomocí CSS 3 vlastnosti outline nastavujeme obrys okolo rámečku HTML elementu.
<u>outline-color</u>	Pomocí CSS 3 vlastnosti outline-color nastavujeme barvu obrysu okolo HTML elementu.
<u>outline-style</u>	Pomocí CSS 3 vlastnosti outline-style nastavujeme styl čáry obrysu okolo HTML elementu.
<u>outline-width</u>	Pomocí CSS 3 vlastnosti outline-width nastavujeme šířku obrysu okolo HTML elementu.

Okraje (margin a padding)

<u>margin</u>	Pomocí CSS 3 vlastnosti margin nastavujeme okraj elementu, vzdálenost mezi rámečkem a okolím elementu.
---------------	--

<u>margin-bottom</u>	Pomocí CSS 3 vlastnosti margin-bottom nastavujeme dolní okraj elementu, vzdálenost mezi dolní hranou rámečku a okolím.
<u>margin-left</u>	Pomocí CSS 3 vlastnosti margin-left nastavujeme levý okraj elementu, vzdálenost mezi levou hranou rámečku a okolím.
<u>margin-right</u>	Pomocí CSS 3 vlastnosti margin-right nastavujeme pravý okraj elementu, vzdálenost mezi pravou hranou rámečku a okolím.
<u>margin-top</u>	Pomocí CSS 3 vlastnosti margin-top nastavujeme horní okraj elementu, vzdálenost mezi horní hranou rámečku a okolím.
<u>padding</u>	Pomocí CSS 3 vlastnosti padding nastavujeme vzdálenost mezi rámečkem a obsahem elementu.
<u>padding-bottom</u>	Pomocí CSS 3 vlastnosti padding-bottom nastavujeme vzdálenost mezi dolní hranou rámečku a obsahem elementu.
<u>padding-left</u>	Pomocí CSS 3 vlastnosti padding-left nastavujeme vzdálenost mezi levou hranou rámečku a obsahem elementu.
<u>padding-right</u>	Pomocí CSS 3 vlastnosti padding-right nastavujeme vzdálenost mezi pravou hranou rámečku a obsahem elementu.
<u>padding-top</u>	Pomocí CSS 3 vlastnosti padding-top nastavujeme vzdálenost mezi horní hranou rámečku a obsahem elementu.

Seznamy a tabulky

<u>border-collapse</u>	Pomocí CSS 3 vlastnosti border-collapse stylujeme rámeček HTML tabulky tak, aby se vykresloval jako jednoduchá čára.
<u>border-spacing</u>	Pomocí CSS 3 vlastnosti border-spacing nastavujeme rozestupy mezi rámečky okolo buněk HTML tabulky.
<u>caption-side</u>	Pomocí CSS 3 vlastnosti caption-side nastavujeme umístění nadpisu HTML tabulky.
<u>empty-cells</u>	Pomocí CSS 3 vlastnosti empty-cells nastavujeme, zda se má skrýt rámeček a pozadí prázdných buněk HTML tabulky.
<u>list-style</u>	Pomocí CSS 3 vlastnosti list-style nastavujeme styl odrážek seznamů.
<u>list-style-image</u>	Pomocí CSS 3 vlastnosti list-style-image nastavíme obrázek odrážkám seznamu.
<u>list-style-position</u>	Pomocí CSS 3 vlastnosti list-style-position nastavíme pozici odrážek seznamu.
<u>list-style-type</u>	Pomocí CSS 3 vlastnosti list-style-type nastavíme typ odrážek seznamu. Máme na výběr různé tvary, číslované i abecední.
<u>table-layout</u>	Pomocí CSS 3 vlastnosti table-layout nastavujeme algoritmus výpočtu šířky sloupců HTML tabulky.

Ostatní vlastnosti

<u>@keyframes</u>	Pomocí CSS 3 pravidla @keyframes nastavujeme jméno animace a v jeho obsahu chování animace.
-------------------	---

<u>@media</u>	Pomocí CSS 3 pravidla @media nastavujeme rozdílné styly pro různá zařízení či různé média typy.
<u>2D Transforms (transformace)</u>	Libovolné natáčení, škálování, posun či natahování libovolného HTML elementu pomocí CSS3.
<u>3D Transforms (transformace)</u>	Libovolné natáčení, škálování, posun či natahování libovolného HTML elementu ve třetím rozměru pomocí CSS3.
<u>align-content</u>	Pomocí CSS 3 vlastnosti align-content nastavujeme vertikální pozicování ohebných elementů.
<u>align-items</u>	Pomocí CSS 3 vlastnosti align-items nastavujeme pozici ohebných sloupců.
<u>align-self</u>	Pomocí CSS 3 vlastnosti align-self nastavujeme pozici pouze daného ohebného sloupce.
<u>animation</u>	Pomocí CSS 3 vlastnosti animation si můžeme vytvářet vlastní animace pomocí pouhého CSS.
<u>animation-delay</u>	Pomocí CSS 3 vlastnosti animation-delay nastavujeme dobu před začátkem animace.
<u>animation-direction</u>	Pomocí CSS 3 vlastnosti animation-direction nastavujeme jakým směrem (a rychlostí) se animace přehraje.
<u>animation-duration</u>	Pomocí CSS 3 vlastnosti animation-duration nastavujeme dobu trvání animace.
<u>animation-fill-mode</u>	Pomocí CSS 3 vlastnosti animation-fill-mode nastavujeme, co se děje s elementem, když je animace pozastavená.
<u>animation-iteration-count</u>	Pomocí CSS 3 vlastnosti animation-iteration-count nastavujeme, kolikrát se bude animace opakovat.
<u>animation-name</u>	Pomocí CSS 3 vlastnosti animation-name nastavujeme jméno animace.
<u>animation-play-state</u>	Pomocí CSS 3 vlastnosti animation-play-state nastavujeme, zda je animace zastavená nebo má běžet.
<u>animation-timing-function</u>	Pomocí CSS 3 vlastnosti animation-timing-function nastavujeme rychlostní křivku animace.
<u>Animations (animace)</u>	Tvorba animací pomocí CSS3.
<u>backface-visibility</u>	Pomocí CSS 3 vlastnosti backface-visibility nastavujeme viditelnost zadní strany elementu.
<u>box-shadow</u>	Pomocí CSS 3 vlastnosti box-shadow nastavujeme stín pod HTML elementem.
<u>column-count</u>	Pomocí CSS 3 vlastnosti column-count nastavujeme počet sloupců, do kterých se text rozdělí.
<u>column-fill</u>	Pomocí CSS 3 vlastnosti column-fill nastavujeme, jak se sloupce budou naplňovat.
<u>column-gap</u>	Pomocí CSS 3 vlastnosti column-gap nastavujeme velikost mezery mezi sloupci.
<u>column-rule</u>	Pomocí CSS 3 vlastnosti column-rule nastavujeme pravidla stylů mezi sloupci.

<u>column-rule-color</u>	Pomocí CSS 3 vlastnosti column-rule-color nastavujeme barvu stylu mezi sloupci.
<u>column-rule-style</u>	Pomocí CSS 3 vlastnosti column-rule-style nastavujeme styl čar mezi sloupci.
<u>column-rule-width</u>	Pomocí CSS 3 vlastnosti column-rule-width nastavujeme šířku čar mezi sloupci.
<u>column-span</u>	Pomocí CSS 3 vlastnosti column-span nastavujeme, jak bude daný element roztažený napříč všemi sloupečky.
<u>column-width</u>	Pomocí CSS 3 vlastnosti column-width si můžeme rozdělit text do několika sloupců podle zadané šířky.
<u>columns</u>	Pomocí CSS 3 vlastnosti columns si můžeme rozdělit text do několika sloupců.
<u>cursor</u>	Pomocí CSS 3 vlastnosti cursor nastavujeme, jaký kurzor myši se má zobrazit nad HTML elementem.
<u>flex</u>	Pomocí CSS 3 vlastnosti flex nastavujeme velikost pružného elementu od ostatních.
<u>flex-basis</u>	Pomocí CSS 3 vlastnosti flex-basis nastavujeme základní velikost pružného elementu.
<u>flex-direction</u>	Pomocí CSS 3 vlastnosti flex-direction nastavujeme směr ohebných elementů.
<u>flex-flow</u>	Pomocí CSS 3 vlastnosti flex-flow nastavujeme směr a chování ohebných elementů při zaplnění toho rodičovského.
<u>flex-grow</u>	Pomocí CSS 3 vlastnosti flex-grow nastavujeme část velikosti rodičovského elementu pružnému elementu.
<u>flex-shrink</u>	Pomocí CSS 3 vlastnosti flex-shrink zmenšujeme pružný element, když všechny elementy jsou větší než ten rodičovský.
<u>flex-wrap</u>	Pomocí CSS 3 vlastnosti flex-wrap nastavujeme, zda se při zaplnění velikosti rodičovského elementu jeho potomci seřadí.
<u>justify-content</u>	Pomocí CSS 3 vlastnosti justify-content nastavujeme horizontální pozicování ohebných elementů.
<u>nav-down</u>	Pomocí CSS 3 vlastnosti nav-down definujeme posunutí označení šipkou dolů.
<u>nav-index</u>	Pomocí CSS 3 vlastnosti nav-index nastavujeme sekvenční navigační pořadí elementu.
<u>nav-left</u>	Pomocí CSS 3 vlastnosti nav-left definujeme posunutí označení šipkou doleva.
<u>nav-right</u>	Pomocí CSS 3 vlastnosti nav-right definujeme posunutí označení šipkou doprava.
<u>nav-up</u>	Pomocí CSS 3 vlastnosti nav-up definujeme posunutí označení šipkou nahoru.
<u>opacity</u>	Pomocí CSS 3 vlastnosti opacity nastavujeme průhlednost HTML elementu. Ukážeme si i jak na průhledné pozadí.
<u>order</u>	Pomocí CSS 3 vlastnosti order nastavujeme pořadí ohebných elementů.

<u>page-break-after</u>	Pomocí CSS 3 vlastnosti page-break-after nastavujeme zalomení stránky po elementu při tisku stránky.
<u>page-break-before</u>	Pomocí CSS 3 vlastnosti page-break-before nastavujeme zalomení stránky před elementem při tisku stránky.
<u>page-break-inside</u>	Pomocí CSS 3 vlastnosti page-break-inside nastavujeme zalomení stránky v elementu při tisku stránky.
<u>perspective</u>	Pomocí CSS 3 vlastnosti perspective nastavujeme vzdálenost mezi osou Z a uživatelem.
<u>perspective-origin</u>	Pomocí CSS 3 vlastnosti perspective-origin měníme 3D elementu pozici dolních bodů.
<u>transform</u>	Pomocí CSS 3 vlastnosti transform můžeme element různě transformovat (natáčet, zkosit, posouvat, natahovat atd.).
<u>transform-origin</u>	Pomocí CSS 3 vlastnosti transform-origin můžeme měnit pozici transformovaného elementu.
<u>transform-style</u>	Pomocí CSS 3 vlastnosti transform-style určujeme vnoření elementu v 3D prostoru.
<u>transition</u>	Pomocí CSS 3 vlastnosti transition můžeme vytvářet různé přechodné efekty např. pro změnu barvy elementu.
<u>transition-delay</u>	Pomocí CSS 3 vlastnosti transition-delay nastavujeme dobu před začátkem přechodného efektu.
<u>transition-duration</u>	Pomocí CSS 3 vlastnosti transition-duration nastavujeme dobu trvání přechodného efektu.
<u>transition-property</u>	Pomocí CSS 3 vlastnosti transition-property nastavujeme, co se bude měnit v přechodném efektu.
<u>transition-timing-function</u>	Pomocí CSS 3 vlastnosti transition-timing-function nastavujeme rychlostní křivku přechodného efektu.
<u>Transitions (přechody)</u>	Použití přechodového efektu při změně vlastnosti elementu.
<u>User interface (rozhraní)</u>	Nové vlastnosti pro změny velikosti elementu, resizování boxů, outline.

6.7 JavaScript

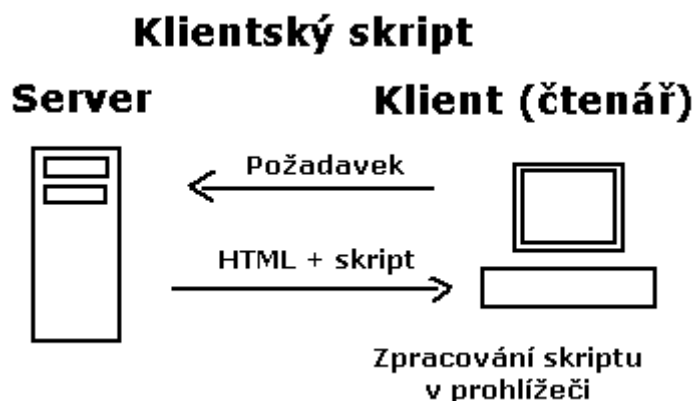
Jak bylo zmíněno, JavaScript (JS) umožňuje vložit akci na straně klienta.³⁸ Využití na straně klientské aplikace je podmíněno schopností HTML kódu „rozumět“ příkazům scriptů (ve smyslu přečtení příkazů) a schopností prohlížeče interpretovat příkazy JS. V praxi je prohlížečem zpracováván HTML kód do chvíle, než je nalezen příkaz pro předání řízení JavaScriptu. Následně jsou zpracovávány příkazy JS, s tím souvisí akceptace syntaxe JS (v době zpracovávání příkazů JS neplatí syntaxe HTML ale syntaxe JS). JS je zpracováván do chvíle, než je předáno opět řízení HTML.

³⁸ Předpokládáme standardní klientskou aplikaci, nikoli serverovou

Vzhledem k poměrně rozsáhlé problematice, vychází tato kapitola z textu publikovaném na www.jakpsatweb.cz, text je převzatý s drobnými úpravami, doporučuji důkladné prostudování uvedených stránek, které současně obsahují množství příkladů.

6.7.1 ÚVOD DO JAVASCRIPTU

JavaScript je programovací jazyk, který se používá v internetových stránkách. Zapisuje se přímo do HTML kódu, což je velká výhoda, protože je to jednoduché. JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván, jak je zřejmé z obrázku 42. (Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky.) Existují i jiné jazyky klientských skriptů, například VBScript. JavaScript je často zaměňován s Javou. Java je samostatný programovací jazyk. Má s JavaScriptem pouze podobnou syntaxi. Pro práci s JS je doporučeno zvládnutí základů HTML a základ programování.



Obrázek 63: Zpracování JS – klientský skript

Zdroj: www.jakpsatweb.cz

6.7.2 CHARAKTERISTIKY JAZYKA

JavaScript – charakteristika jazyka

- Interpretovaný -- nemusí se kompilovat.
- Objektový -- využívá objektů prohlížeče a zabudovaných objektů.
- Závislý na prohlížeči -- funguje ve většině prohlížečů.
- Case sensitivní -- záleží na velikosti písem v zápisu.
- Syntaxí podobný jazykům C, Java a podobným.

Omezení jazyka

- JavaScript funguje pouze v prohlížeči.
- Uživatel může JavaScript zakázat.

- Existují různé odlišné verze jazyka i prohlížečů, což vede k častým chybám.
- Neumí přistupovat k souborům (kromě cookies) ani k žádným systémovým objektům.
- Neumí žádná data uložit (kromě cookies).

Po zvládnutí základů je nejlepší všimnout si cizích skriptů na cizích stránkách. Většina skriptů je zapsána přímo ve zdrojovém kódu stránky, takže se dají zkopírovat.

6.7.3 OBJEKTIVÝ MODEL

JavaScript je jazyk objektový, třebaže nevyužívá všechny možnosti OOP (objektově orientovaného programování). V praxi často znamená "objektovost" javascriptu vlastně jenom to, že všechny vlastnosti a příkazy jsou uspořádány podle nějakého systému.

Objektový model je způsob, jak pojmenovat jednotlivé prvky okna prohlížeče a dokumentu, aby se s nimi dalo pracovat. Studium JavaScriptu je vlastně studium objektového modelu prohlížeče a dokumentu; je třeba se naučit, jak se které prvky zapisují.

Zápis

K adresování objektů se používá tečková syntaxe objektů (v jiných jazycích se používají např. šipky ->). Většina objektů má podobjekty nebo vlastnosti nebo metody; syntaxe je objekt.podobjekt, objekt.vlastnost nebo objekt.metoda().

Metoda objekt.metoda() je sama o sobě příkazem, který něco dělá. Má za sebou závorky. Vlastnost objekt.vlastnost nic nedělá, ale má hodnotu. Hodnota se dá číst nebo zapisovat, některé vlastnosti jsou jen pro čtení, některé jen pro zápis. Podobjekt objekt.podobjekt může mít další metody, vlastnosti a podobjekty. (V literatuře se podobjekt v některých případech považuje také za vlastnost.)

Přístup k objektům

- K objektům okna prohlížeče (window).
- Přes okno prohlížeče k prvkům stránky (window.document), které mají úzkou vazbu na jazyk HTML.
- K zabudovaným objektům (Date, Math, string).
- K vytvořeným objektům (to nepatří do objektového modelu a použije se to zřídka).

Nekompatibilita prohlížečů

Základní úskalí práce s JavaScriptem spočívá v tom, že objektové modely jednotlivých prohlížečů se liší. Některé objekty existují jenom v některých prohlížečích. V praxi je tedy třeba různými podmínkami testovat verzi prohlížeče a na základě toho skript větvit.

Práce s objekty

Stávající přehled není úplným výčtem, autor stránek www.jakpsatweb.cz je však považuje za prioritní, proto je uvádím i ve skriptech.

- Objekt window.
- Metody objektu window.
- Objekt window.event.
- Objekt document.
- Objekt String.
- Objekt Date.
- Objekt Math.

Dále se budeme věnovat zejména objektu window, s dalšími objekty doporučuji se seznámit na uváděných www stránkách.

6.7.4 OBJEKT WINDOW

Disponuje podobjekty: location - history - navigator - screen - frames - event - document - ostatní vlastnosti. Objekt window (okno) je vrcholem hierarchie objektů, se kterými může JavaScript pracovat. Celý objektový model (kromě Date, Math, String) je vlastně seznam podobjektů objektu window.

Pro zjednodušení lze při zápisu u podobjektů (vlastností) „window“ vynechávat. Např. zápis `window.navigator` a `navigator` jsou ekvivalentní.

6.7.5 VLASTNOSTI A METODY OBJEKTU WINDOW.

Objekt `window.location`

Nebo jenom `location`. Je to adresa načteného dokumentu.

Načtení nového dokumentu:

`window.location.href = "http://www.opf.slu.cz"`; načte do okna domovskou stránku OPF.

Aktualizace stránky `reload()`:

`location.reload(false)`; aktualizuje stránku, pokud byla změněna,

`location.reload(true)`; natáhne stránku ze serveru, i když se nezměnila

`location.replace("http://www.seznam.cz")`; natáhne stránku Seznamu, ale současnou stránku nezařadí do historie.

Zjištění adresy dokumentu (například načtením do proměnné adresa)

`adresa = location.href`;

Zjištění fragmentu adresy, například jméno domény:

`domena = location.hostname`;

Pokud by adresa byla třeba `http://www.seznam.cz/cokoliv/`, tak by se do proměnné vložila hodnota "www.seznam.cz".

Objekt `window.history`

Historie prohlížení stránek. Dovoluje vracet se ke dříve zobrazeným stránkám. Obsahuje několik metod, žádné vlastnosti ke čtení. Kvůli tomu jsou možnosti tohoto objektu velmi omezené. Pro spolehlivou složitější práci s historií je nutno ukládat informace do cookies nebo zkusit vlastnost `document.referrer`.

Metody objektu `history`:

`history.back()`; načte do prohlížeče minulou stránku. Je to to samé, jako když uživatel klikne na tlačítko Zpět v prohlížeči.

`history.back(3)`; o tři stránky zpět. Jako trojí kliknutí. Když se číslo neuvede (minulý příklad), je tam jako jednička.

`history.forward(2)`; o dvě stránky vpřed. Většinou je to nepoužitelné, protože metoda `forward` je přístupná jedině když se předtím zacouvalo.

`history.go(-1)`; je také návrat na minulou stránku. Číslo v argumentu může být libovolné celé číslo. Metodou `go()` se dají nahradit `back()` a `forward()`.

OBJEKT `WINDOW.NAVIGATOR`

Slouží ke zjišťování informací o typu a verzi prohlížeče.

`navigator.appName` vrací jméno aplikace prohlížeče

`navigator.appVersion` vrací verzi prohlížeče

`navigator.javaEnabled()` vrací informace o podpoře Java appletů

Většina autorů používá `window.navigator` pro větvení programu: když je `window.navigator` Internet Explorer, provede se skript optimalizovaný pro IE, když je to Netscape, provede se skript pro Netscape. Další vlastnosti objektu `navigator` najdete na zmiňovaných `www` stránkách.

Objekt `window.screen`

Objekt slouží pro zjišťování vlastností obrazovky.

`screen.height`, **`screen.width`** – zjistí výšku a šířku pracovní plochy grafického systému počítače.

`screen.availWidth`, **`screen.availHeight`** – zjistí dostupnou velikost plochy. Prakticky jde o totéž jako `width` a `height`, pouze výška bývá poněkud menší, pokud je stále zobrazen hlavní panel systému (šedá lišta dole). Ve skriptech doporučuji používat raději tyto vlastnosti.

`screen.colorDepth` – udává barevnou hloubku, počet bitů připadajících na jednu barvu. (Systémové nastavení obrazovky.)

Objekt `window.event`

Pomocí tohoto objektu lze nastavovat události myši a události klávesnice

Události myši

event.button - Vrací hodnotu podle stisknutých tlačítek (1 levé, 2 pravé, 4 prostřední).
event.clientX, **event.clientY** - vrací polohu myši vzhledem k levému hornímu rohu okna dokumentu.

event.screenX, **event.screenY** - vrací polohu myši vzhledem k levému hornímu rohu obrazovky.

event.offsetX, **event.offsetY** - souřadnice myši vzhledem k pozicovanému objektu (vrstvě).

Události klávesnice

event.altKey, **event.ctrlKey**, **event.shiftKey** - vracejí true / false (pravda nebo nepravda) podle toho, jestli je stisknutý alt, control resp. shift.

event.keyCode - vrací ascii kód stisknuté klávesy.

event.type – je vlastnost, vrací typ události, která nastala.

event.returnValue – vlastnost, mění předdefinované chování události

event.srcElement - vrací prvek, na němž událost nastala.

event.fromElement, **event.toElement** - vracejí prvky, ze kterého a na který myš při události jela. Vztahují se na vlastnosti onmouseover a onmouseout.

Objekt window.document

Objekt document je nejdůležitějším objektem, ke kterému JavaScript přistupuje. Document v sobě obsahuje všechno, co je nějakým způsobem spojeno s aktuální stránkou. Přes objekt document se přistupuje k obrázkům, formulářům, odkazům, k barvám atd. K objektům dokumentu přistupujeme proto, aby se mohly měnit. Například lze zaměnit obrázek, dá se změnit hodnota v políčku formuláře, spustit hudba, dají se měnit barvy dokumentu atd. To vše v reakci na uživatelské události nebo ještě při načítání stránky.

V praxi není důležité, že to jsou všechno operace na objektu document. Projevuje se to jenom zápisem, který vždy začíná "document. ...".

K jednotlivým prvkům na stránce se dá přistupovat pomocí objektů document.images, document.forms, document.applets, document.link, document.anchors, document.all, document.frames, document.styleSheets, document.scripts, document.selection a dalších.

K jednotlivým prvkům stránky se dá přistupovat více způsoby. Z nich nejzajímavější je metoda **document.getElementById()**. Dokáže zpřístupnit skriptu libovolný prvek na stránce, který má nastavené id="řetězec". Další podobné metody pro zpřístupnění prvků vždy začínají **getElementBy**, případně **getElementsBy**.

Ostatní vlastnosti objektu window

window.defaultStatus - ovlivní text ve stavové řádce

window.status - je skoro totéž, ale vázáno dočasně na událost

window.closed - vrací true, pokud je okno zavřeno (nešlape v IE 3)

window.opener - vrací odkaz (ukazatel) na okno, které jej otevřelo

window.length - vrací počet rámu v okně nebo rámu

window.name - vrací jméno rámu nebo okna

window.parent - vrací odkaz na nadřazený rám (okno)

window.self - vrací sebe samo

window.top - vrací odkaz na hierarchicky nejvyšší rám

window.offscreenBuffering - když se nastaví true, okno se načte, přepočítá a teprve potom zobrazí.

6.7.6 METODY

Metoda znamená, že se provede nějaká akce (na rozdíl od vlastnosti)

window.open() - otevře nové okno. Nejčastěji se používá pro vyskakovací reklamy v nových okénkách, protože umožňuje vypnout zobrazení tlačítek a předepsat přesné umístění a velikost okna.

window.showModalDialog() – umožní otevírat i okna, která čekají na uzavření a teprve potom umožňují vrátit se do původního okna.

window.close() – umožní zavření okna., buďto aktuálního, nebo zavření okna, které bylo dříve otevřeno metodou **window.open** a jeho identifikátor uložen do proměnné.

moveBy() - posouvá oknem o souřadnice.

moveTo() - umístí levý horní roh okna na přesnou souřadnici obrazovky.

resizeBy() - zmenší nebo zvětší velikost okna.

resizeTo() - změní velikost okna na přesně zadané velikosti v pixelech.

scrollBy() - odroluje dokument podle zadaných souřadnic (horizontálně, vertikálně). Nahoru se roluje zápornou druhou hodnotou.

scrollTo() - odroluje dokument na přesnou pozici (horizontální, vertikální). Alternativou je metoda **scroll()**.

V systému Windows je vždy jen jedno okno aktivní (má focus), ostatní jsou v pozadí.

window.blur() přeneso okno do pozadí, deaktivuje okno. Aktivním se stane jiné okno.

window.focus() je opakem blur(). Přeneso okno do popředí (zamodří mu proužek).

Akce se musí provádět z jiného okna (většinou po příkazu window.open) nebo pomocí časování.

Pro jednoduchou formu komunikace pomocí dialogových oken lze použít metody alert(), prompt() a confirm().

window.alert("Text hlášky") - jednoduchý dialog, zobrazí text ze závorčky.

proměnná = **window.prompt("zadej hodnotu", "přednastavený text")** - načte uživatelem zadaný text do proměnné.

confirm() zobrazí dialog s textem a s tlačítkem OK a Storno. Do proměnné se uloží true nebo false (pravda nebo nepravda) podle toho, co uživatel zmáčknul.

print() - vytiskne aktuální okno.

window.setTimeout(), window.clearTimeout(), window.setInterval(),

window.clearInterval() - používají se na časování událostí.

6.7.7 ZÁKLADNÍ DATOVÉ TYPY, PROMĚNNÉ, FUNKCE, OPERÁTORY

Jak bylo zmíněno dříve, JavaScript je plnohodnotný programovací jazyk. Jako takový používá standardní nástroje, proměnné, operátory a funkce. Proměnnou deklaruujeme klíčovým slovem **var** následovaným výpisem použitých proměnných. JS nevyžaduje deklaraci, proměnné se inicializují při prvním přiřazení hodnoty. Není třeba deklarovat typ proměnné (číslo, text apod.), JavaScript určí typ, JS provádí automatickou konverzi při použití. Základní datové typy jsou tyto String (text), Number (číslo s/bez desetinné čárky), Boolean

(pravda/nepravda), objekt a null nebo undefined. S proměnnými lze provádět standardní výpočty. Vzhledem k tomu, že JS je jazyk case-senzitivní, je proměnná x je různá od proměnné X. Všechny proměnné typu text musejí mít hodnoty zadané v uvozovkách nebo v apostrofech. Apostrofy je nutno použít, pokud se celý skript nachází v uvozovkách (in-line zápis).

Např.: ``. Stejně jako u CSS jsou dostupné escape sekvence, to znamená, že znak, který je používán jako klíčový (např. uvozovky, závorky apod.) lze zadávat pomocí znaku zpětné lomítka.

Např: `promenna = "Obsah";`

6.7.8 PROMĚNNÉ

String

String, řetězec, je řada znaků uložená v za sebou jdoucích bajtech paměti. Maximální délka není přesně specifikována, záleží na interpretu. Proměnné typu řetězec mají funkce a vlastnosti, mezi nejčastěji používané patří:

Index - za pomoci indexů lze procházet jednotlivé znaky v řetězci stejně jako v poli. Čísluje se od nuly.

Length - vlastnost vrací délku řetězce ve znacích.

Trim - odstraní specifické znaky (mezery apod.) okolo řetězce, aby nekomplikovali další zpracování.

Replace - funkce nahradí hledanou hodnotu v řetězci jinou.

toUpperCase a toLowerCase - změní všechna písmena v řetězci na velká nebo na malá:

Concat - spojuje dva a více řetězců. Tato funkce je volána automaticky když se pro spojení omylem

Substring a Substr - funkce vrací vybranou část řetězce, které se říká podřetězec. Obě funkce dělají v podstatě to samé, avšak liší se významem parametrů.

Split - rozdělí řetězec na pole řetězců pomocí určitého znaku.

IndexOf, LastIndexOf a Search - vrací pozici daného podřetězce.

Match - vrací shodu s výrazem v řetězci. Používá se hlavně pro regulární výrazy (Regex).

Číslo

Číslo lze zapsat dvěma způsoby, standardně nebo pomocí vědecké (inženýrské, exponenciální) notace, `var x = 10; var x = 10e5; // 10**5 = 100 000.`

S čísly lze provádět většinu známých základních operací jako sčítání, násobení, dělení, existují však i složitější výpočetní operace, např. zbytek po dělení (tzv. modulo). Všechna čísla jsou v JavaScriptu ukládána jako 64-bit double a jsou počítána s přesností na 15 čísel.

Obdobně, jako u textových řetězců, i u čísel existují funkce a vlastnosti čísel.

isNaN(hodnota) - zjistí, zda je objekt v parametru funkce číslo či nikoli. Vrací true nebo false (pravda/nepravda) podle toho, zda je parametrem číslo. NaN Označuje zkratku Not a Number.

toFixed(x) - ořízne číslo na danou přesnost.

toString() - převede číslo na plnohodnotný řetězec znaků.

toExponential(x) - změni zápis čísla na exponenciální (vědeckou) notaci.

Number.MAX_VALUE a ***Number.MIN_VALUE*** - vrací největší / nejmenší možné číslo.

Boolean

Logický datový typ, nabývá pouze dvou hodnot 0-1 (true-false). Jeho hodnota je tedy buď pravdivá, nebo nepravdivá. Používá se např. pro vyhodnocení podmínek a cyklů. Hodnot se zapisují bez uvozovek.

Null / Undefined

Oba dva datové typy zastupují prázdnou proměnnou. NULL je nevytvořený objekt, UNDEFINED naopak objektem není.

6.7.9 FUNKCE

Funkce je v podstatě blok kódu, který jednou napíšeme a potom ho můžeme libovolně volat bez toho, abychom ho psali znovu a opakovali se. Funkci deklarujeme pomocí klíčového slova ***function*** a obsahuje blok kódu ve složených závorkách.

V případě potřeby zpracování segmentu kódu deklarovaného jako funkce, funkci tzv. zavoláme. Toto lze provést až po tom, co funkci deklarujeme, jinak by ji prohlížeč neznal.

Funkce s parametry

Funkce může mít libovolný počet vstupních parametrů, které píšeme do závorky v její definici a podle nich ovlivňujeme její chování.

Syntaxe deklarace funkce v JavaScriptu:

function jmenoFunkce(parametry) {tělo funkce};

nebo podrobněji zapsáno:

function jmenoFunkce(parametr, parametr)

{

příkaz; příkaz; return hodnota

};

Volání funkce:

jmenoFunkce(hodnota, hodnota);

Velmi často se funkce volají na základě událostí dokumentu přímo z HTML kódu, například:

Obsah

Při kliknutí na slovo "Obsah" se vyvolá funkce `upozorneni()` s hodnotou parametru "hlavní stránka". Předtím samozřejmě musí být funkce inicializovaná.

Pokud funkce vrací hodnotu (deklarace obsahuje `return` hodnota), dá se funkce volat zápisem

```
proměnná = jmenoFunkce(parametry);
```

Proměnné ve funkci

Proměnná deklarovaná ve funkci klíčovým slovem **var** je lokální. Lokální proměnné jsou i parametry funkce (to, co je v závorce za jménem funkce). Pokud se ve funkci použije jméno jiné nedeklarované proměnné, jde o proměnnou globální. Výhoda funkcí je tedy v přehlednosti a úspornosti (můžeme napsat nějakou věc jednou a volat ji vícekrát na různých místech skriptu). Když se rozhodneme funkci změnit, provedeme změnu jen na jednom místě a tato změna se projeví všude, což značně snižuje riziko chyb.

Uložení funkce do proměnné

JavaScript se liší od jiných jazyků tím, jak pracuje s funkcemi. Funkce zde umí ještě více, funkci můžeme totiž uložit do běžné proměnné a z této proměnné ji později volat. Všechny funkce v JavaScriptu jsou vnitřně proměnné. Funkci můžeme definovat přímo v přiřazení do proměnné, hovoříme potom o tzv. anonymní funkci

Zabudované funkce JavaScriptu

JavaScript obsahuje předdefinované funkce. Tyto funkce se dají chápat také jako metody objektu `window`, které se dědí na všechny rodičovské elementy.

eval - vychází slova vyhodnotit (evaluate). Funkce vezme svoje argumenty a vyhodnotí je jako by to byl kus programu. Používá se zejména pro dynamickou změnu kódu. Tuto funkci mají v oblibě všichni programátoři JavaScriptu, protože umožňuje zápis konstrukcí běžných z vyšších jazyků, které by JavaScript jinak nevzal.

escape a unescape - funkce `escape()` umí zakódovat řetězce pro přenos. Rozkódování se provede funkcí `unescape()`. Používá se zejména v souvislosti s cookies a předávání parametrů z formulářů. Jde o to, aby při přenosu nenastaly problémy např. s diakritikou a speciálními znaky.

isFinite - funkce vrátí `false`, pokud je číslo v argumentu nekonečné. To se může stát třeba při dělení nulou nebo při počítání logaritmu z nuly:

isNaN - vrací `true` (pravdu), pokud argument není číslo, `false` pokud je číslo.

parseFloat a parseInt - převodní a zaokrouhlovací funkce. Argumentem je řetězec, který je převeden na číslo. **parseInt** z něj vrátí jenom celou část, **parseFloat** i s desetinou částí. Funkce **parseInt** má parametry dva, první je co se parsuje, druhý je základ (2 až 36).

6.7.10 OPERÁTORY

Operátory vycházejí z jazyka Java resp. C. Kromě níže uvedených existují ještě bitové operátory, které se prakticky nepoužívají.

Operátory přiřazení

Slouží k nastavení hodnot proměnných. Standardním operátorem je =, ale používají se i jiné operátory.

= přiřazení; += přičtení, ale také připojení řetězce; *=, -=, /= přinásobení, odečtení, přidělení; ++ přičtení 1; -- odečtení 1.

Obdobný význam mají i méně využívané operátory, <<=, >>=, >>>=, &=, ^= a |=,

Početní operátory

+ sčítání, spojování řetězců; - odčítání, unární negace; * násobení; / dělení.

Logické operátory

== rovnost (dvě rovnítka); != nerovnost; <, <=, >=, > aritmetické srovnání; && logické AND (a zároveň); || logické OR (nebo); ! logické NOT (negace); ? : podmínkový výběr (ternární operátor), logické spojení (třeba v zápisu parametrů funkcí).

Logické operátory se používají zejména při větvení programů na stanovení podmínek. Podmínky nabývají hodnot true (pravda) a false (nepravda). Je potřebné si uvědomit, že rovnost je realizována dvěma rovnítky (jedno rovnítko znamená přiřazení).

6.7.11 VĚTVENÍ

Základem každého programového kódu je využívání opakujících se sekvencí. JS, stejně jako jiné programy, umožňuje větvení programu a definování cyklů. Základní způsoby větvení a cyklů v JavaScriptu jsou příkazy *if*, *while* a *for*.

if(podmínka) {program} - jestliže je splněna podmínka, vykonává se program

while(podmínka) {program} - program se opakuje, dokud trvá podmínka

for(iniciace; podmínka; navýšení){program} cyklus (vykonávaný většinou přesně několikrát)

Jako podmínka se musí uvést výraz, jehož logická hodnota je true nebo false (pravda nebo nepravda).

Je-li v těle podmínky *If* jeden příkaz, nemusí se uzavírat do složených závorek. Také lze vynechat *else*, není-li potřeba. Časté chyby, se kterými se u *if* můžeme setkat, jsou: špatný zápis podmínky (pro rovnost se musejí používat dvě rovnítka), chybějící středníky nebo složené závorky, nadbytečné středníky, mezery většinou nehrají v zápisu roli.

Rozhodovací operátor ?

Tzv. ternární operátor, umožňuje rychlejší zápis rozhodování, pokud chci pouze přiřadit hodnotu proměnné. Syntaxe příkazu je *proměnná = podmínka ? hodnota1 : hodnota2*;

Pokud je podmínka pravdivá, má proměnná hodnotu *hodnota1*, pokud je podmínka nepravdivá, má proměnná hodnotu *hodnota2*.

Přepínač

switch case – slouží pro větvení do více alternativ. Syntaxe příkazu je:

```
switch(proměnná) {  
    case hodnota : příkaz; break;  
    case hodnota2 : příkaz2; break;  
    ...  
    default : příkazx;  
}
```

While – lze použít ve dvou tvarech, jako cyklus s podmínkou na začátku, nebo s podmínkou na konci. Cyklus s podmínkou na začátku provádí sekvenci příkazů (uzavřených ve složených závorkách, následujících za **while**) tak dlouho, dokud platí podmínka. Jakmile podmínka neplatí, vnitřek příkazu **while** se už nevykoná a program bude pokračovat pod koncem sekvence **while**. Podmínka na konci má tvar **do{sekvence příkazů}while (podmínka)**

V souvislosti s cykly je ještě nutné zmínit příkaz **break**, který předčasně ukončí cyklus **while** nebo **for** a příkaz **continue**, který skočí na začátek cyklu.

Uváděný přehled příkazů je redukována množina, sloužící k nástinu vlastností a možností JavaScriptu. Doporučuji pro pochopení odzkoušet příklady vztahující se k textu na stránkách jakpsatweb.cz. Z těchto stránek byly většinou převzaty i texty vztahující se k Css a JavaScriptům.

SHRNUTÍ KAPITOLY



V této kapitole jste se seznámili s principy hypertextu a jeho použití pro publikování WWW stránek. Dozvěděli jste se jak stránky formátovat a jak vložit dynamiku do stránek.

7 WEBDESIGN, WEBMARKETING, OPTIMALIZACE



RYCHLÝ NÁHLED KAPITOLY

Následující kapitola stručně nastíní principy tvorby webových stránek. Tvůrci www si většinou neuvědomují, že zveřejněním www stránek se stávají autory zodpovědnými za formu i obsah www. Současně se v této kapitole seznámíte s pravidly a zásadami doporučenými pro vytváření www stránek.



CÍLE KAPITOLY

Cílem kapitoly je naučit studenty vytvářet stránky designově i obsahově kvalitní, dobře zobrazitelné a podle požadavků uživatelů.



ČAS POTŘEBNÝ KE STUDIU

Tato část je převážně doporučená, nikoli závazná, student se s obsahem seznámí během 1 hodiny.



KLÍČOVÁ SLOVA KAPITOLY

webdesign

Vytvořením webové aplikace nebo jen webových stránek proces často nekončí. Je důležité mít webovou aplikaci správně navrženou, s využitím zpětné vazby od uživatelů. Dále je samozřejmě velmi důležité v dnešním velkém světě na nově vytvořenou webovou aplikaci uživatele nebo potenciální zákazníky vhodně upozornit. S tímto pak také souvisí problematika optimalizace webu jak pro vyhledávače, tak pro různé typy zařízení.

V této kapitole se tedy dozvíte více o principech používaných při návrhu profesionálních webových aplikací, v další části pak budou rozebrány možnosti a nástroje marketingu na internetu, a v neposlední řadě budou zmíněny základní aspekty optimalizace webu pro vyhledávače.

7.1 Trendy současného webdesignu

Webdesign je v podstatě proces tvorby webových stránek od vstupní analýzy, přes návrh až po realizaci projektu. Můžeme jej rozdělit na dvě základní oblasti [1]:

7. **Programování back-endu**, které zahrnuje proces programování podpůrných funkcí webu, funkčnost webu jako takového a veškeré technické aspekty
8. **Návrh front-endu**, který se týká především návrhu vzhledu, jehož pomocí se identifikuje společnost nebo jednotlivec, grafický vzhled a celkový finální dojem z webu samotného.

Při návrhu webových stránek je třeba mít na paměti několik oblastí, jako jeho funkcionality, konzistentnost zobrazení v různých prohlížečích, dostupnost vysokorychlostního připojení k internetu, kompatibilita s vyhledávací nebo estetika stránek. Všechny tyto oblasti lze řešit různými způsoby. Všechny jsou však odvozeny od dvou základních přístupů. První z nich je nakloněn používání nových technologií, využívání multimediálních funkcí apod., zatímco druhý přístup preferuje zcela funkční web, bez jakýchkoliv novinek. Oběma přístupům budou věnovány následující řádky této kapitoly. Webdesign jako takový je v praxi především o hledání kompromisu mezi těmito dvěma základními přístupy.

7.1.1 VÝVOJ WEBDESIGNU

Chceme-li se podívat na počátky webdesignu, musíme začít od vzniku webu jako takového. V 90. letech minulého století se na vzniku webu podílel Tim Berners-Lee, vědec ze švýcarského výzkumného střediska CERN (Berners-Lee, 2000). Ve svých počátcích se web skládal z textových stránek propojených pomocí hypertextových odkazů. Web byl tvořen především jazykem HTML, důraz byl kladen především na důležitost informací, pojem webdesign nebyl v té době ještě na světě. To vyplývalo v té době také z technických možností, které byly k dispozici. Rychlost připojení k internetu byla minimální, počítače nebyly schopny zpracovávat objemnější data, a především počítačové sítě nebyly schopny přenášet grafické informace a multimedia. S postupem času se však z textových prohlížečů vyvinuly prohlížeče grafické (první grafický prohlížeč vzniknul v roce 1993). Kolem roku 1995 byl používán tzv. Statický web, zaměřený především na obsah. Funkčnost webu byla pevně zakódována, stránky byly tvořeny jazykem HTML, pouze jako zpestření byla používána grafika. S postupem času a s vývojem technologií však bylo možno vytvářet také složitější, a především graficky bohatší, webové prezentace. Začal se rozšiřovat tzv. Dynamický web, pro nějž je charakteristická integrace webu samotného s databází. Ve velké míře je již využívána grafika, formátování pomocí stylů a další technologie. V současné době se projeví trend vytvářet web jako takový v podobě aplikace, velmi populární je vytváření aplikačních uživatelských rozhraní v prostředí webu.

7.2 Základní pravidla webdesignu

Vezmeme-li v potaz vše, co bylo zmíněno výše, můžeme nabyté vědomosti shrnout do několika základních pravidel webdesignu (Eccher, 2010):

Použití nových technologií

Použití nových technologií by mělo být vždy zvažováno s rozmyslem. Důležité pro takové rozhodnutí je, zda vůbec nové technologie opravdu přinesou přidanou hodnotu danému webu. Použití nových technologií totiž může snížit výkon webu, může také odradit uživatele (ne vždy jsou nové technologie ve svých počátcích podporovány všemi zařízeními pro prohlížení webu, o prohlížečích nemluvě).

Každý web je jedinečný

Co platí pro jeden web, nemusí platit pro druhý. Toto konstatování se týká nejen použitých technologií. Vždy by se rozhodování o webdesignu mělo opírat o analýzu uživatelské skupiny, která bude k webu přistupovat.

Největší váhu má názor uživatele

Vždy budou existovat různé názory na vzhled webu. Jinak se bude dívat na design webdesigner samotný, jiný názor bude mít jeho nadřízený a zcela jiný názor může mít zadavatel. V neposlední řadě je zde spousta uživatelů, kteří k webu přistupují, a kteří budou mít také každý svůj názor na web. Ne vždy je jednoduché vyhovět všem, často se jedná o kompromis mezi názory jednotlivých skupin. Pokud to však jde, měl by mít větší váhu názor uživatele, protože pro ně je většinou web primárně vytvářen.

Webdesignér by měl mít znalosti z více oborů

Webdesigner by měl mít minimálně znalosti z oblasti grafiky či výtvarného oboru, měl by však také znát technické aspekty webdesignu tak, aby byl schopen posoudit, zda jsou jeho návrhy vůbec technicky realizovatelné.

Učit se od ostatních

Internet je prostředí neustále se vyvíjející, což platí samozřejmě také pro technologie používané pro tvorbu webu. Internet je zároveň tím nejlepším zdrojem učení. Vždy tedy budou existovat atraktivnější weby nebo weby s novějšími technologiemi, což se dá samozřejmě využít k inspiraci.

Není možné se zalíbit všem

Tento bod úzce souvisí s bodem 3. Každý uživatel může mít jiný názor na design webu. Chceme-li dosáhnout spokojenosti většiny, je velmi důležité provést analýzu uživatelů (analýzu cílové skupiny). Na základě zjištěných výsledků pak můžeme efektivněji přizpůsobit funkční a estetické aspekty návrhu webu.

Dodržování specifikací a standardů

Chceme-li, aby byl web přístupný pokud možno co největší skupině uživatelů, a aby byl uživatelsky pokud možno co nejpřívětivější (nejen pro uživatele, ale také pro zařízení, která jej zobrazují), je vhodné se zajímat o specifikace a standardy vyvíjené v této oblasti. Je dobré mít přehled o nových technologiích, případně se zajímat o současné dění.

Další metody

Mezi další metody můžeme zařadit například blogování, přispívání do diskusí se vkládáním odkazu na web, internetová videa, podcasty, inzeráty, tvorbu internetových aplikací apod.

7.2.1 NÁSTROJE PRO OPTIMALIZACI

Optimalizace pro vyhledávače vyžaduje zapojení člověka, a to především do tvorby analýz a úprav webů. Existuje však několik užitečných nástrojů, které dokáží pomoci. Např. Seo Administrator³⁹ je aplikace obsahující soubor nástrojů vhodných k optimalizaci. Nabízí zjištění ranků, popularity zpětných odkazů, kontrolu indexování, návrh klíčových slov a mnohé další funkce. Jedná se o proprietární software. Pro práci v týmu je vhodný např. Collabim⁴⁰. Usnadňuje spolupráci a kontrolu výměny zpětných odkazů, vyhledává a srovnává zpětné odkazy dle tematické podobnosti a navrhuje nová klíčová slova. Dále existuje mnoho online i offline nástrojů pro analýzu kódu a on-page faktorů (<http://seo.webber.cz>, Adobe Web Premium⁴¹). Nástroje zjistí, zda jsou na stránkách titulky, meta tagy, mapa stránek, popisky u obrázků a odkazů, zda jsou stránky validní, apod. Také pro analýzu klíčových slov existuje mnoho nástrojů, např. na Seznam.cz, či na specializovaných webech⁴². Jako poslední zde zmíníme ještě nástroje, které umí zjistit, co vadí robotům a vypíše tyto nedostatky⁴³ či nástroje, které vytvoří po zadání URL adresy do formuláře mapu webu⁴⁴.

SHRNUTÍ KAPITOLY



.Tato kapitola stručně seznámila studenty s pravidly designu tvorby www stránek.

³⁹ Seo Administrator <http://www.seoadministrator.com/>

⁴⁰ Collabim nástroj <http://www.collabim.cz/>

⁴¹ http://www.adobe.com/mena_en/products/creativesuite/design-web-premium.html

⁴² Zjištění hustoty klíčových slov <http://www.webjectives.com/keyword.htm>

⁴³ na <http://www.feedthebot.com/tools/spider/index.php>

⁴⁴ Sitemap generátor http://www.seoutil.com/it/tools/google/sitemap_generator.aspx

LITERATURA

BEDNÁŘ, V. Internetová publicistika. 1. vyd. Praha: Grada Publishing a.s., 2011, 216 s. ISBN 978-80-247-3452-1

BERNERS-LEE, T. Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. New York: HarperCollins Publishers Inc., 2000. 241 s. ISBN 0-06-251587-X.

Dostálek, L., Velký průvodce protokoly TCP/IP, 2. aktualizované vydání, Computer Press, a.s., 2010, 592 s.,

ECCHER, C. Profesionální webdesign: Techniky a vzorová řešení v CSS a XHTML. Brno: Computer Press, a.s., 2010. 672 s. ISBN 978-80-251-2677-6.

KABELOVÁ, A., DOSTÁLEK, L. Velký průvodce protokoly TCP/IP a systémem DNS 5. aktualizované vydání, Computer Press, Brno, 2012

KOSEK, J. XML pro každého, Grada, 2000

KRUG, Steve. Webdesig : Nenutíte uživatele přemýšlet!. 2. aktualiz.vyd. Brno: Computer Press, a.s., 2006. 168 s. ISBN 80-251-1291-8.

MILLER, Michael. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. Indianapolis: QUE, 2008, str. 8. 292 s. ISBN 978-0-7897-3803-5.

NIELSEN, Jakob, TAHIR, Marie. Použitelnost domovských stránek. 1. vyd. Brno : Zoner Press, 2005. 323 s. ISBN 80-86815-18-8.

ŘÍHA P. Tvorba webových aplikací s využitím WYSIWYG editorů, dostupné na <http://www.petrpexa.cz/diplomky/riha.pdf>, online, 30.1.2013

STUHLÍK, Petr; DVOŘÁČEK, Martin. Marketing na Internetu. Praha : Grada Publishing, 2000. 248 s. ISBN 80-7169-957-8

VELTE, Anthony T. Cloud Computing: Praktický průvodce. Brno: Computer press, a.s., 2011, str. 28. 344 s. ISBN 978-80-251-3333-0

Další zdroje, které byly použité nebo citované části v textu nebo jsou vhodné jako další zdroje informací:

Internet pro všechny, Internet, připojení k němu a možný rozvoj, 2006, <http://www.internetprovsechny.cz/internet-pripojzeni-k-nemu-a-mozny-rozvoj-cast-5-internet-v-cr/>, online, 30.1.2013

Stránky firmy NIX, <http://www.nix.cz/cs/services#peering>, online, 30.1.2014

TCP/IP, <http://www.gvp.cz/local/new/ucebnice/vyptech/internet/tcpip.htm>, online, 30.1.2014

Počítačové sítě, <http://site.the.cz/?id=2>, online, 30.1.2014

Peterka, J., eArchiv.cz, Aplikační protokoly TCP/IP, <http://www.earchiv.cz/l223/>, online, 30.1.2014

Přenos souborů v Internetu, <http://4iz110.vse.cz:41004/4iz110/cv4/cv4.html>, online, 30.1.2014

Studentům je dále doporučeno nastudování problematiky v alternativních odborných textech, zejména v:

LUBBERS, P. HTML5: programujeme moderní webové aplikace. Brno: Computer Press, 2011. ISBN 978-80-251-3539-6.

ECCHER, C. Profesionální webdesign: Techniky a vzorová řešení v CSS a XHTML. Brno: Computer Press, a.s., 2010. ISBN 978-80-251-2677-6.

DOSTÁLEK, L. Velký průvodce protokoly TCP/IP. Praha: Computer Press, 2001. ISBN 80-7226-513-X.

GOLDSTEIN, A. HTML5 a CSS3 pro webové designéry. Brno: Zoner Press, 2011. ISBN 978-80-7413-166-0.

HOGAN, B. HTML5 a CSS3: výukový kurz webového vývojáře. Brno: Computer Press, 2011. ISBN 978-80-251-3576.

BRÁZDA, J., KABELOVÁ, A. Internet, tvorba a propagování WWW. Praha: Grada, 2003. ISBN 80-247-0699-7.

RAHMEL, D. Joomla: podrobný průvodce tvorbou a správou webů. Brno: Computer Press, 2010. ISBN 978-80-251-2714-8.

NARAMORE, E. a kol. PHP5, MySQL, Apache - Vytváříme webové aplikace. Brno: Computer Press, 2009. ISBN 978-80-251-2767-4.

.

SHRNUTÍ STUDIJNÍ OPORY























Studijní opora slouží jako podpůrný text pro předmět *Informační a internetové technologie*.

. Poskytuje seznámení s hardwarem a softwarem počítačů, s vývojovými trendy a počítačovými architekturami, dále seznamuje se základními službami internetu a zájemcům o webové technologie a webové aplikace poskytuje základní přehled principů používaných na internetu, včetně praktických návodů.

Struktura textu je zvolena tak, aby se studenti seznámili s principy komunikace na internetu tak, aby mohli v maximální míře využívat služeb, které Internet poskytuje. V jednotlivých kapitolách jsou popsány základy fungování vybraných aplikačních protokolů včetně vybraných aplikací. Dále byla pozornost věnována internetovým systémům a zásadám webhostingu a hostingů internetových portálů. V neposlední řadě pak byla rozebrána problematika webdesignu, webmarketingu a optimalizace webových aplikací. Text uzavírají další vybrané služby, jako např. trendy webdesignu.

Studijní opora je doplněna celou řadou demonstrativních ukázek a obrázků, které mají studentům usnadnit orientaci v problematice. Vzhledem k šíři probírané látky doporučuji praktické odzkoušení a prostudování příkladů uvedených na citovaných www zdrojích vztahujících se k probírané látce

PŘEHLED DOSTUPNÝCH IKON

	Čas potřebný ke studiu		Cíle kapitoly
	Klíčová slova		Nezapomeňte na odpočinek
	Průvodce studiem		Průvodce textem
	Rychlý náhled		Shrnutí
	Tutoriály		Definice
	K zapamatování		Případová studie
	Řešená úloha		Věta
	Kontrolní otázka		Korespondenční úkol
	Odpovědi		Otázky
	Samostatný úkol		Další zdroje
	Pro zájemce		Úkol k zamyšlení

Pozn. Tuto část dokumentu nedoporučujeme upravovat, aby byla zachována správná funkčnost vložených maker. Tento poslední oddíl může být zamknut v MS Word 2010 prostřednictvím menu Revize/Omezit úpravy.

Takto je rovněž omezena možnost měnit například styly v dokumentu. Pro jejich úpravu nebo přidávání či odebrání je opět nutné omezení úprav zrušit. Zámek není chráněn heslem.

Název: **Informační a internetové technologie**

Autor: **Ing. Josef Botlík**

Vydavatel: Slezská univerzita v Opavě
Obchodně podnikatelská fakulta v Karviné

Určeno: studentům SU OPF Karviná

Počet stran: 17675

Tato publikace neprošla jazykovou úpravou.