TEORIE INFORMACE

- objekty okolního světa jsou charakterizovány hmotou a tvarem. Existují dva druhy integrace mezi objekty:

  - **HMOTNÉ –** mění se hmota
  - **NEHMOTNÉ –** mění se forma

ZÁKLADNÍ SCHÉMA

Informační zdroj ⇨ zpráva ⇨ přenosový kanál (signál/šum) ⇨ příjemce zprávy

- **ZPRÁVA** je uspořádaný soubor znaků, který je sestavován informačním zdrojem.
- Rozlišitelné prvky se nazývají symboly a jsou relativně nedělitelné.
- **SYMBOL** je elementární zpráva. Symboly, kterým je přiřazeno grafické znázornění se nazývají znaky.
- Množina všech využitelných symbolů, ze které vybírá informační zdroj při sestavování zprávy, se označuje jako **ABECEDA**. Jestliže je tato množina konečná, hovoříme o diskrétním zdroji zpráv.
- Pravidla výběru znaků jsou dány syntaxe. Abeceda zdroje spolu se syntaxí tvoří kód informačního zdroje**.**
- **KÓDOVÁNÍ** jsou transformační pravidla, podle kterých jsou vysílané znaky přiřazovány vnitřnímu stavu informačního zdroje. Kódování je tedy převedení zprávy do formy signálu. Už samotná znalost kódu představuje určitou zprávu. Zná-li příjemce kód informačního zdroje, může usuzovat o jeho vnitřním stavu.

- Zpráva je souhrnem určitého množství informace.
- Znalost, kterou příjemce před přijetím zprávy neměl (a vedla ke změně jeho vnitřního stavu) je definována jako **INFORMACE**. Množství informace ve zprávě je relativní a vždy závisí na příjemci a konkrétní situaci.
- **SIGNÁL** je hmotný nositel zprávy.
- **PŘENOSOVÁ CESTA** je fyzikální prostředí, ve kterém se pohybuje signál mezi informačním zdrojem a příjemcem.
- **PŘENOSOVÝ KANÁL** je soubor technických prostředků nutných k zabezpečení přenosu signálu od zdroje k příjemci.

- **DATY** se rozumí zprávy určené pro strojní zpracování případně výsledek tohoto zpracování.
- **ÚDAJ** je zpráva získaná jako produkt jistého postupu (výstup měřícího přístroje, senzoru). Převedením údajů do podoby vhodné pro strojní zpracování dostaneme data.

  - Zpracováním **dat, či údajů** získáme **informace**

SYSTEMATIZACE VÝPOČETNÍ TECHNIKY

- **POČÍTAČ** je zařízení schopné přijmout data, aplikovat na ně předepsaný proces a vydat výsledky. Jinými slovy – počítač je elektrický stroj pro zpracování dat (a získávání informací). Data jsou v počítači zobrazena pomocí určité fyzikální veličiny. Nemusí to být nutně elektrické napětí. Tato veličina může mít charakter diskrétní nebo spojitý. Na základě vnitřní reprezentace dat dělíme počítače na číslicové, analogové a hybridní.

- Existuje systematizace počítačů podle použití součástkové základny: generace počítačů:

  - **relé**
  - **elektronky:** 1953 – R4, P1k, štítky, páska, zpožďovací linky, mb buben
  - **tranzistory:** 1959 – R5, P10k, mg páska, feritová jádra
  - **integrované obvody:** 1964 – R6, P1m, mg páska, ferity
  - **IV. generace** – definice sporná
  - **V. generace** – projekt Japonska na začátku 90. let (JIPDEC)

Základní technologie výroby integrovaných obvodů:

- **bipolární** – nejrozšířenější, jediné napájení, logický zisk, rychlost, cena, TTL
- **unipolární** – proudový spínač, ECL, CMOS

HISTORIE VÝPOČETNÍ TECHNIKY

STROJE

1. MARK 1

- spíše než počítač to byl počítací stroj
- postaven v roce 1944 na Hardwarské univerzitě **(Howard Hathaway Aiken 9.3.1900 – 14. 3. 1973)**

- pracoval s čísly o délce 23 míst, prováděl 4 základní operace a měl speciální program na výpočet logaritmu a trigonometrických funkcí
- vstupem byla děrná páska, výstup na děrné štítky
- základem bylo 72 dekadických rotačních čítačů a mechanické relé
- čas pro vynásobení 2 čísel byl 3 – 5 sekund
- původní název ASCC – automatic sequence controller calculator

2. ENIAC

- Electrical numeric integrator and calculator, Pensylvánská univerzita
- používal dekadické desetimístné slovo
- 19000 elektronek, 1500 relé, váha 30 tun, délka 30 metrů, hloubka 1 metr, výška 3 metry
- vstup a výstup děrné štítky
- poprvé použita rychlá paměť: sada registrů s dobou přístupu cca 0,2 msec
- používal hodinové synchronizační impulsy a počítal pulsy
- výpočetní výkon 5000 operací za vteřinu (HP45), spotřeba 175 kw
- konstrukčně 30 nezávislých bloků, 12 akumulátorů (1 akumulátor = sčítačka + střadač)
- Master programmer, který realizoval vložené cykly
- některé výpočty byly prováděný paralelně
- programování se provádělo změnou propojení jednotlivých modulů (6 žen)
- doba práce mezi poruchami byla 5,6 hodiny
- **VYLEPŠENÍ:** motorgenerátor – jednotka pro posuny a 100 slov mag. paměti
  výpočet balistických tabulek:
  - trajektorie 60 sec
  - člověk - 20 hodin
  - analogový Bushův diferenciální
  - analyzátor - 15 minut
  - ENIAC – 30 vteřin

- odpojen ve 23:45 2. října 1955, pracoval celkem 80223 hodiny
- **řešené problémy:** balistika, předpovědi počasí, atomová bomba, kosmické záření, náhodná čísla
- původní rozpočet $61 700, konečný 486 804,22$

3. EDVAC, ODVAC

- změnou vnitřní reprezentace čísel na BCD se změnil počet elektronek z 19 000 na 5000.
- výkon 71 000 součtů nebo 1000 násobení za vteřinu

4. M1

- byl jednoúčelovým stroj pro řešení trojrozměrné Fourierovy transformace
- byl používán při výzkumu krystalových struktur
- pracoval v binárním kódu asi 40 operací za vteřinu
- spuštěn do provozu v roce 1952 v ústředním ústavu matematické v oddělení matematických strojů

5. SAPO

- počítač s binární aritmetikou v plovoucí řádové čárce, slovo o délce 32 bitů
- magnetická bubnová paměť o kapacitě 1024 slov, výkon 5 operací za sekundu

LIDÉ

John Louis von Neumann (1903 – 1957)

- vystudoval chemii v Berlíně a techniku v Curychu
- diplom chem. inženýra v roce 1926
- v červnu 1945 publikoval „First draft of a report to the EDVAC computer“
- fundamentální práce definující principy činnosti počítače:
    - počítač je tvořen řadičem, ALU, pamětí a obvody vstupu výstupu
    - struktura počítače je neměnná a jeho práce je řízena programem uloženým v paměti
    - paměť obsahuje jak data, tak instrukce
    - paměť je tvořena paměťovými místy, do kterých se dá psát nebo z nich číst. Adresa je dána pořadovým číslem tohoto místa
    - program je tvořen posloupností instrukcí, které určují elementární změnu stavu počítače
    - pořadí prováděných instrukcí je sekvenční, výjimku tvoří skokové instrukce
    - instrukce, adresy i data jsou kódovány binárně

Alan Mathison Turing (1912 - 1954)

- britský matematik
- v roce 1936 publikoval článek „On computable numbers“, ve kterém navrhl stroj, dnes známý jako Turingův stroj (konečný automat)
- testující osoba komunikuje přes počítačový terminál s člověkem a zároveň se systémem umělé inteligence, ale neví, který komunikující je na kterém kanále. Pokud není svými dotazy schopen identifikovat umělý systém, je tento systém inteligentní.
- podílel se (znalostmi) na stavbě COLOSSUS
- Turingův test (1950)
- věřil, že každou činnost lze algoritmizovat

Grace Murray Hooper (1906 – 1992)

- prvá programátorka stroje Navy MARK 1, později UNIVAC
- autorka termínu BUG:MARK 2, 9.9.15:45
- konzultant firmy DEC

Antonín Svoboda

- ČVUT, Př.FUK, 1936 doktor technických věd
- zaměřovač protiletecký – analogový stroj
- 1940 emigroval do USA, prezident ABAT, řád USA
- po válce Strojovka Brno – kalkulační děrovač
- vedoucí laboratoře matematických zdrojů
- 1958 založen VÚMS – vedoucí výzkumu
- projekty počítačů SAPO, EPOS
- 1964 opět emigroval do USA, pracoval u GE
- známé Svobodovi mapy
- přednášel na Kalifornské univerzitě, později profesor
- za zásluhy o rozvoj počítačů, čestný člen IEEE

PRŮMYSL

- historie výpočetní techniky není historií lidí nebo firem, je to historie strojů
- ***jediná výjimka je firma IBM (International business machine),*** založena Hermanem Holleritem (1896 Tatralating machine, 1924 IBM). Rok 1964 byl rokem uvedení do prodeje systému 360. Použití monolitických a hybridních IO – nová architektura a organizace se s malými změnami používala více než 20 let. Podstatou byla stavebnicová struktura, škálovatelnost, jednotný instrukční soubor a struktura dat a unifikovaný způsob připojování periferií. Adresování paměti probíhalo po bajtech, slovo bylo v násobcích bajtů. Instrukční soubor obsahoval 170 instrukcí včetně instrukcí s plovoucí řádovou čárkou. Řídící paměť byla rozdělena na permanentní a semipermanentní, což umožňovalo emulaci jiných počítačů. Adresovací systém pracoval s bázovými registry – snadná relokace programů v paměti. Ochrana paměti byla zajištěna v blocích 2k, chybné čtení vedlo k vyvolání specifického přerušení. Stav stroje byl indikován speciálním stavovým slovem, které obsahovalo údaje o poslední prováděné instrukci. Periferie byly připojené pomocí dvou, autonomně pracujících kanálů – multiplexním (až 256 zařízení) a selektorovém

    - STROJE: 360/20,30,40,50,60,75,80,95, další řady:370,3090,39

OPERAČNÍ SYSTÉM 360

- umožňoval poprvé v historii běh několika úloh současně
- nejprve pevný, později proměnný počet úloh
- TSO, vzdálené vstupy z terminálů
- programové vybavení pro řízení práce stroje JCL a pro vývoj aplikací: asembler, ALGOL, FORTRAN, COBOL RPG a PC/1

další operační systémy: PCP, MFT, (TSS), VM/370

- také firma DEC (1957) začala v té době vyrábět unifikovanou řadu malých počítačů pod názvem PDP A PDP 11 (1970 programmed dataprocesor, 16 bitů). Konkurence producentům sálových počítačů především cenou (PDP-1 $ 120 000, IBM $ 1 000 000). Byly určeny především k automatizaci a řízení procesů, jak v průmyslu, tak i ve vědě a výzkumu. Úspěšným pokračováním řady PDP 11 byla série VAX (32 bitů). Výroba těchto počítačů byla ukončena zhruba v roce 1985, firma DEC dávno zanikla, ale části počítačů této architektury vyrábí firma MENTEC dodnes!!!!

    - STROJE: PDP – 11/5,10,20....94

PROGRAMOVACÍ JAZYKY

- Pod pojmem programovací jazyk rozumíme prostředek pro zápis programů, jež mohou být provedeny na počítači. V tomto smyslu je programovací jazyk komunikačním nástrojem mezi uživatelem počítače, který jeho jazykovými prostředky specifikuje algoritmus řešení daného problému, a počítačem, jenž svými technickými prostředky algoritmus interpretuje a realizuje tak transformaci vstupních údajů na výstupní.

Assembler

- Příznačným rysem jazyka symbolických instrukcí je jeho strojová závislost. Syntaxe i sémantika instrukcí odráží takové rysy konkrétního počítače, jako je soubor operací a způsoby adresování operandů. Výhody jsou rychlost překladu a zpracování přeloženého programu a možnost programování na elementárních úrovních, která je v některých případech nepostradatelná a ve vyšším programovacím jazyce nedostupná. Často se používají při vytváření programů operačního systému.

Fortran

- Od fimy IBM
- Není standardizován
- Určen pro vědeckotechnické výpočty
- Je to vyšší programovací jazyk
- Tím že ho všichni používali a že nebyl standardizován, začalo vznikat spoustu verzí, to vede k úpadku (nikdo neuměl pracovat se všemi verzemi..)
- Ve své první podobě byl Fortran opravdu snadno naučitelným jazykem vedoucím k maximálně efektivnímu programu. Programování tak přestalo být výsostnou doménou skupiny úzce specializovaných odborníků. Fortran byl však původně jazykem určeným pro vědeckotechnické výpočty, proto nemohl vyřešit úplně všechno.

Algol

- Univerzální programovací jazyk
- Nemá firemní podporu + vznikl v evropě -> nerozšířil se
- První programovací jazyk, který poskytoval ucelenou a jednotnou množinu jazykových prostředků pro popis algoritmů. Byl to první programovací jazyk, který obsahoval explicitní řídící příkazy pro reprezentaci posloupnosti příkazů, iteraci a výběr alternativ. To umožňovalo zápis programů v takovém tvaru, že bylo možno zřetelně a jasně sledovat pořadí zpracování jeho příkazů. Bylo možno začít vytvářet programy metodou shora dolů, postupným zjemňováním zpočátku abstraktních akcí. To vedlo ke vzniku strukturovaného programování.

Cobol

- Byl vyvinut společným úsilím výrobců a uživatelů počítačů ve spolupráci s ministerstvem obrany USA.
- Jazyk by měl umožňovat
    a) *sestavení programů v minimálním čase s minimálním programovacím úsilím*
    b) *zápis programů v jazyce blízkém angličtině*
    c) *snadný převod programů na nové typy počítačů*
    d) *úplnou dokumentaci programů*
- přenositelný z PC na PC
- je standartizován

PL1

- určen pro počítače firmy IBM. Jazyk PL/I je aplikovatelný na různé okruhy problémů a v praxi se ho hodně používalo; postrádá však systematickou strukturu s jednotnou koncepcí.
- Univerzální programovací jazyk
- Neujal se
- Velké množství příkazů => složitý (komplikovaný)

PASCAL

- Byl navržen v roce 1971 pro potřeby výuky programování. Dnes má Pascal ve světě nejen dominantní postavení při výuce programování, ale velmi se používá i pro běžné programování
- Borland – Turbo pascal
- Nahradil Algol
- Vysoká rychlost kompilace

Delphi

- Je to vývojové prostředí
- Vychází z Turbo pascalu
- Použití komponent – program se sestavuje na obrazovce, softwarové stavební díly

C

- Byl vyvíjen společně s OS UNIX, v prapůvodní podobě se nazýval B-jazyk, C-jazyk je unikátní, neboť je jakési střední úrovně. Není to jazyk vysoké úrovně jako BASIC, PL/I nebo PASCAL, není to však ani jazyk nízké úrovně jako assembler. Jazyky střední úrovně jsou jazyky, které nahrazují assembler a přitom umí úkony jazyků vysoké úrovně.
- Systémový programovací jazyk
- Jednoduchý zápis
- Univerzální
- Jazyk střední úrovně
- Unix byl přepsán do C
- Spustitelný na jakékoli platformě

C++

- Odvozený od C rozšířením o oběktivně orientované programování
- Nejrozšířenější na platformě windows (C# - microsoft)

Basic

- Univerzhální jazyk
- Vyrobený Microsoftem
- Jednoduché programování
- Je to programovací jazyk DOSu
- Velké množství verzí
- Použit pro analýzu systému

Java

- Univerzální jazyk
- Sun microsystem
- Ovládání domácích spotřebičů
- Interpretovaný (vyrobený na pomalé procesory, omezenou velikost paměti)
- Vychází z C++
- Používá se na oživení webových stránek

Java skript

- Je to interpret
- Vyrobený firmnou Netscape
- Je standartizován

ADA

- Byl vyvinut na podnět amerického Ministerstva národní obrany. Důvodem byl prudký růst nákladů na programové vybavení vestavných počítačů. Nešlo přitom jen o náklady na vývoj nových systémů, ale i oběžné a

stálé náklady na jejich udržování. Ukázalo se, že hlavní příčinou těchto neúměrně vysokých nákladů je nedostatek standardizace; zjistilo se totiž například, že v oblasti zájmu tohoto ministerstva se používalo k programování systémů a úloh více než 350 různých jazyků. Bylo proto rozhodnuto přikročit k vývoji standardního jazyka.

- Vyšší programovací jazyk
- Vyroben s pomocí ministerstva obrany
- Je univerzálný
- Je náročný na výkon PC
- Řízení procesů v reálném čase
- spolehlivý

ČÍSELNÁ SOUSTAVA

- rozumíme soubor určitých znaků a pravidel sloužících k zobrazení čísla. V zásadě známe:
    A. ***poziční*** (polyadické) – význam znaku závisí na místě
    B. ***nepoziční*** (nepolyadické) – znak je na pozici nezávislý

- nejznámější je polyadická soustava dekadická nebo binární. Hexadecimální nebo oktalyová je jen zkrácený zápis binární soustavy

- pro zobrazení čísel v počítači používáme polyadickou binárně kódovanou soustavu

- číslo může být reprezentováno:
    - **přirozeným tvarem** – pevná řadová čárka – počet řádků je konstantní
    - **semilogaritmickým tvarem** – pohyblivá řadová čárka – počet řádů mantisy a charakteristiky je pevně stanoven (čísla REAL)

- nezávisle na kódu je základní vlastností zobrazení čísel konečný počet zobrazitelných míst (256, 65536, 4.294.367.296)
- volba reprezentace dat je jedna z nejdůležitějších etap návrhu architektury počítače, protože zásadním způsobem ovlivňuje i organizaci stroje.
- pro zobrazení čísel v pevné řadové čárce jsou většinou používány jednoduché váhové binární kódy. Číslo je reprezentováno kombinací bitů, kde je každému bitu přidělena jeho váha.
- řadová čárka bývá nejčastěji úplně vpravo – obecně může být umístěna kdekoli – závisí to na architektuře počítače. Výhodou této reprezentace je jednoduchost konstrukce a shodnost realizace pomocí elektronických obvodů

BINÁRNÍ ČÍSLA SE ZNAMÉNKEM

(existuje několik způsobů reprezentace)
- přidání znaménkového bitu
- zobrazení s posunem – přičtení pozitivní konstanty (AD převodníky)
- dvojkový doplněk – negace + 1

BCD čísla

(binary coded decimal)

- přímý binární kód, kde jsou dekadické číslice ukládány do čtveřic bitů, které reprezentují právě jednu dekadickou číslici

- ***VÝHODY:*** nejsou nutné převody – odstranění nepřesnosti a konverzních chyb

- ***NEVÝHODY:*** asi 20% redundance (může zabírat víc místa než je nutné). Znaménko: +1100, -1101 (jsou volné kombinace)

- ***chybové stavy*** – přetečení a podtečení

- při práci v pevné řadové čárce je počet bitů, ve slovně, pevně stanoven a tím omezuje počet zobrazitelných čísel (aritmetiku čísel s konečnou délkou)

- je potřeba sledovat nejen velikost operandů, ale i pořadí prováděných operací.

- neplatí obecně zákon asociativní a distributivní
    - PŘ.: 700+(400-300) = (700+400) – 300 !!!!!! - na 3 místa

- čísla v plovoucí řádové čárce mají tyto vlastnosti:

- - volný rozsah zobrazitelných čísel
- - čísla v semilogaritmickém tvaru netvoří kontinuum (existuje mezera)
- - stejná přesnost pro všechna čísla
- - nerovnoměrné pokrytí číselné osy
- - nutnost zaokrouhlení nevyjádřitelných čísel

- existuje několik způsobů zobrazení čísel s plovoucí řádovou čárkou:

- ***přímé –*** mantisa i exponent jsou opatřeny znaménky uloženými zcela vlevo, řádové čárky jsou zcela vpravo, nejsrozumitelnější zobrazení, nejpřesnější pro celá čísla

- ***exponent s posuvem***

- ***se skrytým bitem*** – musí být vždy normalizované

- správná volba mantisy a poměru mantisy k exponentu má klíčový význam při návrhu architektury stroje. Pro standardizaci práce s čísly v plovoucí řádové čárce bylo proto vytvořeno doporučení
  - IEEE-754. Není závazné, ale většina výrobců jej dodržuje

PROCESORY INTEL

- - jednoduchá přesnost : 32 bitů 0,1.2x10-38 3.4x10+38
- - dvojitá přesnost : 64 bitů 0,2.3x10-38 1.1x10+308
- - zvětšená přesnost : 80 bitů 0,3.4x10-4932 1.1x10+4932
  - posunutí exponentu 7F, 3FF, 3FFF

  - existují i hodnoty + - nekonečno a NaN (Not a Number) a nenormalizované (tiny) čísla

DATA S AUTOMATICKOU IDENTIFIKACÍ

- ***tagované paměti*** – snadná korekce chyb, konverze, obecné instrukce
- ***deskriptory dat*** – obsahují informace o typu, přesnosti, rozměrech, umístění

ZOBRAZENÍ DAT V POČÍTAČI

- počítač je stroj, který je schopen přijmout informace, aplikovat na ně předepsaný proces. Informace je v počítači uložena v paměti, přemísťována pomocí komunikačních kanálů a přeměněna v procesoru

- stroj musí pracovat s nějakou symbolikou, která bude:
  - srozumitelná pro stroj
  - reprezentovaná pomocí fyzikálních veličin
  - převoditelná do symbolů srozumitelných člověku
  - dostatečně obecná i pro popis složitých nehmotných představ

- data jsou v počítači zapsána pomocí kódu. Z hlediska organizace počítače dělíme kódy na:
  - **komunikační** (zde se sleduje především odolnosti vůči chybě při přenosu)
  - **kódy pro provádění matematických operací** (sledujeme především jednoduchost provádění úkonů, účinnost zobrazení)

- základním objektem, se kterým je v počítači manipulováno je ***datový typ.***
  - místo uložení
  - rozsah možných hodnot (množina povolených hodnot)
  - soubor pravidel pro transformace (množina povolených transakcí)
  - soubor pravidel definující způsoby a oprávnění přístupu

- každý datový typ musí explicitně (přímo vyjádřeno) nebo implicitně (vyplývá) obsahovat výše uvedené vlastnosti

ZÁKLADNÍ DATOVÉ TYPY:

- logická hodnota (0 = false, cokoli = true)
- znak abecedy
- číslo

literál

- je lexikální jednotka, která přímo reprezentuje hodnotu
- je hodnota definována sama sebou (konstanta)

- informace je v počítači reprezentována množinou bitů a kódována
- kód přenese příslušnou informaci na data v binární soustavě vhodnou k zpracování strojem
- k reprezentaci alfanumerických symbolů se používají kódy zadané tabulkou

  - EBCDIC – extended binary coded decima interchange code
  - ASCII – american standart code for information interchange
  - UNICODE – kód, který se snaží zavést konsurcium firem
    - základem je čtyřbitový váhový kód BCD pro desítkové číslice
    - rozšířením na 8 bitů = 256 možností. ASCII prakticky totéž, ale původně jen 7 bitů = 128 možností
    - rozšířením na 8 bitů dalo možnost přidat národní abecedy (Latin 2)

ZÁPORNÁ ČÍSLA:

znaménkový bit – nevýhodné pro stroj
prvý dvojkový doplněk – nevýhodné pro člověka, jsou dvě nuly
druhý dvojkový doplněk – nevýhodné pro člověka, asymetrický rozsah

- Při manipulaci s binárními čísly ve dvojkovém doplňku platí následující pravidla:

1. mají – li čísla různá znaménka, potom k přeplnění nemůže dojít
2. dojde – li k přeplnění, potom výsledek má opačné znaménko jako operandy

- Reálná čísla ve formátu s pohyblivou řádovou tečkou – standard IEEE-754. Tento standard specifikuje následující:
    - - způsob zobrazení reálných čísel v počítači
    - - přesnost výsledků aritmetických operací a operací srovnání
    - - konverze mezi celými a desetinnými čísly
    - - konverze mezi reálnými čísly s různou přesností
    - - definuje podmínky vzniku zvláštních situací a způsob manipulace s nimi

ZVLÁŠTNÍ HODNOTY:

Nula

- není přímo zobrazitelná (skrytý bit je vždy 1). Je definována jako exponent i zlomková část samé nuly. Existuje kladná a záporná nula – obě mají stejnou platnost

Nekonečno

- je reprezentováno jako exponent samé 1 a zlomková část samé nuly. Existuje kladné i záporné nekonečno. Norma definuje výsledky operací s nekonečnem a nulou

Nečíslo

- NaN je speciální hodnota, která vznikne jako výsledek operací matematicky nedefinovaných (0/0,nekonečno - nekonečno,nekonečno x 0, nekonečno/nekonečno)

BINÁRNÍ ARITMETIKA

REÁLNÁ ČÍSLA

ČÍSLO = (-1) ^ sign x 2 ^ (exponent – 127) x (1.mantisa)

4 0 A 0 0 0 0 0

0 |100 0000 1 |010 0000

1 x 4 x 1.25 = 5

4 2 F E

0 |100 0010 1 |111 1110

1 x 64 x 1.98 = 63

4 2 7

0 |100 0010 0 |111

1 x 32 x 1.875 = 60

4 1 2 8

0 |100 0001 0 |010 1000

1 x 8 x (1/4 + 1/16) = 1 x 8 x 1.3125 = 10.5

Binární čísla bez znaménka

bajt	8 bitů	byte
slovo	16 bitů	word
dvojslovo	32 bitů	doubleword (486)
čtyřslovo	64 bitů	quadword
dvojité čtyřslovo	128 bitů	double quadword (PIII)

Celá čísla kladná nebo záporná

položka	bez znaku	-	+
bajt	255	128	127
slovo	65.536	32.768	32.767
dvojslovo	4.294.967.295	2.147.483.648	2.147.483.647
čtyřslovo	1.844x10^19		

Reálná čísla

položka	bitů	znak	j-bit	mantisa	exponent	řádů	
jednoduchá přesnost 32	1	Ne	24		8	127	
dvojitá přesnost	64	1	Ne	52	11		1.023
rozšířená přesnost 80	1	Ano	63		15	16.383	

- vícebajtové datové položky jsou v paměti zapsány ve formátu méně významný bajt na nižší adrese (little endian). Platí pouze pro architekturu 1A-32 (neplatí paušálně u Itania). Datové položky mají být v paměti počítače vyrovnány na přirozenou hranici typu. Tím se rozumí, že slovo by mělo ležet na adrese dělitelné dvěma, dvojslovo na adrese dělitelné čtyřma a čtyřslovo na adrese dělitelné osmi. Nerespektování této skutečnosti v lepším případě způsobí zmenšení výkonu stroje – procesor bude potřebovat dva přístupy do paměti k přičtení takto uložené datové položky. V horším případě (záleží na instrukci) to může způsobit chybový stav (#GP – general – protection Exception). Tento fakt je důležitý pro programátora v asembleru, ve vyšším programovacím jazyku je to věcí překladače.

- při sčítání a odčítání binárních čísel platí následující

```
0+0=0              0-0=0
0+1=1              0-1=1 a výpůjčka
1+0=1              1-0=1
1+1=0 a přenos     1-1=0

 01101       110110
 10111     - 10111
100100      1010111
```

PAMĚŤ

- paměť je zařízení, které je schopné přijmout informace, uchovat je po danou dobu a na požádání je vydat
- paměť je složena z adresovatelných jednotek – paměťových míst (buněk, lokací)

- ***paměť můžeme charakterizovat:***
    A. funkcí (čtení, zápis)
    B. kapacitou (bity, bajty)
    C. vybavovací doba, cyklus

ROZDĚLENÍ PAMĚTÍ PODLE PŘÍSTUPU K DATŮM:

- **RAM (random access memory)** – paměť s libovolným přístupem. Doba odezvy je konstantní a nezávisí na adrese, která je použitá pro čtení a zápis
- **SAM (serial access memory)** – paměť se sekvenčním přístupem – zpožďovací linky
- **DAM (direct access memory)** – paměť s přímým přístupem (disk)
- **CAM (content addressable memory)** – paměť adresovaná obsahem

ROZDĚLENÍ PAMĚTÍ PODLE IMPLEMENTACE:

- **RWM (read/write memory)** – paměti pro čtení a zápis na libovolnou adresu
- **ROM (read only memory)** – paměti schopné pouze číst
- **SRAM (static random access memory)** – statické paměti RAM
- **DRAM (dynamic random access memory)** – dynamické paměti RAM
- **PROM (programmable read only memory)** – jednou programovatelné

HIERARCHICKÉ USPOŘÁDÁNÍ PAMĚTÍ:

- **REGISTRY** – vysoká rychlost přístupu – realizace přímo jakou součást procesoru
- **CACHE** – vyrovnávací paměť – co nejblíže procesoru (10 nanosekund)
- **HLAVNÍ** – fyzická paměť počítače pro instrukce a data (10 – 100 nanosekund)
- **ODKLÁDACÍ** – ukládání stránek hlavní paměti – disk (milisekundy)
- **SEKUNDÁRNÍ** – vnější paměti s náhodným přístupem (disky, diskety)
- **ARCHIVNÍ** – trvalé ukládání velkých objemů dat, ke kterým se přistupuje zřídka

ORGANIZACE PAMĚŤOVÝCH JEDNOTEK:

- **paměti se širokým slovem** – do datového registru se předává několikanásobek potřebné informace. Nutným předpokladem je následné čtení dat či instrukcí

- **paměti s prokládanými cykly** – rozdělení pamětí na moduly (banky) a střídavý přístup k informacím v různých modulech

KONTROLA A KOREKCE CHYB

- parita příčná a podélná. Samoopravné kódy (Hammingův)

HIERARCHICKÉ USPOŘÁDÁNÍ PAMĚTI

- vychází ze skutečnosti, že přístupy do paměti mají tendenci shlukovat se do skupin

- lokalita referencí:
    A. ***časová*** – reference, která byla právě použita a bude v krátké době použita znovu
    B. ***prostorová*** – reference následující bude pravděpodobně ležet poblíž právě použité

- mechanizmus funkce je následující – paměť je rozdělena na úseky, tvořené posloupnosti adresovatelných jednotek. Podle požadavků procesoru je obsah hierarchicky nižší paměti vyměněn s vybranou oblastí hierarchicky vyšší paměti

- ***mapovací mechanizmy*** – stránkovaná a segmentovaná paměť

NĚKTERÉ DŮLEŽITÉ ORGANIZACE VÝBĚRU Z PAMĚTI

- zásobníková organizace (STACK) je pamětí LIFO. Může být realizovaná jako zvláštní součást operační paměti i jako samostatná paměť na bázi registrů. Je adresovaná přes zvláštní registr – SP (stack pointer). Základními operacemi se zásobníkem je operace uložení a vybrání položky (PUSH, POP)

OCHRANA PAMĚTI

- mechanizmus řízení přístupu ke specifickým částem operační paměti. Narušení ochrany se projeví jako přerušení, které obsluhuje operační systém. Užívá se v jednouživatelském i víceuživatelském režimu práce.

- ***může být realizováno takto:***
  - **- mezními registry** – obsah MAR se porovnává s limitními hodnotami
  - **- maskou** – paměť je rozdělena na stránky. Maska = číslo stránky
  - **- klíče** – prakticky totéž co maska. Srovnání provádí software
  - **- metabity** – pomocné bity přidané k datovému obsahu paměti.

PC – PAMĚTI

- systémová paměť může být tvořena dvěma jakostními typy fyzické paměti:
  - ***DRAM (DYNAMIC RAM)***
  - ***SRAM (STATIC RAM)***

- paměťové buňky čipu DRAM jsou tvořena dvojicí malého kondenzátoru a tranzistoru
- paměťové buňky čipu SRAM jsou tvořeny 6 tranzistory SRAM = 2M = DRAM 64 MB

ZÁKLADNÍ TYPY PAMĚŤOVÝCH MODULŮ DRAM

- **FPM – Fast page mode**
- **EDO – Extended data ant**
- **SDRAM – Synchronous dynamic RAM**
- **RDRAM – Rambus DRAM**
- **DDR SDRAM – Double data rate SDRAM**

FYZICKÉ USPOŘÁDÁNÍ PAMĚTI RAM

- **SIMM – single inline memory module** – 30 a 72 vývodů
- **DIMM – dual inline memory module** – 168 a 1284 vývodů (DDRDRAM)
- **RIMM – Rambus inline memory module** – 184 vývodů

BUŇKY PAMĚTÍ

- nejmenší množství paměti, které může být najednou adresováno
- šířka paměti musí vždy odpovídat šířce procesorové sběrnice

VÝVODY MODULŮ

- povrch kontaktů je pokryt zlatem nebo cínem. Nelze doporučit používání konektorů a vývodů modulů s různým pokovením (fretting corrosion)

SPOLEHLIVOST PAMĚŤOVÉHO SUBSYSTÉMU

- chyby můžeme rozdělit na logické a fyzické

- ***FYZICKÉ*** – projeví se trvalým selháním modulu
- ***LOGICKÉ*** – projeví se nepravidelně a mohou být způsobeny: poruchami napájení, rušením, statickými výboji, zářením nebo nevhodným použitím paměťového modulu (výpadky časování)

- ochranou je použití paritního bitu nebo ECC (Error correcting code), což zvyšuje cenu paměťového modulu

PROPOJOVACÍ SYSTÉMY

- propojovací subsystémy můžeme rozdělit na vnitřní a vnější. Každé propojení je charakterizováno rychlostí, šířkou pásma a pořizovací cenou

- propojovací systém je většinou hierarchický. Význačnou vlastností je způsob šíření informace mezi jednotlivými jednotkami – organizace subsystému

- ***směrová*** – vždy bude aktivována zvláštní cesta (část subsystému e neúčastní)
- ***sběrnicová*** – informace jsou předávány po obecných cestách

    - ***sériový kanál*** – informace je přenášena po elementárních cestách
    - ***paralelní kanál*** – přenášejí se větší jednotky informace

SMĚROVÁ ORGANIZACE

- propojovací síť je složena z uzlů, které komunikují prostřednictvím spojů. Každý spoj je dvoubodové spojení – na jednom spoji se nemůže podílet více uzlů. Způsob propojení jednotlivých jednotek sítě se nazývá TOPOLOGIE propojovacího systému. Vlastnosti navržené topologie určují, jakým způsobem budou jednotky sdílet zdroje a co toto sdílení bude stát. Nejjednodušší topologie je propojení dvou jednotek (bod – bod). Komunikační strategie je algoritmus směrování zpráv na cestě od uzlu k síti

SBĚRNICOVÁ ORGANIZACE

- všechny jednotky jsou propojeny jednou nebo několika společnými sběrnicemi
- komunikace je rozdělena do diskrétních transakcí mezi vysílačem a přijímačem. Transakce začne, když některý účastník získá kontrolu nad sběrnicemi a stane se řídící jednotkou (master). Často existuje více jednotek, které se mohou stát řídící.

ARBITRÁŽNÍ PROTOKOL

- je uplatnění, pokud více jednotek ve stejný okamžik usiluje o získání řízení sběrnice, transakce na sběrnici je řízena komunikačním protokolem

ASYNCHRONNÍ PROTOKOL

- protokol, při kterém přednost elementu informace může začít kdykoli

SYNCHRONNÍ PŘENOS

- může být zahájen jen v přesně stanovený okamžik. Provoz na sběrnici je vždy řízen taktovacím kmitočtem (hodinové impulzy). Výkon sběrnicového systému je určen dvěma parametry:
    1. přenosovým časem
    2. šířkou pásma (bity za sekundu)

ARBITRÁŽ POŽADAVKŮ O SBĚRNICI

- sběrnice je sdílené médium – dochází k situacím, kdy o použití soupeří několik jednotek. Pravidla při rozhodování o přidělování sběrnice jsou v zásadě dvojí:

    - **1, centrální arbitráž** – priorita, pořadí příchodu (LIFO) nebo náhodné
    - **2, decentralizovaná arbitráž** – prioritní linka (daisy chain), cyklické (tokem) a kolizní

- centrální arbitráž zajišťuje zvláštní jednotka, která vyhodnocuje přicházející žádosti

- decentralizovaná arbitráž používá rozhodovací algoritmy vestavěné do každé připojené jednotky. Jednotky mají vyšší inteligenci a potřebují jistý čas na provedení arbitráže a výměnu zprávy o jejím výsledku.

- libovolný propojovací systém bude vyžadovat jistou programovou obsluhu. V případě srovnatelných rychlostí obou komunikujících jednotek bude pro programátora transparentní. Je – li rozdíl významný, je třeba komunikaci řešit tak, aby rychlejší jednotka byla komunikačním procesem zatížena co nejméně.

- ***existují následující způsoby práce:***
  - programová obsluha
  - obsluha s využitím přerušení
  - pomocí blokové přímého přístupu do paměti (DMA)
  - samotné kanálové procesory

- mechanizmus přerušení je v zásadě vždy stejný – přerušení práce procesoru na vnější podnět, vykonání obsluhy přerušení a návrat k původní práci

- obvyklý způsob vyvolání programu obsluhy přerušení je nucený skok na adresu, kde je program uložen, pomocí tabulky vektorů, ukazující na příslušné obsluhy přerušení (skutečnost je komplikovanější)

- přenos se v počítači uskutečňuje buď po jednotlivých datových elementech (bajty) nebo po blocích. Pomalá zařízení (tj. taková, u kterých je odezva na přenos podstatně delší než vlastní přenos), používají většinou přenos po datových elementech (asynchronně) a rychlá pak blokový (synchronní) přenos

BLOKOVÉ PŘENOSY – DIRECT MEMORY ACCESS (DMA)

- někdy je výhodné, aby se procesor řízení komunikačního procesu neúčastnil
  - procesor je řadičem DMA odpojen od sběrnice (a případně i zastaven)
  - řadič DMA zpomalí procesor zastavením hodin (kradení cyklů)

- KANÁLOVÉ PŘENOSY

- snaha o nezávislost komunikace na procesoru vedla ke vzniku komunikačních procesorů. Procesor je rychlým spojem propojen se specializovaným procesorem, kterému pouze předá požadavek na přemístění určitého množství dat

- komunikační procesory jsou speciálně navržená zařízení s vlastní instrukční sadou orientovanou na efektivní přesuny dat. Rychlému spoji říkáme komunikační kanál a komunikačnímu procesoru kanálový

SIGNÁLY, KÓDOVÁNÍ, PŘENOS DAT A PŘENOSOVÉ CESTY

- sítí rozumíme účelové propojení výpočetních systémů. Účelem propojení je sdílení zdrojů (a poskytování služeb). Organizace sítí je většinou směrová (existují i výjimky).

- přenosové prostředky

- komponentami sítě jsou datové stanice, které provádějí přenos informací ve formě dat (datovou komunikaci) po přenosové cestě.

přenosová cesta
- jednosměrná (přenosový kanál)
- obousměrná (kruh)

signál
- je nositelem informace
- analogový
- digitální (mění se v definovaných časových okamžicích a může nabývat pouze určitých hodnot).

okruhy (kanály)
- pevné
- přepínané (komutované)

fyzická realizace přenosové cesty
- drátová komunikace (kovový vodič, skleněné vlákno)
- komunikace bezdrátová (vzduchem)

kabely

- jak je třeba kabely pokládat uvádí specifikace EIA/TIA 568-570 a 606. Správně provedená kabeláž je základním předpokladem dobře fungující sítě. V nových budovách se provádí jako strukturovaná kabeláž. Postatou je jednotné provedení rozvodů a společné umístění spolu souvisejících kabelů, což má umožnit snadnou případnou rekonfiguraci struktury.

symetrický kabel (kroucený pár – Twisted pair)
- je složený z párů zkroucených vodičů.
- **Používají se dva typy:**
    - stíněný STP
    - nestínění UTP

- nestíněný kabel může mít 2,4,22,24, nebo 26 párů vodičů (specifikuje AWG – American Wire Gauge). **Kvalitu nestíněných kabelů vyjadřuje kategorie, do které je kabel zařazen:**
    - 1=žádná výkonností kriteria
    - 2=do 1Mhz (telefonní rozvody)
    - 3=do 16 Mhz
    - 4=do 20Mhz (token ring)
    - 5=do 100 Mhz
    - existují i předběžné definice kategorie 6 (200 Mhz) a 7 (600 Mhz)

koaxiální kabel
- je tvořen dvěma vodiči v provedení, kdy vnější obaluje vnitřní (většinou měděný vodič), po kterém se přenáší aktivní signál (nesymetrický přenos). Vodiče jsou odděleny dielektrikem a zaizolovány obalem kabelu. Kabel

dobře chrání proti elektromagentickému rušení (proti magnetickému méně). Odolnost snižují vyrovnávací proudy – je žádoucí, aby připojená zařízení byla izolována (minimální styk se zemí). Používá se tlustý (4 vrstvy izolace) a tenký koaxiál. Hodnota impedance kabelu 50 ohmů

optické kabely

- dovolují dosáhnout větších přenosových rychlostí, dokonalého galvanického oddělení, jsou odolné proti rušení a nelze je odposlouchávat. Základním nedostatkem je složitější konstrukce (převod elektrického signálu na světlo) a značně dražší propojovací konektory (a také podstatně dražší nářadí, potřebné pro realizaci spoje)

Multiplexor

- zařízení, která umožňují účinnější využití přenosových, kapacit se nazývají multiplexory. Využívají nějakou metodu sdružování nezávislých datových toků na jedno přenosové médium a jejich zpětného oddělení v demultiplexoru na opačné straně. **Pracují na dvou základních principech:**
    - ***kmitočtové dělení*** spočívá v rozdělení kmitočtového pásma na dílčí podpásma oddělením kmitočtovými filtry a jejich přidělení příslušným kanálům.
    - ***časové dělení*** využívá rozdělení na časové rámce a časové úseky, které tvoří části datových kanálů. Každé připojené zařízení komunikuje pouze v a po určený časový interval. Což předpokládá synchronizaci. Komplikovanější variantou jsou statické časové muliplexory (asynchronní). Přidělují časové rámce na základě momentální potřeb. (asynchronní multiplexory s vyrovnávací pamětí patří do skupiny sdružovacích prostředků a nazývají se koncentrátory)

- přenosové prostředky, které zajišťují spojení nebo propojení kanálů či okruhů, tvoří skupinu spojovacích zařízení. Přepojování je realizováno jako přepínání okruhů nebo jako přepínání paketů.
- Přepínání okruhů lze rozdělit na:
    - časové (dělení čas plus směr)
    - prostorové (spojování, ústředny)

- přenosové prostředky – shrnutí

- stavebními prvky sítí jsou:
- koncová zařízení (datové stanice)
- přenosové cesty (hmotní či nehmotné)
- spojovací (propojovací zařízení)

Kódování:

- kódováním rozumíme převedení zprávy do formy podoby signálu. **Většinou kódováním sledujeme i jiná hlediska:**
- přizpůsobení přenosovým vlastnostem kanálu (modulace)
- zvýšení účinnosti přenosu (odstranění stejnosměrné složky)
- zabezpečení přenosu proti chybám (bezpečnostní složky)

- při přenosu zpráv dochází většinou k několikanásobnému kódování. Zpracovávaná data jsou reprezentována v nějaké číselné soustavě. Z praktického hlediska se používá binární soustava. **Podle složitosti zabezpečení můžeme kódy pro přenos zpráv rozdělit, na:**
    - bezpečnostní
    - jednoduché:
    - rovnoměrné
    - nerovnoměrné
- jsou navrženy tak, aby přenos byl co nejefektivnější. Nejpoužívanější znaky mají nejkratší délky. Nevýhodou přenosu datových posloupností různých délek je nutnost modifikace začátku znaku (synchronizační impulsy). Realizace nerovnoměrných kódů je složitější a mají i nižší schopnost proti rušivým signálům. Tyto nedostatky rovnoměrné kódy nemají (snadné rozlišení části zprávy)

Číslicové signály

- převedení binárního kódu na elektrický signál je důležité nejen z pohledu přesností zpráv, ale jsou sledovány (většinou) i jiné cíle (energetická účinnosti, odolnost proti rušení).

- podle polarity dělíme signály, na:

  - ***unipolární*** - je signál o jedné polaritě. Označuje se také jako RZ (return to zero). Může být realizován s mezerami (snadné rozlišení jednotlivých bitů) nebo bez mezer (pro větší rychlosti). Přiřazení 1/0 je dosaženo buď změnou šířky impulsu nebo přítomnosti či nepřítomnosti impulsu (toto se často používá při přenosu dat pomocí optických vláken)

  - ***polární*** - je tvořen impulsy s kladnou nebo zápornou polaritou. Neobsahuje žádné mezery a nemá nulovou úroveň. Označuje se jako NRZ (non return to zero). Při delší sekvenci log 0 nebo 1 dochází k zhoršení identifikace bit. Což odstraňuje kódování NRZI. Výhodou tohoto způsobu kódování je dobré potlačení stejnosměrné složky signálu.

  - ***bipolární (pseudotermální)*** **-** *zpracovávají se tři napěťové úrovně***:** kladná, záporná a nulová. Logická 1 je reprezentována impulsem libovolné polarity, logická 0 pak nulovou úrovní (mezerou). Pro volbu polarity impulsu logické 1 existují různé algoritmy (časové nebo poziční řízení polarity). Tento způsob kódování nejlépe potlačuje stejnosměrnou složku signálu (při pozičním řízení polarity a sudého počtu přenášených bitů je nulová)

ZÁKLADNÍ TECHNOLOGIE VÝROBY INTEGROVANÝCH OBVODŮ

Monolitické obvody

- Základem je monokrystal z velmi čistého polovodiče (křemík). Monokrystal musí být velmi čistý (bez poruch). Proto se provádí čistění (za vysokých teplot). Hotový monokrystal má válcový či doutníkový tvar. Po-té se nařeže na velmi tenké plátky. Které se musí dokonale vyleštit. Na plátcích se vytvářejí masky, na nezamaskovaná místa se přidávají různé příměsi, které vytvoří polovodič (typu N nebo P). Po vytvoření struktury obvodu se na povrch navaří tenká vrstvička kovu, která se za pomocí masky odleptá, takže na určených místech vzniknou kovové kontakty. Na jedné destičce se většinou vytvoří více řad stejných obvodů, které se otestují a rozřežou. Čipy lze vrstvit. Celý obvod je zapouzdřen do pouzdra.

Hybridní obvody

- Skládají se z tenké keramické destičky, na kterou jsou naneseny vodivé spoje a přilepeny křemíkové destičky s polovodičovými součástkami nebo s jednoduššími monolitickými IO. Po-té se provede kontaktování polovodičových součástek a uzavření do pouzdra.

Rozdělují se na:

- **bipolární** – nejrozšířenější, jediné napájení, logický zisk, rychlost, cena, TTL
- **unipolární** – proudový spínač, ECL, CMOS

ARCHITEKTURA POČÍTAČE

ARCHITEKTRURA ORGANIZACE A IMPLEMENTACE POČÍTAČŮ

- architektura stoje je globální pohled na všechny podstatné vlastnosti počítače
- původní definice praví, že je to souhrnný přehled množiny registrů, paměti, instrukčního souboru, datových formátů a adresovacích módů
- v podstatě je to popis počítače z hlediska programátora ve strojovém kódu (poprvé definoval Gene Amdahl, hlavní architekt OS 360, IBM)

4 POPISY STROJŮ

- - struktura - propojení jednotlivých funkčních bloků
- - organizace - dynamická interakce funkčních bloků
- - implementace - návrh a obvodová realizace
- - funkce - popis stroje jako funkčního celku

- **ORGANIZACE** stroje specifikuje logické seskupení a vzájemné vztahy mezi subsystémy počítače. Je to projekce architektury jednotlivých funkčních jednotek

- **IMPLEMENTACE** stoje je obvodový návrh a realizace počítače pomocí elektronických prvků

OBECNÝ MODUL POČÍTAČE:

- ***základem logické části počítače tvoří:***
  - - paměť - pasivní zařízení k uložení dat
  - - procesor - aktivní prvek, který je schopen interpretovat program
  - - propojení - přenáší data, mění umístění dat nikoli obsah
  - - transduktor - mění reprezentace dat, jiný způsob kódování

- dále je nutno definovat instrukční soubor, způsoby adresace a prezentace dat

- uvedené skutečnosti popisují sériový neboli skalární počítač: stroj, který postupně (sekvenčně) zpracovává instrukce uložené v paměti. Stejně postupně probíhá zpracování jedné instrukce: čte příkaz, potom operandy, potom vykoná instrukci, potom uloží do paměti výsledky operace

OBECNÝ MODEL POČÍTAČE

- ***v architektuře počítačů lze vysledovat dva koncepční přístupy:***
  - - je-li operační paměť společná pro data i programy hovoříme o **klasické architektuře von Neumannově**
  - - je-li paměť rozdělená, jedná se o **architekturu hardvarskou**

OBECNÝ MODEL POČÍTAČE – NĚKTERÉ ZÁKLADNÍ POJMY:

- **adresa** je označení místa uložení instrukce nebo dat v paměti stroje. je to pořadové číslo udávající pozici adresovatelné jednotky od začátku lineárního prostoru paměti
- **instrukce** je kódovaný příkaz, který způsobí, že stroj vykoná určitou činnost
- vykonání jedné instrukce se nazývá **instrukčním cyklem**
- vykonání jedné instrukce je složeno z provedení jistého počtu elementárních úkonů
- **hodinový cyklus** je časový interval mezi dvěma následujícími čely hodinového impulsu – je to nejmenší rozlišitelná časová jednotka instrukčního cyklu
- **strojový jazyk** je úplný soubor strojových instrukcí (také instrukční sada, soubor)
- **počítač je** stavový stroj, pracuje synchronně, činnost je řízena hodinovým signálem

- ***k vykonání instrukce jsou potřebné alespoň tři kroky:*** vytažení, dekódování a provedení (fetch, decode, execute)

- ***jsou-li data uložena v paměti potřebujeme ještě další 2 kroky:*** vytáhnout data a uložit výsledek. Obecně lze provedení instrukce rozbít na 5 logických kroků, což však nic neříká o počtu hodinových textů k provedení instrukce nutných!!!!

SHRNUTÍ

- základními funkčními bloky počítače je paměť, ALU, zařízení vstupu a výstupu a řídící jednotka (řídící logika realizovaná jako řadič)
- řídící jednotka počítače používá dvoustavovou binární logiku reprezentovanou el. napětím (stavy: přítomnost/nepřítomnost el. napětí)
- struktura počítače je neměnná, činnost počítače je řízena programem, což je posloupnost instrukcí stroje. Změny v činnosti stroje je dosaženo změnou programu
- program přestavuje předpis (elementárních úkonů), jak manipulovat s daty, aby byla získána (nová) informace. Skalární počítač používá přímo sekvenční předpis.
- sled instrukcí nemusí být lineární (přeskočení, opakované užití části programu)
- podstatou zpracování dat počítače je aplikace matematických a fyzických operací
- program je uložen v paměti počítače v binární podobě
- data jsou v paměti počítače uložena v binární podobě

PROCESORY SUBSKALÁRNÍ, SKALÁRNÍ, SUPERSKALÁRNÍ

- vývoj mikroprocesorů prošel zhruba třemi vývojovými etapami

1. etapa:

- vyznačuje se sekvenčním vydáváním a sekvenčním prováděním instrukcí. Doba pro vykonání určitého programu je dána součtem trvání jednotlivých instrukcí. Jedná se o klasické Voj Neumannovo schéma **(subskalární procesor).** Charakteristickým rysem je, že v jednom hodinovém taktu je vykonána méně než jedna instrukce.

2. etapa:

- Ve druhé etapě bylo sekvenční zpracování instrukcí změněno vykonáváním paralelním. Čehož je dosaženo buď replikací funkčních jednotek (několik stejných jednoteek) nebo pomocí zřetězeného zpracování instrukce). Hovoříme o **skalárních procesorech** a jejich charakteristickým rysem je vykonání nejvýše jedné instrukce v jednom taktu hodin. Protože se instrukce provádějí překryvně nebo dokonce paralelně, je obtížné vypočítat dobu vykonávání programu. Podstatný je však fakt, že v důsledku konfliktů (datový a skokový) je počet hodinových taktů potřebných k realizaci vždy větší než počet instrukcí v programu.

3. etapa:

- Zatím poslední etapu představují **superskalární** mikroprocesory. Používají paralelně vydávání i paralelní zpracování. V zásadě se jedná o to, že sekvenčně pracující vydávací jednotka nestačí zásobovat paralelně pracující funkční jednotku. Prvotním řešením byla architektura VLIW (very long instruction word). Procesory s dlouhým instrukčním slovem vyžadovaly, aby skupiny současně vydávaných instrukcí byly předpřipraveny v době kompilace programu (problémy se zpětnou kompatibilitou). Vývojem bylo dosaženo dynamického plánování vydávání – procesor je schopen v průběhu vykonávání programu seskupovat instrukce do skupin, které je možno paralelně zpracovat (a speciální kompilátor tady není nutný).

- základní hnací silou rozvoje mikroprocesorů je zvyšování výkonu výpočetního systému. Podstatou procesoru je zvyšování stupně paralelizmu procesoru. Zatím poslední inovací je použití více jader v jednom procesoru. Je však třeba zdůraznit, že prostá replikaci funkčních jednotek nestačí. Zásadní význam má i postupná integrace ostatních částí výpočetního systému přímo do procesoru. Prognózy předpokládají zásadní změnu v architektuře mikroprocesorů při dosažení hranice jedné miliardy tranzistorů na čipu. Nezadatelný vliv na výkon (stabilitu, bezpečnost) výpočetního systému má i hardwarová podpora některých funkcí, původně zajišťovaných zcela OS

- základním nedostatkem klasické von Neumannovy koncepce počítače (skalární počítač) jsou omezené možnosti zvyšování výkonu zdroje. To je důsledek sekvenčního způsobu práce prakticky všech komponent počítače. Zpracování informace je realizováno jako posloupnost elementárních úkonů, datová položka prochází řadou transformací, přičemž nová operace začne, až ta předchozí skončí. Jinak řečeno je-li položka zpracovávána v n funkčních jednotkách, je každá jednotka (n-1) intervalů nevyužita a čeká na příchod další datové položky.

- bylo by velmi efektivní, kdyby v libovolný časový okamžik, byly plně využity všechny funkční jednotky počítače

-způsoby zvýšení výkonu stroje:

- překrývání (overlopping)
  - nová instrukce se začne zpracovávat dříve, než je rozpracovaná instrukce úplně dokončena.

- dokonalejší možností je proudové (zřetězené) zpracování (pipelining).
  - metoda je to poměrné stará as vývojem architektury počítačů existuje tendence, používat tuto metodu na stále nižší úrovni.

PROUDOVÉ ZPRACOVÁNÍ MŮŽEME APLIKOVAT NA PROVEDENÍ:

- aritmeticko-logických operací (proudová sčítačka)
- na realizaci instrukcí (výběr, dekódování, výpočet adres operandů provedení, zápis výsledků)
- realizaci procesů (prokládaná paměť, sběrnice s rozdělenými transakcemi)

- proudové zpracování sice zvyšuje výkon stroje, ale také přináší nové problémy. Předpis pořadí zpracování instrukcí není jednosměrný (skoky, smyčky), což vede k datovému a skokovému konfliktu.
  - **datový konflikt** - vznikne, když rozpracovaná instrukce nemůže být dokončena, protože potřebuje pracovat s daty, jejichž modifikací provádí předchozí (taktéž nedokončena) instrukce.

  - **skokový konflikt** - vznikne při zjištění, že právě dokončená instrukce mění posloupnost vykonávaných instrukcí a všechny rozpracované instrukce je třeba zrušit (prováděly se zbytečně).

- -uvedené problémy je možné řešit softwarovou nebo hardwarovou cestou. Optimalizující kompilátor je jedno z možných (a poměrně nedokonalých) řešení (co třeba s existujícími, už přeloženými programy). Hardwarová řešení předpokládají vyřešení konfliktů za běhu programu tak, aby nedošlo k významnému snížení výkonu stroje. Používají se technologie umožňující předpovídání (predikci) vícenásobného větvení a analýzu datových závislostí, což vede k vykonávání instrukcí mimo pořadí. Stroje s velmi dlouhým instrukčním slovem (VLIW) řeší problém konfliktů kombinací obou přístupů HW + SW
- proudové zpracování předpokládá rozdělení nějaké činnosti na posloupnost kroků. Každý krok je prováděn samostatnými technickými prostředky, které realizují určitou operaci. V jednom okamžiku je prováděno několik činností, každá v jiném stupni rozpracovanosti a každá využívá jiný stupeň řetězu technických prostředků.

- základní předpoklady pro proudovou realizaci nějaké činnosti jsou následující:
  - soustavný přísun dat, nad kterými se provádí příslušná činnost
  - činnost musí být rozdělitelná na sekvenci nezávisle proveditelných operací
  - trvání každé jednotlivé operace musí být přibližně stejné

- v procesorech se proudové zpracování používá jednak při provádění aritmetických operací (ALU) a jednak při vykonávání instrukcí (CPU)

- proudové pracování se objevilo v superpočítačích zhruba v 60. letech minulého století. Nejprve při vykonávání instrukcí (CDC 6600, 1964) a následně v aritmetických jednotkách (IBM 360/91, 1967).

INSTRUKČNÍ SOUBOR

RISC – reduced instruction set computer
CISC – complex instruction set computer

RISC:

- minimální instrukční soubor
- jednoduché způsoby adresování
- pevný formát instrukce
- vykonání instrukce v jednom strojovém cyklu
- datové operace pouze nad registry
- styk s paměti výlučně instrukcemi load/store
- zřetězená realizace instrukcí
- nutností je optimalizující kompilátor

CISC:

- Snaha zachovat kompatibilitu
- Soubor instrukcí většinou přes 200
- Málo registrů
- Řadič mikroprogramový (každá nová instrukce nový mikroprogram, nebo jejich sled)
- Určitá skupina bitů v instrukcích má různý význam. Jedna skupina určuje, co druhá vlastně znamená.
- Proměnlivý formát instrukcí

ETAPY VÝVOJE RISC:

1. ***etapa (1975 - 1982)***
   - pouze expocimentální stroje, realizován pouze procesor
   - procesor IBM 801

2. ***etapa (1982 – 1985)***
   - 1. generaci prakticky použitelných strojů

3. ***etapa (1985 -)***
   - Seymour Cray (CDC)
   - John Cocke (IBM)
   - David Pattersen (Berkeley)
   - John Menessy
   - Sun SPARC
   - Acorn ARM
   - Motorola 88000
   - DEC APLHA21064A

CISC

- Systémy s rozsáhlým komplexním souborem instrukcí. Pro tyto procesory je typická implementace architektury pomocí mikroprogramování. Výsledkem je velký počet specializovaných typů instrukcí, z časového pohledu mohou trvat až 300 strojových cyklů. Mikroprogramování poskytuje možnost nabízet spektrum strojů se stejnou architekturou, ale přesto rozdílnou hardwarovou realizací. S rostoucím objemem sady instrukcí však bylo pro překladače překládající programy do strojového kódu využít celou škálu speciálních instrukcí. Z toho vyplývá, že komplexní instrukce jsou používány jen zřídka. Při zpracování programu z vyššího programovacího jazyka se velká část času spotřebuje na čtení informace z pracovní paměti a naopak.

RISC

- RISC procesor vystačí pouze se 30 typickými instrukcemi. RISC počítač musí s tímto souborem pracovat častěji než CISC, ale díky většímu počtu registrů může program proběhnout rychleji, protože většina operací se koná přímo mezi registry a pamětí. CISC CPU má sadu obvykle 16 registrů, zatím co RISC má až 100 volných registrů. Z toho důvodu není centrální jednotka vůbec zatěžována a mimo to je mikrokód zbytečný. Instrukce jsou implementovány hardwarově (nejsou dekódované). Podstatné vlastnosti RISC - architektury umožňující vysokou propustnost dat, mají malý počet jednoduchých instrukcí, instrukce s pevnou délkou a pevným formátem, přímá hardwarová interpretace instrukcí, jednocyklový příkazový režim a díky internímu režimu Pipeling překrývané provádění po sobě následujících instrukcí. Hlavní nárůst rychlosti RISC byl dosažen použitím velkého množství registrů. Tím, že odpadly mikroprogramy, mohla být uvolněna celá oblast na povrchu čipu, která byla tradičně používaná pro mikroprogramy a použitá pro velmi rychlou paměť. U procesorů RISC je jednodušší a rychlejší změna návrhu čipu, než u komplikovaných a komplexních struktur CISC. Architektura RISC má slabinu ve výpočtech v plovoucí desetinné čárce, proto byl zaveden speciální koprocesor pro numerické výpočty.

INSTRUKCE

INSTRUKCE:

- co se má udělat, s čím se to má udělat, kam se má uložit výsledek a kde je následující instrukce

- všechny čtyři části instrukce musí být jednoznačně přímo nebo nepřímo určené. Každá instrukce má tedy 4 log. části a 5 polí: operační kód, prvý operand, druhý operand, místo uložení výsledku a adresu následující instrukce.

ČTYŘADRESOVÉ INSTRUKCE:

- obsahují všechny čtyři části instrukce
- historicky se používali u strojů, které jako paměťové médium měly mag. buben
- dnes jsou občas interně používány v některých řídících strukturách procesoru (mikrokódem řízený kontrolér)

TŘÍADRESOVÉ INSTRUKCE:

- neobsahují pole adresy následující instrukce. Ta je určena nepřímo pomocí registru programového čítače (také instrukční registr), který je modifikován prováděnou instrukcí (délkou nebo obsahem) tak, že vždy ukazuje na následující instrukce

DVOUADRESOVÉ INSTRUKCE:

- instrukce nemají přímo vyjádřenou adresu výsledku, předpokládá se, že výsledek bude uložen na místo jednoho z operandů. Dnes nejčastější varianta instrukce

JEDNOADRESOVÉ INSTRUKCE

- předpokládají užití akumulátorů, což je speciální registr, ve kterém je vždy jeden z operandů a do kterého i bude uložen výsledek prováděné instrukce

BEZ ADRESOVÉ INSTRUKCE

- bez adresové instrukce zásobníkově orientovaných strojů nepotřebují adresy. Výsledek i operand jsou vždy uloženy v zásobníku. Instrukce naplnění zásobníku (PUSH) a uložení obsahu zásobníku (POP) však mají adresu. Tyto stroje jsou vhodné jen pro určité výpočty

- programový kód pro tříadresové stroje je nejkompaktnější (nejkratší). Vykonání instrukce je však časově nejnáročnější (dlouhá instrukce – opakované čtení z paměti)
- naopak programový kód pro zásobníkově orientované stroje je nejdelší avšak vykonání instrukce je nejrychlejší (instrukce jsou krátké a většinou není třeba vypočítávat adresy operandů)

- principální výhodou strojů s akumulátorem a zásobníkově orientovaných je fakt, že výsledek operace zůstává v procesoru a může být opakovaně použit v další operaci bez nutnosti čtení či zápisu do paměti. Bohužel akumulátore je jen jeden, což vedlo ke konstrukci strojů s registry pro všeobecné použití. V procesoru je sada registrů (nebo i několik sad), které slouží k uchování mezivýsledků nebo adres o a tudíž eliminují následné přístupy do paměti. Dalším efektem použití registrů je zkrácení délky instrukce.

INSTRUKCE:

- je řetězec symbolů, které při interpretaci procesorem způsobí jednoznačnou a definovanou změnu stavu stroje

- ***obsahuje:***
  - operační kód (určuje, jaká změna stavu nastane)

- explicitně nebo implicitně zadané identifikátory v adresovém prostoru stroje, které mají být v dané akci použity jako parametry

- provedení instrukce je jediný způsob jak změnit stav stroje. Vykonaná změna stavu se promítá do dvou oblastí – výsledků operace a nastavení příznaků. Příznak charakterizuje ukončenou instrukci a může být použit jako vstupní parametr následující operace:

OPCODE	01	Ö2	O3	OX

- instrukci lze také chápat jako datový typ – vykazuje všechny jeho vlastnosti

- ***instrukční repertoár –*** množina pravidel integrovaná v hardware procesoru, která zcela přesně určuje jednotlivé stavy stroje (jinak také instrukční soubor)

instrukční soubor lze dělit podle různých kritérií:

- - instrukce privilegované (nejsou dostupné každému programu)
- - instrukce neprivilegované (jsou dostupné každému programu)
- - instrukce přenosu dat (paměti, registry, zásobníky)
- - diadické instrukce (aritmetické a logické operace)
- - monodické instrukce (nulování, inkrement, negace, rotace, posuny)
- - instrukce větvení, skoků a cyklů
- - instrukce volání podprogramů
- - instrukce vstupu, výstupu
- - instrukce pro řízení stroje

- většina je implementována jako řada podobných instrukcí nebo jako jedna instrukce, která bude mít různé modality (variace operačního kódu)

ZPŮSOB ADRESACE

- Určuje, jak bude k datům nebo instrukcím přistupováno
- adresa uvedená v instrukci často není adresou finální (neukazuje přímo na místo uložení)
- stroj musí vykonat jistý postup (předepsaný v instrukci), aby získal skutečnou, konečnou adresu – efektivní adresu

pro existenci různých způsobů adresace je nejméně:

1. ***dobrý důvod***
   - čím variabilnější jsou módy operace, tím elegantnější a efektivnější by měl být programový kód

2. ***špatný důvod***
   - spletitý postup získání konečné adresy komplikuje konstrukci procesoru a prodlužuje dobu potřebnou pro vykonání instrukce)

BEZPROSTŘEDNÍ ADRESACE:

- Instrukce obsahuje operand. Data jsou přímo součástí instrukce. Vložené hodnoty jsou neměnné (konstantní), proto hovoříme o instrukcích pro práci s literálem (literál je datová položka, která nemá vlastní identifikátor)

PŘÍMÁ ADRESACE:

- instrukce obsahuje adresu operandu. Tato adresa je adresou efektivní

NEPŘÍMÁ ADRESACE:

- Instrukce obsahuje adresu, na které je adresa operandu

REGISTROVÁ PŘÍMÁ ADRESACE:

- Instrukce obsahují číslo (malou adresu) registru, který obsahoval operand. Velmi efektivní instrukce – doba potřebná k vykonání je zvláště krátká

REGISTROVÁ NEPŘÍMÁ ADRESACE:

- Instrukce obsahuje číslo registru, ve kterém je uložena adresa operandu. Tento adresový mód je obzvláště efektivní, potřebujeme-li adresu operandu plynule měnit (práce s bloky dat). Čehož možno dosáhnout prostým přičítáním nebo odečítáním k registru, ve kterém je uložena adresa (a také ji příslušně modifikovat)

REGISTROVÁ NEPŘÍMÁ ADRESACE S POSUNEM:

- Instrukce obsahuje číslo registru, ve kterém je uložena adresa a posun. Efektivní adresa operandu se získá sečtením těchto dvou hodnot. Posun musí být konstanta nebo číslo registru (proměnná hodnota uložená v registru)

- při adresování instrukcí se používá relativní adresace (vzhledem k programovému čítači). Instrukce pro řízení chodu programu (například podmíněný a nepodmíněný skok, volání programů) obsahují konstantu („vzdálenost místa skoku“), jejímž přičtením k programovému čítači získáme adresu následující instrukce (je to jistá modifikace bezprostřední adresace). POZOR, konstanta neudává, kolik, instrukcí se má přeskočit, ale kolik nejmenších adresovatelných jednotek (většinou bajtů se přeskočí)!!!! Velikost pro konstantu (počet bitů) limituje minimální délka skoku (1 bajt ≈ skok ± 127 bajtů)

PROBLÉMY SPOJENÉ SE ZVYŠOVÁNÍM VÝPOČETNÍHO VÝKONU STROJE

- **stupeň integrace:** čas přepnutí logického elementu (což určuje dobu instrukčního cyklu) je limitující do hodnoty cca 1 nasec
- **rychlost signálu vodičem:** rychlost světla: 0,3m za 1 nasec
- počítač nemůže být větší než kopací míč, což přináší další problém – odvod produkovaného tepla
- **Moore Gordon E.** (1964, ředitel Faichild Semiconductor, jeden ze zakladatelů Intelu)
- **počet komponent integrovaných na čipu se každých 18 měsíců zdvojnásobí**
- počítač slouží k řešení problému reálného světa. Musíme nějak reálný svět do počítače dostat. To je modelování reálného. Dosažením lepších výsledků dosáhneme jak zlepšením modelu, tak zlepšením výpočetního výkonu. Zlepšením modelu je většinou spojeno bezprostředně s pokrokem v ostatních vědních oborech. Zvyšováním početního výkonu při stávající technologii není možné do nekonečna. Reálná jsou omezení daná fyzikálními principy. V budoucnu budou počítače pracovat na jiných principech (chemické, kvantové, elektrické nanopočítače)

PROPOJOVACÍ SUBSYSTÉMY

- propojovací subsystémy můžeme rozdělit na vnitřní a vnější. Každé propojení je charakterizováno rychlostí, šířkou pásma a pořizovací cenou

- propojovací systém je většinou hierarchický. Význačnou vlastností je způsob šíření informace mezi jednotlivými jednotkami – organizace subsystému

- ***směrová*** – vždy bude aktivována zvláštní cesta (část subsystému e neúčastní)

- ***sběrnicová*** – informace jsou předávány po obecných cestách

    - ***sériový kanál*** – informace je přenášena po elementárních cestách
    - ***paralelní kanál*** – přenášejí se větší jednotky informace

SMĚROVÁ ORGANIZACE

- propojovací síť je složena z uzlů, které komunikují prostřednictvím spojů. Každý spoj je dvoubodové spojení – na jednom spoji se nemůže podílet více uzlů. Způsob propojení jednotlivých jednotek sítě se nazývá TOPOLOGIE propojovacího systému. Vlastnosti navržené topologie určují, jakým způsobem budou jednotky sdílet zdroje a co toto sdílení bude stát. Nejjednodušší topologie je propojení dvou jednotek (bod – bod). Komunikační strategie je algoritmus směrování zpráv na cestě od uzlu k síti

SBĚRNICOVÁ ORGANIZACE

- všechny jednotky jsou propojeny jednou nebo několika společnými sběrnicemi
- komunikace je rozdělena do diskrétních transakcí mezi vysílačem a přijímačem. Transakce začne, když některý účastník získá kontrolu nad sběrnicemi a stane se řídící jednotkou (master). Často existuje více jednotek, které se mohou stát řídící.

ARBITRÁŽNÍ PROTOKOL

- je uplatnění, pokud více jednotek ve stejný okamžik usiluje o získání řízení sběrnice, transakce na sběrnici je řízena komunikačním protokolem

ASYNCHRONNÍ PROTOKOL

- protokol, při kterém přednost elementu informace může začít kdykoli

SYNCHRONNÍ PŘENOS

- může být zahájen jen v přesně stanovený okamžik. Provoz na sběrnici je vždy řízen taktovacím kmitočtem (hodinové impulzy). Výkon sběrnicového systému je určen dvěma parametry:
    3. přenosovým časem
    4. šířkou pásma (bity za sekundu)

ARBITRÁŽ POŽADAVKŮ O SBĚRNICI

- sběrnice je sdílené médium – dochází k situacím, kdy o použití soupeří několik jednotek. Pravidla při rozhodování o přidělování sběrnice jsou v zásadě dvojí:

    - **1, centrální arbitráž** – priorita, pořadí příchodu (LIFO) nebo náhodné
    - **2, decentralizovaná arbitráž** – prioritní linka (daisy chain), cyklické (tokem) a kolizní

- centrální arbitráž zajišťuje zvláštní jednotka, která vyhodnocuje přicházející žádosti

- decentralizovaná arbitráž používá rozhodovací algoritmy vestavěné do každé připojené jednotky. Jednotky mají vyšší inteligenci a potřebují jistý čas na provedení arbitráže a výměnu zprávy o jejím výsledku.

- libovolný propojovací systém bude vyžadovat jistou programovou obsluhu. V případě srovnatelných rychlostí obou komunikujících jednotek bude pro programátora transparentní. Je – li rozdíl významný, je třeba komunikaci řešit tak, aby rychlejší jednotka byla komunikačním procesem zatížena co nejméně.

- ***existují následující způsoby práce:***
  - programová obsluha
  - obsluha s využitím přerušení
  - pomocí blokové přímého přístupu do paměti (DMA)
  - samotné kanálové procesory

- mechanizmus přerušení je v zásadě vždy stejný – přerušení práce procesoru na vnější podnět, vykonání obsluhy přerušení a návrat k původní práci

- obvyklý způsob vyvolání programu obsluhy přerušení je nucený skok na adresu, kde je program uložen, pomocí tabulky vektorů, ukazující na příslušné obsluhy přerušení (skutečnost je komplikovanější)

- přenos se v počítači uskutečňuje buď po jednotlivých datových elementech (bajty) nebo po blocích. Pomalá zařízení (tj. taková, u kterých je odezva na přenos podstatně delší než vlastní přenos), používají většinou přenos po datových elementech (asynchronně) a rychlá pak blokový (synchronní) přenos

BLOKOVÉ PŘENOSY – DIRECT MEMORY ACCESS (DMA)

- někdy je výhodné, aby se procesor řízení komunikačního procesu neúčastnil
  - procesor je řadičem DMA odpojen od sběrnice (a případně i zastaven)
  - řadič DMA zpomalí procesor zastavením hodin (kradení cyklů)

- KANÁLOVÉ PŘENOSY

- snaha o nezávislost komunikace na procesoru vedla ke vzniku komunikačních procesorů. Procesor je rychlým spojem propojen se specializovaným procesorem, kterému pouze předá požadavek na přemístění určitého množství dat

- komunikační procesory jsou speciálně navržená zařízení s vlastní instrukční sadou orientovanou na efektivní přesuny dat. Rychlému spoji říkáme komunikační kanál a komunikačnímu procesoru kanálový

PAMĚŤOVÝ SUBSYSTÉM

- paměť je zařízení, které je schopné přijmout informace, uchovat je po danou dobu a na požádání je vydat
- paměť je složena z adresovatelných jednotek – paměťových míst (buněk, lokací)

- ***paměť můžeme charakterizovat:***
  D. funkcí (čtení, zápis)
  E. kapacitou (bity, bajty)
  F. vybavovací doba, cyklus

ROZDĚLENÍ PAMĚTÍ PODLE PŘÍSTUPU K DATŮM:

- **RAM (random access memory)** – paměť s libovolným přístupem. Doba odezvy je konstantní a nezávisí na adrese, která je použitá pro čtení a zápis
- **SAM (serial access memory)** – paměť se sekvenčním přístupem – zpožďovací linky
- **DAM (direct access memory)** – paměť s přímým přístupem (disk)
- **CAM (content addressable memory)** – paměť adresovaná obsahem

ROZDĚLENÍ PAMĚTÍ PODLE IMPLEMENTACE:

- **RWM (read/write memory)** – paměti pro čtení a zápis na libovolnou adresu
- **ROM (read only memory)** – paměti schopné pouze číst
- **SRAM (static random access memory)** – statické paměti RAM
- **DRAM (dynamic random access memory)** – dynamické paměti RAM
- **PROM (programmable read only memory)** – jednou programovatelné

HIERARCHICKÉ USPOŘÁDÁNÍ PAMĚTÍ:

- **REGISTRY** – vysoká rychlost přístupu – realizace přímo jakou součást procesoru
- **CACHE** – vyrovnávací paměť – co nejblíže procesoru (10 nanosekund)
- **HLAVNÍ** – fyzická paměť počítače pro instrukce a data (10 – 100 nanosekund)
- **ODKLÁDACÍ** – ukládání stránek hlavní paměti – disk (milisekundy)
- **SEKUNDÁRNÍ** – vnější paměti s náhodným přístupem (disky, diskety)
- **ARCHIVNÍ** – trvalé ukládání velkých objemů dat, ke kterým se přistupuje zřídka

ORGANIZACE PAMĚŤOVÝCH JEDNOTEK:

- **paměti se širokým slovem** – do datového registru se předává několikanásobek potřebné informace. Nutným předpokladem je následné čtení dat či instrukcí

- **paměti s prokládanými cykly** – rozdělení pamětí na moduly (banky) a střídavý přístup k informacím v různých modulech

KONTROLA A KOREKCE CHYB

- parita příčná a podélná. Samoopravné kódy (Hammingův)

HIERARCHICKÉ USPOŘÁDÁNÍ PAMĚTI

- vychází ze skutečnosti, že přístupy do paměti mají tendenci shlukovat se do skupin

- lokalita referencí:
  C. ***časová*** – reference, která byla právě použita a bude v krátké době použita znovu
  D. ***prostorová*** – reference následující bude pravděpodobně ležet poblíž právě použité

- mechanizmus funkce je následující – paměť je rozdělena na úseky, tvořené posloupnosti adresovatelných jednotek. Podle požadavků procesoru je obsah hierarchicky nižší paměti vyměněn s vybranou oblastí hierarchicky vyšší paměti

- ***mapovací mechanizmy*** – stránkovaná a segmentovaná paměť

NĚKTERÉ DŮLEŽITÉ ORGANIZACE VÝBĚRU Z PAMĚTI

- zásobníková organizace (STACK) je pamětí LIFO. Může být realizovaná jako zvláštní součást operační paměti i jako samostatná paměť na bázi registrů. Je adresovaná přes zvláštní registr – SP (stack pointer). Základními operacemi se zásobníkem je operace uložení a vybrání položky (PUSH, POP)

OCHRANA PAMĚTI

- mechanizmus řízení přístupu ke specifickým částem operační paměti. Narušení ochrany se projeví jako přerušení, které obsluhuje operační systém. Užívá se v jednouživatelském i víceuživatelském režimu práce.

- ***může být realizováno takto:***
  - **- mezními registry** – obsah MAR se porovnává s limitními hodnotami
  - **- maskou** – paměť je rozdělena na stránky. Maska = číslo stránky
  - **- klíče** – prakticky totéž co maska. Srovnání provádí software
  - **- metabity** – pomocné bity přidané k datovému obsahu paměti.

PC – PAMĚTI

- systémová paměť může být tvořena dvěma jakostními typy fyzické paměti:
  - ***DRAM (DYNAMIC RAM)***
  - ***SRAM (STATIC RAM)***

- paměťové buňky čipu DRAM jsou tvořena dvojicí malého kondenzátoru a tranzistoru
- paměťové buňky čipu SRAM jsou tvořeny 6 tranzistory SRAM = 2M = DRAM 64 MB

ZÁKLADNÍ TYPY PAMĚŤOVÝCH MODULŮ DRAM

- **FPM – Fast page mode**
- **EDO – Extended data ant**
- **SDRAM – Synchronous dynamic RAM**
- **RDRAM – Rambus DRAM**
- **DDR SDRAM – Double data rate SDRAM**

FYZICKÉ USPOŘÁDÁNÍ PAMĚTI RAM

- **SIMM – single inline memory module** – 30 a 72 vývodů
- **DIMM – dual inline memory module** – 168 a 1284 vývodů (DDRDRAM)
- **RIMM – Rambus inline memory module** – 184 vývodů

BUŇKY PAMĚTÍ

- nejmenší množství paměti, které může být najednou adresováno
- šířka paměti musí vždy odpovídat šířce procesorové sběrnice

VÝVODY MODULŮ

- povrch kontaktů je pokryt zlatem nebo cínem. Nelze doporučit používání konektorů a vývodů modulů s různým pokovením (fretting corrosion)

SPOLEHLIVOST PAMĚŤOVÉHO SUBSYSTÉMU

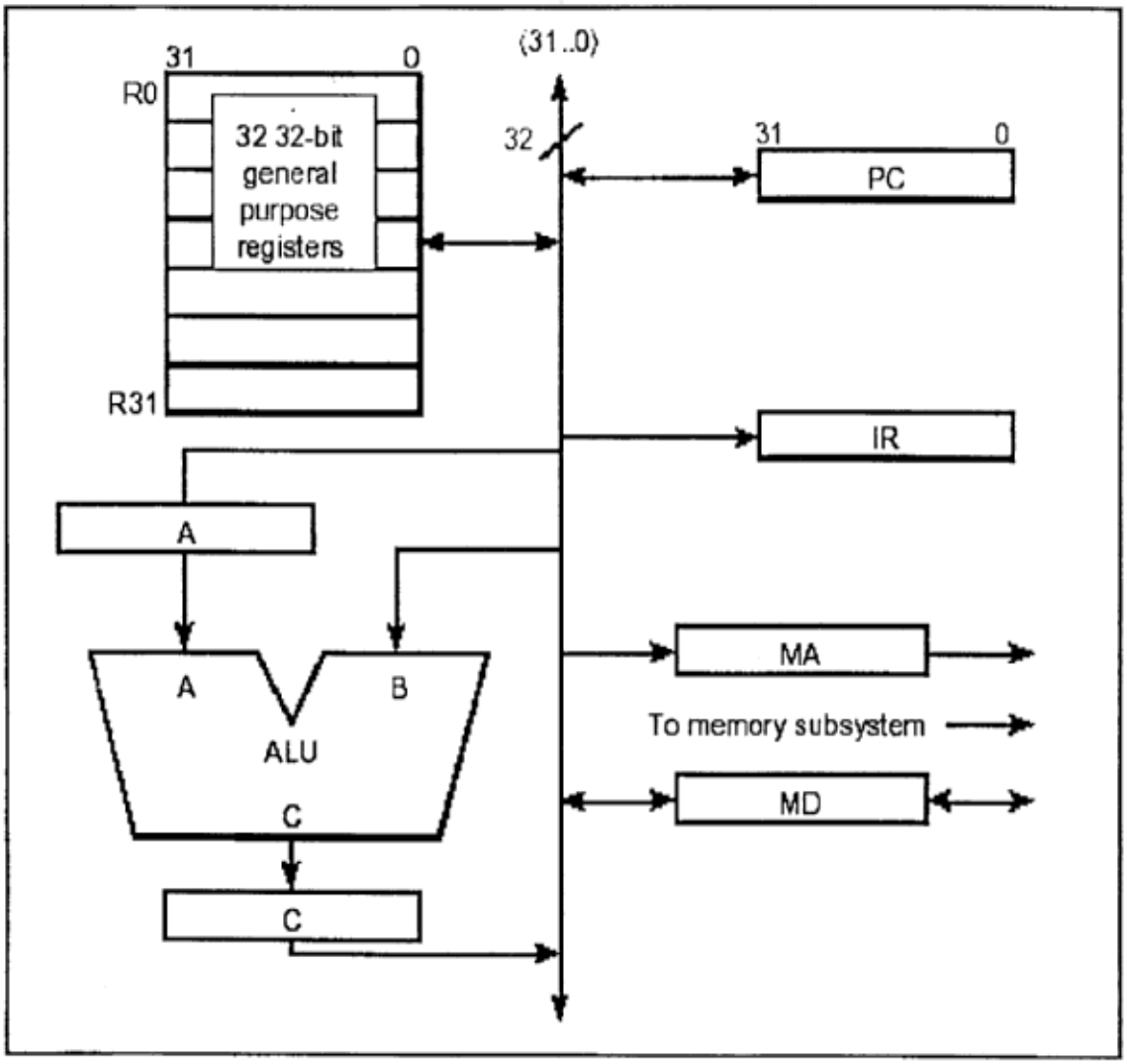
- chyby můžeme rozdělit na logické a fyzické

- ***FYZICKÉ*** – projeví se trvalým selháním modulu
- ***LOGICKÉ*** – projeví se nepravidelně a mohou být způsobeny: poruchami napájení, rušením, statickými výboji, zářením nebo nevhodným použitím paměťového modulu (výpadky časování)

- ochranou je použití paritního bitu nebo ECC (Error correcting code), což zvyšuje cenu paměťového modulu

OBECNÝ PROCESOR

PROCESORY.

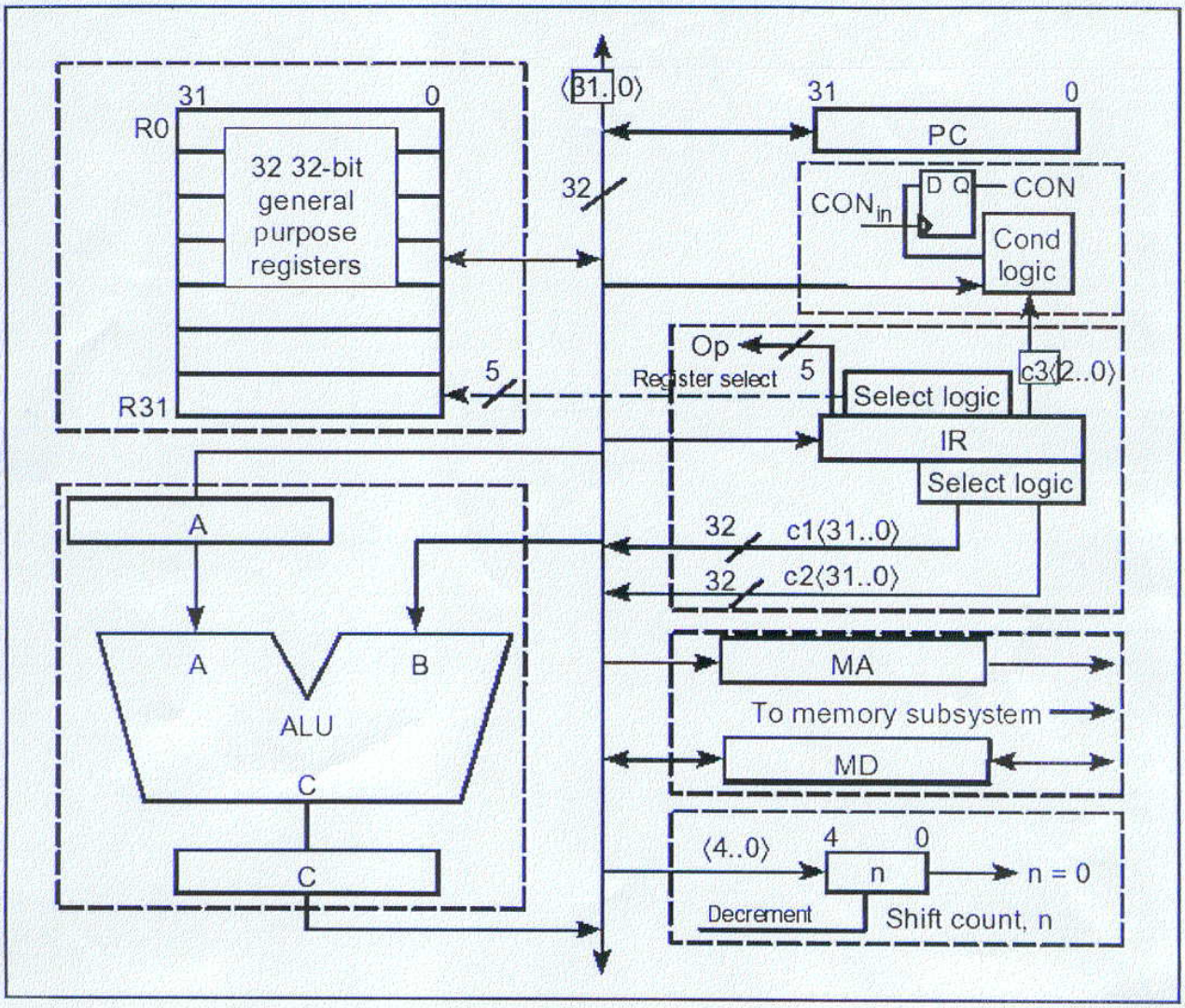


Obecný procesor je složen z částí:

1) Soubor 32 dvaatřicetibitových registrů
2) Aritmeticko logická jednotka
3) Realizace podmíněných skoků
4) Dekódování instrukce a extrakce konstant
5) Rozhraní pro styk s operační pamětí
6) Čítač posuvů

PROCESORY IV.

Logická struktura procesoru.



Logická struktura procesoru je složena z bloků :

1. Soubor 32 dvaatřicetibitových registrů
2. Aritmeticko logická jednotka
3. Realizace podmíněných skoku
4. Dekódování instrukce a extrakce konstant
5. Rozhraní pro styk s operační pamětí
6. Čítač posuvů

PŘEHLED INSTRUKCÍ.

Nr.	Binárně	Mnemonika	Slovní popis	Instrukce	
0	0 0000	nop	Žádná operace		
1	0 0001	ld	Naplnit registr	ld	rA K
2	0 0010	ldr	Naplnit registr relativně	ldr	rA, D
3	0 0011	st	Uložit obsah registru	st	rA, K
4	0 0100	str	Uložit obsah registru relativně	str	rA, D
5	0 0101	la	Naplnit registr adresou	la	rA, K
6	0 0110	lar	Naplnit registr adresou relativní	lar	rA, D
7	0 0111				
8	0 1000	br	Předání řízení na návěští	br	rB, rC, P
9	0 1001	brl	Předání řízení s návratem	brl	rA, rB, rC, P
10	0 1010	een	Povolit přerušení		
11	0 1011	edi	Zakázat přerušení		
12	0 1100	add	Sčítání	add	rA, rB, rC
13	0 1101	addi	Přičtení konstanty	addi	rA, rB, K
14	0 1110	sub	Odčítání	sub	rA, rB, rC
15	0 1111	neg	Druhý dvojkový doplněk	neg	rA, rC
16	1 0000	svi	Uložit kontext při přerušení	svi	rA, rB
17	1 0001	ri	Vrátit kontext	ri	rA, rB
18	1 0010				
19	1 0011				
20	1 0100	and	Logický součin	and	rA, rB, rC
21	1 0101	andi	Logický součin s konstantou	andi	rA, rB, K
22	1 0110	or	Logický součet	or	rA, rB, rC
23	1 0111	ori	Logický součet s konstantou	ori	rA, rB, K
24	1 1000	not	Prvý dvojkový doplněk	not	rA, rC
25	1 1001				
26	1 1010	shr	Posun obsahu registru doprava	shr	rA, rB, P
27	1 1011	shra	Aritmetický posun reg. doprava	shra	rA, rB, P
28	1 1100	shl	Posun obsahu registru doleva	shl	rA, rB, P
29	1 1101	shc	Rotace obsahu registru doleva	shc	rA, rB, P
30	1 1110	rfi	Návrat z přerušení		
31	1 1111	stop	Zastavit stroj		

EFEKTIVNÍ ADRESA:

- počítána nebo platná po vykonání instrukce
- máme definovány dva způsoby adresace operandů – absolutní: umístění je dáno hodnotou K (bezprostřední adresace, když rB =0 nebo bázovaná, když rB ≠ 0) a relativní, když je výsledné umístění dáno jako součet programového čítače a konstanty K

INSTRUKCE PŘENOSU DAT:

- zajišťují přemísťování operandů (data nebo adresy). Žádná nová informace nevzniká. Základem jsou instrukce LOAD (přemístění dat do procesoru) a STORE (uložení dat z procesoru do paměti)

KROK	INSTRUKCE LOAD	INSTRUKCE STORE
T0-T2	Vytažení instrukce	Uložení instrukce
T3	A ← ((rB=0) → 0 : (rB≠ 0) → R[rB]	A ← ((rB=0) → 0 : (rB≠ 0) → R[rB]
T4	C ← A+K[IR]	C ← A+K[IR]
T5	MA ← C	MA ← C
T6	MD ← M[MA]	MD ← R[rA]
T7	R[rA] ← MD	M[MA] ← MD

- instrukce mají celkem tři varianty: absolutní nebo relativní adresu pro data a přesun adresy

Aritmetické a logické instrukce

- pomocí této skupiny instrukcí vzniká v procesoru nová informace. Aritmetické operace se provádějí s operandy numerického charakteru, charakter operandů logických operací je většinou nenumerický (práce se symboly, zpracování grafiky) V této skupině jsou jak instrukce monodické (negace) tak diadické.

Instrukce addi rA, rB, K (bezprostřední adresace)

KROK	POPIS ČINNOSTI
T0-T2	Vytažení instrukce
T3	A ← R[rB]
T4	C ← A+K
T5	R[rA] ← C

Instrukce neg rA, rC (monodická instrukce)

KROK	POPIS ČINNOSTI
T0-T2	Vytažení instrukce
T3	C ← -R[rC]
T4	R[rA] ← C

instrukce shr rA, rB, P nebo shr rA, rB, Rc

KROK	POPIS ČINNOSTI
T0-T2	Vytažení instrukce
T3	n ← P
T4	n = 0 → (n ← R[rC])
T5	C ← R [rB]
T6	Opakuj krok T5 n krát
T7	R[rA] ← 0

- provedení kroku T6 závisí na variantě instrukce posuvu (formální popis je poměrně složitý)

Instrukce řízení chodu programu

- slouží k měnění posloupnosti vykonávaných příkazů

- žádná nová informace nevzniká

- podle provedení je dělíme na:
    a. instrukce bez návratu (podmíněné a nepodmíněné skoky)
    b. instrukce s návratem (volání podprogramů)

- instrukce nepodmíněného skoku předá řízení jiné části programového kódu (je to podmíněný skok, jehož podmínka je splněna vždy)

- skupina instrukcí podmíněných skoků potřebuje k rozhodnutí, zda se krok provede splnění určité podmínky

- někdy je žádoucí, aby řízení programu vrátilo po vykonání požadovaného úkonu zpět a původní činnost mohla pokračovat. Předpokladem je zapamatování návratové adres (adresy instrukce následující za skokem) a nějaký mechanizmus (instrukce návratu), který obnoví chod původní větve.

instrukce br rB, rC, P {2..0}

KROK	POPIS ČINNOSTI
T0-T2	Vytažení instrukce
T3	CON ← cond (R[rC])
T4	COND → PC ←R[rB])

instrukce brl rA, rB, rC, P {2..0}

KROK	POPIS ČINNOSTI
T0-T2	Vytažení instrukce
T3	R[rA] ← PC
T4	CON ← cond (R[rC])
T5	COND → PC ←R[rB])

P {2..0}	***POPIS PODMÍNKY***	***MNEMONIKA***	
000	neskočí nikdy	brnv	brlnv
001	skočí vždy	br	brl
010	skočí, když r[rc] = 0	brzr	brlzr
011	skočí, když r[rc] ≠ 0	brnz	brlnt
100	skočí, když r[rc] ≥ 0	brpl	brlpl
101	skočí, když r[rc] < 0	brmi	brlmi

ŠESTNÁCTIBITOVÉ MIKROPROCESORY

ZÁKLADNÍ CHARAKTERISTIKA

8086

ZÁKLADNÍ VLASTNOSTI:

- - úplná 16bitová paralelní univerzální procesorová jednotka
- - vyroben unipolární technologií HMOS
- - pouzdro DIL 40
- - ekvivalent cca 29000 tranzistorů
- - napájení +5 V, 275 mA max.
- - vstupy a výstupy TTL úrovně
- - hodinový signál jednofázový
- - 20bitová adresová sběrnice dovoluje adresovat 1M paměti
- - 16bitová interní a 16bitová (8bitová u 8088) externí datová sběrnice
- - základem architektury jsou 2 zřetězené subprocesory. Pracují poměrně nezávisle a využívají překrytí fáze zápisu, čtení i výběru instrukce s fázemi vykonání instrukce předchozí. Výsledkem je zvětšení rychlosti a zmenšení nároků na rychlost hl. paměti

JEDNOTKA BIU

- zajišťuje styk s vnější sběrnicí, výpočet adresy a asynchronní výběr instrukcí (realizuje instrukční fázi: VYTÁHNI)

- ***funkčními bloky jednotky jsou:***
  - 0bitová sčítačka pro výpočet adresy (offset + segment) = adresa
  - programový čítač
  - vnitřní komunikační registry
  - čtyři segmentové registry (CS, DS, ES, SS)
  - logika řízení vnější sběrnice a fronta připravených instrukcí (realizována jako 6-ti bajtová paměť FIFO)

VÝKONNÁ JEDNOTKA EU

- vykonává připravené instrukce

- ***funkčními bloky jednotky jsou:***
- ALU s třemi registry pro dočasné uložení operandů
- registr příznaku
- skupina registrů pro všeobecné použití (čtyři registry HL ,dva registry ukazatelů,dva indexové registry), logika instrukcí provedená jako mikroprogram, logika řízení vnitřní sběrnice a vnitřní synchronizace, který mění vnější hodinový signál v posloupnost interních řídících impulsů

- instrukční soubor obsahuje 133 instrukcí, které dovolují zpracovávat jak 8bitové, tak i 16bitové datové položky. Instrukce může mít jeden, dva nebo žádný operand. Délka instrukce je proměnná v rozmezí od 1 do 6 bajtů. Prvý bajt vždy obsahuje instrukční kód, ostatní bajty nesou informace o adresách a operandů. Prvý bajt také obsahuje speciální pole (bit W, = 1 slovo) určující, zda instrukce zpracovává bajt nebo slovo. Procesor je schopen realizovat většinu známých adresový módů. Použití segmentových registrů je předdefinováno. Speciálním prefixem lze předepsat (poručit) jiný způsob (například adresu datové položky

nikoli k registru DS, ale třeba CS). Způsob adresace operandů určuje (většinou) druhý bajt instrukce. Obsahuje pole režimu (mód, 2 bity), což určuje, je-li použito posunutí. Pole registr obsahuje číslo registru (malou adresu, 3bity). V poli r/m je zakódováno, jakým způsobem se vypočte efektivní adresa. Je očividné, že doba vykonání jedné a téže instrukce, závisí na způsobu výpočtu efektivní adresy.

REZERVOVANÉ OBLASTI PAMĚTI:

- některé adresové lokace paměti jsou rezervované pro speciální operace CPU. V dolní části paměti RAM adresy 0-3FFH je vyhrazená oblast pro vektory přerušení. Celkem 255 vektorů x 4 bajty = 1KB. Podobně v každé horní části od adresy FFFF=0H do konce paměti je prostor 16 bajtů pro RESET BOOTSTRAP. Klasické IBM PC má 640 KB paměti RAM. Zbytek do 1 MB je vyhrazen pro adaptery zařízení a moduly ROM (BIOS)

START MIKROPROCESORU:

- se vzestupnou hranou signálu RESET procesor ukončí svou činnost a provede:
    - vyčištění fronty instrukcí
    - vynulování registrů příznaků a DS, ES, SS a IP, do registru CS se zapíše hodnota FFFFH
    - sestupnou hranou signálu RESET mikroprocesor zahájí činnost vždy od adresy FFFFOH (CS = FFFFH a IP = 0000). Délka impulsu RESET je minimálně 4 hodinové cykly

80286

- procesor s pokročilou architekturou podporující práci ve dvou režimech
    - **reálný režim –** je plně kompatibilní s I 8086 což znamená, že je možno spouštět programy určené původně pro 8086 bez jakékoli modifikace
    - **chráněný režim –** poskytuje vlastnosti podporující zpracování více úloh najednou

- dovoluje adresovat až 16 MB reálné paměti a až 1 GB virtuální paměti (adresová sběrnice je 24 bitová)

ARCHITEKTURA:

- procesor je složen ze čtyř nezávislých paralelně pracujících jednotek:
    - **sběrnicová jednotka** – zajišťuje přístup k reálné paměti – vybírá data, která ukládá do fronty dlouhé šest slabik (BU)
    - **instrukční jednotka** – dekóduje vybrané instrukce a ukládá je do fronty (IU)
    - **prováděcí jednotka** – realizuje dekódované instrukce (EU)
    - **adresová jednotka** – je správcem paměti. Zajišťuje ochranu paměti a přepočítává virtuální adresy na reálné

REGISTRY

8086

- ***Registry pro všeobecné použití (skupina HL)***
  - 4 registry, každý adresovatelný jako jeden 16bitový nebo dva nezávislé 8bitové

AH AL střadač
BH BL báze
CH CL čítač
DH DL data

  - Tyto registry se používají převážně pro aritmetické operace. Speciálně je pro ně určen registr AX (akumulátor), řada instrukcí pracuje převážně jen s tímto registrem. BX se používá především k výpočtu adresy v některých adresových módech. CX je čítač, používá se u instrukcí s opakováním. DX je datový registr a nemá speciální funkci.

- ***Skupina ukazatelů a indexových registrů (skupina PI)***
  - mohou se podílet na většině aritmetických a logických operací. Všechny registry vyhovují pojmu střadač

SP ukazatel zásobníku
BP ukazatel báze
SI index zdrojové adresy
DI index cílové adresy

  - Tyto registry se nejčastěji používají pro adresaci dat. Indexové registry mají specifikovaný způsob využití. SP (ukazatel zásobníku) obsahuje hodnotu offsetu zásobníku mikroprocesoru a používá se při práci se zásobníkem a voláním podprogramu. BP je určen k adresování dat hlavní paměti. Při spolupráci s vyšším programovacím jazykem je využíván jako ukazatel na parametry volané procedury. SI a DI se využívají při přenosech boku dat a při přístupu do paměti.

- ***Segmentové registry***

CS segmentový registr programu (Code Segment)
DS segmentový registr dat (Data Segment)
SS segmentový registr zásobníku (Stack Segment)
ES pomocný registr (Extra Segment)

  - CS je nejdůležitější, jelikož obsahuje segmentovou část adresy právě běžícího programu. DS je určen k adresování dat v hlavní paměti, tzn., že pokud instukce pracuje s daty v hlavní paměti, je určen právě tento registr. SS ukazuje na segment strojového zásobníku. SS a SP je přesná adresa strojového zásobníku v hlavní paměti. Je využit u instrukcí, které pracují se zásobníkem. ES je využíván při přesunech dat, jinak je volně k dispozici. IP - tento registr obsahuje offsetovou část adresy právě zpracovávané instrukce.

- ***Příznakové registry - FLAGS***
  - výkonná jednotka má 9 jednobitových indikátorů uspořádaných v 16bitovém slově

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

CF – přenos (carry)
PF – parita výsledků je sudá (parity)
AF – přenos z nižšího nibble AL registru (auxiliary)
ZF – výsledek je nulový (zero)
SF – výsledek je záporné znaménko (sign)
OF – aritmetické přetečení (overflow)
DF – autodekrementace (direction)
IF – povolení přerušení (interrupt)
TF – krokovací režim (trap)

80286

- struktura registrů 8086 zůstala zachována. Přibyl 16 bitový registr MSW (machine status word) – jsou použity pouze spodní čtyři bity – PE, MP, EM, TS. Registr příznaků byl rozšířen o bity IOPC = 13,12 : určuje úroveň oprávnění pro operace V/V a bit NT = 14 : určuje práce instrukce IRET

ADRESACE

8086

SEGMENTACE PAMĚTI

PŘÍKLADY:

DS	2	0	0	0	
SI		2	2	1	3
	2	2	2	1	3

CS	1	0	0	0	
IP		2	2	1	3
	1	2	2	1	3

SS	3	0	0	0	
SP		F	F	F	F
	3	F	F	F	F

- paměť je bajtově orientovaná – základní adresovatelnou jednotkou je jeden bajt. Slovo o délce 16 bitů lze ukládat na libovolné adresy. Uloží se tak, že nižší bajt přijde na specifikovanou adresu a vyšší bajt na následující vyšší adresu. Doporučuje se však slova přednostně ukládat na sudé adresy – čtení i zápis slova se realizuje v jednom cyklu paměti (při liché adrese je na tutéž operaci třeba dvou cyklů paměti)

- Šířka adresové sběrnice je 20 bitů (adresuje 1 MB), adresa má dvě části – segment a offset.

- Vytvořená adresa se skládá ze dvou šestnáctibitových částí. Z této tzv. logické adresy se fyzická vytváří tak, že se nejprve posune o 4 bity vlevo (odpovídá násobení 16) a k takto vzniklému číslu se přičte offset. Tím vzikne 20bitová adresa, ukazující na konkrétní místo v paměti.

- Důsledkem adresování je paměť rozdělena na jednotlivé bloky o velikosti 64 kB (segmenty). Segmentová část adresy po vynásobení šestnácti (připsání čtyř nul na nejnižší bity) ukazuje na začátek segmentu a offset je pozice v segmentu vzhledem k jeho začátku. Segmentové registry udávají segmentovou část adresy.

- ***Počet taktů potřebných při různých způsobech adresace:***
  - přímá 6
  - bázová 5
  - báze + posunutí 1
  - báze + index 7
  - báze + index + posunutí 11

ZPŮSOBY ADRESACE OPERANDŮ

1. ***Přímý (bezprostřední) operand*** – konstanta je součástí instrukce
   ```
   mov     ax, 12h
   ```

2. ***Přímá adresa*** – hodnota offsetové části adresy je součástí instrukce
   ```
   mov     ax, ADR1
   mov     ax, ds:[100]
   ```

3. ***Nepřímá adresa*** – offsetová část adresy je uložena na adrese, která je součástí instrukce
   ```
   mov     ax, [bx]
   mov     ax, [ADR1]
   ```

4. ***Bázová adresa*** – získá se jako součet přímé a nepřímé adresy v bp,bx
   mov ax, [bx]+4
   mov ax, [ADR1]+4

5. ***Indexová adresa*** – součet přímé adresy a obsahu indexového registru si,di

6. ***Kombinovaná adresa*** – báze + index
   mov ax, [bx+di]

7. ***Kombinovaná adresa*** - přímá + báze + index
   mov ax, [bx+si] + 4
   mov ax, [bx+si] + ADR1

- v případě lze pro danou instrukci zrušit implicitní přiřazení segmentového registru
  mov ax, es: [bx+si] + 8

FYZICKÁ ADRESA – 20 bitů

LOGICKÁ ADRESA – CS + IP = 32 bitů

REGISTRY – 16 bitů

OFFSET: 0 – 65.535

SEGMENT 16 – 1M

80286

- ochrana paměti pracuje pouze v chráněném režimu. Nový rys procesoru. Paměť je rozdělena na segmenty.

- segment je souvislý paměťový prostor o délce až 64 KB. Každý segment je určen těmito parametry:
  - **báze segmentu** (adresa začátku)
  - **limit segmentu** (délku ve slabikách)
  - **přístupové práva**- čtyři úrovně oprávnění což umožňuje vyjádřit množství „důvěry“ poskytnuté určitému procesoru (0-3).

- ***existují také 4 typy segmentu:***
  - pouze provádět
  - pouze číst
  - provádět a číst
  - číst i psát

INSTRUKCE

8086

OPCODE	CÍL	ZDROJ

OPCODE – co se má udělat
OPERANDY – s čím se to má udělat

ZPŮSOB ADRESACE OPERANDŮ

- - bezprostřední mov al, 12H bo 12
- - registrová mov al, bl 80 d8
- - přímá mov [550H], al a2, 5005
- - registrová nepřímá mov dl, [si] 80a14
- - bázovaná mov ax, [bx+4] 8b4704
- - indexová mov [di+8], bl 885d08
- - kombinovaná – báze + index mov [bp+si], ab 8822
- - kombinovaná – báze + index + posunutí mov cl, [bx+di+2] 8a4902

ROZDĚLENÍ INSTRUKCÍ PODLE ÚČELU:

- instrukce pro manipulaci s daty
- aritmetické instrukce
- instrukce pro manipulaci s bity
- instrukce pro řízení chodu programu
- řídící instrukce pro procesor

INSTRUKCE PRO MANIPULACI S DATY:

- pro přesun dat
- pro práci s řetězci dat
- pro práci se zásobníkem
- pro obsluhu vstupu a výstupu

- s výjimkou instrukcí pro práci s řetězci a v/v je možno použít libovolný způsob operace

INSTRUKCE PŘESUNU DAT

PRO VŠEOBECNÉ POUŽITÍ:

1. instrukce přesunu dat — mov — cíl — zdroj
2. instrukce uložení do zásobníku — push — zdroj
3. instrukce vyjmutí ze zásobníku — pop — cíl
4. instrukce záměny — xchg — cíl — zdroj
5. instrukce transformace podle tabulky — xlat — (al = as: [bx+al])

PRO PRÁCI S ADRESAMI:

6. uložení offsetové části adresy — lea — cíl — zdroj
7. naplnění adresou – segment ds — lds — cíl — zdroj
8. naplnění adresou – segment es — les — cíl — zdroj

PRO PRÁCI S REGISTREM PŘÍZNAKU:

9.
10.
11.
12.

PRO OPERACE VSTUPU A VÝSTUPU:

13. vstup dat z portu — in — cíl — zdroj
14. výstup dat na port — out — port — zdroj

INSTRUKCE PRO PRÁCI S ŘETĚZCI ZNAKŮ

REP	repeat until		cx <>0
REPE	repeat while	equal and	cx <>0
REPZ	repeat		
REPNE	repeat while	not equal and	cx <>0
REPNZ	repent		

MOVS	copy	es : di < - ds : si
MOVSB	no operands	
MOVSBV	no operands	

COMPS	compare	es : di < - ds : si
COMPSB	no operands	
COMPSV	no operands	

SCAS	compare	es : di - ax
SCASB	no operands	
SCASV	no operands	

LODS	copy	ax < - ds : si
LODSB	no operands	
LODSV	no operands	

STOS	copy	es : di < - ax
STOSB	no operands	
STOSV	no operands	

CLD, STD Set / reset direction flag (+,-si,di)

```
EXAM 1 :    cld
            mov cx, 16
            lea si, BLC1
            lea di, BLC2
      rep movs B
```

```
EXAM 2 :    cld
            xor ax, ax
            mov cx, 255
            lea di, BLC1
$1          stosB
            inc al
            loop $1
```

INSTRUKCE PRO PRÁCI S BITY

Logical	destination, source
AND	Finde the logical AND two operands 2-7
OR	Performs logical inclusive OR operation 2-7
XOR	Performs logical exclusive OR operation 2-3
NOT	Reverse each bit of byte, word 2-7
TEST	Performs logical compare operation 2-6
BSF	Bit scan forward 10+3n
BSR	Bit scan reverse 10+3n
BT,C,S,R	Bit test 3.13

Shift	destinatiion, count
SAL	Shift arithmetic left 2-5
SAR	Shift arithmetic right 2-5
SHL	Shift logical left 2-5
SHR	Shift logical right 2-5

Rotate	destination, count
ROL	rotate left 2-5
ROR	rotate right 2-5
RCL	rotate left through carry 2-5
RCR	rotate right through carry 2-5
SHLD	double precisiion shift left 3-7
SHRD	double precisiion shift right 3-7

EXAM 1 : shrd ax, bx, 12 ; double precision shift right

```
EXAM 2 :    mov cl,2
            shl ax, cl ; multiply an unsigned number by 4
            sal ax, cl ; multiply a signed number by 4
            shr ax, cl ; divide an unsigned number by 4
            sar ax, cl ; divide a signed number by 4
```

```
EXAM 3 : ; multiply by 10
            move bx, ax ; save contents of ax
            shl ax, 1 ; multiply by 2
            shl ax, 1 ; multiply by 4
            add ax, bx ; multiply by 5
            shl ax, 1 ; multiply by 10
```

INSTRUKCE ŘÍZENÍ CHODU PROGRAMU

ja	jnbe	>	if above	/ not below or equal
jae	jnb	> =	if above or aqual	/ not below
jb	jnae	<	if below	/ not above or equal
jba	jna	< =	if below or equal	/ not above
je			if carrz	
je	jz	= 0	if equal 0	
jg	jnle	> +	if greater	/ not less or equal
jge	jnl	> = +	if greater or equal	/ not less
jl	jnge	< +	if less	/ not greater or equal
jle	jng	< = +	if less or equal / not greater	
jnc			if no carry	
jne	jnz	≠ 0	if not equal 0	
jno			if not overflow	
jnp	jpo		if not parity	/ parity add
jns		+	if not sign	
jo			if overflow	
jp	jpe		if paritz	/ paritz even
js			if sign	

INSTRUKCE CYKLU

jc x 2		= 0	if cx equal 0
loop		≠	loop while cx not equal 0
loope	loopz	= 0	loop while cx equal 0
loopne	loopnz	≠ 0	loop while not equal

INSTRUKCE SKOKU

jmp	jump unconditionally

INSTRUKCE ŘÍZENÍ CHODU PROGRAMU II

INSTRUKCE VOLÁNÍ PODPROGRAMU

call	call procedure
ret	return from procedure
retn	return from near procedure
retf	return from far procedure

INSTRUKCE ŘÍZENÍ PROCESORU

INSTRUKCE PRO PRÁCI S REGISTREM

cle	clear carry
cld	clear direction (auto – increment)
cli	clear interrupt - enable flag
cmc	complement carry
stc	set carry flag
std	set direction (auto – dekrement)
sti	set interrupt – enable flag

INSTRUKCE ŘÍZENÍ STROJE

hlt	halt procesor
lock	lock the bus
wait	wait to coprocesor
esc	escape to coprocesor
nop	no operation

INSTRUKCE PŘERUŠENÍ CHODU PROGRAMU

int	call to interrupt routine
into	call to interrupt routine if overflow
iret	return from interrupt routine

PŘERUŠOVACÍ SYSTÉM

- přerušení je obecně asynchronní událost, která způsobí pozastavení probíhajícího procesu a umožní spuštění procesu jiného (proces obsluhy přerušení)

- ***zdroj signálu inicializujícího přerušení může být:***
  - vnější (třeba z periferie)
  - vnitřní (softwarové přerušení)

- po příchodu signálu přerušení je běžná sekvence instrukcí dočasně přerušena řízení je tím předáno rutině obsluhy (interrupt handler).

- obsluha přerušení vyžaduje provedení určitých operací navíc (režim přerušení), což spotřebovává jak paměť, tak čas. Nákladné je hlavně zabezpečení neměnnosti pracovního prostředí přerušeného procesu (uložení kontextu), tak, aby byla zajištěna kontinuita procesu (protože procesor 1-86 má jen jednu sadu registrů, je třeba jejich obsah uložit a před návratem k původnímu procesu opět obnovit).

- určitou dobu také trvá samotná operace přepnutí procesů (pro 1-86 od 50 až 61 hodinových cyklů). Zdrojů přerušení je většinou vícero. Každý zdroj přerušení má svou obslužnou rutinu. Adresy vstupních bodů těchto rutin se nazývají vektory přerušení a jsou uspořádány do tabulky uložení v paměti od adresy nula. Vnější přerušení z různých zdrojů mohou běžně vzniknout ve stejný časový okamžik. Pořadí důležitosti, v jakém se přerušení obsluhuje, určuje priorita přerušení a je řízena hardwarově (PIC).

HARDWAROVÉ PROVEDENÍ

- procesor 1-86 má pro příjem signálů vnějšího přerušení dva vývody.

  Vývod 17 (NMI - nemaskovatelné přerušení)

  - slouží k signalizaci fatálních událostí. Signál nelze ignorovat (zamaskovat, odstínit), protože oznamuje katastrofické stavy systému (chyba parity paměti, sběrnice), na které procesor musí okamžitě reagovat (obvykle chybová hláška a HALT)

  Vývod 18 (INTR)

  - na tento vývod je připojen programovatelný kontrolér přerušení (PIC). Má 8 vstupů a obvody lze je řadit do kaskády (běžně 2, maximálně až 64). Další přerušení realizované hardwarově slouží k ladění. Nastavením TF v registru příznaků přijde procesor do krokovacího režimu, což znamená, že po každém vykonání instrukce dojde k přerušení. Obslužná rutina aktivuje ladící systém, který stanoví, co se vykonáním instrukce změnilo (a oznámí to uživateli). Další čistě hardwarové přerušení je akce na dělení nulou a přetečení.

- tabulka rezervovaných vektorů přerušení (firma Intel) obsahuje 32 vektorů (32 rutin obsluhy související se stavem procesoru, adresy 0 až 7 FH). Nejdůležitější jsou následující: 0 – dělení nulou, 1 – krokování (trap), 2 – NMI, 3 – ladění (breakpoint), 4 – přetečení (overflow)

PŘERUŠOVACÍ SYSTÉM – POPIS FUNKCE

- při vzniku přerušení se provedou následující operace:

  - dokončí se rozpracovaná instrukce, přerušení se uplatní až po skončení instrukce. Uloží se do zásobníku registr příznaků.
  - v registru příznaků se vynulují bity IF a TF.
  - do zásobníku se uloží registr CS, který se naplní z adresy N*4+2
  - do zásobníku se uloží registr IP, který se naplní z adresy N*4

- programátor v rutině obsluhy přerušení musí uchovat obsah všech registrů, které hodlá použít (modifikovat obsah). Přerušovací systém je možno při obsluze přerušení aktivovat (instrukce STI) a dovolit přerušení obsluhy přerušení. Návrat k přerušenému procesu se provede instrukcí IRET. Ta obnoví ze zásobníku IP, CS a registr příznaků (to uvede přerušovací systém do původního stavu)

KÓD	***TYP***	***FUNKCE***
0	P	Dělení nulou
1	P	Ladění – krokování
2	P	Nemaskovatelné přerušení NMI
3	P	Ladění – breakpoint
4	P	Aritmetické přetečení
5	S	Print screen
6	P	Chybný operační kód
7	P	Koprocesor nepřítomen
8	H	IRQ 0 – časovač
9	H	IRQ 1 – klávesnice
A	H	IRQ 2 – kaskáda
B	H	IRQ 3 – sériový adaptér 2 nebo 4
C	H	IRQ 4 – sériový adaptér 1 nebo 3
D	H	IRQ 5 – tiskárna LPT2
E	H	IRQ 6 – kontrolér floppy disku
F	H	IRQ 7 – tiskárna LPT1
10	S	Služby video
11	S	Seznam zařízení
12	S	Velikost operační paměti

80286

- přidány privilegované instrukce (cca šestnáct instrukcí), některé staré byly upraveny
- ***nejvyšší priorita :***
    - LGDT, CIDT, LLDT, LTR, LMSN, CLTS, MCT

- operace vstupu a výstupu patří v chráněném módu mezi „částečně“ privilegované instrukce (smí použít „důvěryhodný“ proces – nesmí být tedy procesorem privilegovaným)

PROPOJOVACÍ SUBSYSTÉM

IDE (Integrated Drive Electronics)

- Disková jednotka se zabudovaným řadičem, dřívější jednotky bez řadiče, ten je na základní desce.
- U procesoru 8086 se přenáší po 8 bitové ISA sběrnici (40 pinový konektor)

První verze: ATA-1

- První verzi ATA vyvinuly firmy CDC, Compaq a Western Digital v osmdesátých letech. Na konci 80. let byla přijata jako ANSI standard, díky čemuž se sjednotil přístup výrobců k tomuto rozhraní. Předtím mnoho výrobců produkovalo své vlastní varianty, což pak způsobovalo problémy s kompatibilitou. Některé oblasti standardu zůstaly výrobcům poměrně otevřeny pro jejich vlastní příkazy. Pro nízkoúrovňové formátování pak bylo nutné mít program přizpůsobený pro jednotky od konkrétního výrobce.

VSTUPY A VÝSTUPY

- mikroprocesor pro styk s vnějšími zařízeními používá V/V brány adresové odděleně od hlavní paměti (isolated IO). Může komunikovat buď po 8, nebo 16 bitech. Má přístup k 64K 8bitových nebo 32K 16 bitových bran. Pro adresy bran (sudá-lichá) platí stejná doporučení a omezení jako při adresování paměti.

IA - 32

ZÁKLADNÍ CHARAKTERISTIKA (80386, 80486)

80386

- 32bitová adresová sběrnice

- procesor s 32bitovou architekturou podporující práci ve třech režimech
  - **reálný režim** – v tomto režimu je plně kompatibilní s I 8086 což znamená, že je možnou spouštět programy původně pro 8086 bez jakékoli modifikace (bez rekompilace)
  - **chráněný režim** – je pro procesor nativní a je plně slučitelný s I 80 286.
  - **virtuální režim V86** – spustitelný v chráněném módu, dovoluje provozovat několik virtuálních strojů I 8086 na jednom procesoru 80 386

- může adresovat až 4 GB reálné paměti a až 64 TB virtuální paměti.

ZÁKLADNÍ VYLEPŠENÍ:

- pracuje s 32 bitovými operandy (dvojslovo)
- segmenty mohou mít velikost až 4 GB
- má podporu pro stránkovací mechanizmus pamětí

procesor se vyráběl ve dvou variantách

- DX
- SX – liší se od DX tím, že má obvody pro styk s okolím pouze 16bitové

ARCHITEKTURA:

- procesor je složen ze 6 nezávislých paralelně pracujících jednotek:
  - **sběrnicová jednotka** – realizuje styk s okolí, organizuje činnost sběrnice
  - **jednotka předvýběru** – získává z paměti slabiky a sestavuje instrukce do fronty
  - **instrukční jednotka** – přivádí instrukce na mikroinstrukce a ty ukládá do fronty
  - **prováděcí jednotka** – zpracovává mikroinstrukce
  - **segmentační jednotka** – převádí logické adresy na lineární (případně na fyzické)
  - **stránkovací jednotka** – transformuje lineární adresu na fyzickou

REGISTRY

- *struktura registrů I 8086 zůstala v zásadě zachována.*

- 6 16 bitových segmentových registrů (CS, DS, SS, ES, FS, GS)
- registry indexové a pro všeobecné použití jsou 32 bitové a mohou být užity jako 8,16 nebo 32 bitové
- skupina registrů systémových adres (GDTR, LDTR, IDTR, TR) je stejná jako na I 80 286, jen báze byla rozšířena na 32 bitů (LDTR a TR má selektor 16 bitů)
- přibyly 32 bitové řídící registry CRO, CR1, CR2, CR3 (CRO = MSW)
- registr příznaků je 32 bitový a byl rozšířen o bity:
  - **RF = 16** – maskuje ladící přerušení
  - **VM = 17** – zapíná režim V86
- nové 32bitové ladící registry DR0, DR1, DR2, DR3, které slouží pro uložení lineárních adres ladicích bodů
- registr DR6 je stavový a DR7 příkazový
  - pro testování a ladění mechanizmu stránkování jsou určeny 32bitové registry TR6 a TR7

80486

- vyladěné jádro 386, ke kterému byla integrovaná aritmetická jednotka pro práci s čísly v plovoucí řádové čárce a interní cache o velikosti 8 kB. Tato cache je společná pro data i instrukce a může být doplněna vnější pamětí cache. Maximální pracovní frekvence byla 50 MHz. Instrukce jsou do značného stupně prováděny proudově – procesor je schopen provádět některé instrukce v jednom cyklu (především často používané instrukce). Procesor se vyráběl v několika variantách. Procesor 80 486 SX se lišil od základní varianty tím, že nemá funkční jednotku pro práci s čísly v pohyblivé řádové čárce. Dalším vylepšením bylo násobení vnitřní frekvence procesoru (over drive). Verze DX 2 pracuje vnitřně s dvojnásobkem vnější frekvence (25,33 MHz), verze DX se čtyřnásobkem. Zdvojením taktovací frekvence stoupne výkon procesoru asi o 50%. Dalším rozdílem bylo snížené napájecí napětí (3,3V)

ARCHITEKTURA:

- IA – 32. Plně dvaatřicetibitový procesor. Pro zvýšení výkonu byly implementovány některé rysy architektury RISC. Instrukční sada tímto faktem nedotčena.

REGISTRY:

- struktura registrů stejná jako u 80 386
    - registry pro všeobecné použití a indexové (EAX, EBX, ECX, EDX, ESI, EAI, ESP, EBP)
    - segmentové registry – celkem šest (CS, SS, DS, ES, FS, GS)
    - čítač instrukcí a registry příznaků (EIP, EFLAGS)
    - řídící registry (CR0, CR1, CR2, CR3 (CRO=MSW)
    - registry systémových adres (GDTR, LDTR, IDTR, TR)
    - ladící registry (DR0, DR1, DR2, DR3, DR6, DR7)
    - testovací registry – (TR3, TR4, TR5), pro testování paměti cache a pro ladění stránkování paměti (TR6 a TR7)

- registr příznaků je 32bitový a byl rozšířen o bit AC=18: zapíná generování přerušení při odkazu na objekt, který není zarovnán na příslušnou hranici (například zápis nebo čtení 16bitového slova na liché adrese)

ZÁKLADNÍ CHARAKTERISTIKA (PROCESORY PENTIUM)

- prvý typ procesoru Pentium se objevil na trhu v roce 1993. Pentia se sice vyrábí dodnes (Pentium 4), ale za obchodním názvem je schováno několik generací procesorů architektury IA-32 různých produktových řad

- základním hnacím motorem inovací není jen zvyšování výpočetního výkonu procesoru (počet operací za jednotku času), ale neméně důležitá je podpora (hardwarová) systémových funkcí výpočetního systému (funkce operačního systému, podpora grafiky, atd). Toto vyžaduje více elementů na čipu (Moore), více elementu ⇒ větší energetická náročnost ⇒ větší nároky na chlazení. Důsledkem je neustálé snižování napájecího napětí procesoru, stále tenčí propojovací vodiče (litografické čáry), časté změny v zapouzdření procesoru (počet vývodů, typ pouzdra a patice).

- dalším podstatným rysem je jistá specializace procesorů. Původně se vyráběl prostě procesor – jeho použití bylo poměrné úzce ohraničeno (pokud ho někdo chtěl použít jinak, měl smůlu – musel vzít to, co bylo na trhu). Dnes se procesory používají v nejrozmanitějších zařízeních, což klade na procesor různé, často protichůdné požadavky. Tak vznikly produktové řady – procesory pro stolní počítače, výkonné servery, mobilní zařízení. Generací procesoru se rozumí určitá typová řada, která vykazuje stejné architektonické a výkonnostní rysy. V zásadě jsou generace procesorů definovány podle procesorů vyráběných firmou Intel (funguje to tak zhruba do 6. generace)

P5

- postupně se vyrábělo ve třech verzích. Realizace litografickou technologii s tloušťkou čary 0.8,0.6 a 0.35 um. Počáteční pracovní frekvence činila 60 nebo 66 Mhz a byla shodná s frekvencí základní desky (MB). Pozdější verze pracovaly na násobku frekvence MB až do 266 Mhz. Napájení 5V a spotřeba zhruba 16 W. Později bylo napětí sníženo na 3.3 až 2.8V (což vyžadovalo vždy nový MB).

- v poslední verzi došlo k rozšíření instrukční sady o 57 nových instrukcí pro zpracování grafiky, videa a zvuku (MMX – instrukce typu SIMD – jedna instrukce zpracovává více datových položek najednou) a ke zvětšení instrukční cache na 16kB. Původní patice S4 nahrazena S5 a S7. Na MB přidán modul regulátoru napětí pro procesor (delší použitelnost určitého typu desky). Procesory této generace mohly pracovat pouze v symetrickém multiprocesingu (což znamená maximálně dva procesory na jedné základové desce).

P6

pentium pro (1995)

- bylo prvním členem této řady
- **v pouzdře procesoru jsou integrovány dvě jednotky:**

- vlastní procesor
- paměť cache L2 o velikosti 256, 512, či 1MB.

- samotný procesor obsahuje cache L1 o velikosti 16 KB. Instrukční cache je dvoucestná asociativní o velikosti 8 KB. Datová cache je realizována jako čtyřcestná asociativní taktéž 8 KB.

- základní inovací je architektura DIB – procesor má dvě nezávislé sběrnice (celková propustnost sběrnic se zvýší až 3x). Paměť cache L2 je připojena k procesoru samostatnou sběrnicí, která pracuje na plné rychlosti procesoru. Uvedené řešení umožňuje vytvoření mulitprocesorového systému až čtyřech procesorů.

- další inovací je dynamické vykonávání instrukcí. Podstatou je zpracování instrukcí spíše na základě logiky, než pouze podle seznamu. **Výsledný efekt je dosažen kombinací tří metod:**
  - predikce vícenásobného větvení
  - analýza dat
  - spekulativního vykonávání instrukcí

- šířka datové sběrnice zůstala stejná (64 bitů), adresová byla rozšířena na 36 bitů. Napájecí napětí 3.3V později 3.1V. Příkon až 25 W. Taktovací frekvence maximálně 200 Mhz. Tento procesor nepodporuje, (neumí) instrukční sadu MMX.

procesor pentium II (1997)

- je sice jen vylepšené Pentium Pro, ale způsobem zapouzdřením se jedná o zcela nový procesor (poprvé bylo užito pouzdro s hranovým kontaktním polem SECC). Vylepšením je zvětšení L1 cache na 32 KB (2x16). Vyráběl se litografickou technologii s tloušťkou čáry nejprve 0.35 později 0.25 um. To, společně se snížením napájecího napětí z 2.8 V až na 2 V, se odrazilo ve snížení spotřeby procesoru [(450 MHZ procesor (27 W) má menší spotřebu než 233 Mhz (35W)]. Maximální taktovací frekvence 450 Mhz. Procesor podporuje instrukční sadu MMX a pouze symetrický multiprocesing.

procesor pentium III (1999)

- je posledním modelem řady P6. Vylepšením bylo další rozšíření instrukční sady SSE (70 instrukcí).

- **ostatní inovace jsou spíše kosmetické :**
  - - výrobní číslo procesoru
  - - teplotní čidlo pro detekci teploty jádra
  - - ochrana proti přetaktování procesoru.

- -počáteční litografická technologie pracovala s tloušťkou čary 0.25, později se přešlo na 0.18 um. Původně byla, L2 cache o velikosti 512 KB, umístěna mimo jádro a pracovala na jeho poloviční rychlosti. Později bylo 256 KB L2 cache integrováno na čip, kde pracovala na plné rychlosti jádra. Maximální taktovací frekvence byla 1 Ghz. Podpora pouze pro symetrický multiprocesing.

procesory celeron

- představují levnější variantu Pentia II/III. Předpokládá se použití především v počítačích pro kancelářské aplikace (segment trhu low-end). V podstatě se jedná o procesor bez paměti L2 cache. Starší verze tohoto procesoru obsahují jádro Pentium II novější pak Pentium III. Aby byla cena procesoru co nejnižší, používá se i jiný způsob zapouzdření. Nejprve to bylo pouzdro SEPP, později pak PPGA a FC-PGA pro paticí S370. Procesor se postupně vylepšoval – typy s pracovní frekvencí 300 a více MHz už mají integrovánu paměť L2 cache o velikosti 128 MB. Vyráběny byly litografickou technologii 0.25 později 0.18 um. Spotřeba Celeronu je menší než odpovídajícího procesoru Pentium (= menší chladič), ostatní parametry shodné nebo mírně lepší.

procesory xeon

- představují výkonnější variantu Pentia II/III. Předpokládá se použití v grafických stanicích serverech (segment trhu high-end).

- **rozdíly jsou následující:**
  - - procesor je zapouzdřen ve zvětšeném pouzdře SEC
  - - paměť L2 cache pracuje vždy na plné rychlosti jádra a byla zvětšena až na 2 MB (proto větší pouzdro).
  - - řídící čip cache umí vypočítávat odkazy na plných 64GB systémové paměti.
  - Ostatní parametry jsou stejné či lepší.

Pentium II/III a Celeron MObile

- je určen pro použití v přenosných počítačích. Základním parametrem je spotřeba procesoru (při vyhovujícím výkonu).

- **snížení spotřeby se dosahuje kombinací několika technologií:**
  - především je to výrobní postup, litografická technologie 0.18 nebo 0.13 um a použití měděných vodičů (příklad: procesory vyrobené 0.13 um technologií mají spotřebu až o 40% nižší).
  - snížením napájecího napětí jádra až na 1.6V ve verzi Mobile a až na 1.1V ve verzi ULV.
  - použitím technologie SpeedStep, která řídí napětí a frekvenci procesoru při běhu na baterie
  - technologie QuickStart (procesor je převeden do spánku i mezi dvěma stisku klávesnice).

- příklad: LV Mobile pentium III na frekvenci 900 MHz má průměrnou spotřebu menší než 1W.

- **uvedené technologie se dále vylepšují**:
  - Enhanced SpeedsStep řídí napájecí napětí a taktovací frekvenci podle zatížení procesoru (dříve pouze ve dvou stupních)
  - pro zapouzdření se používá technologie BGA a nebo mini-cartridge (IMM – modul, který má v pouzdře procesoru integrován čip North Bridge).

- dnes se procesory pro mobilní zařízení označují jako Pentium III-M.

Pentium 4

- procesor je prvá realizace mikroarchitektury Netburst (sedmá generace procesorů Intel).

- **základními inovacemi jsou:**
  - dvoustupňová jednotka proudového zpracování instrukcí
  - nová architektura vyrovnávacích pamětí cache
  - aritmeticko-logická jednotka pro práci s celými čísly pracující na dvojnásobné rychlosti
  - vylepšená (kompaktní) jednotka pro práci s čísly v pohyblivé řádové čárce

- rozšiřující sada instrukcí (SSE-2) podporující 128bitovou aritmetiku (plovoucí i pevná řádová čárka)
- zrychlená systémová sběrnice.

- procesor je vyroben litografickou CMOS technologií s tloušťkou čáry 0.18 um. Na čipu o ploše 217 mm2 je umístěno 42 miliónů tranzistorů. Spotřeba při základní taktovací frekvenci 1.5 GHz činí 55W. Systémová sběrnice má šířku 64 bitů, pracuje na 100MHz a má propustnost 3.2 GB/sec (4 přenosy na jeden takt). Funkční jednotky procesoru pracují v různých taktovacích frekvencích (polovina, čtvrtina). Volba pracovní frekvence funkční jednotky je výsledek kompromisu mezi požadovaným výkonem jednotky a možnostmi obvodové realizace.

- **procesor se skládá ze čtyř základních funkčních částí:**

1. *vstupní část*

- zpracovává instrukce v programem předepsaném pořadí
- zajišťuje přísun instrukcí z paměti cache L2
- upravuje instrukce tak, aby byly zpracovatelné v prováděcí jednotce.

- **skládá se z:**
  - cache L1 (stopovací vyrovnávací paměť)
  - jednotky predikce skoků
  - dekódovací jednotky, která převádí instrukce IA-32 na mikroinstrukce
  - paměti ROM mikrokódu.

2. *blok pro řízení zpracování mikroinstrukcí mimo pořadí*

- **skládá se z:**
  - jednotky alokace vyrovnávacích pamětí
  - jednotky přejmenování registrů
  - jednotky plánování provádění mikroinstrukcí mimo pořadí.

3. *blok výkonných jednotek*

- **má tyto funkční jednotky:**
  - dvě celočíselné ALU
  - dvě jednotky pro práci s čísly v plovoucí řádové čárce
  - jednotky generace adres operandů a datovou cache L1.

4. *paměťový subsystém*

- cache L1 má velikost 256 KB a je společná pro data i instrukce. Je realizována jako 80cestná asociativní paměť s délkou řádku 128 bajtů. Používá strategii zpožděného zápisu.
- Špičkový výkon (při 1.5 Ghz) je 48 GB/sec. S pamětí cache L2 bezprostředně spolupracuje jednotka předvýběru, monitoruje přístupy a usiluje se doplňovat instrukce tak, aby měla optimální náskok 256 bajtů. Snaží se detekovat nezávislé programové proudy – dbá, aby v cache bylo co má být a nebylo, co se potřebovat nebude (pamatuje si historii, učí se z chyb – cache miss)

patice

- uvedením nové patice pro procesor sleduje firma Intel několik cílů. Především spolehlivější (méně poruchové) uchycení procesoru. Se zvyšování počtu vývodu na pouzdře, se zmenšují mezery mezi nimi (platová izolační destička s otvory praská), vývody jsou tenčí (ohýbají se) a (miniaturní) přitlačovací mechanizmus se láme. Dále, na procesor se nasazoval mohutný chladič a občas, na některé základové desce s nešťastným rozmístěním součástek v okolí procesoru, nešel nasadit. Napájení procesoru bylo vedeno ze „severu“ a zemnění z „jihu“ což

zhoršovalo distribuci napájení v čipu a komplikovalo návrh základové desky. Nová patice Socket T (LGA 775) je realizována jako pole se 775 kontakty. Procesor nemá vývody (nožičky), ale kontaktní plošky a na kontakty patice je přitlačován masivní kovovou sponou. Celé upevnění je podstatně robustnější (stabilnější), místo kolem procesoru je přesně definováno, takže nejsou problémy s chladičem. Napájecích kontaktů je 250 a zemních 274 (Socket 478 ⇒85 napájecích a 180 zemnících). Napájení je vedeno ze „severu a východu“. Vývoj trval 2 roky.

OPERAČNÍ MÓDY

- procesor s 32bitovou architekturou podporující práci ve třech režimech
    - **reálný režim** – v tomto režimu je plně kompatibilní s I 8086 což znamená, že je možnou spouštět programy původně pro 8086 bez jakékoli modifikace (bez rekompilace)
    - hodnota segmentového registru se násobí šestnácti (posune o čtyři binární řády). Fyzická adresa je dána jako součet takto získaného čísla a ofsetu.
    - **chráněný režim** – je pro procesor nativní a je plně slučitelný s I 80 286.
    - hodnota segmentového registru je jen ukazatel (selektor) do tabulky popisovačů objektů (deskriptorů). V této tabulce je (mimo jiné) uložena počátečná adresu bloku v paměti (x bajtů). Fyzická adresa je stejně jako v předešlém případě dána jako součet takto získaného čísla a ofsetu.
    - **virtuální režim V86** – umožňuje v rámci chráněného módu spouštět programy určené původně pro procesor 8086 bez nutnosti změn v programovém kódu (rekompilace). Z pohledu uživatele režim V86 znamená provozovat jeden nebo i více procesorů 8086 na jednom procesoru 80386 (nebo vyšším). Ochranné mechanizmy chráněného režimu do probíhajícího procesu 8086 nezasahují, hlídají však akce, které mohou ovlivnit ostatní probíhající procesy. Přístup na zařízení V/V je omezeno mapou povolených portů. Je-li příslušný bit nastaven, není přístup povolen. Přerušení je obsluhováno přepnutím do chráněného režimu podle příslušného popisovače. (Režim V86 je známé DOSovské okno v systému Windows)

ADRESACE

- mikroprocesory Intel bez ohledu na typ procesoru používají pro dostup k paměti adresu složenou ze dvou částí (segment a ofset). Kombinací obou složek dostaneme fyzickou adresu adresovatelné jednotky v lineárním paměťovém prostoru. Fyzická adresa je binární číslo použité na adresových linkách rozhraní CPU – paměť. Jinak řečeno, procesor vidí paměť rozdělenou na bloky (segmenty), paměťová jednotka jako lineární prostor (flat model). Dělení paměti procesorem na bloky není samoúčelné, protože podporuje programátorský pohled na operační paměť (kód, data přemístitelnost programu v paměti). Hodnota ofsetové části se interpretuje vždy stejně – je to vzdálenost od začátku bloku paměti, který je jistým způsobem definován segmentovou částí adresy. **Existují dva mechanizmy, které převádějí segmentovou část adresy na binární číslo, udávající začátek bloku v paměti stroje:**

reálný režim

- hodnota segmentového registru se násobí šestnácti (posune o čtyři binární řády). Fyzická adresa je dána jako součet takto získaného čísla a ofsetu.

chráněný režim

- hodnota segmentového registru je jen ukazatel (selektor) do tabulky popisovačů objektů (deskriptorů). V této tabulce je (mimo jiné) uložena počátečná adresu bloku v paměti (x bajtů). Fyzická adresa je stejně jako v předešlém případě dána jako součet takto získaného čísla a ofsetu.

- důležité je uvědomit si spojitost mezi šířkou adresové sběrnice (určuje maximální velikost operační paměti) a šířkou pracovních registrů (programátorovi viditelné i neviditelné části). Počet binárních řádů čísla udávajícího konečnou fyzickou adresu nemůže být větší než šířka adresové sběrnice paměti.

formát ukazatele

- segmentový registr má šířku šestnáct bitů:
  - bity 0 a 1 – úroveň oprávnění (RPL requester´s prvilege level)
  - bit 2 – volba tabulky popisovačů (TI table indicator)
  - bity 3 - 15 – index do tabulky popisovačů objektů

- pozor skutečná šířka segmentových registrů je 64 bitová

- objekty umístěné v operační paměti dělíme do dvou tříd: systém a segment

existují tři typy tabulek popisovačů objektů:

- globální (GDT)
- lokální (LDT)
- přerušení (IDT)

globální tabulka

- je unikátní a mohou v ní být umístěny objekty libovolné třídy, které mají všeobecnou platnost. Počátek této tabulky a její limit v paměti je dám obsahem systémového registru GDTR.

lokální tabulka

- lokálních tabulek popisovačů objektů může být vícero a mohou obsahovat pouze objekty mající místní (task) platnost. Pro výběr aktivní tabulky slouží systémový registr LDTR, který udává svou viditelnou částí 16bitový ukazatel a neviditelnou počátek a limit.

tabulka pro přerušení

- je také unikátní a může obsahovat pouze objekty třídy systém obsluhující přerušení. Začátek tabulky a její limit udává systémový registr IDTR.

- tabulka popisovačů objektů obsahuje jednotlivé položky. Každá položka je dlouhá osm bajtů. Obsahuje čtyři pole různé délky

- tabulka popisovačů objektů obsahuje jednotlivé položky. Každá položka je dlouhá osm bajtů. Obsahuje čtyři pole různé délky

Báze bity 24-31	Atributy 16 bitů	Báze bity 0-23	Limit16 bitů

0

bity pole atributů mají následující význam:

G	DB	O	Av1	Li9	Li8	Li7	Li6	P	DPL	DPL	DT	CD	EC	WR	A
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

G – Granulita – limit je v bajtech nebo stránky 4kB
Db – default 16/32
Av1 – volný bit pro použití návrháři systému
P – objekt existuje v paměti
DPL – přístupová práva popisovače
DT – typ popisovače systém/segment Dt = 1 systém
CD – typ segmentu kód/data Cd = 1 kód
EC a WR – interpretují se podle toho, jedná-li se o kód nebo data
A – indikuje přístup (použitím popisovače). Používá systém k sledování aktivity

- je-li nataven bit 4 prvého bajtu atributů popisovače, je definován systémový objekt. Typ objektu určují spodní čtyři bity tohoto bajtu. **Třída obsahuje celkem dvanáct objektů:**

0	Nepovolená hodnota	
1	TSS neaktivního procesu	(80286)
2	LDT	
3	TSS aktivního procesu	(80286)
4	Brána pro předání řízení	(80286)
5	Brána zpřístupňující TSS	(80286)
6	Brána pro maskovatelné přerušení	(80286)
7	Brána pro nemaskovatelné přerušení	(80286)
8	Nepovolená hodnota	
9	TSS neaktivního procesu	(80386)
A	Nepovolená hodnota	
B	TSS aktivního procesu	(80386)

C	Brána pro předání řízení	(80386)
D	Nepovolená hodnota	
E	Brána pro maskovatelné přerušení	(80386)
F	Brána pro nemaskovatelné přerušení	(80386)

režim virtuální 8086

- umožňuje v rámci chráněného módu spouštět programy určené původně pro procesor 8086 bez nutnosti změn v programovém kódu (rekompilace). Z pohledu uživatele režim V86 znamená provozovat jeden nebo i více procesorů 8086 na jednom procesoru 80386 (nebo vyšším). Ochranné mechanizmy chráněného režimu do probíhajícího procesu 8086 nezasahují, hlídají však akce, které mohou ovlivnit ostatní probíhající procesy. Přístup na zařízení V/V je omezeno mapou povolených portů. Je-li příslušný bit nastaven, není přístup povolen. Přerušení je obsluhováno přepnutím do chráněného režimu podle příslušného popisovače. (Režim V86 je známé DOSovské okno v systému Windows)

- níže uvedený postup je spuštěn v okamžik, kdy dojde ke změně obsahu libovolného segmentového registru (velmi zjednodušeno)

1. vložená hodnota je interpretována jako ukazatel (selektor) do tabulky popisovačů objektů. **Má tři logické části:** přístupová práva, typ tabulky a index

2. bit 2 (T1 – Table indicator) určuje, zda bude použití globální (=0) nebo lokální (=1) tabulka. Index (horních 13 bitů) ukazuje na konkrétní položku v tabulce

3. z příslušného registru (GDTR, IDTR, LDTR) zjistí procesor, kde je tabulka v paměti uložena – bázi a limit (tabulky jsou vždy rezidentní)

4. každá položka tabulky specifikuje příslušný objekt, je dlouhá 8 bajtů a **obsahuje tři logické části:**bázi (32 bitů), limit (20 bitů) a atributy (vlastnosti) objektu (12 bitů)

5. procesor zkontroluje bit 7 prvého bajtu atributů. Je-li nastaven (=1) je položka tabulky platná. Pokud položka není platná (=0) dojde k přerušení (11 nebo 12)

6. procesor zkontroluje bit 4 prvého bajtu atributů a určí, o jaký typ objektu se jedná. Je-li bit nulový, jedná se objekt typu systém, je-li nastaven, pak je to typ segment.

7. další postup závisí na typu objektu. Jedná-li se o objekt systém, pak spodní čtyři bity prvého bajtu atributů určují konkrétně, o jaký druh systémového objektu se jedná (TSS, brána). V případě objektu segment pak tyto bity určují způsob manipulace s objektem (pouze vykonat, číst nebo vykonat, pouze číst, číst i psát)

8. zkontrolují se přístupová práva – bity 0 a 1 selektoru a bity 5 a 6 prvého bajtu atributů. V případě selhání je generováno příslušné přerušení (13 – general protection)

9. položka z tabulky popisovačů je zanesena do neviditelné části segmentového registru

doplnění:

- Maximální počet položek v tabulce popisovačů je 8.192
- Nulový registr v GDT (index = 0, T1 = 0) je neplatný selektor a má speciální význam
- Globální a tabulka pro přerušení je v systému unikátní (vždy pouze jedna a jedna)
- Lokálních tabulek popisovačů může být vícero (mnoho)

- GDTR a IDTR je realizován jako 48 bitový registr a obsahuje bázi (32 bitů) a limit (16 bitů) tabulky
- LDTR má viditelnou 16bitovou část – selektor (do GDT) a neviditelný deskriptor

Stránkování

- mechanizmus stránkovací paměti dovoluje programům užívat virtuální paměť, která je větší než skutečná fyzicky přítomná operační paměť.

- virtuální paměť je rozdělena na stránky délky (4kB, 4MB, 2MB), které jsou uloženy na externím zařízení

- fyzicky přítomná paměť je rozdělena na rámce konstantní délky (stejné jako stránky)

  - operační systém zajišťuje, aby programem potřebná stránka byla natažena z externího média do rámce v operační paměti. Procesor poskytuje tomuto procesu potřebnou hardwarovou podporu (stránkovací jednotka procesoru). Především se jedná o podporu přepočtu lineární (virtuální) adresy na fyzickou a signalizace stavu při nepřítomnosti potřebné stránky v paměti (přerušení #PF – page fault). Transformace virtuální adresy na fyzickou se provádí při každém přístupu do paměti.

  - způsobu provedení transformace existuje více, vždy se jedná o časově náročnou operaci. Procesor má proto implementovanou vyrovnávací paměť (asociativního typu) pro posledně transformované adresy (TLB – Translation Lokk-aside Buffer).

- **stránkovací jednotka procesoru užívá následující registry:**
  - CR3 – obsahuje fyzickou adresu adresáře stránek
  - CR2 – v tomto registru je v případě výpadku stránky její lineární adresa
  - CR0 – v tomto registru se zapíná stránkování nastavením bitu 31

  - velikost stránky (rámce) je standardně 4k. Na procesorech Pentium je možno použít i stránky o velikosti 2 a 4 MB (zde je situace komplikovanější, protože od procesoru Pentium Pro se používá 36bitová adresová sběrnice ⇒operační paměť až 64 Gb). Velikost stránky se ovládá bity 4 a 5 v registru CR4 (bity PSE a PAE)

Přepočet adresy se provádí následujícím algoritmem:

  - horních 20 bitů se použije jako ukazatel do adresáře stránek. Počáteční adresa adresáře stránek je uložena v registru CR3 (musí být vždy dělitelná 4k). Adresář stránek je struktura 1024 čtyřbajtových položek. Horních 20 bitů je ukazatel do tabulky stránek, spodních 12 pak atributy. Důležitý je především bit 0, který určuje platnost položky. Je-li nenastaven, pak požadovaná stránka s tabulkou stránek není v paměti. Vznikne přerušení (#PF) a rutina obsluhy se postará o natažení stránky do paměti. Tabulka stránek má 1024 čtyřbajtových položek. Horních 20 bitů je finální adresa a spodních 12 atributy. Bit 0 opět určuje platnost položky. Neplatnost znamená, že stránka není v paměti a horních 31 bitů položky určuje místo uložení stránky na vnějším nosiči (většinou). Přepočet adresy vyžaduje 2x číst z paměti – zrychluje TLB

SKUPINY REGISTRŮ

- Procesor 8086 měl pouze 16bitové registry, tzn. neměl jejich 32bitová rozšíření charakterizovaná předponou E. Tato 32bitová rozšíření se objevila až u procesorů 80386.

Univerzální

- Tyto registry jsou 32bitové s možností přístupu buď k celému registru, nebo k jeho spodním 16 bitům, nebo k vyššímu a nižšímu bytu spodních šestnácti bitů. K horním 16 bitům registru přistupovat nejde.

- Například k registru EAX lze přistupovat 32bitově (přístup k celému registru), nebo 16bitově k jeho spodní polovině (tato část se nazývá registr AX). Vyšších 8 bitů registru AX se nazývá registr AH (chová se jako osmibitový registr), jeho nižších 8 bitů se nazývá AL. Obdobně se chovají všechny ostatní univerzální registry (EBX - BX - BH - BL, ...).

- Univerzální registry může programátor využít jakkoli, zároveň má každý z nich nějakou zvláštní funkci:
    - EAX - akumulátor (řada instrukcí ho má jako implicitní operand)
    - EBX - bázový registr (tj. dá se využít pro adresaci)
    - ECX - čítač (tj. určený pro počítání cyklů)
    - EDX - rozšíření akumulátoru

Indexové

- Tyto registry slouží primárně pro adresaci v paměti. Jsou 32bitové, lze však samostatně přistupovat k jejich spodním 16 bitům (ty tvoří 16bitové registry). Např. spodních 16 bitů registru ESI se nazývá registr SI.
    - ESI - source index - index pro zdroj (tj. pro čtení)
    - EDI - destination index - index pro cíl (tj. pro zápis)
    - EBP - určen jako ukazatel na záznam aktivní procedury na zásobníku (tím, že se implicitně spojoval s SS)
    - ESP - ukazatel vrcholu zásobníku
    - EIP - ukazatel kódu následující instrukce. Nelze k němu přímo přistupovat (jen pomocí instrukcí skoků)

Segmentové

- Slouží k ukládání adresy segmentu – pomocí nich se adresuje paměť. Jejich viditelná (přístupná) část je pouze 16bitová.
    - CS - segment kódu
    - DS - datový segment
    - ES - extra segment
    - SS - zásobníkový (stack) segment
    - FS a GS - přibyly u novějších procesorů. Nemají zvláštní název (písmena byla vybrána podle abecedy pro doplnění řady CS, DS, ES, FS, GS)

EFLAGS (registr příznaků)

- Je 32bitový, jeho spodních 16 bitů se nazývá FLAGS. Ukládají se do něj informace o stavu procesoru, úspěšnosti provedených instrukcí, atd.

řídící registry

- CR0 – v tomto registru se zapíná stránkování nastavením bitu 31
- CR2 – v tomto registru je v případě výpadku stránky její lineární adresa
- CR3 – obsahuje fyzickou adresu adresáře stránek
- CR4 – určuje velikost stránky

registry systémových adres

- GDTR – 48bitů, ukazatel do globální tabulky popisovačů
- LDTR – 16bitů (viditelných), ukazatel do lokální tabulky popisovačů
- IDTR – 48bitů, ukazatel do tabulky přerušení

ladící registry

- DR0
- DR1
- DR2
- DR3
- DR6
- DR7

testovací registry

- TR3
- TR4
- TR5
- TR6 a TR7 - pro testování paměti cache a pro ladění stránkování paměti

PŘERUŠOVACÍ SYSTÉM

- dojde-li v systému k události, která vyžaduje neodkladnou obsluhu, hovoříme o přerušení. Přesněji, jedná se o přerušení nebo obsluhu výjimek. Přerušením rozumíme žádost o obsluhu, generovanou jak hardware (signálem) tak i programovými prostředky (INT X). Výjimečná situace (výjimka) nastává, detekuje- li procesor nějaký chybový stav (na Pentiu je situace složitější: APIC).

- **Existují výjimky tří typů:**
  - fault
  - trap
  - abort

- prvé dvě výjimky lze lokalizovat, mohou být proto ošetřeny a vykonání úlohy může pokračovat. Třetí typ je fatální. Ošetření chybového stavu není proveditelné – úloha je ukončena.

- v reálném módu v obsluze není změn. **V chráněném módu lze obsluhu přerušení a výjimek realizovat buď jako:**

proceduru (interrupt handler)

- je jednodušší
- do zásobníku se uloží registr příznaků, návratová adresa a předá se řízení do segmentu obsluhy přerušení.

úlohu

- probíhá zcela stejně jako jakákoli jiná úloha v systému – vykonávání stávající úlohy je pozastaveno (suspend, uložení kontextu) a dojde k přepnutí na úlohy obsluhy příslušného přerušení nebo výjimky

mechanizmus obsluhy je následující:

- IDTR (registr) ukazuje na tabulku popisovačů přerušení IDT (cosi jako tabulku vektorů přerušení). V tabulce mohou být umístěny pouze objekty třídy systém, konkrétně pouze brány pro předání řízení (objekt číslo 5, 6, 7).

tabulka nejdůležitějších přerušení:

Vector no.	mnemotic	description	type	Error code	Source
0	#DE	Divide error	fault	No	DIV and DIV instructions
1	#DB	reserved	Fult trap	no	For intel use only
2	-	NMI interrupt	interrupt	no	
3	#BP	breakpoitn	trap	No	
4	#OF	overflow	trap	No	
5	#BR	Bround range exceeded	fault	No	
6	#UD	Invalid opcode (undefined opcode)	fault	No	
7	#NM	Device not available (no math coprocessor)	fault	No	
8	#DF	Double fault	abort	Yes (zero)	
9		Coprocessor segment overrun (reserved)	fault	No	
10	#TS	Invalid TSS	fault	yes	

výjimečná situace typu:

- fault – řízení je vráceno na instrukci, která stav způsobila
- trap – řízení je vráceno za instrukcí, která stav způsobila
- abort – většinou nelze jednoznačně identifikovat, kde událost vznikla

SYSTÉM OCHRAN

- procesoru spuštěnému v chráněném režimu je v závislosti na stupni důvěry přidělena určitá úroveň oprávnění (Privilege Level). Architektura IA-32 umožňuje používat až čtyři úrovně oprávnění. Platí čím vyšší číselná hodnota tím menší oprávnění. **Možné rozdělení:**

- úroveň 0 jádro OS – řízení procesoru
- úroveň 1 služby OS – přidělování prostředků, plánování úloh
- úroveň 2 systémové programy z knihoven, souborový systém
- úroveň 3 uživatelské aplikace

- každá úloha je sestavena z kódu a dat, které jsou uloženy v příslušném segmentu. Každému segmentu je přiřazena v závislosti na jeho obsahu jistá úroveň oprávnění uvedená v popisovači příslušného segmentu. **Existují následující indikátory oprávnění:**

- DPL – přístupová práva popisovače segmentu (Descriptor Privilege Level)
- CPL – úroveň oprávnění procesu – selektor CS (Current Privilege Level)
- RPL – požadovaná přístupová práva – selektor segmentu (Requested Privilege Level)
- EPL – efektivní přístupová práva – numerické maximum CPL a RPL

- datový segment bude procesoru zpřístupněn, platí-li Max (CPL, RPL) < DPL
- proces smí předat řízení do segmentu (kódu) pouze platí-li: CPL=DPL
- pomocí bány (call gate) je možné předat řízení do segmentu s vyšším oprávněním, do segmentu s nižším oprávněním nelze předat řízení nikdy!!!!!!!!!!

brána (gate) je objekt uložený v tabulce popisovačů, který může plnit čtyři funkce:

- předání řízení do segmentu s vyšším oprávněním (Call Gate)
- předání řízení pro nemaskovatelné přerušení (Trap Gate)
- předání řízení při maskovatelném přerušení (Interrupt Gate)
- zpřístupnění segmentu stavu procesu (Task Gate)

- řízení můžeme předat instrukcemi JMP, CALL a RETF (vzdálené (FAR) varianty uvedených instrukcí, kdy se změní jak segmentová tak i ofsetová část adresy!!!!)

- každý proces má přidělena určitá přístupová práva, která charakterizují jeho důvěryhodnost (spolehlivost). **Přístupová práva jsou definována jednak v:**

- segmentovém registru kódu – CPL
- selektoru popisovače – RPL
- deskriptoru objektu – DPL

- v případě segmentu kódu je důležitý ještě bit 2 prvého bajtu atributů v DPL (bit C – conforming), který určuje, zda se segment dokáže či nedokáže přizpůsobit změně priority

mohou nastat následující případy:

1. Je přistupováno k datům. Přístup je povolen pouze tehdy, když DPL je numericky větší nebo rovný maximální hodnotě z CPL, RPL. V opačném případě dojde k přerušení (#GP).

2. Řízení je přímo předáno do jiné části operační paměti (JMP, CALL, RET instrukce typu FAR). Pokud má CPL numericky stejnou nebo větší hodnotu jako DPL a bit C je nastaven (přizpůsobivý), není co řešit (volající má stejnou nebo menší prioritu než volaný proces). Jinak dojde k přerušení (#GP). Hodnota RPL se v tomto případě neprověřuje.

3. Řízení je přímo předáno do jiné části operační paměti, která je označena atributem nepřizpůsobivý. Akce se uskuteční pouze v případě, že CPL = DPL. Hodnota RPL má limitující účinek – musí být numericky menší nebo rovna CPL, aby byla akce proveditelná.

v chráněném módu jsou prováděny následující kategorie kontrol činnosti:

- limit objektu
- typ objektu
- úroveň oprávnění
- omezení pro adresovatelné oblasti
- omezení pro vstupní body procedur
- omezení v použití instrukčního souboru

přehled některých polí a atributů použitých při kontrolách:

platnost popisovače	bit 7	1 bajt atributů
třída popisovače	bit 4	1 bajt atributů
typ objektu	bit 0-3	1 bajt atributů
limit	bity 0-15 a 16-19	1,2,7 bajt
zrnitost adresace	bit 7	2 bajt atributů
DPL	bit 5 a 6	1 bajt atributů
RPL	bit 0 a 1	ukazatel na popisovač
CPL	bit 0 a 1	registr CS

kontroly pro adresovatelné oblasti:

- zapisuje se pro popisovač do správného registru – segment definuje data nebo kód, povoleno psát nebo jen číst, povoleno vykonání nebo i číst, kód je přizpůsobivý nebo nepřizpůsobivý. Pokus použít nulový ukazatel pro registr CS nebo SS vyvolá #GP, s registry DS, ES, FS, GS lze operaci provést, ale použití registru vede k #GP. Kontroly pro vstupní body procedur: řízení je možno předat pouze do segmentu kódu nebo na objekt typu GATE či TSS a to instrukcemi JMP, CALL, SYSENTER a SYSEXIT, (předpokládají se FAR varianty instrukcí JMP a CALL). Při předávání řízení v rámci segmentu (NEAR varianta) se žádné kontroly neprovádějí. Kontroluje se taktéž nastavení bitu C (bit 2 prvý bajt atributů popisovače). V případě nastavení C=1 pracuje volaná procedura na úrovni oprávnění volajícího (přizpůsobivá procedura CONFORMING)

PŘEPÍNÁNÍ ÚLOH

- chráněný režim umožňuje, aby bylo zavedeno do paměti několik úkolů, z niž právě jeden je aktivní. **Jako úkol můžeme spustit:**
  - program (službu OS)
  - obsluhu přerušení
  - výjimečné situace.

- předávání řízení mezi jednotlivými úlohami zajišťuje mechanizmus přepínání úloh (zajišťuje hardware!!). **Každá úloha je složena ze dvou logických částí:**
  - výkonného prostoru (segmenty kódu a dat)
  - segmentu stavu úlohy (TSS Task-state Segment)

- úloha je jednoznačně identifikována ukazatelem na její TSS.

- velikost TSS pro 32bitové prostředí je minimálně 103 bajty (67h) a slouží k uchování kontextu (stavy prostředí) úlohy v momentě přepnutí. Pro 16bitové prostředí je to 42 bajtů. Přepnutí úlohy lze realizovat některým ze způsobů předání řízení.

přímé přepnutí

- znamená bezprostřední aktivaci objektu TSS (instrukce FAR, JMP, CALL – ukazatel na popisovač TSS). Popisovač TSS musí být uložen v GDT.

nepřímé (chráněné) přepnutí

- předpokládá použití brány (TG – task gate). Popisovač brány (TG) může být uložen GDT, LDT i IDT a má sice standardní formát popisovače (8 bajtů), ale významné jsou jen bajty tři. Spodní dva bajty báze se interpretují jako ukazatel na popisovač TSS úlohy a bity spodního bajtu atributů udávají třídu objektu (TG = 5), přístupová práva a platnost položky. Důvodem použití TG je umožnit úlohám selektivní přístup k určitým procesům (třeba službám OS). Architektura IA-32 neumožňuje rekurzivní volání úloh (TSS úlohy je unikátní)

tss (task state segment)

– je struktura, která slouží k uložení stavu (kontextu) procesu v momentě přepnutí.

Data zde uložená dělíme na:

- statické (nastavují se při inicializaci procesu)
- dynamické (mění se při přepnutí)

logické členění rozeznává čtyři skupiny:

- zpětný ukazatel je selektor přerušeného procesu
- ukazatele zásobníků úrovní 2 až 0 (SS, SP)
- registry procesoru
- selektor LDT (selektor položky GDT specifikuje LDT procesu)

brána

- systémový objekt, který umožňuje nepřímé, chráněné předání řízení mezi procesy. Zabezpečuje předání řízení mezi procesy s různou prioritou nebo umožňuje předání řízení mezi procesy, které jsou realizovány v různém

programovém kódu (16 bitový a 32 bitový). Používají se také k předání řízení při přerušení, avšak v tomto případě se nepředávají žádné parametry (trap a interrupt gate). **Popisovač brány plní následující funkce:**

- specifikuje platnost popisovače
- definuje vstupní bod procesu (selektor segmentu kódu a ofset), kterému bude předáno řízení
- určuje prioritu tohoto procesu
- dochází-li k přepnutí zásobníků, určuje kolik parametrů (slov) a jaké šířky bude mezi zásobníky kopírováno (určuje to typ brány a D bit popisovače CS volajícího procesu)

- k přepnutí zásobníku dojde vždy, když se předává řízení nepřizpůsobivému procesu s vyšší prioritou. Přepíná se na nový, dočasný zásobník!!! Hlavním důvodem je snaha zabránit havárii procesu s vyšší prioritou v důsledku podtečení zásobníku a také eliminovat jakýkoli úmyslný či neúmyslný přístup procesu s nižší prioritou k datům procesu s vyšší prioritou (přes sdílený zásobník). Tedy procesy bezezbytku odizolovat.
- A také eliminovat jakýkoli úmyslný či neúmyslný přístup procesu s nižší prioritou k datům procesu s vyšší prioritou (přes sdílený zásobník). Tedy procesy bezezbytku odizolovat.

INSTRUKČNÍ SOUBOR

- instrukční soubor procesorů Intel prošel významným vývojem. Základem IA-32 je procesor I-80386 jehož instrukční sada má 128 instrukcí rozdělitelných do patnácti skupin

No.	název skupiny	počet	No.	název skupiny	počet
1	přesun dat	7	1	práce s bity	6
2	konverze	6	2	řízení procesoru	4
3	operace vstupu a výstupu	2	3	řízení chodu programu	26
4	práce s adresami	6	4	obsluha přerušení	5
5	aritmetické operace	18	5	podpora vyšších jazyků	3
6	práce s řetězci znaků	10	6	chráněný mód	14
7	logické operace	5	7	práce s příznaky	11
8	posuny a rotace	5	8	celkem instrukcí	128

- s procesorem I-80486 přibylo dalších 7 instrukcí (řízení cache, konverze malý/velký indián).

- počáteční verze Pentia přidaly dalších 5 instrukcí.
- podstatná je instrukce CPUID a RDTSC (práce s 64bitovým registrem „časového razítka“).

- pentium Pro má sadu rozšířenou o 5 instrukcí (CMOV,FCMOV – problém řídící závislosti)

- pentium II přidalo další 4 instrukce
- zajímavé jsou instrukce SYSENTER a SYSEXIT pro zrychlení přístupu ke službám operačního systému.

- pentium III rozšířilo sadu o 9 instrukcí (převážně pro práci s FPU)

- pentium 4 – 8 instrukcí
- zajímavé jsou prefixy pro řízení predikce skoků

- toto je přehled víceméně drobných změn. Podstatné změny ve struktuře instrukční sady:
- rozšíření MMX (47) od Pentia P55C
- SSE (70 instrukcí pro práci se 128bitovými položkami, přidáno osm registrů XMM) od Pentia III
- SSE2 (144 instrukcí, P4 130nm) a SSE3 (13 instrukcí, P4 90nm)

- určeno převážně pro multimédia

- o složitosti architektury soudobého procesoru svědčí instrukce CPUID. Je určena k získání informací o procesoru jak všeobecných (sériové číslo procesoru, typ, šarže) tak i o struktuře a organizaci (cache- velikost a organizace, TLB a stránkování).

- další informace poskytuje SMM (system management mode) – správa napájení a teplotních poměrů

PAMĚŤOVÝ SUBSYSTÉM

- cache – rychlá, malá paměť vložená mezi procesor a hlavní paměť s cílem zkrátit efektivní vybavovací dobu hlavní paměti. Většinou se používá dvou a více úrovňový systém. Důvodem pro existenci tohoto typu paměti je naše (momentální) neschopnost vyrobit dostatečně velkou a rychlou paměť za přijatelnou cenu. Tento stav je dočasný. Intel používá cache od procesoru I-80486. Délka bloku byla 16 bajtů (486), 32 bajtů (pentium P6), 128 bajtů (pentium 4) a 64 bajtu (pentium M). Většinou se používá 4cestná asociativní paměť pro kód a 2cestnou asociativní pro data (L1,P6), 8cestná je použita v L.2 cache Pentia 4. Plně asociativní mapování se používá pouze pro TLB (stránkování). Pro výměnu bloku se výhradně užívá algoritmus LRU. Obecně platí, že L1 cache je dělená (split cache) na kód a data zatím co L2 cache je společná (unified cache). Do Pentia Pro se jedná o blokující paměť (po stavu MISS se zablokuje) následné procesory už mají neblokující se (má zásadní význam při zpracování instrukcí mimo pořadí (out-of-order))

- operační paměť je většinou příliš pomalá, aby stačila svou rychlostí procesoru. Tak vzniklo trojúrovňové schéma paměťového subsystému. Mezi procesor a hlavní paměť je vložena další úroveň paměti – rychlá (a drahá), ale co do objemu malá vyrovnávací paměť (cache). Strategie přesouvání dat mezi těmito paměťmi je založena na skutečnosti, že přístupy do operační paměti mají jistou časovou a místní lokalitu. Přístup k datovým položkám ale vykazuje poněkud odlišné závislosti než přístup k instrukcím. Vyrovnávací paměť je proto dnes poměrně často rozdělená na cache pro data a cache pro instrukce.

úspěšné použití paměti předpokládá vyřešit:

- problém mapování - přiřazení bloků hlavní paměti blokům cache
- problém aktualizace - vhodný algoritmus výměny (vytěsňování) dat z cache
- problém konzistence - zabezpečení shody obsahu (zvláště při více procesorech)

organizace paměti cache:

- **plně asociativní mapování** – libovolný blok hlavní paměti může být umístěn v libovolném bloku paměti cache. Porovnání adres bloků v paměti cache s adresou požadovanou je plně asociativní. Nejlepší a také nejdražší řešení.
- **přímé mapování** - nejjednodušší řešení. Určité bloky hlavní paměti jsou přiřazeny právě jednomu bloku paměti cache. Do velikosti cache cca 900 bajtů nejefektivnější organizace (není třeba vpočítávat, který blok bude vytěsněn z paměti cache)
- **částečně asociativní paměť** - kompromisní řešení, kombinace výše uvedených metod. Dnes nejrozšířenější organizace pamětí cache (několikachodé paměti cache)

- strategie výměny dat mezi hlavní a vyrovnávací pamětí řeší zbylé dva problémy. Pro uvolňování dat z paměti cache se většinou používá mechanizmus LRU (Least Recently Used) nebo FIFO. Zajistit konzistentnost dat je při použití paměti cache největší problém. Je žádoucí dosáhnout toto, aby se každá změna co nejdříve objevila v hlavní paměti. Nejjednodušší varianta je zapisovat data společně do obou pamětí (write through), což je pomalé. Jinou možností je zapisovat (uklízet změny do hlavní paměti při vyřazování bloku (write back). Existují tři varianty této strategie – zápis vždy nebo podle příznaku.

- většinou se používá nějaká varianta asociativní paměti. **V zásadě se jedná o tabulku:**
  - v jednom sloupci jsou klíče (tagy), podle kterých se v tabulce hledá
  - ve druhém příslušná data
  - případně v dalším služební data (potřebná ke správě cache).

- „hledání“ v paměti probíhá tak, že se vstupní hodnota klíče (argument) porovnává po bitech vstupní hodnoty a každého klíče v tabulce (najednou!!! – logika z hradel XOR a NOR). Tak pracují plně asociativní paměti. Je zřejmé, že jsou použitelné pro relativně malé systémy (velké množství komparátorů, dlouhý tag).

- krajním zjednodušením je přímo mapová cache. Hlavní paměť je rozdělena na třídy. Třída je tvořena sadou datových položek, z nichž může být v paměti cache právě jedna. Hledání je jednoduché – adresa třídy ukazuje na příslušný řádek, zde uložená hodnota (tag) se srovná s argumentem a hotovo. Výhoda tohoto schématu je očividná – nejsou potřeba žádná rozhodovací kritéria, která položka bude v daném řádku paměti cache nahrazena (je tam jen jedna). Pro malé paměti cache nejefektivnější řešení. Kombinací obou přístupů jsou vícecestné (také n-chodé) asociativní paměti cache (tabulka má více sloupců).

charakteristiky jednotlivých řešení:

- **plně asociativní organizace cache** - je vhodná pouze pro nevelké systémy. Prověření, zda se položka nachází v paměti cache předpokládá porovnat najednou adresu se všemi tagy v cache uloženými (porovnání po bitech). Což je obvodově složité, tudíž nákladné, a co je horší, i pomalé.
- **přímo mapovaná paměť cache** - je druhý extrém – je jednoduchá proto nenákladná a pracuje velmi rychle. Bohužel, pravděpodobnost úspěšného nalezení požky (HIT) v cache je nižší.
- **n-cestá asociativní paměť** - je kombinací obou přístupů
  - je třeba provést n porovnání k získání výsledku

paměť pracuje v několika režimech

- pokud jsou data v cache nalezena (HIT) ⇒jsou následně dodána

- **pokud nalezena nejsou (MISS) máme dvě možnosti:**
  - v cache je ještě volné místo – z hlavní paměti
  - přečte blok dat a příslušné slovo se dodá do CPU

strategie výměny bloku

- je třeba rozhodnout, který blok bude v paměti uvolněn (případně jak) a je nutné to udělat rychle a efektivně (minimalizovat ztrátový čas). Používá se nejčastěji nějaká (zjednodušená) modifikace strategie FIFO nebo LRU.

- **paměť cache může být umístěna v různých místech výpočetního systému:**
  - -je-li umístěna přímo v procesoru hovoříme o prvé úrovně (L1)
    - často je L1 cache rozdělena na datovou a instrukční.
  - - tvoří-li jí samostatné obvody umístěné na základové desce jedná se o cache druhé úrovně (L2).

  - je také žádoucí umístit cache co nejblíže k procesoru. Na procesorech Pentium je situace podstatně složitější (čtyři typy cache)

- zajištění konzistentnosti dat v hierarchickém paměťovém systému představuje závažný problém. Zejména a při zápisu je třeba umísti shodu dat mezi hlavní pamětí a vyrovnávacími pamětmi různé úrovně. **Což řeší zapisovací strategie:**

strategie současného zápisu (write-through)

- modifikuje data současně jak v cache tak v hlavní paměti. Hlavní paměť je však pomalá a brzdí cache (řešení: třeba uložením zapisovaného do speciálních vyrovnávacích registrů). Stejně, ale po jistý čas je nesoulad mezi daty v hlavní paměti a cache. Řeší se to různě, podstatou ne nastavení příznaku neplatnosti dat. Nevýhodou strategie jsou zvýšené nároky na šířku pásma paměťové sběrnice.

strategie zpožděného zápisu (write-back)

- ukládá změny pouze do bloku v paměti cache. Blok je přitom označen jako modifikovaný (dirty). V hlavní paměti se změny projeví až při uvolnění bloku z cache (úklidu). Uklízejí se pouze modifikované bloky!!! Tato strategie snižuje komunikaci mezi hlavní pamětí a cache (zvláště žádoucí v multiprocesorovém prostředí), ale vyžaduje složité mechanizmy pro zajištění platnosti dat mezi pamětmi různé hierarchie (nevýhodou je komplikovaná implementace). Kombinací obou metod je strategie předběžného zápisu.

NUMERICKÝ PROCESOR

- původně byla jednotka realizována jako samostatný čip (matematický koprocesor 8087, 80 287, 80 387)
- od procesoru 80 486 se stává součástí procesoru (FPU). Způsob ovládání (programování) zůstal nezměněn

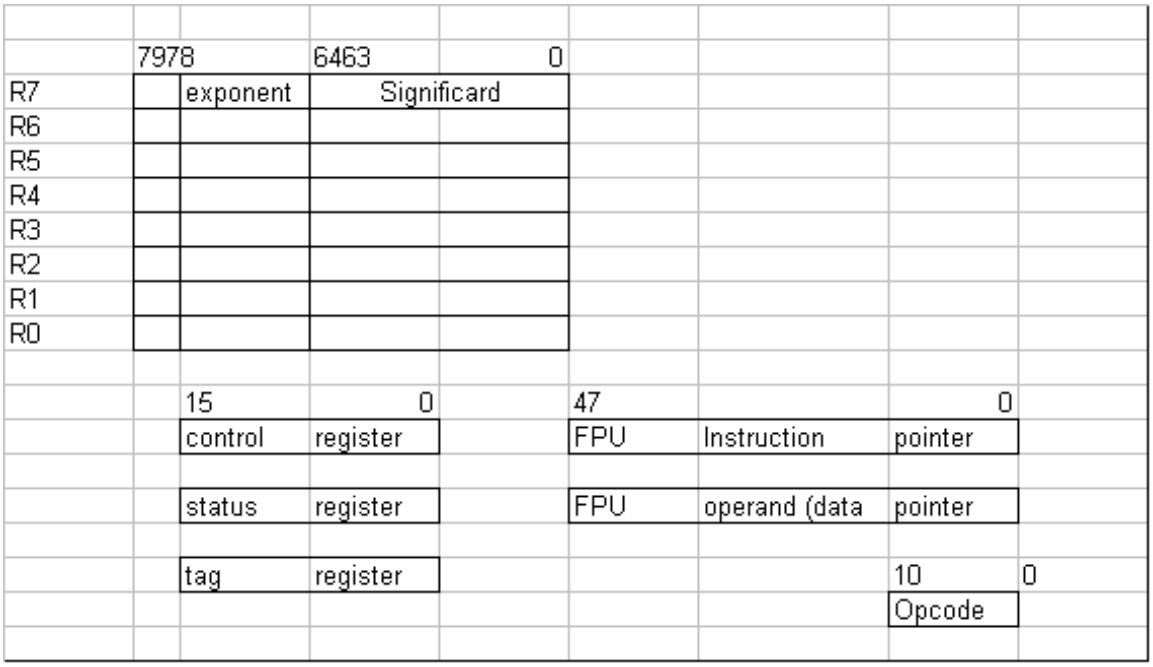
- jednotka interně pracuje výhradně s reálnými čísly o délce 80 bitů ve standardu IEEE – 754 a 854. Numerické výpočty prováděné jednotkou jsou nejméně o řád rychlejší než softwarová emulace s odpovídající přesností.

- jednotka operací v pohyblivé řadové čárce pracuje nezávisle na jednotce pro práci s celými čísly. Instrukce pro dekódování je předána jedné či druhé jednotce – výpočet se provádí paralelně. Pro úplnost nutno dodat, že v závislosti na procesoru, se mění počet jednotek. Procesor 80 486 má jednu IU a jednu FPU. Pentium má dvě IU a jednu FPU a Pentium Pro jednotky 2+2. Dosahuje se tak vyššího výpočetního výkonu – z hlediska programování procesoru se však nic nemění

jednotka umí pracovat s následujícími datovými položkami:

- celá čísla se znaménkem (dvojkový doplněk) o délce 16,32 a 64 bitů
- reálná čísla o délce 32 a 64 bitů a BCD čísla se znakem ve zhuštěném formátu o délce 18 číslic

- uvedené typy čísel jsou na vstupu automaticky konvergovány do vnitřního zobrazení a při výstupu opět přivedeny na požadovaný tvar. Je-li to žádoucí je možný i výstup dat ve vnitřním formátu (80-ti bitová položka --- se zvýšenou přesností)



- jednotka pro práci s reálnými čísly obsahuje 6 služebních a 8 datových registrů. Datové registry lze použít samostatně (jednotlivé a adresovatelné registry, 1-7) nebo jako zásobník – záleží na vykonávané instrukci

- stavový registr charakterizuje stav jednotky – obsahuje bity příznaků a bity signalizující chybové stavy. Bity 11 až 13 určují, který datový registr je vrchol zásobníku

- doplňující registr (tag) upřesňuje obsah každého datového registru (8 x 2 bity) – platný, nula, speciální (nečíslo), prázdný

řídící registr plní 2 funkce:

- umožňuje nastavit některé parametry FPU – přesnost výpočtu, způsob zaokrouhlování (2 x 2 bity) a případně maskovat některé chybové stavy (ztráta přesnosti, podtečení, přetečení – celkem 6 možností)

- dva 48 bitové registry ukazatelů ukazují (ofset, segment) na poslední neřídící instrukci a její data – slouží k řešení chybových stavů. Registr operačního kódu obsahuje prvé 2 bajty poslední neřídící instrukce – podobně jako předchozí, poslouží při řešení chybových stavů.

INSTRUKČNÍ SOUBOR – MANIPULACE S DATY

prostý přesun

FLD	- load real
FST	- store real
FSTP	- store real and pop
FILD	- load integer
FIST	- store integer
FISTP	- store integer and pop
FBLD	- load BCD
FBSTP	- store BCD and pop
FXCH	- exchange registers

podmíněný přesun

FCMOVE	move if equal
FCMOVNE	move if not equal
FCMOVB	move if below
FCMOVBE	move if below or equal
FCMOVNB	move if not below
FCMOVNBE	move if not below or equal
FCMOVU	move if unordered
FCMOVNV	move in not unoredered

INSTRUKČNÍ SOUBOR – ARITMETICKÉ INSTRUKCE

základní

FADD	add real
FADDP	

doplňující

FABS	absolute value
FCHS	change sign
FRNDINT	round to integer
FSCALE	
FSQRT	square root
FXTRACT	extract exponent and significand

INSTRUKČNÍ SOUBOR – ŘÍDÍCÍ INSTRUKCE

FINCSTP	increment FPU register stack pointer
FFREE	Free floating – point registr
FCLEX	clear exception flags after checking for error conditions
FNCLEX	clear exception flags without checking for error conditions
WAIT	wait for FPU

funkce goniometrické

FSIN	sine
FCOS	cosine
FSINCOS	sine and cosine
FPTAN	pertial tangent
FPATAN	partial arctangent

funkce transcendentní

FYL2X	y.log 2x
FYL2XP1	y.log2(x+1)

funkce ostatní

F2XM1	2x-1

Příklad:

dat product = (5.6 x 2.4) + (3.8 x 10.3)

1. instrukce FLD VALUE1 uloží na vrchol zásobníku FPU [ST(0)] hodnotu prvé proměnné [5.6]. Ukazatel vrcholu zásobníku ukazuje na datový registr číslo 4

2. instrukce FMUL VALUE2 násobí hodnotu na vrcholu zásobníku [ST(0)] hodnotou druhé proměnné [2.4], která je uložena v paměti. Výsledek je na vrcholu zásobníku

3. instrukce FLD VALUE3 uloží na vrchol zásobníku hodnotu třetí proměnné [3.8]

4. instrukce FMUL VALUE4 násobí hodnotu na vrcholu zásobníku hodnotou čtvrté proměnné [10.3]. Výsledek je opět na vrcholu zásobníku

5. instrukce FADD(1) sečte hodnotu na vrcholu zásobníku a obsah registru ST(1), což je fyzicky registr R4. Výsledek je opět na vrcholu zásobníku

V případě, že struktura uložených hodnot nedovoluje provést požadovanou operaci přímo, jo možné instrukcí FXCH prohodit obsahy registrů zásobníků. Některé instrukce existují i v reverzní verzi (např. FDIV – ST(1)/ST(0) a FDIVR – ST(6)/ST(1)

PROPOJOVACÍ SUBSYSTÉM

FSB

- FSB (Front Side Bus) nebo System Bus je fyzická obousměrná datová sběrnice, která přenáší veškeré informace mezi CPU a ostatními zařízeními uvnitř systému jako jsou RAM, grafické AGP karty, PCI, hard disky, paměť obsahující systém BIOS, atd.

- Některé počítače mají L2 nebo L3 vyrovnávací paměti, které jsou k procesoru připojeny přes Back Side Bus. Tato sběrnice a vyrovnávací paměť se připojují rychleji než přístup do paměti přes FSB.

- Maximální teoretická šířka pásma FSB sběrnice je určena z výsledku šířky frekvence a množství dat přenesených za časový úsek.

AGP

- Accelerated Graphics Port (též Advanced Graphics Port), zkratka AGP, je vysokorychlostní „point-to-point“ kanál (nejde tedy o sběrnici v pravém slova smyslu) pro připojení grafické karty k základní desce počítače. Některé základní desky mají několik nezávislých AGP slotů. AGP je pomalu nahrazováno novějším rozhraním PCI Express.

- Intel představil první verzi AGP (nazvanou „AGP specification 1.0“) v roce 1997. Zahrnovala obě rychlosti 1× a 2×. Novější verze AGP zvýšily přenosovou rychlost z 2× na 8×. **Tyto verze jsou:**
    - AGP 1×, používající 32bitový kanál běžící na 66 MHz umožňující maximální datový tok 266 MB/s, zdvojnásobený z 133MB/s rychlosti sběrnice PCI 33 MHz / 32bitů; 3.3V.
    - AGP 2×, používající 32bitový kanál běžící na 66MHz (double pumped) na efektivních 133MHz umožňující maximální přenosovou rychlost 533MB/s; 3.3V;
    - AGP 4×, používající 32bitový kanál běžící na 66MHz (quad pumped) na efektivních 266MHz umožňující maximální přenosovou rychlost 1066MB/s (1GB/s); 1.5V;
    - AGP 8×, používající 32bitový kanál běžící na 66MHz (8 bitů za takt) na efektivních 533MHz umožňující maximální přenosovou rychlost 2133MB/s (2GB/s); 0.8V.

PCI

- je počítačová sběrnice pro připojení periferií k základní desce, která není omezená na platformu osobních počítačů PC. Používá paralelní přenos dat (šířka 32 nebo 64bitů) a je orientovaná na přenos zpráv místo přímé komunikace. Od zbytku systému je oddělena pomocí PCI mostů, které zprostředkovávají komunikaci s připojenými kartami. V jednom počítači je jedna nebo i více nezávislých PCI sběrnic.

- Sběrnice PCI je běžná v moderních osobních počítačích, kde jako standardní rozšiřující sběrnice nahrazuje sběrnici ISA a VESA Local Bus, ale objevuje se také v mnoha jiných typech počítačů. Nízká propustnost PCI sběrnice vedla k vytvoření specializovaného portu AGP určeného pro grafické karty, který je rychlejší, než PCI sběrnice a zároveň přinesl další vylepšení. PCI a AGP byly dlouhou dobu součástí většiny vyráběných základních desek pro PC. V současné době jsou PCI i AGP nahrazovány sběrnicí PCI Express, která používá sériový přenos. Specifikace PCI se zabývá fyzickými rozměry sběrnice (včetně rozestupu vodičů), elektrickými charakteristikami, časováním sběrnice a protokoly.

- Specifikace PCI protokolu zajišťuje, že přerušení mohou být sdílená, takže o jedno přerušení se může dělit více karet. Ovladače jsou pak vyvolány v sérii a tak je zajištěno, že je příslušné zařízení obslouženo. Nevýhodou je však vyšší latence a režie obsluhy přerušení, a proto se obvykle velmi aktivní zařízení umisťují na samostatná přerušení (typicky řadič disků a síťová karta). Sběrnice PCI obsahuje čtyři linky přerušení a všechny z nich jsou dostupné každému zařízení. Každému zařízení tak mohou být při inicializaci sběrnice až čtyři přerušení, což však není běžné. Zároveň jsou k dispozici pouze čtyři přerušení, které označujeme INTA#, INTB#, INTC# a INTD#,

které jsou v PCI mostu asociovány s vnějšími přerušeními procesoru (jejich konkrétní přiřazení není obvykle uživatelsky nastavitelné).

PCI sběrnice používaná v běžných domácích PC

- šířka sběrnice 32 bitů
- hodiny s kmitočtem 33.33 MHz a synchronním přenosem
- maximální teoretická přenosová rychlost 133 MB za sekundu (33.33 MHz × 32 bitů ÷ 8 bitů/byte = 133 MB/s)
- přenosovou kapacitu sdílí všechna připojená zařízení
- prakticky je maximální rychlost nižší (část se spotřebuje na režii sběrnice)
- běžný pevný disk dokáže při sekvenčím čtení přesáhnout 50 MB/s
- připojíme-li na sběrnici PCI kartu IDE řadiče se čtyřmi disky, nebude již klasická PCI sběrnice stačit
- 32 bitová adresová sběrnice (adresace až 4 GB paměti RAM)
- 32 bitové adresy I/O portů
- 256 bajtů konfiguračního prostoru

PCI sběrnice pro výkonné stanice a servery

- šířka sběrnice 64 bitů
- hodiny s kmitočtem 66 MHz nebo 133 MHz
- maximální teoretická přenosová rychlost 266 MB nebo 532 MB za sekundu

Verze

- Pozdější verze PCI umožňují (a v nejposlednější verzi vyžadují) 3,3 V sloty na základních deskách (jiné klíčování resp. zářezy) a umožňují kartám, aby byly buď dvakrát klíčované pro obě napětí, nebo jen pro 3,3 V
- PCI 2.2 umožňuje 66 MHz signalizaci (vyžaduje 3,3 V signalizaci - nejvyšší možná rychlost přenosu 533 MB/s)
- PCI 2.3 dovoluje použít 3,3 V a univerzální klíčování, ale nedovoluje klíčování pro 5 V.
- PCI 3.0 je konečný oficiální standard pro PCI sběrnici, byla úplně odstraněna možnost 5 V.
- PCI-X zdvojnásobuje šířku na 64 bitů, upravuje protokol a zvyšuje maximální signalizační frekvenci na 133 MHz (nejvyšší přenosová rychlost 1014 MB/s)
- PCI-X 2.0 povoluje 266 MHz kmitočet (nejvyšší přenosová rychlost 2035 MB/s) a také 533 MHz, rozšiřuje konfigurační prostor na 4096 bajtů, přidává šestnáctibitovou variantu a umožňuje signalizaci na 1,5 V.
- Mini PCI je nová forma PCI 2.2 pro použití v přenosných počítačích.
- CardBus je karta PC pro 32 bitovou, 33 MHz PCI
- Compact PCI používá moduly velikosti Eurokarty
- PCI/104-Plus je průmyslová sběrnice která používá signály PCI s různými konektory

PŘEHLED VÝVOJE PROCESORŮ INTEL

8086

- 20-bitová adresová sběrnice – až 1 MB operační paměti
- 16-bitová datová sběrnice
- 16-bitové registry
- bez cache, externí numerický koprocesor (FPU)

80286

- 24-bitová adresová sběrnice – až 16 MB operační paměti
- 16-bitová datová sběrnice
- 16-bitové registry
- bez cache, externí numerický koprocesor (FPU)

80386

- 32-bitová adresová sběrnice – až 4 GB operační paměti
- 32-bitová datová sběrnice
- 32-bitové regisry
- bez cache, externí numerický koprocesor (FPU)

80486

- 32-bitová adresová sběrnice – až 4 GB operační paměti
- 32-bitová datová sběrnice
- 32-bitové registry
- ALU a FPU
- interní L1 cache 8-16 KB

pentium P5

- 32-bitová adresová sběrnice – až 4 GB operační paměti
- 64-bitová datová sběrnice
- 32-bitové registry
- dvě ALU a FPU
- interní dělená L1 cache (2x) 8 – 16 KB

pentium P6

- 36-bitová adresová sběrnice – až 64 GB operační paměti
- 64-bitová datová sběrnice
- 32-bitové registry
- dvě ALU a FPU
- interní dělená L1 cache (2x) 16 KB
- interní L2 cache 128 KB – 2 MB
- architektura DIB (FSB,BSB)

pentium 4 (P7)

- 36-bitová adresová sběrnice – až 64 GB operační paměti
- 64-bitová datová sběrnice
- 32-bitové registry
- dvě ALU a FPU
- interní dělená L1 cache 8 + 16 KB

- interní L2 cache 128 – 512 KB

Itanium (P8)

- 44-bitová adresová sběrnice – až 16 TB operační paměti
- 64-bitová datová sběrnice (128)
- 64-bitové registry
- dvě ALU a dvě FPU
- interní dělená L1 cache (2x) 16 KB
- interní L2 cache 96 KB
- interní L3 cache 2 – 4 MB

TECHNOLOGIE MIKROPROCESORŮ

TECHNOLOGIE PROCESORŮ INTEL P5 A P6

- prvý typ procesoru Pentium se objevil na trhu v roce 1993. Pentia se sice vyrábí dodnes (Pentium 4), ale za obchodním názvem je schováno několik generací procesorů architektury IA-32 různých produktových řad

- základním hnacím motorem inovací není jen zvyšování výpočetního výkonu procesoru (počet operací za jednotku času), ale neméně důležitá je podpora (hardwarová) systémových funkcí výpočetního systému (funkce operačního systému, podpora grafiky, atd). Toto vyžaduje více elementů na čipu (Moore), více elementu ⇒ větší energetická náročnost ⇒ větší nároky na chlazení. Důsledkem je neustálé snižování napájecího napětí procesoru, stále tenčí propojovací vodiče (litografické čáry), časté změny v zapouzdření procesoru (počet vývodů, typ pouzdra a patice).

- dalším podstatným rysem je jistá specializace procesorů. Původně se vyráběl prostě procesor – jeho použití bylo poměrné úzce ohraničeno (pokud ho někdo chtěl použít jinak, měl smůlu – musel vzít to, co bylo na trhu). Dnes se procesory používají v nejrozmanitějších zařízeních, což klade na procesor různé, často protichůdné požadavky. Tak vznikly produktové řady – procesory pro stolní počítače, výkonné servery, mobilní zařízení. Generací procesoru se rozumí určitá typová řada, která vykazuje stejné architektonické a výkonnostní rysy. V zásadě jsou generace procesorů definovány podle procesorů vyráběných firmou Intel (funguje to tak zhruba do 6. generace)

P5

- postupně se vyrábělo ve třech verzích. Realizace litografickou technologii s tloušťkou čary 0.8,0.6 a 0.35 um. Počáteční pracovní frekvence činila 60 nebo 66 Mhz a byla shodná s frekvencí základní desky (MB). Pozdější verze pracovaly na násobku frekvence MB až do 266 Mhz. Napájení 5V a spotřeba zhruba 16 W. Později bylo napětí sníženo na 3.3 až 2.8V (což vyžadovalo vždy nový MB).

- v poslední verzi došlo k rozšíření instrukční sady o 57 nových instrukcí pro zpracování grafiky, videa a zvuku (MMX – instrukce typu SIMD – jedna instrukce zpracovává více datových položek najednou) a ke zvětšení instrukční cache na 16kB. Původní patice S4 nahrazena S5 a S7. Na MB přidán modul regulátoru napětí pro procesor (delší použitelnost určitého typu desky). Procesory této generace mohly pracovat pouze v symetrickém multiprocesingu (což znamená maximálně dva procesory na jedné základové desce).

P6

pentium pro (1995)

- bylo prvním členem této řady
- **v pouzdře procesoru jsou integrovány dvě jednotky:**

- vlastní procesor
- paměť cache L2 o velikosti 256, 512, či 1MB.

- samotný procesor obsahuje cache L1 o velikosti 16 KB. Instrukční cache je dvoucestná asociativní o velikosti 8 KB. Datová cache je realizována jako čtyřcestná asociativní taktéž 8 KB.

- základní inovací je architektura DIB – procesor má dvě nezávislé sběrnice (celková propustnost sběrnic se zvýší až 3x). Paměť cache L2 je připojena k procesoru samostatnou sběrnicí, která pracuje na plné rychlosti procesoru. Uvedené řešení umožňuje vytvoření mulitprocesorového systému až čtyřech procesorů.

- další inovací je dynamické vykonávání instrukcí. Podstatou je zpracování instrukcí spíše na základě logiky, než pouze podle seznamu. **Výsledný efekt je dosažen kombinací tří metod:**
  - predikce vícenásobného větvení
  - analýza dat
  - spekulativního vykonávání instrukcí

- šířka datové sběrnice zůstala stejná (64 bitů), adresová byla rozšířena na 36 bitů. Napájecí napětí 3.3V později 3.1V. Příkon až 25 W. Taktovací frekvence maximálně 200 Mhz. Tento procesor nepodporuje, (neumí) instrukční sadu MMX.

procesor pentium II (1997)

- je sice jen vylepšené Pentium Pro, ale způsobem zapouzdřením se jedná o zcela nový procesor (poprvé bylo užito pouzdro s hranovým kontaktním polem SECC). Vylepšením je zvětšení L1 cache na 32 KB (2x16). Vyráběl se litografickou technologii s tloušťkou čáry nejprve 0.35 později 0.25 um. To, společně se snížením napájecího napětí z 2.8 V až na 2 V, se odrazilo ve snížení spotřeby procesoru [(450 MHZ procesor (27 W) má menší spotřebu než 233 Mhz (35W)]. Maximální taktovací frekvence 450 Mhz. Procesor podporuje instrukční sadu MMX a pouze symetrický multiprocesing.

procesor pentium III (1999)

- je posledním modelem řady P6. Vylepšením bylo další rozšíření instrukční sady SSE (70 instrukcí).

- **ostatní inovace jsou spíše kosmetické :**
  - - výrobní číslo procesoru
  - - teplotní čidlo pro detekci teploty jádra
  - - ochrana proti přetaktování procesoru.

- -počáteční litografická technologie pracovala s tloušťkou čary 0.25, později se přešlo na 0.18 um. Původně byla, L2 cache o velikosti 512 KB, umístěna mimo jádro a pracovala na jeho poloviční rychlosti. Později bylo 256 KB L2 cache integrováno na čip, kde pracovala na plné rychlosti jádra. Maximální taktovací frekvence byla 1 Ghz. Podpora pouze pro symetrický multiprocesing.

procesory celeron

- představují levnější variantu Pentia II/III. Předpokládá se použití především v počítačích pro kancelářské aplikace (segment trhu low-end). V podstatě se jedná o procesor bez paměti L2 cache. Starší verze tohoto procesoru obsahují jádro Pentium II novější pak Pentium III. Aby byla cena procesoru co nejnižší, používá se i jiný způsob zapouzdření. Nejprve to bylo pouzdro SEPP, později pak PPGA a FC-PGA pro paticí S370. Procesor se postupně vylepšoval – typy s pracovní frekvencí 300 a více MHz už mají integrovánu paměť L2 cache o velikosti 128 MB. Vyráběny byly litografickou technologii 0.25 později 0.18 um. Spotřeba Celeronu je menší než odpovídajícího procesoru Pentium (= menší chladič), ostatní parametry shodné nebo mírně lepší.

procesory xeon

- představují výkonnější variantu Pentia II/III. Předpokládá se použití v grafických stanicích serverech (segment trhu high-end).

- **rozdíly jsou následující:**
  - - procesor je zapouzdřen ve zvětšeném pouzdře SEC
  - - paměť L2 cache pracuje vždy na plné rychlosti jádra a byla zvětšena až na 2 MB (proto větší pouzdro).
  - - řídící čip cache umí vypočítávat odkazy na plných 64GB systémové paměti.
  - Ostatní parametry jsou stejné či lepší.

Pentium II/III a Celeron MObile

- je určen pro použití v přenosných počítačích. Základním parametrem je spotřeba procesoru (při vyhovujícím výkonu).

- **snížení spotřeby se dosahuje kombinací několika technologií:**
  - především je to výrobní postup, litografická technologie 0.18 nebo 0.13 um a použití měděných vodičů (příklad: procesory vyrobené 0.13 um technologií mají spotřebu až o 40% nižší).
  - snížením napájecího napětí jádra až na 1.6V ve verzi Mobile a až na 1.1V ve verzi ULV.
  - použitím technologie SpeedStep, která řídí napětí a frekvenci procesoru při běhu na baterie
  - technologie QuickStart (procesor je převeden do spánku i mezi dvěma stisku klávesnice).

- příklad: LV Mobile pentium III na frekvenci 900 MHz má průměrnou spotřebu menší než 1W.

- **uvedené technologie se dále vylepšují**:
  - Enhanced SpeedsStep řídí napájecí napětí a taktovací frekvenci podle zatížení procesoru (dříve pouze ve dvou stupních)
  - pro zapouzdření se používá technologie BGA a nebo mini-cartridge (IMM – modul, který má v pouzdře procesoru integrován čip North Bridge).

- dnes se procesory pro mobilní zařízení označují jako Pentium III-M.

Pentium 4

TECHNOLOGIE NetBurst

- NetBurst je mikro-architektura procesorů Pentium 4 firmy Intel. Spoléhá (co se týče výkonu) na vysoké frekvence. Procesory s touto architekturou mají hyperskalární architekturu (řetězení instrukcí) a také superskalární (2 fronty). Různé komponenty v procesoru mají různou frekvenci. Konstrukce obvodů jsou provedeny tak, aby bylo provádění instrukcí co nejrychlejší. Architektura používá funkci "Front End". Dále pak tyto procesory obsahují jednotku "Out-of-Order", která umí "předělat" program tak, aby neutrpěla jeho logika, tzn. nelze-li nějakou instrukci provést, protože nejsou k dispozici data, jednotka provede jinou instrukci (přičemž bere v úvahu hardwarové prostředky) pro kterou data má a tím se sníží zpoždění. Procesory pak mají ještě jednotku "Retirement Section" (zpětně uspořádavají posloupnosti instrukcí - souvisí s předvídáním skoků).
- V současné době (červenec 2007) je již architektura NetBurst téměř minulostí a je nahrazena architekturou Core.

- hyperzřetězená technologie (Hyper Pipelined Technology), která umožňuje provádění softwarových instrukcí ve 20-ti fázovém datovodu oproti 10-ti fázovému datovodu u Pentia III. Hyperzřetězená technologie podporuje novou úroveň taktovacích kmitočtů (zatím 1,5 a 1,4 GHz se značnou rezervou pro budoucnost).

TECHNOLOGIE CORE

- Core je architektura x86 procesorů firmy Intel uvedená v roce 2006.

- Architektura Core je významně postavena na návrhu procesorů Pentium M. Proti architektuře Netburst kterou nahrazuje, byla výrazně zkrácena instrukční pipeline. Následkem toho procesory Core nedosahují pracovních frekvencí starších procesorů, jejich reálný výkon je ale vyšší. V důsledku snížení pracovních frekvencí bylo omezeno produkované teplo.

- Více procesorů v jednom.

Na této architektuře jsou založeny procesory

- Core 2 Duo pro stolní počítače (jádro Conroe)
- Core 2 Extreme
- Core 2 Quad
- Core 2 Duo pro notebooky (jádro Merom)
- některé Xeon pro víceprocesorové systémy

TECHNOLOGIE OPERAČNÍCH SYSTÉMŮ

ZÁKLADNÍ FUNKCE A TYPY OPERAČNÍCH SYSTÉMŮ

- je základní řídící program, který ovládá všechny vnitřní funkce stroje a poskytuje uživateli prostředky pro řízení provozu počítače. Výchozí činností OS je správa prostředků výpočetního systému (paměť, procesor, zařízení vstupu výstupu, souborový systém, uživatelé, služby..). **Má dvě hlavní složky: hardwarově závislou část (jádro) a hardwarově nezávislou (uživatelskou) část.** Podstatnou funkcí OS je poskytování služeb aplikačnímu softwarovému vybavení (API – aplikační rozhraní systému). Principem poskytování služeb je odstínění aplikací od rutinních, standardních činností (například otevření souboru, přidělení paměti, atd..). Jinými slovy, OS je soubor aplikací a technického vybavení počítače. Aplikační programy slouží k řešení určité skupiny úkolů – mechanizace kancelářských prací, manipulace se strukturovanými daty (databáze), grafika, komunikace, atd. Ke své činnosti potřebují podporu (služby) OS.

- OS je programové vybavení, jímž se řídí chod programů v počítači (ČSN). Základní funkcí OS je správa prostředků výpočetního systému.

- Tyto prostředky dělíme na:
  - **hardwarové** (také fyzické)
  - **prostředky nemateriální podoby**, které se většinou nezývají logické.

  - správou prostředků rozumíme jak jejich evidenci, tak i zpřístupnění ostatním komponentám výpočetního systému. Zpřístupnění je třeba především chápat jako odstínění od rutinní, většinou i dost složité a nepodstatné (pro dané účel) činnosti. OS řídí chod výpočetního systému, musí tedy existovat nějaké rozhraní, které zajišťuj komunikaci s člověkem.

  - V zásadě jsou možné dva způsoby: **textová nebo grafická forma.**

  - výpočetní systémy nebyly vybaveny OS vždy (patrně i dnes většina vestavěných výpočetních systému pracuje bez OS.

Z historického hlediska můžeme OS rozdělit na:

- **monitory** - jednoduché programy, které umožňují zavádět do počítače aplikační programy v binární formě a popřípadě je také odlaďovat.
- **dávkový jednouživatelský systém** - zpracovává úlohy (batch, processing), pomocí souboru příkazů definující zpracování zakázky v některém skriptovacím jazyce pro řízení úloh (OS 360, JCL – Job Control Language). Základní výhodou i nevýhodou tohoto typu OS je nepřítomnost aplikačního programátora při zpracování úlohy.
- **dávkový víceúlohový systém** - umožňuje zpracování více úloh takřka současně. Základním cílem těchto OS je odstranění prostojů procesu (nejdražší části počítače) při dávkovém zpracování úloh. Podstatou je systém s několika úlohami a přepínáním kontextu podle připravenosti úlohy (a dostupnosti zdrojů). Řízení systémů v reálném čase – řízení technologických procesů (především vojenská zařízení). Klíčovým parametrem těchto OS je doba odezvy a spolehlivost.
- **systémy se sdílením času a interaktivním přístupem uživatele** - většina moderních univerzálních OS vychází z této koncepce. Jinak řečeno je to OS pracující v reálném čase, schopný zpracovat pomocí mechanizmu sdílení času více úloh najednou, který umožňuje bezprostřední (interaktivní) komunikaci s uživatelem (nebo i uživateli)

  - fundamentální práce v oblasti OS a programování vůbec jsou spojeny s panem Edsgarem W. Dijkstrou (Holaňdan, OS T.H.E, zhruba rok 1968

- klíčovým elementem OS je proces – posloupnost událostí definována svým účelem nebo účinkem, uskutečňovaná za daných podmínek.

- proces je vykonávající se program (program vykonávaný v daném kontextu, prakticky totéž co úloha, při přepínání úloh).

- -v systému musí existovat vhodný signál, který pozastaví nebo znovu aktivuje příslušný proces. Dojde-li k pozastavení procesu, musí existovat nějaký mechanizmus uložení kontextu procesu a jeho zpětné obnovení při reaktivaci procesu.

- každý proces má unikátní identifikátor (zvláště ve více úlohovém nebo víceuživatelském prostředí). Většinou je to binární číslo (16 nebo 32 bitů). Proces může zrodit další proces (procesy). Což znamená, že procesy mají hierarchickou strukturu (rodič – potomek).

soubor

- posloupnost záznamů, které lze zpracovat jako celek (množina souborů je hierarchicky uspořádána:kořenový adresář, adresáře (složky nebo knihovny) a soubory). I soubory mají své identifikátory – názvy (pro uložení na paměťovém médiu) a binární čísla (file descriptor nebo handler) při manipulaci.

zařízení v/v.

- znakově orientované
- blokově orientované

- OS také vždy definuje standardní zařízení pro vstup a výstup. Zařízení V/V jsou často reprezentována jako speciální soubory. Komunikaci mezi procesy zajišťuje speciální druh souborů zvaných roury (pipe).

os má typicky čtyři základní komponenty:

- správa procesů
- správa operační paměti
- správa souborového systému
- správa periférií.

- spravování je základní činnost. OS má definováno několik (nejméně 2) rozhranní, které určují (umožňují) spolupráci s okolím. Rozhraní pro styk s uživatelem umožňuje řízení systému. Aplikační programový interface (API) zpřístupňuje služby OS ostatním programům.

struktura os

- zpočátku měly OS strukturu jako libovolný jiný program (the big mess). Později se přešlo na vrstevnatou strukturu (Dijkstra). Následovala koncepce virtuálního stroje (OS 3700). Vývoj postupně vytlačoval kód do vyšších vrstev OS, což vedlo k minimalizaci jádra OS (kernel). Výsledkem je model OS typu klient-server. Rozdělení OS na relativně autonomní části (servery), spravující plně určitou činnost a nemajíce dostup k hardware znamenitě vylepšilo spolehlivost a přenositelnost.

KOMPONENTY OS

PROCESY

sekvenční proces

- je posloupnost výpočetních kroků v určitém kontextu

paralelní sekvenční proces

- je množina sekvenčních procesů, které běží současně a vzájemně mezi sebou komunikují.

- disponoval-li by výpočetní systém více procesory mohl-by každý proces (teoreticky), pracovat na svém procesoru (to je pravý paralelismus). Většinou je k mání jen procesor jeden a iluze současného provádění několika úloh najednou je dosaženo rychlým přepínáním mezi úlohami (pseudoparalelismus). Procesor v takovém případě v krátkých časových intervalech (milisekundy) přechází od vykonávání instrukcí jednoho procesu k vykonávání instrukcí jiného procesu (přepnutí kontextu).

procesy se během svého života nacházejí v různých stavech:

- **proces je naplánován** (scheduled), je připraven ke spuštění, má stanoven okamžik zahájení práce a přiděleny potřebné zdroje
- **proces běží** (running), je právě vykonáván
- **proces je ve stavu čekání** (waiting), nejsou-li splněny podmínky pro pokračování (čeká na uvolnění zdroje nebo dokončení činnosti požadované na jiném procesu)
- **proces je pozastaven** (seuspend), když by sice byl schopen práce, ale provádí se jiný proces, protože nastalo přepnutí kontextu

- proces tedy běží nebo neběží, je připraven nebo blokován, čeká na spuštění nebo na zdroj nebo na svůj čas. OS (většinou manažer procesů nebo plánovač) musí mít přehled v jakém že stavu se proces momentálně nachází (většinou popisuje PCB – process control block).

- přepínaní procesů, předpokládá komunikaci mezi procesy a dynamickým přidělováním zdrojů. Což může představovat značný problém. V programovém kódu procesu mohou být kritické sekce, ve kterých může docházet k časově závislým chybám v důsledku sdílení prostředků (race conditions). Chybové stavy tohoto typu vznikají pouze za určitých podmínek (časová koincidence). Musí proto existovat jistý arbitrážní (rozhodčí) mechanizmus, synchronizace komunikace mezi procesy, který zajistí vykonání kódu kritické sekce ve správné časové posloupnosti. Dynamické přidělování zdrojů vede k situacím, kdy několik procesů soupeří o tentýž systémový zdroj a výsledkem je buďto uvíznutí (mrtvý bod – deadlock) nebo vyhladovění (starvation).

- základním problémem pseudoparalelizmu je prokládání procesů a s tím související časově závislé chyby, které vznikají pouze při určité koincidenci časových průběhů procesů.

známe několik možných řešení tohoto problému:

- **zákaz přerušení** - možnost víceméně teoretická, použitelná pouze v jednouživatelském OS, protože nevhodně provedený zákaz přerušení zablokuje celý systém. Dlouhodobější umrtvení přerušení může vést k tomu, že procesor některé přerušení propásne, což pro příslušný proces má většinou fatální následky
- **aktivní čekání** - primitivní způsob synchronizace založený na cyklickém testování logické proměnné (zvané zámek). Proměnná má hodnotu false, je-li kritická sekce volná. Vstoupí-li proces do kritické sekce, musí nastavit proměnnou na hodnotu true a po výstupu opět na false. Základní

nedostatkem uvedené konstrukce je, že jednoduché naprogramování problém neřeší. Nehledě také nato, že neustálé olizování logické proměnné je mrháním procesorového času.

- **instrukce tsl** - hardwarová realizace výše uvedené logické proměnné (Test Set and Lock). Bývá součástí některých procesorů (Pentium – instrukce SB). Zajistí čtení i zápis do paměťové buňky nepřerušitelným způsobem.
- **pomoci synchronizačních prostředků jádra os**
  - semafor - je objekt, který obsahuje celočíselný čítač a frontu procesů. Nad semaforem jsou definovány tři operace: down, up a init. Podstatné je, že tyto operace jsou atomické (nerozložitelné)
  - monitor - je objekt (programový modul), který má deklarovány jistě proměnné a procedury, které výhradně mohou těmito proměnnými manipulovat. Jádro OS zajistí, že žádný proces nemůže zavolat monitor, pokud se nějaký jiný proces nachází v monitoru
  - předávání zpráv - je nejuniverzálnější metoda synchronizace. Nepotřebují společnou paměť. Pokud proces vyšle zprávu a pokračuje v činnosti, jedná se o strategii asynchronního předávání zpráv (RT OS). Při synchronním předávání zpráv musí být přijímací proces připraven okamžitě vyslanou zprávou přijmout (jinak je vysílající proces pozdržen, dokud přijímač neakceptuje zprávu – rendezvous). Asynchronní varianta předávání zpráv je složitější.
-

- rozvrhovač (scheduler) je proces jádra OS, jehož úkolem je rozvrhování přidělování systémových prostředků jednotlivým procesům.

Dvě základní strategie řízení chodu procesů:

- dobrovolná výměna (nonpreemptive)
  - nonpreemtivní
  - nejjednodušší a nejsnadněji realizovatelná strategie
  - proces pracuje tak dlouho, dokud se sám neodstaví. Což je vhodné pro systémy s jedním řídícím procesem (databázové systémy), ale pro univerzální stroje zcela nepoužitelné.
- nucená výměna (preemptive)
  - preemtivní
  - všestrannější
  - Na konci přiděleného časového intervalu je procesu procesor nekompromisně odebrán. Rozvrhovač přitom postupuje podle určitého (plánovacího) algoritmu a může mít na zřeteli různá hlediska (například prioritu uživatele, využití systému nebo konečnost termínu).
  -

Plánovací algoritmy používané při preemptivní strategii jsou následující:

- ***cyklická obsluha*** – každému procesu je přidělen stejný časový úsek (kvantum), což se cyklicky opakuje. Poměr velikosti kvanta k režijním nákladům spojeným s přepnutím kontextu je důležitý parametr (20/25 nebo 500/501). Zvětšování kvanta se zhoršuje odezva systému. (Round Robin Scheduler)
- ***podle priority*** - proces má podle stanoveného kriteria přiřazenu prioritu, z množiny k běhu připravených procesu se spustí ten, který má největší prioritu. Priorita může být fixní nebo přidělována dynamicky (například podle skutečně spotřebovaného času: 100/2 = 50). Užívají se i kombinované algoritmy – procesy jsou rozděleny do tříd podle priority a uvnitř třídy se použije cyklická obsluha.
- ***fronty s prioritou*** - máme několik front s klesající prioritou. Fronta s nejvyšší prioritou přiděluje procesu jedno kvantum času, fronta s nižší prioritou přiděluje procesu vždy dvojnásobek kvanta nadřazené fronty. Proces startuje s nejvyšší prioritou a po odstavení je zařazen na konec fronty s nižší prioritou. Výhody: krátký proces bude rychle vykonán, dlouhý bude méněkrát přerušen. (Multiple Queues nebo Feedback Queues Scheduler)

- v některých systémech je vhodné, aby proces měl určitý vliv na přidělenou prioritu. Dosáhne se toho tím, že mechanizmus rozvrhování ovládá rozvrhovač a politiku přidělování ovlivňuje proces sám (zejména u procesů se vztahem rodič – potomek)

- **volba algoritmu pro správu procesů OS je závislá na mnoha faktorech. Typ OS má zásadní význam.**

V podstatě máme tři základní typy OS:

- systém dávkového zpracování
- interaktivní systém
- práce v reálném čase

nicméně některé požadavky jsou všem systémům společné:

- spravedlivé přidělování CPU
- schopnost prosazovat definovanou politiku přidělování
- rovnoměrné využití a vytížení všech zdrojů výpočetního systému

systémy dávkového zpracování

- zde je důležitá propustnost (počet vykonaných úloh za jednotku času), průměrná doba obrátky (doba potřebná k vykonání úlohy) a stupeň využití CPU

- **používají nejrůznější algoritmy správy procesů:**
  - primitivní, podle pořadí (kdo dřív přijde, ten dřív mele) nebo nejkratší úloha první (small is beautiful)
  - komplikované, jako tříúrovňový plánovač (vstup – vhodná směs úloh: využití CPU a operace I/O, využití pamětí: vytěsňování a počet spuštěných procesů v paměti)

interaktivní systémy

- zde je významná především doba odezvy a proporcionalita (uspokojení očekávání uživatelů)

- využívají jak jednoduché algoritmy (cyklická obsluha), tak i složitější na základě priorit (často podle využití posledního přiděleného kvanta: například kvantum je 100 msec, proces využil 1 msec a v následujícím běhu má prioritu 100, využil 25 msec, příště má prioritu 4, tedy KV/Ti) či kombinovaná schémata používající fronty s prioritami.

systémy reálného času

- zde je vždy nutno dodržet požadovanou dobu odezvy (k zamezení ztráty dat musí akce nastat do určitého času) a zabránit degradaci poskytovaných služeb.

- v systémech reálného času je situace složitější - proces musí na požadavek odpovědět v určitém časovém limitu a musí to provést co nejrychleji – je žádoucí, aby proces byl co nejkratší. Obsluha požadavků může být periodická (vznik požadavků je určitelný) nebo aperiodická (požadavky jsou nepředvídatelné). V prvém případě je realita většinou složitější, v systému často existuje několik proudů událostí s různou periodou.

vlákna (threads)

- proces může spustit jiný proces (vztah rodič – potomek). Spuštěný proces sice zdědí obsah kontextu rodičovského procesu, ale další vývoj je zcela nezávislý. Mnohdy to není žádoucí a bylo by naopak výhodné, kdyby potomek měl s rodičem společnou část kontextu (dat, semafory, dětské procesy). K tomuto účelu slouží vlákna.

PAMĚŤ

- operační paměť je základní prostředek výpočetního systému. Správa pamětí je tudíž důležitou částí OS. Máme dva zásadově odlišné způsoby zprávy:
    - bez vytěsňování procesů z operační paměti na vnější médium
    - s vytěsňováním (virtuální paměť a stránkování).

- **další důležité skutečnosti mající podstatný vliv na správu paměti jsou:**
    - zda OS podporuje zpracování více úloh najednou
    - zda OS podporuje práci více uživatelů najednou (nesložitější varianta)

jednoúlohový, jednouživatelský os bez vytěsňování

- správa paměti na poměrně primitivní úrovni. Paměť je pouze sdílená mezi OS a uživatelský program. Správu tvoří pouze:
    - funkce umožňující zavedení programu do paměti
    - funkce přidělení či uvolnění bloku paměti
    - informační služby (kolik užito, kolik volné paměti).

- potřebuje-li program více paměti než fyzicky existuje, řeší se to nejčastěji metodou překrývaných segmentů (overlays). Kořenová část programu je trvale v paměti, překryvné části se natahují z vnějšího média podle potřeby. Metoda je založena na skutečnosti, že celý program nemusí být najednou v operační paměti. (což ovšem nemusí být vždy platné)

víceúlohový jednouživatelský os

- správa operační paměti se komplikuje o bezpečnostní funkce, které odstiňují procesy od vzájemného vlivu a mechanizmus relokace paměti

- relokací se rozumí přemístění programového kódu v paměti, což předpokládá úpravu paměťových odkazů tak, aby bylo možno po přemístění kód provést.

- základním předpokladem je nepoužívání absolutní adresace (adresa musí být báze/ofset – adresové odkazy jsou vztaženy relativně k řídící sekci procesu). Takřka vždy se používá i nějaký vytěsňovací mechanizmus. Uvedený typ OS se používá nejčastěji na osobních počítačích.

víceúlohový, víceuživatelský os

- je už jen komplikovanější verzí předchozího

metody alokace paměti

- paměť je procesům přidělována podle určitého algoritmu. Správce paměti musí mít přehled o obsazení paměti (bitová mapa, svázané seznamy).
    - nejjednodušším postupem je metoda **prvního vhodného výběru** (First – fit)
    - modifikací je **následný vhodný výběr** (Next – fit)
    - jinými postupy jsou metody**- nejlepšího výběru** (Best – fit)
    - **nejhoršího výběru** (Worst – fit).

- cíl je jednoduchý – dosáhnout co nejmenší fragmentace paměti a co nejlepšího využití paměťového prostoru.

- jednoduché operační systémy, které zpracovávají v daný moment pouze jednu úlohu, mají i nekomplikovanou správu operační paměti. **Základní funkce všech systémů správy paměti jsou následující:**

- udržování přehledu o volné a obsazené paměti
- přidělování paměti podle požadavků
- uvolnění bloku paměti
- poskytování informací o stavu paměti (kolik celkem a kolik volné)

- někdy je možné volit (zjistit) alokační strategii. Další funkce jednoduché OS nemají a v podstatě je ani nepotřebují. Pokud například program potřebuje více operační paměti, než je fyzicky v systému přítomno, lze to řešit i s pomocí tak jednoduchých funkcí jak bylo výše uvedeno. Charakteristickou technickou je metoda překryvných segmentů (overlays). Program se rozdělí na části, které nemusí být najednou v operační paměti a kořenový segment, který je v paměti trvale a natahuje překryvné segmenty z vnějšího média podle potřeby. Rozdělení programu na samostatné nezávislé části je záležitostí programátora (ostatně, ví nejlépe co je v jeho programu nezávislé). Mechanizmus provedení této techniky musí být podporován prostředky programovacího jazyka (operační systém tento proces v žádném případě neřídí).

- podstatným problémem správy paměti je relokace adres. Instrukce procesů identifikují operandy v operační paměti pomocí adres. Číslo, které je přeneseno na adresní část systémové sběrnice v momentě vykonání instrukce se nazývá fyzická adresa. Paměťový adresový prostor je lineární. Proces získání finální, fyzické adresy (výpočet efektivní adresy) může být poměrně komplikovaný. Další komplikací je fakt, že v momentě překladu programu do strojového kódu, nejsou hodnoty adres známy, protože se neví, do jaké části operační paměti bude kód při vykonání skutečně zaveden. Veškeré v programu uvedené adresy (instrukcí, proměnných), musí být proto zapsány takovým způsobem, aby je bylo možné při zavádění (vykonání) snadno a tudíž rychle modifikovat (relokovat)

- jiným zásadním problémem je ochrana částí paměti před nežádoucím zásahem. V systémech umožňujících souběžné zpracování se toto stává problémem kardinálním. Nežádoucím zásahem se rozumí jakákoli neautorizovaná změna. Nezáleží na tom, zda je úmyslná či neúmyslná (v důsledku chyby). Cílem veškerého snažení je správnost prováděných operací, důvěryhodnost, spolehlivost a robustnost operačního systému.

- **-v zásadě známe dva základní způsoby ochrany bloků operační paměti:**
-

metoda zámků a klíčů

- poprvé byla použita v Systému 360 Ibm.

  - **1. varianta (rozdělení paměti na bloky)** - paměť byla rozdělena na bloky o délce 2kB a každému bloku byl přidělen čtyřbitový ochranný kód (klíč). V registru stavového slova programu (PSW) byla čtyřbitová kombinace (zámek). Pokud se program pokoušel přistupovat k bloku, jehož klíč neodpovídal zámku, byl hardwarově zablokován.
  - **2. varianta (použití registrů)** - použití speciálních registrů zvaných báze a limit. Při inicializaci programu (procesu) se do registru báze zanese počáteční adresu přiděleného adresového prostranství a do registru limit délka (velikosti) prostranství.

- výhodou tohoto způsobu ochrany je, že použití báze řeší i problém relokace adres – finální adresa se získá jakou součet relativní adresy (od začátku bloku, offset) a báze.

- nevýhodou je nutnost provést jedno sčítání a jedno porovnání při každém přístupu k operační paměti (časově náročné – zjednodušenou variantu používá Intel)

metody evidence použití operační paměti

  - **1. varianta (metoda bitové mapy)** - metoda bitové mapy předpokládá rozdělení pamětí na bloky konstantní délky – alokační jednotky. S každou jednotkou je spojen jeden bit v bitové mapě.

Nastavení bitu signalizuje, že blok je použit, nulová hodnota udává, že je volný. Volba velikosti alokační jednotky je velmi důležitý parametr. Je-li to jen několik bajtů je bitová mapa veliká (spotřebuje hodně paměti). Pokud je alokační jednotka několik kilobajtů je bitová mapa poměrně malá, ale přidělují se často bloky zbytečně velké, což znamená plýtvání paměti. Jistou nevýhodou této metody je poměrně složitý postup při vyhledávání volné bloku paměti určité velikosti.

- **2. varianta (spojový seznam)** - druhou možností je použití zřetězeného seznamu (spojový seznam). Což je seznam, jehož datové položky jsou v paměti rozptýleny, avšak každá nese informaci, kde je následující položka. Používá se seznam (většinou) tříděny podle adresy.
  - **každá položka obsahuje alespoň následující pole:**
    - charakteristika bloku
    - délka bloku
    - adresa následující položky seznamu
  - třídění podle adresy je výhodné při uvolnění bloku
  - třídění podle velikosti bloku je výhodné při přidělování podle požadavků
  - uvolnění bloku realizuje složitější algoritmus – každý blok má dva sousedy (s výjimkou prvého a posledního) a jsou tedy **čtyři možné varianty:**
    - obsazeno na obou stranách, tedy díra
    - volno vlevo
    - volno vpravo
    - volno na obou stranách, tedy vznikne jeden souvislý blok místo tří původních

virtuální paměť

- velikost stránky a rychlost přepočtu virtuální adresy na reálnou jsou zásadní parametry ovlivňující efektivitu stránkování. Současné mikroprocesory mají šířku nejméně 32 bitů. Při velikosti stránky 4KB (12 bitů) máme 1 milion stránek. Potřebujeme-li pro definic jedné položky (stránky) 4 bajty je to celkem 4 MB paměti. Vzhledem k tomu, že každý proces v systému aktivovaný, má své vlastní adresové prostranství, je takto spotřebované značné množství paměti. Zbytečně. Pokud má reálná (fyzicky přítomná) paměť velikost například 512 MB, více než 85% položek nevyužito. Jistý problém představuje i to, kam bude tabulka stránek uložena. Musí se nacházet trvale v paměti, v části, která není vytěsňována na vnější médium. Omezit velikost tabulky je možné pomocí víceúrovňového překladu – tabulka stránek se rozdělí na adresář tabulek a jednotlivé tabulky stránek. Adresář tabulek je poměrně nevelký a je proto možné, aby po většinu času byl uložen v operační paměti. Dále, vzhledem k tomu, že procesy většinou nevyužívají celé virtuální adresové prostranství, je možné přidávat položky do adresáře tabulek podle potřeby (omezení režie stránkování)

- existuje i jiná metoda překladu adres - inverzní tabulka stránek – tabulka je organizována podle rámů a nikoli podle stránek (pro jednoznačné přiřazení adresy se používá hardwarově realizovaná hash funkce). Používají procesory firmy IBM (S 370, PowerPC)

- bohužel, víceúrovňový překlad má i jednu velmi nepříjemnou vlastnost – na jednu transformaci adresy je třeba minimálně dvakrát (záleží na počtu úrovní) číst z paměti. Protože adresa musí být transformována při každém přístupu do paměti. Postrádá užití stránkování bez hardwarové podpory prakticky smysl.

- mechanizmus výměny stránek je aktivován v případě výpadku stránky (Page Fault – požadovaná stránka není v operační paměti). V případě, že v systému ještě existuje volná paměť (rám), je reakce jednoduchá – stránka je natažena z vnějšího média a uložena do prvého volného rámu. Pokud volná paměť není k dispozici, je třeba nejprve některý rám uvolnit. Otázkou je, který. Patrně ten, jehož obsah byl nejdéle nepoužíván.

algoritmus nru (not recentyl used)

- s každým rámem jsou asociovány dva bity – R a M. Bit R je nastaven, když je stránka použita (čtení nebo zápis). Bit M je nastaven, když je obsah rámu modifikován. Bity musí být nastaveny při každém přístupu k rámu, je žádoucí hardwarová realizace. V pravidelných časových intervalech (třeba 20 msec) jsou všechny bity R vynulovány. Takto získáme čtyři skupiny rámů: 0 – nepoužitý a nemodifikovaný. V případě výpadku stránky se náhodně uvolňuje rám ve skupině 0,1,2,3. Algoritmus je jednoduchý, poměrně snadno realizovatelný, ale poskytuje méně uspokojivé výsledky.

algoritmus lru (the last recentyl used)

- metoda výměny založená na předpokladu, že dlouho nepoužitá stránka bude ještě dlouho nepotřebná a proto je možné ji odstranit. Realizace algoritmu je nákladná na zdroje). Předpokládá setříděný svázaný seznam (frontu) všech stránek v paměti přítomných. V čele fronty je právě použitá a v týlu nejdéle nepoužitá stránka. Při každém přístupu do paměti je třeba frontu aktualizovat. Softwarová realizace není použitelná. Možné hardwarové řešení předpokládá 64bitovbý čítač, který je vždy po vykonání instrukce inkrementován. Při přístupu ke stránce je stav čítače kopírován do příslušné položky tabulky stránek (8 bajtů). Možností realizace je vícero (bitová matice n x n bitů). Excelentní výsledky, ale velmi náročná realizace algoritmu.
-

algoritmus fifo (first in – first out) a jeho modifikace

- principem je fronta – nejstarší stránka je prvá, přidává se na konec. Vyměňuje se stránka, která je prvá. Takto primitivní algoritmus nedává uspokojivé výsledky. Modifikací je verze „druhá šance“ a „hodiny“, což předpokládá přidání bitu R, pro každý rám (podobně jako u metody NRU)

ZAŘÍZENÍ V/V

- **zařízení V/V můžeme rozdělit na dvě základní skupiny:**

znakově orientované

- data se přenášejí po znacích (character devices)
- nejmenší adresovatelnou jednotkou je jeden znak (bajt)
- klávesnice, sériový komunikační adaptér

blokově orientované

- data se přenášejí po blocích (block devices)
- nejmenší adresovatelnou jednotkou je jeden blok (řada bajtů)
- disková nebo pásková jednotka

- existují samozřejmě i zařízení, které se nedají zařadit ani do jedné z uvedených kategorií (například časovač nebo hodiny reálného času). Nicméně, tento přístup umožňuje značně zjednodušit správu zařízení V/V

zařízení V/V se obecně skládá ze dvou částí:

- mechanické
- elektronické

- modulární členění na řídící kontrolér, které obsahuje obecnou části řídící elektroniky a mechaniku (s nezbytnou speciální elektronikou) je užitečné, protože umožňuje spravovat podobné zařízení stejným způsobem. Rozhraní mezi kontrolérem a mechanikou je (takřka vždy) standardizováno (EIDE, PCI). Takové řešení zhusta dává příležitost připojit k jednomu kontroléru několik (běžně 2-8) mechanik. Nezanedbatelnou výhodou je i láce (levnost) produktů.

- kontrolér je řídící jednotka zařízení vstupu-výstupu. Rozhraní mezi kontrolérem a mechanikou zařízení většinou pracuje na velmi nízké úrovni. Příklad u pevného disku, který je klasicky formátován (sektor 512 bajtů) se při operaci čtení přenáší sériový proud dat o délce 4568 bitů – hlavička (preamble), data a kontrolní suma (ECC). Úkolem kontroléru je konvergovat proud bitů na bajty, které uloží do vyrovnávací paměti. Zkontroluje, pomocí ECC, zda byly data přeneseny

bezchybně. Pokud došlo k chybě, pokusí se jí opravit. Následně dodá data a zprávu o výsledku operace (nebo oznámí selhání). Každý kontrolér potřebuje registry pro komunikaci s nadřízenou jednotkou (CPU). Většinou se jedná o registr datový, řídící, stavový. Datový registr obvykle bývá jeden (pro čtení i zápis), řídích a stavových může být vícero.

- způsoby jak CPU přistupuje k registrům zařízení vstupu-výstupu jsou v zásadě dva:
  - registry jsou přímo mapovány do adresového prostoru operační paměti (DEC, PDP-11)
  - registry mají vlastní adresový prostor, do kterého se přistupuje pomocí příslušeného typu instrukcí (IBM, S360)

- existuje i kombinované řešení – datové registry jsou mapovány do operační paměti a řídící a stavové registry mají samostatný adresový prostor a příslušné instrukce pro komunikaci (Intel Pentium – datové registry v oblasti od 640 KB – 1 MB). Každé uvedené řešení má samozřejmě své výhody i nevýhody. Některé programovací jazyky (C, C++) nemají instrukce vstupu-výstupu, ktakže je nutné napsat část kódu v asembleru. To je zjevná nevýhoda. Základní výhodou registrů mapovaných přímo do operační paměti je, že s nimi lze pracovat jako s libovolnou jinou proměnnou. Což znamená, že mohu použít stejné instrukce i stejné mechanizmy řízení přístupu.

metody přesunu dat

- známe tři základní metody pro přesun dat mezi procesorem a zařízením vstupu-výstupu. Každá metoda má své klady a zápory. Aby věc byla komplikovanější, v důsledku evoluce se priority přesouvají.

  - **programová obsluha** - představuje nejjednodušší variantu přenosu dat. Je vhodná pro jednotky, které pracují přibližně stejnou rychlostí. **Postup je jednoduchý:** řídící jednotka vyšle požadavek a v čekací smyčce čeká na jeho splnění. Podřízená jednotka provede žádané (nebo neprovede, pokud z nějakého důvodu nemůže) a oznámí, nastavením příslušných bitů ve stavovém registru svůj stav (nebo důvod neprovedení).
  - **obsluha pomocí přerušení** - je poněkud komplikovanější. Začátek operace je stejný jako v předchozím případě. Řídící jednotka vyšle požadavek, ale nečeká na jeho splnění a pokračuje v činnosti (tedy, může-li, má-li jinou práci). Když podřízená jednotka splní požadované, vyvolá přerušení. Rutina obsluhy přerušení provede potřebné (třeba přesune data z/do vyrovnávací paměti) a signalizuje ukončení operace. Tento způsob obsluhy je zhusta časově méně náročná než programová obsluha (úměrně rozdílu rychlosti práce jednotek). Jistou komplikací je nutnost nejprve iniciovat (nastavit) přerušovací systém. Naopak vítanou výhodou může být fakt, že podřízená jednotka může být schopna (je-li tak konstruována) aktivně přivolat pozornost nadřízené jednotky (pokud to potřebuje).
  - **přímý přístup do paměti** - představuje nejelegantnější, ale také nejkomplexnější způsob datových přenosů mezi zařízením V/V a procesorem. Základním předpokladem je existence kontroléru přímého přístupu do paměti (DMA). Ten může být součástí základní desky (nejčastěji) nebo samotné periferie. Přenáší se vždy blok dat (lze přenést i jeden bajt, postrádá to však smysl). **Operace je zahájena naprogramováním kontroléru:** kolik se má přenést, odkud a kam. Další činnost už řídí kontrolér. **Přenos se v zásadě realizuje dvěma způsoby:**
    - řízení nad sběrnicí převezme kontrolér vždy jen na přenos jednoho slova – **metoda kradení cyklů** (cycle stealing) – procesor se o sběrnici dělí s kontrolérem DMA.
    - při druhém způsobu převezme kontrolér DMA řízení nad sběrnicí na celou dobu **přenosu (bloku) dat**. Blok dat se přenese „najednou“, procesor je odstaven od sběrnice na delší dobu, což může způsobit jistí problémy (burst mode – shlukový přenos, také intervalový)

Správa zařízení

- je poměrně rozsáhlá a různorodá komponenta operačního systému. Systém spravuje velmi rozdílná zařízení, přesto (nebo právě proto) je třeba dosáhnout toho, aby se tak dělo jednotným způsobem. Software je proto rozdělen do vrstev, které spolu komunikují standardním způsobem. Každá vrstva plní určitou funkci a poskytuje služby vrstvě sousední. Tak lze dosáhnout toho, že většinu činností, které bezprostředně souvisí s hardware je

možno soustředit do jedné (nejspodnější) vrstvy. Vyšší vrstvy pak na hardware nezávisí, jejichž činnost má víceméně obecnější charakter a může být proto společná pro vícero zařízení stejného typu. Některé funkce, které mají obecný charakter:

- **jednotná identifikace** všech zařízení pomocí znakového řetězce (A.,C.,PRN,COM – Uniform Naming)
- **správa chybových stavů**
- **správa vyrovnávacích pamětí** (blokově orientovaná zařízení nepřenášejí obvykle data přímo na místo určení, ale do vyrovnávacích pamětí a teprve následně, až je celý blok dat připraven se předá uživateli)

- aplikační program (nejvyšší vrstva) používá pro přístup k zařízení V/V služby operačního systém (například, přiřazení zařízení, čtení, zápis). Operační systém přijaté komplexní požadavky rozdělí na úkony obecného charakteru a speciální (na hardware závislé), které následně předá ke splnění ovladači příslušného zařízení. Ovladač (Device Driver) postupně podobně rozdělí požadavek na elementární operace, které následně předává řízenému zařízení. Po provedení (splnění) požadavku informuje o výsledku požadovaného.

- nejspodnější vrstvou je obsluha přerušení (Interrupt Handler). Vykonává relativně triviální činnost. Ovladač je v podstatě proces, který čeká na úkoly. Po spuštění operačním systémem, zadá požadavky příslušnému zařízení a je pozastaven (nemůže pokračovat, čeká na data). Rutina obsluhy přerušení potom jen nastaví příslušný semafor, že obsluha požadavku je skončena. Ovladač má splněny podmínky pro pokračování, tudíž pokračuje.

- přerušení je obecně nepříjemná událost. Na klasických procesorech se vznik přerušení testuje obvykle v poslední fázi vykonání instrukce, kdy je stav procesoru přesně definován. U procesorů, které
- zpracovávají instrukce proudovým způsobem je situace mnohem komplikovanější (u superskalárních je ještě mnohem horší). Vzniká tak zvané nepřesné přerušení. Navíc, velmi vzrostly náklady na ošetření přerušení. Musí se vyprázdnit proudová linka, instrukční cacha, přepnout stránkování (TLB a načíst stránky).

SOUBOROVÝ SYSTÉM

- v současnosti nedokážeme vyrobit operační paměť dostatečně levnou a dostatečně velkou, která by neztrácela data po vypnutí napájení. Procesy zpracovávají data, která musí někde získat a která je nutno po ukončení práce (většinou) uchovat pro další použití.

- **systém uchovávání dat výpočetních systémů musí splňovat tři základní požadavky:**
  - musí být schopen uložit značná množství dat
  - uchovat je přiměřeně dlouhou dobu
  - umožnit k nim (vícenásobný) přístup.

- data uchováváme v souborech (File). Soubor je (pojmenovaná) posloupnost záznamů, se kterou lze manipulovat jako s celkem (přemístit, kopírovat, smazat). Soubory se většinou nacházejí na vnějších nosičích (zhusta discích). Správa souborového systému je klíčová složka operačního systému. Správou rozumíme softwarové procesy, které se zabývají umisťováním a organizací souborů na nosiči, přidělují jim atributy a přístupová práva, řídí a kontrolují přístup k datům.

- soubor je abstraktní mechanizmus umožňující uložení a znovupoužití dat. Každému souboru je přiřazen název (což je abstrakce), který zajišťuje identifikaci dat v systému. Pravidla pro vytváření jmen souborů se systém od systému liší, nicméně řetězec osmi znaků je většinou postačující. Některé systémy rozlišují mezi velkými a malými písmeny (Unix) a některé nikoli (DOS). Délka řetězce znaků může být různě dlouhá, až 255 znaků (Windows). Nejznámější je konvekce 8.3 – dvousložkový název souboru, maximálně osm znaků název a tři znaky rozšíření (původně DOS převzato rannými verzemi Windows). Rozšíření blíže specifikuje použití souboru

dat, v některých systémech je to jen konvence (Unix), v některých nikoli (Windows – provede akci podle (registrovaného) rozšíření).

- každý datový soubor má jistou strukturu.

v zásadě máme tři možnosti:

- **nestrukturováno** znamená, že data jsou uložena jako proud bajtů. Je zcela na libovůli uživatele, jak bude s daty manipulovat. Což však znamená, že mu v tom operační systém nika nepomůže. Tedy maximální flexibilita minimální pomoc.
- druhou možností je, že data jsou uložena jako **sekvence záznamů** pevné délky – malé omezení i malá pomoc.
- třetí možností je **struktura organizovaných záznamů** (v případě proměnné délky). Soubor záznamů je uspořádán (setříděn) podle klíče, což je pevná část záznamu (vždy na stejném, přesně definovaném místě). Způsob přístupu k datům je pevně dán, pomoc operačního systému je velmi vysoká (a tedy i nákladná)

- soubor je identifikovatelná posloupnost záznamů, které lze zpracovávat jako celek. Slouží k trvalému uložení informace. Systém souborů je sada souborů včetně informace pro jejich nalezení.
- **má tři základní složky:**
  - lokátor
  - data
  - informace o nosném médie

  - správa souborů je proces zajišťující umístění souborů na paměťovém médiu, kontrolou přístupu a případně i jiné funkce servisního charakteru.
- **nosné médium svou konstrukcí určuje způsob přístupu k uloženým datům:**
  - postupný (sekvenční,mg.páska)
  - přímý s přístupem k libovolné datové položce (pevný disk)

lokátor souboru

- slouží k identifikaci souboru
- specifikuje umístění dat souboru na médiu. Většinou obsahuje popisovač souboru, ve kterém jsou uloženy základní informace o souboru – název, velikost, datum a čas vzniku a poslední modifikace, atributy atd.

struktura souboru

- je fyzické uspořádání určující metodu přístupu k uloženým datům
- **základní struktury jsou tři:**
  - nestrukturováno
  - pevná délka záznamu
  - organizované záznamy s pevnou nebo proměnnou délkou

- první dvě struktury podporuje většina OS, poslední pouze některé

- struktura souboru určuje také náklady, které musí OS vynaložit na manipulaci se souborem. Data jsou do souboru uložena v určitém logickém formátu (text, grafická data). Název souboru často naznačuje, jaký formát uložená data mají (přípona, koncovka názvu souboru). Logický formát (typ) souboru většinou určuje způsob manipulace se souborem (předurčení souboru – spustitelný, pro překladač jazyka C, atd.)

implementace systémového souboru

- **jak je realizován souborový systém:**
  - způsob organizace souborů a adresářů (složek) na médiu
  - mechanizmus spravování paměťového média

- efektivnost a spolehlivost práce systému

- souborový systém je uspořádán hierarchicky (DOS strom, UNIX acyklický graf). Základem je adresář, což je seznam (rejstřík) souborů a podadresářů uložených na médiu. Důvodem pro takové uspořádání je přehlednost a robustnost řešení. Správa paměťového média řeší především problém jak rychle a efektivně rozdělovat souborům místo na médiu. Přidělení souvislého prostoru je jednoduché i efektivní. Bohužel v moment vzniku souboru je zřídka známa finální velikost souboru. Jiným řešením je zřetězený seznam nebo ještě lépe zřetězený seznam s použitím indexů (také alokační tabulka). Což odstraňuje problémy při přímém přístupu k datům.

typy souborů:

- většina operačních systémů podporuje obyčejně několik typů souborů. Základními typy jsou řádný soubor (regular) a knihovna (directory). Knihovny (také složky) jsou systémové soubory, které slouží k vytvoření struktury souborového systému. Řádné soubory můžeme rozdělit na textové a binární. V zásadě lze na textové soubory pohlížet jako na data, která je možno přímo vytisknout. Což dnes platí jen omezeně. Tiskárna nemusí kódování znaků rozumět (dokud se používalo jen ASCII kódování byla situace jednoduchá). Takže, pokud soubor není textový je binární.

textové soubory

- soubory ve formátu PS a PDF (tyto soubory zcela jistě nejsou sekvence znaků) lze na vhodné tiskárně přímo tisknout. Vhodnou se rozumí vhodný hardware a vhodný ovladač. Hardware je dáno a většinou umí jak PostScript tak PCL (nebo PDL), ovladač si zvolíme.

binární soubory

- jsou, zjednodušené řečeno, sekvence bajtů. Zvláštní postavená mají tak zvané spustitelné (executable) soubory. Jedná se o binární soubory, které mají určitý specifický formát. Každý operační systém musí podporovat (rozeznat) minimálně jeden typ souboru – jeho vlastní (nativní) spustitelný soubor (některé rozeznají i vícero typů). Formát spustitelného souboru je v různých operačních systémech sice obecně různý, ale základní struktura je podobná: hlavička, text, data, seznam relokací a tabulka proměnných. Hlavička obsahuje identifikaci spustitelného souboru (magické číslo), délky jednotlivých částí (text, data, relokace, proměnné) a adresu vstupního bodu. Spustitelný soubor může být také tvořen (složen) několika cílovými moduly, každý modul má potom svou hlavičku.

metody přístupu

- **sekvenční přístup** - znamená, že k datům uloženým na médiu se dostaneme postupně. Když chceme přečíst n-tou položku, musíme přečíst všechny položky předcházející. Můžeme také říci, že k datům se vždycky přistupuje tak, jak byla zapsána. Klasickým případem jsou data uložena na magnetické pásce.
- **přímý přístup** (náhodný, random) - umožňuje číst data z určitého místa podle potřeby (tedy přímo nebo „náhodně) bez nutnosti číst položky předcházející.

SPRÁVA UŽIVATELŮ

- Myšlenka nechat s jedním počítačem pracovat najednou více uživatelů pochází již z raných časů sálových počítačů, kde se muselo mnoho uživatelů podělit pouze o malé množství prostředků. Počítače byly dříve především na univerzitách, v menším množství pak ve větších fabrikách. K jejich obsluze sloužily terminály, které sestávaly z klávesnice a monitoru a byly propojené s hlavním počítačem.

- Schopnost operačního systému spravovat více uživatelů je dosaženo pomocí souboru různých opatření. Pod tím si člověk může představit např. správu privátních přístupových práv a osobních předvoleb pro každého uživatele. Od každého víceuživatelského systému se takové očekává přehled všech přístupů, který je dělen dle času a uživatelů.

Multitasking

- Oproti Multitaskingu nepředstavuje pojem „víceuživatelský“ vždy souběžnou práci několika uživatelů. Dnešní víceuživatelské systémy jsou multitaskingu schopny, ale dříve byly avšak také počítače, kde mohli uživatelé pracovat pouze v určitých pracovních oknech, která následovala vždy po sobě (neběžela současně)

- Termínem multitasking (z angličtiny, multi = mnoho, task = úloha) se v informatice označuje schopnost počítače, resp. operačního systému provádět (přinejmenším zdánlivě) několik úloh současně. Dnešní operační systémy jsou typicky víceúlohové – sem patří např. Microsoft Windows či Linux. Naopak MS-DOS je příkladem jednoúlohového systému, na kterém vždy běží pouze jediný program a teprve po jeho ukončení je možno spustit jiný.

- Multitasking může být realizován mnoha způsoby. Základní dělení je na skutečný multitasking, kdy je hardware počítače opravdu schopen v každém okamžiku zpracovávat více úloh současně, a na zdánlivý multitasking, kdy se dojmu současného běhu více programů dosahuje tím, že se tyto programy velice rychle v běhu střídají, ale v každý jednotlivý okamžik běží pouze jediná úloha. Opravdu 100% skutečný multitasking se v praxi příliš neobjevuje, běžné operační systémy podporují druhou zmiňovanou techniku, ale pokud je počítač vybaven více procesory, jsou úlohy mezi tyto procesory rozděleny tak, aby alespoň některé úlohy mohly běžet současně.

- Podle způsobu přidělování a odebírání časových kvant se rozlišují dva základní způsoby zdánlivého multitaskingu: kooperativní multitasking a preemptivní multitasking.

DOS

- textově orientovaný operační systém osobních počítačů IMB PC

různé varianty:

- MS-DOS
- PC (IBM)
- Dr. (Digital Research)

celkem 6 základních verzí:

- **verze 1**
  - 1981
  - 4000 řádků v asembleru = 12 kB
  - tvořen 3 programy:
  - diskový a souborový manažer
  - diskový a znakový systém vstupu a výstupu
  - příkazový interpret
  - drivery pro standardní zařízení V/V byly součástí HW (ROM-BIOS)
  - podpora pouze jednostranných disků (160 kB)
- **verze 2**
  - 1983
  - 20 kB řádků v asembleru
  - kompletně přepsáno
  - vzorem UNIX
  - přidána podpora pro 360 kB disky
- **verze 3**
  - 1984
  - 40 kB řádků v asembleru
  - podpora pro HDD 10 MB
- **verze 4**
  - rekce na neúspěch OS/2
  - podpora pro HDD do 2 GB
- **verze 5**
  - překonání hranice 640 kB RAM využitím rozšíření paměti

shrnutí:

- technicky zastaralý, uživatelský "nepřátelský OS"
- za 15 let prodáno 5 miliónů legálních kopií

základní rysy:

- po spuštění PC je řízení předáno interpretu příkazů
- interpret příkazů se skládá ze tří částí:
  - obsluha přerušení, kritická chybová hlášení, rutiny pro spuštění procesu typu potomek
  - rutiny pro zpracování dávkových souborů a procedury přidělování paměti pro zavádění programů
  - realizace interpretu příkazů OS (provádí veškeré interní příkazy)

- dva typy spustitelných souborů (*.com, *.exe)
- com – přiděleno maximálně 64 kB operační paměti
- správa paměti
- paměť je v zásadě omezena na 640 kB – lze obejít pomocí driverů paměti
- přístupové segmenty 64 kB
- paměť přidělována v paragrafech (16 B)

souborový systém

- hierarchické uspořádání
- lokátor – FAT
- názvy o délce 8+3 znaků ber rozlišení velkých a malých písem

WINDOWS

- Microsoft Windows je řada grafických víceúlohových operačních systémů společnosti Microsoft. Až na výjimky jsou určené pro osobní počítače (PC) s procesory Intel kompatibilními.

16bitové

- Microsoft uvedl první Windows (odtud i název - okna) na trh v roce 1985, tehdy jen jako nadstavbové grafické uživatelské prostředí (GUI) nad svým tehdejším standardním 16bitovým operačním systémem MS-DOS. První verzí, která došla mezi dobovou konkurencí podobných produktů, byla verze 3 (1990), mezi uživateli se však vyšší oblibě těšila až verze 3.1 (1992). Verze Windows 3.11 pak přinesla sdílení disků (označení Windows for Workgoups) a je doposud mnohde používána na starších počítačích.

- Změna vzhledu (výměna správce oken) ve Windows 95 (1995) přinesla intuitivnější ovládání a vyšší zájem uživatelů. Přímo do instalace systému bylo integrováno dříve samostatně dostupné rozšíření Win32s pro podporu 32bitových aplikací. Stejně tak i přímou podporu protokolu TCP/IP, což znamenalo umožnění přímého přístupu k Internetu bez instalace doplňků od jiných dodavatelů (např. Trumpet Winsock), ale i nebývalý nárůst požadavků na operační paměť RAM. Dalším vylepšením byla verze Windows 98 (1998) a posléze Windows Me (Millennium Edition - vydání k novému tisíciletí) v roce 2000, což byla i poslední verze této řady.

- I když byla zřejmá podpora 32bitových aplikací a ovladačů, některé části operačního systému zůstávaly 16bitové, což vedlo k nestabilitě systému, který mohl být snadno ohrožen nesprávně fungujícím programem. Na rozdíl od svého konkurenta OS/2 totiž tato větev nikdy zcela nevyužila možnosti 32 bitových procesorů, čímž byli uživatelé více nuceni přejít na plně 32 bitové systémy, které byly prodávány souběžně.

32bitové

- V roce 1993 byla uvedena do prodeje nová řada operačních systémů firmy Microsoft, která byla založena na plně 32bitovém jádře NT (New Technology), která byla původně míněna jako plně profesionální produkt pro desktop. Prvním zástupcem byly Windows NT 3.1, verze 3.5 (1994) a verze 3.51 (1995), která byla považována za stabilní.

- Úspěch nového grafického vzhledu převzatého z Windows 95 převzaly Windows NT 4.0 (1996). Dále následovaly Windows 2000 (2000), Windows XP (2001, eXPerience - zkušenost, zážitek), Windows Server 2003 (2003) a Windows Vista (pro výrobce na konci roku 2006, pro uživatele až v roce 2007).

- Do Windows 2000 byly prodávány verze pro server i desktop současně, v poslední době jsou již odděleny. Za serverovou edici se tak považují Windows Server 2003, kdežto Windows XP a Windows Vista jsou určena pro desktop.

64bitové

- První 64bitovou verzí pro procesory Intel byly po dlouhých odkladech Windows XP vydané v roce 2005. Nicméně již při vydání Windows XP existovala verze pro platformu Itanium (IA-64).

- Dřívější verze Windows NT existovaly i pro procesory Alpha, ale ty tento 64bitový procesor přepínaly do 32bitového režimu, což vedlo k degradaci výkonu a posléze i k předčasnému ukončení podpory ze strany firmy Microsoft.

UNIX A LINUX

- UNIX (původně Unics, podle Unary Information and Computing Service) je víceúlohový a víceuživatelský operační systém, který je implementován na mnoha hardwarových platformách.

- Na bázi UNIXu je založeno velké množství dnes používaných operačních systémů na serverech, na pracovních stanicích a na osobních počítačích.

- UNIX byl vytvořen Kenem Thompsonem a Dennisem Ritchiem roku 1969 v Bell Laboratories, kde byl vyvíjen až do konce sedmdesátých let. Poté probíhal vývoj UNIXu ve dvou hlavních větvích:
  - pod vedením AT&T: SYSTEM III, SYSTEM V
  - na Kalifornské universitě v Berkeley: BSD Unix

- Název vznikl (v původní podobě Unics) jako protiklad k systému z roku 1960, Multics (Multiplexed Information and Computing Service). Později se zápis změnil na Unix či UNIX.

Charakteristika

- Systém založený na Unixu je charakteristický tím, že je:
  - jednoduchý
  - víceúlohový (implementuje multitasking)
  - víceuživatelský
  - (téměř) vše je soubor
  - důraz je kladen více na vztahy mezi programy, než na programy samotné (dělba práce)
    - sada jednoúčelových nástrojů, které dobře plní svůj specifický úkol
    - propojování nástrojů do kolon
    - využívání hotových programů jinými programy
  - orientovaný na zpracování textů

Filosofie sytému

- Operační systém Unix se většinou symbolicky znázorňuje jako kulovitý útvar, uprostřed s jádrem, obaleným různými vrstvami:
- **jádro (kernel)** - slouží zejména pro obhospodařování hardware počítače, tj. pro přidělování jeho prostředků (zdrojů) různým uživatelům a různým úlohám. Zejména jde o přidělování:
  - času jednoho či více procesorů
  - operační paměti
  - diskového prostoru, ve kterém organizuje souborový systém
  - různých dalších periferií
- **interpret příkazů (shell)** – slouží ke komunikaci systému s jedním či více uživateli, zpravidla pomocí příkazové řádky
- **aplikace (aplikační software)** – nejrůznější programové vybavení, sloužící zejména jednak pro administraci (správu) vlastního systému, jednak pro vlastní užitečnou práci
- **grafické uživatelské** rozhraní – GUI, Graphic User Interface. S nástupem výkonnějších počítačů byla pro pohodlnější práci a pro zpracování grafických úloh vytvořena řada různých grafických prostředí. Nejznámější z nich je X Window System (1984) (neplést se známějšími Microsoft Windows - verse 1.0 z roku 1985).

Standardizace

- Bouřlivý vývoj různých klonů UNIXu si vynutil vznik různých standardů. Nejznámějšími jsou POSIX a Single UNIX Specification.

Linux

- Linux je jádrem několika počítačových operačních systémů. Je známým příkladem svobodného softwaru a vývoje open source softwaru – narozdíl od proprietárních operačních systémů jako Windows či Mac OS je celý jeho zdrojový kód volně k dispozici pro veřejnost a kdokoli jej může svobodně používat, upravovat a dále distribuovat.

- Ačkoliv termín Linux značí Linuxové jádro, často se používá pro označení celých unixových operačních systémů (známých jako GNU/Linux), které sestávají z Linuxového jádra a zároveň z knihoven a nástrojů z projektu GNU, ale i z dalších zdrojů. V nejširším významu GNU/Linuxová distribuce uceleně spojuje základní systém s velkým balíkem aplikačního softwaru, a navíc často zajišťuje uživatelsky přívětivou instalaci a následné aktualizace.

- Zpočátku byl Linux vyvíjen a používán zejména jednotlivými nadšenci. Časem ale získal podporu velkých společností jako IBM, Hewlett-Packard a Novell pro využití na serverech, a poslední dobou získává popularitu i na desktopovém trhu. Zastánci a analytici připisují jeho úspěch nezávislosti na dodavateli, nízkých nákladech, flexibilitě, bezpečnosti a spolehlivosti.

- Linux byl původně vyvíjen pro počítače s procesory architektury i386 (tedy 80386 a kompatibilními). Dnes ale podporuje všechny populární počítačové architektury i mnoho z těch méně obvyklých. Používá se v řadě zařízení od embedded systémů (jako mobilních telefony, roboti či multimediální přehrávače) přes osobní počítače až po superpočítače.

SÍTĚ_OBECNĚ

DATOVÁ KOMUNIKACE

- sítí rozumíme účelové propojení výpočetních systémů. Účelem propojení je sdílení zdrojů (a poskytování služeb). Organizace sítí je většinou směrová (existují i výjimky).

- přenosové prostředky

- komponentami sítě jsou datové stanice, které provádějí přenos informací ve formě dat (datovou komunikaci) po přenosové cestě.

přenosová cesta
- jednosměrná (přenosový kanál)
- obousměrná (kruh)

signál
- je nositelem informace
- analogový
- digitální (mění se v definovaných časových okamžicích a může nabývat pouze určitých hodnot).

okruhy (kanály)
- pevné
- přepínané (komutované)

fyzická realizace přenosové cesty
- drátová komunikace (kovový vodič, skleněné vlákno)
- komunikace bezdrátová (vzduchem)

kabely

- jak je třeba kabely pokládat uvádí specifikace EIA/TIA 568-570 a 606. Správně provedená kabeláž je základním předpokladem dobře fungující sítě. V nových budovách se provádí jako strukturovaná kabeláž. Postatou je jednotné provedení rozvodů a společné umístění spolu souvisejících kabelů, což má umožnit snadnou případnou rekonfiguraci struktury.

symetrický kabel (kroucený pár – Twisted pair)
- je složený z párů zkroucených vodičů.
- **Používají se dva typy:**
    - stíněný STP
    - nestínění UTP

- nestíněný kabel může mít 2,4,22,24, nebo 26 párů vodičů (specifikuje AWG – American Wire Gauge). **Kvalitu nestíněných kabelů vyjadřuje kategorie, do které je kabel zařazen:**
    - 1=žádná výkonností kriteria
    - 2=do 1Mhz (telefonní rozvody)
    - 3=do 16 Mhz
    - 4=do 20Mhz (token ring)
    - 5=do 100 Mhz
    - existují i předběžné definice kategorie 6 (200 Mhz) a 7 (600 Mhz)

koaxiální kabel
- je tvořen dvěma vodiči v provedení, kdy vnější obaluje vnitřní (většinou měděný vodič), po kterém se přenáší aktivní signál (nesymetrický přenos). Vodiče jsou odděleny dielektrikem a zaizolovány obalem kabelu. Kabel dobře chrání proti elektromagentickému rušení (proti magnetickému méně). Odolnost snižují vyrovnávací proudy

– je žádoucí, aby připojená zařízení byla izolována (minimální styk se zemí). Používá se tlustý (4 vrstvy izolace) a tenký koaxiál. Hodnota impedance kabelu 50 ohmů

optické kabely

- dovolují dosáhnout větších přenosových rychlostí, dokonalého galvanického oddělení, jsou odolné proti rušení a nelze je odposlouchávat. Základním nedostatkem je složitější konstrukce (převod elektrického signálu na světlo) a značně dražší propojovací konektory (a také podstatně dražší nářadí, potřebné pro realizaci spoje)

Multiplexor

- zařízení, která umožňují účinnější využití přenosových, kapacit se nazývají multiplexory. Využívají nějakou metodu sdružování nezávislých datových toků na jedno přenosové médium a jejich zpětného oddělení v demultiplexoru na opačné straně. **Pracují na dvou základních principech:**
    - ***kmitočtové dělení*** spočívá v rozdělení kmitočtového pásma na dílčí podpásma oddělením kmitočtovými filtry a jejich přidělení příslušným kanálům.
    - ***časové dělení*** využívá rozdělení na časové rámce a časové úseky, které tvoří části datových kanálů. Každé připojené zařízení komunikuje pouze v a po určený časový interval. Což předpokládá synchronizaci. Komplikovanější variantou jsou statické časové muliplexory (asynchronní). Přidělují časové rámce na základě momentální potřeb. (asynchronní multiplexory s vyrovnávací pamětí patří do skupiny sdružovacích prostředků a nazývají se koncentrátory)

- přenosové prostředky, které zajišťují spojení nebo propojení kanálů či okruhů, tvoří skupinu spojovacích zařízení. Přepojování je realizováno jako přepínání okruhů nebo jako přepínání paketů.
- Přepínání okruhů lze rozdělit na:
    - časové (dělení čas plus směr)
    - prostorové (spojování, ústředny)

- přenosové prostředky – shrnutí

- stavebními prvky sítí jsou:
- koncová zařízení (datové stanice)
- přenosové cesty (hmotní či nehmotné)
- spojovací (propojovací zařízení)

Kódování:

- kódováním rozumíme převedení zprávy do formy podoby signálu. **Většinou kódováním sledujeme i jiná hlediska:**
- přizpůsobení přenosovým vlastnostem kanálu (modulace)
- zvýšení účinnosti přenosu (odstranění stejnosměrné složky)
- zabezpečení přenosu proti chybám (bezpečnostní složky)

- při přenosu zpráv dochází většinou k několikanásobnému kódování. Zpracovávaná data jsou reprezentována v nějaké číselné soustavě. Z praktického hlediska se používá binární soustava. **Podle složitosti zabezpečení můžeme kódy pro přenos zpráv rozdělit, na:**
    - bezpečnostní
    - jednoduché:
    - rovnoměrné
    - nerovnoměrné
- jsou navrženy tak, aby přenos byl co nejefektivnější. Nejpoužívanější znaky mají nejkratší délky. Nevýhodou přenosu datových posloupností různých délek je nutnost modifikace začátku znaku (synchronizační impulsy). Realizace nerovnoměrných kódů je složitější a mají i nižší schopnost proti rušivým signálům. Tyto nedostatky rovnoměrné kódy nemají (snadné rozlišení části zprávy)

Číslicové signály

- převedení binárního kódu na elektrický signál je důležité nejen z pohledu přesností zpráv, ale jsou sledovány (většinou) i jiné cíle (energetická účinnosti, odolnost proti rušení).

- podle polarity dělíme signály, na:

  - ***unipolární*** - je signál o jedné polaritě. Označuje se také jako RZ (return to zero). Může být realizován s mezerami (snadné rozlišení jednotlivých bitů) nebo bez mezer (pro větší rychlosti). Přiřazení 1/0 je dosaženo buď změnou šířky impulsu nebo přítomnosti či nepřítomnosti impulsu (toto se často používá při přenosu dat pomocí optických vláken)

  - ***polární*** - je tvořen impulsy s kladnou nebo zápornou polaritou. Neobsahuje žádné mezery a nemá nulovou úroveň. Označuje se jako NRZ (non return to zero). Při delší sekvenci log 0 nebo 1 dochází k zhoršení identifikace bit. Což odstraňuje kódování NRZI. Výhodou tohoto způsobu kódování je dobré potlačení stejnosměrné složky signálu.

  - ***bipolární (pseudotermální)*** **-** *zpracovávají se tři napěťové úrovně***:** kladná, záporná a nulová. Logická 1 je reprezentována impulsem libovolné polarity, logická 0 pak nulovou úrovní (mezerou). Pro volbu polarity impulsu logické 1 existují různé algoritmy (časové nebo poziční řízení polarity). Tento způsob kódování nejlépe potlačuje stejnosměrnou složku signálu (při pozičním řízení polarity a sudého počtu přenášených bitů je nulová)

Z hlediska funkčnosti rozeznáváme dvě základní třídy zařízení:

- ***koncové datové zařízení*** (DTE, Data Terminal Equipment) - využívají komunikačních služeb pro vlastní činnost, která je jiného charakteru (počítače, terminály, tiskárny).
- ***ukončující datové zařízení*** (DCE, Data Circuit-terminating Equipment) - poskytuje přístup ke komunikačním prostředkům nebo je přímo implementuje. Provádí přesouvání dat a poskytuje rozhraní. Zakončuje a propojuje okruhy (ústředny, přepínače, modemy).

rozhraní

- představuje prostředky a pravidla pro vzájemné fyzické, signálové, funkční a protokolové propojení různých zařízení. Každé rozhraní musí být pokud možno normalizováno. Dosáhne se tak úplné slučitelnosti zařízení různých výrobců. Doporučení CCITT/ITU se týká jen aspektů signálových, funkčních a protokolových. Netýká se to konektorů!! Konektory specifikuje příslušná norma ISO (2110 pro CANNON 25 kolíků).

datové stanice

- v zásadě máme dva základní druhy datových stanic:
  - ***server*** - je počítač poskytující jistý druh služby
  - ***pracovní stanice*** - počítač uživatele, který může služby poskytované obslužnou stanicí (serverem) využívat.

  - tak funguje síť typ klient – server.

- existují také sítě, kde není rozdílu mezi stanicemi (sítě peer-to-peer, rovný s rovným). Každá stanice může poskytovat nějakou službu, kterou mohou ostatní stanice využívat.

výhody:
- úspora specializované stanice
- jejich provoz je většinou levnější (není třeba specialistů při instalaci a správě)

nevýhody:
- omezený repertoár služeb (sdílení souborů, tiskové služby, internet)

  - nedokonalá správa
  - slabé mechanizmy zabezpečení (uživatelé mají větši pravomoci).

- jsou vhodné pro nevelké lokální sítě. Zhruba tak v rozmezí 10 až 20 stanic.

topologie sítě

- popisuje vzájemné uspořádání síťového systému.
- **rozlišujeme dva základní způsoby organizace sítě:**

sběrnicová organizace

- všechny jednotky propojuje společná spojová cesta (stuhová topologie)

výhody:
  - užití jednoho vedení (většinou koaxiální kabel, snadná realizace)
  - dovoluje prosté přidání či odebrání stanice ze soustavy

nevýhody:
  - menší robustnost systému (přerušení kabelu v jednom bodě způsobí kolaps celé sítě).

směrová organizace

- jsou často řešeny jako hvězda (ta má centrální uzel) nebo kruh, případně kombinace uvedeného (stromová struktura, kombinace několika hvězd).

Výhody:
  - menší náchylnost k poruchám (kabelů)
  - jednoduché přenosové protokoly

Nevýhody:
  - podstatně větší spotřeba kabelů (a také práce při jejich pokládání)

OBECNÝ MODEL ARCHITEKTRURY SÍTĚ

- datová komunikace a její řízení představuje složitý systém sestávající z celé řady dílčích problémů a úkolů. Přistoupilo se proto k rozdělení na problémové skupiny nazývaných vrstvy.

- členění do vrstev odpovídá hierarchii činností, které se při datové komunikace vykonávají. Každá vrstva síťové architektury je definována službou, kterou je schopna poskytnout vyšší vrstvě a funkcemi, které vykonává v rámci protokolu. Rozčlenění do vrstev má významné výhody. Nejpodstatnější výhodou je možnost snadné výměny protokolu v rámci jedné vrstvy beze změny ostatních. Funkce vrstvy je množina činností prováděných v dané vrstvě, které jsou charakterizované společným cílem, účinkem nebo účelem. Metoda funkce je pak algoritmus vedoucí ke splnění příslušné funkce. Řízení datové komunikace předpokládá koordinované vykonávání činností ve všech zúčastněných subsystémech. To vyžaduje výměnu údajů mezi stejnolehlými vrstvami komunikujících jednotek, která je řízena protokoly. Souhrn pravidel, formátů a procedur definující výměnu dat mezi dvěma komunikujícími prvky se nazývá protokol.

- síťová architektura je popsána systémem vrstev, služeb, funkcí a protokolů. Je to rozvržení síťových funkcí do jednotlivých vrstev. Jiná než vrstvová architektura se u distribuovaných systémů a sítí ani nepředpokládá. Soustava protokolů je realizována programově. Síťová architektura je dána strukturou síťového (programového) vybavení.

existují dva standardy:

- referenční model OSI (open systems interconnection) od ISO
- X.25 od CCITT

- norma ISO definuje vrstvový model, které je členěn na sedm vrstev (seven-layer reference model). Model popisuje obecný princip propojení otevřených systémů, účel a hierarchické uspořádání jeho vrstev. Konkrétní protokoly jsou zpracovány samostatně. Pokud je síťové vybavení reálného komunikačního systému v souladu s OSI, pak jde o reálný otevřený systém.

- X.25 je standard síťové architektury, která je zaměřena čistě přenosové funkce. Je to tedy standard pro tak zvané veřejné datové sítě, což je digitální obdoba veřejné telefonní sítě. Funguje na principu přepojování okruhů nebo přepojování paketů (příkladem posledního je síť ARPANET, vybudovaná ministerstvem obrany USA v roce 1969)

referenční model osi

- každá vrstva vykonává skupinu jasně definovaných funkcí. Pro svoji činnost využívá služeb sousední, hierarchicky nižší vrstvy. Své služby poskytuje sousední hierarchicky vyšší vrstvě. Vrstva je složena z dále nedělitelných entit, které vykonávají odpovídající funkce – poskytují služby. Komunikace mezi entitami je řízena protokolem.

- poskytování služby se děje prostřednictvím přístupového bodu služby

služby dělíme na:

- povinné
- volitelné provozovatelem (nemusí být poskytovány)
- volitelné uživatelem (poskytované na žádost)

- interakce subjektů v rámci služby se děje pomocí služebních primitivů

celkem jsou čtyři:

- žádost (vyvolání nějakého postupu)
- oznámení (indikace že k vyvolání došlo),

- odpověď (završení vyvolaného postupu)
- potvrzení (o ukončení).

- protokol je definován jako množina syntaktických (formáty zpráv, příkazy a odpovědi) a sémantických (používání příkazů a odpovědí) pravidel, které určují chování funkčních jednotek (PDU) včetně časové specifikace výskytu události nebo jejich posloupností.

vrstvy referenčního modelu OSI jsou:

1. fyzická (physical)
2. spojovací (data link)
3. síťová (network)
4. transportní (transport)
5. relační (session)
6. prezentační (prezentation)
7. aplikační (application)

fyzická vrstva

- je jedinou vrstvou, která zajišťuje komunikaci. Definuje připojovací konektory, elektrické úrovně signálů, kódování, topologie sítě. Přenosové prostředí (kabely) nejsou součástí referenčního modelu! Poskytuje služby zahajování, zprostředkování a ukončování spojení. Oznamuje spojové vrstvě případné chybové stavy.

spojová vrstva

- určuje způsob předávání zpráv v síti. Rozpoznává rámce a plní funkce zabezpečení spolehlivého spojení. Formátuje přenášená data (zprávy, pakety) do datových rámců přenosového protokolu. Každý rámec osahuje data a speciální kódy určené k synchronizaci, rozpoznání rámců, zabezpečení proti chybovosti a informace nutné pro adresování uzlů na lince (služební údaje). Vrstva provádí obsluhu chybovosti, číslování rámců případně opakování přenosu poškozených rámců. Zprávy, které nelze přečíst v jednom rámci (dlouhé) rozdělí na části (pakety) a doplní odpovídající služební údaje.

síťová vrstva

- zajišťuje síťové spojení. Definuje způsob, jakým se pakety pohybují v síti. Poskytuje transportní vrstvě nezávislost na směrování. Funkce směrování určuje vhodnou cestu uzly sítě od zdroje k cíli. Plní služby síťového adresování. Zahajuje, vytváří a ukončuje síťová spojení, identifikuje koncové body, řídí datový tok. Síťová služba může být se spojením (spolehlivá)

transparentní vrstva

- poskytuje transparentní a spolehlivý přenos v požadované kvalitě, optimalizuje síťové služby. Je vložena mezi uživatele a síť. Základními funkcemi je převod transparentních adres na síťové, multiplexování, rozvětvování či spojování spojení.

relační vrstva

- organizuje a synchronizuje dialog mezi entitami prezentační vrstvy. Vytváří časové intervaly (relace), ve kterých probíhá vlastní komunikace. Představuje logické rozhraní pro aplikační programy, které používají služeb sítě. Identifikuje uživatele, ověřuje přístupová práva, eviduje provoz případně ho také tarifikuje (účtuje poplatky)

prezentační vrstva

- transformuje datové položky na reprezentaci srozumitelnou aplikačním entitám. Převádí data do formy vhodné pro přenos, transformuje kódy a formáty mezi nekompatibilními počítači. Komprimuje data. Utajovaná data zašifruje.

aplikační vrstva

- umožňuje aplikačním procesům přístup ke komunikačnímu systému pomocí podpůrných aplikační funkcí (ASE, Application Service Element). Ty jsou rozděleny na všeobecné (CASE, Common ASE) a specifické (SASE, Specific ASE). Služby SASE jsou vázány k určité skupině aplikací (přenos souborů FTAM, elektronická pošta MHS). Služby CASE jsou univerzálnější (navázání spojení ACSE, transakce CCR).

    - v praxi je výše uvedená struktura sítě vzácností

- v síti jsou zařazeny:

- Opakovače, které obnovují signál a mají pouze fyzickou vrstvu
- Směrovače potřebují získat adresu a mají proto fyzickou, linkovou a síťovou vrstvu

- přenos mezi sítěmi s různými protokoly zajišťují brány na úrovni prezentační vrstvy. V reálu může být při dvoubodovém spojení mezi koncovými stanicemi různý počet uzlů s různým počtem protokolových vrstev.

architektura sna

- prakticky nejstarší firemní síťová architektura. Byla vytvořena firmou IBM. Slouží k připojení sítě terminálů k centrálnímu počítači. Pomocí speciálních komunikačních řadičů a poměrně primitivních terminálů (pracují většinou v textovém režimu) získá uživatel vzdálený přístup k výpočetnímu výkonu sálového počítače. Pro komunikaci se užívají protokoly SDLC. Přestože se jedná o značně zastaralou koncepci typu host-terminal, přežívá dodnes.

architektura tcp/ip

- vytvořena jako náhrada za původní protokoly sítě ARPANET. Zhruba od roku 1972 je oficiálním komunikačním prostředím této sítě, která je přímým a bezprostředním předchůdcem dnešního Internetu. Protože vznikla dříve než specifikace RM-OSI, není s ním vnitřně kompatibilní.

model TCP/IP definuje čtyři základní vrstvy:

- síťové rozhraní
- vrstva internet
- transportní vrstva
- aplikační vrstva

platí zhruba následující srovnání model OSI:

- síťová vrstva = fyzická + spojovací
- internet = síťová
- transportní=transportní
- aplikační = relační+prezentační+aplikační

    - způsobem komunikace tato architektura napodobuje dvojbodový kanál. Používají se tzv. aplikační porty. Všechny přenosy (z a do aplikačního procesu) jsou realizovány jako volání služeb TCP/IP. Vždy tedy spolu komunikují konkrétní aplikace, které samy vědí nejlépe, co která data znamenají. Tím se stávají některé vrstvy RM-OSI zbytečně (prezentační, relační).

- pojem architektura je často používaný, ale zřídka pochopen. Určitý systém je vystavěn nebo provozován tak, že vykazuje jisté rysy společné určité architektuře. Návrh naopak specifikuje, jaké konkrétní prvky budou použity a kde a jak budou umístěny. Architektura může být normalizována (má všeobecnou platnost), může se stát standardem de facto (je natolik rozšířena, že je obecně uznávána) nebo má omezenou (firemní) platnost. Síťové architektury jsou založeny na vrstvových modelech. Každá vrstva poskytuje služby vrstvě vyšší a využívá služby nižší. Sladění činností zajišťují komunikační protokoly, což jsou pravidla a formáty definující srozumitelnost komunikace. Výhodou vrstvových modelů je nezávislost na reálním provedení a možnost výměny protokolů jedné vrstvy bez zásadního vlivu na vrstvy ostatní.

VRSTVOVÝ MODEL OSI

- každá vrstva vykonává skupinu jasně definovaných funkcí. Pro svoji činnost využívá služeb sousední, hierarchicky nižší vrstvy. Své služby poskytuje sousední hierarchicky vyšší vrstvě. Vrstva je složena z dále nedělitelných entit, které vykonávají odpovídající funkce – poskytují služby. Komunikace mezi entitami je řízena protokolem.

- poskytování služby se děje prostřednictvím přístupového bodu služby

služby dělíme na:

- povinné
- volitelné provozovatelem (nemusí být poskytovány)
- volitelné uživatelem (poskytované na žádost)

- interakce subjektů v rámci služby se děje pomocí služebních primitivů

celkem jsou čtyři:

- žádost (vyvolání nějakého postupu)
- oznámení (indikace že k vyvolání došlo),
- odpověď (završení vyvolaného postupu)
- potvrzení (o ukončení).

- protokol je definován jako množina syntaktických (formáty zpráv, příkazy a odpovědi) a sémantických (používání příkazů a odpovědí) pravidel, které určují chování funkčních jednotek (PDU) včetně časové specifikace výskytu události nebo jejich posloupností.

vrstvy referenčního modelu OSI jsou:

8. fyzická (physical)
9. spojovací (data link)
10. síťová (network)
11. transportní (transport)
12. relační (session)
13. prezentační (prezentation)
14. aplikační (application)

fyzická vrstva

- je jedinou vrstvou, která zajišťuje komunikaci. Definuje připojovací konektory, elektrické úrovně signálů, kódování, topologie sítě. Přenosové prostředí (kabely) nejsou součástí referenčního modelu! Poskytuje služby zahajování, zprostředkování a ukončování spojení. Oznamuje spojové vrstvě případné chybové stavy.

spojová vrstva

- určuje způsob předávání zpráv v síti. Rozpoznává rámce a plní funkce zabezpečení spolehlivého spojení. Formátuje přenášená data (zprávy, pakety) do datových rámců přenosového protokolu. Každý rámec osahuje data a speciální kódy určené k synchronizaci, rozpoznání rámců, zabezpečení proti chybovosti a informace nutné pro adresování uzlů na lince (služební údaje). Vrstva provádí obsluhu chybovosti, číslování rámců případně opakování přenosu poškozených rámců. Zprávy, které nelze přečíst v jednom rámci (dlouhé) rozdělí na části (pakety) a doplní odpovídající služební údaje.

síťová vrstva

- zajišťuje síťové spojení. Definuje způsob, jakým se pakety pohybují v síti. Poskytuje transportní vrstvě nezávislost na směrování. Funkce směrování určuje vhodnou cestu uzly sítě od zdroje k cíli. Plní služby síťového adresování. Zahajuje, vytváří a ukončuje síťová spojení, identifikuje koncové body, řídí datový tok. Síťová služba může být se spojením (spolehlivá)

transparentní vrstva

- poskytuje transparentní a spolehlivý přenos v požadované kvalitě, optimalizuje síťové služby. Je vložena mezi uživatele a síť. Základními funkcemi je převod transparentních adres na síťové, multiplexování, rozvětvování či spojování spojení.

relační vrstva

- organizuje a synchronizuje dialog mezi entitami prezentační vrstvy. Vytváří časové intervaly (relace), ve kterých probíhá vlastní komunikace. Představuje logické rozhraní pro aplikační programy, které používají služeb sítě. Identifikuje uživatele, ověřuje přístupová práva, eviduje provoz případně ho také tarifikuje (účtuje poplatky)

prezentační vrstva

- transformuje datové položky na reprezentaci srozumitelnou aplikačním entitám. Převádí data do formy vhodné pro přenos, transformuje kódy a formáty mezi nekompatibilními počítači. Komprimuje data. Utajovaná data zašifruje.

aplikační vrstva

- umožňuje aplikačním procesům přístup ke komunikačnímu systému pomocí podpůrných aplikační funkcí (ASE, Application Service Element). Ty jsou rozděleny na všeobecné (CASE, Common ASE) a specifické (SASE, Specific ASE). Služby SASE jsou vázány k určité skupině aplikací (přenos souborů FTAM, elektronická pošta MHS). Služby CASE jsou univerzálnější (navázání spojení ACSE, transakce CCR).

    - v praxi je výše uvedená struktura sítě vzácností

- v síti jsou zařazeny:

- Opakovače, které obnovují signál a mají pouze fyzickou vrstvu
- Směrovače potřebují získat adresu a mají proto fyzickou, linkovou a síťovou vrstvu

- přenos mezi sítěmi s různými protokoly zajišťují brány na úrovni prezentační vrstvy. V reálu může být při dvoubodovém spojení mezi koncovými stanicemi různý počet uzlů s různým počtem protokolových vrstev.

ARCHITEKTURA SÍTÍ INFORMAČNÍCH SYSTÉMŮ

- síť je účelové propojení výpočetních systémů. Smyslem činnosti je získání nové informace ve formě akceptovatelné pro člověka. **Logicky se tak děje v několik provázaných krocích:**
    - spouštění aplikačního procesu
    - zpracování dat
    - prezentace výsledků

- v distribuovaném systému se jednotlivé datové stanice různou měrou podílejí na finálním výsledku. Způsob spolupráce datových stanic je v zásadě vždy stejný – kdosi služby poskytuje a kdosi je využívá. Architektura sítě je potom dána inteligencí datových stanic.

architektura host-terminal

- centralizovaný systém
- všechny služby jsou soustředěny na jednu datovou stanici (počítač střední třídy nebo mainframe) s příslušným operačním systémem. Veškerou činnost řídí a provádí centrální jednotka (fyzicky to může být více strojů).
- koncová zařízení jsou terminály (nebo emulátory terminálů) s malou inteligencí

- **hostitelské prostředí:**
    - operační systém (většinou Unix)
    - firemní aplikace (například SAP/R3)
    - výkonný databázový systém (Oracle, Informix)

- systém může obsloužit i tisíc uživatelů (používají často banky)

architektura file-server

- částečně centralizovaný systém.
- datové soubory jsou uloženy na souborovém serveru
- aplikační procesy běží na pracovních stanicích většinou na bázi personálních počítačů mající stejnou inteligenci jako centrální server.
- architektura se celkově jeví jako multiprocesorový systém a je základem takzvaných lokálních sítí
- **operační systémy:**
    - NetWare (Novell)
    - Windows NT (Microsoft)

architektura klient-server

- plně decentralizovaný systém

- skládá se nezávislých, avšak spolupracujících serverů (databázový, aplikační a prezentační) a pracovních stanic (personální počítač, grafické stanice)

- principem činnosti je rovnoměrné, pružné rozložení výpočetního výkonu mezi datovými stanicemi a minimalizace přesunu větších objemů dat po síti. (Například hledání v databázi probíhá na databázovém serveru, vytříděné položky zpracuje aplikační program na pracovní stanici a finální prezentaci připraví na prezentačním serveru)

- **podle rozložení jednotlivých funkcí výpočetního systému mezi servery a klienta rozlišujeme architekturu:**
  - dvou vrstvou
  - tří vrstvou
  - či obecně vícevrstvou

LOKÁLNÍ A ROZLEHLÉ SÍTĚ

ZÁKLADNÍ POJMY

- síť je systém rozprostřený v jistém ohraničeném prostoru. Technické parametry systému ovlivňuje rozloha tohoto prostoru. Zavedením parametru „rozlehlost sítě“, který je definován jako $a = \tau/t$, kde τ je zpoždění signálu v síti mezi krajními stanicemi a t je střední doba přenosu jednoho paketu, můžeme vliv rozlohy sítě charakterizovat. Podle velikosti parametru rozlehlost můžeme sítě rozdělit na lokální ($a < 1$) a rozlehlé ($a > 1$)

lokální síť (lan, local area network)

  - je komunikační síť umístěna u uživatele v ohraničené geografické oblasti. Maximální vzdálenost mezi koncovými stanicemi většinou nepřesahují 1000 m. Vždy jsou v soukromé správě. **Přenosový výkon sítě je omezen:**
    - kapacitou přenosového média
    - počtem uživatelů
    - maximální vzdáleností mezi stanicemi

- většina lokálních sítí pracuje v režimu bez spojení a dovoluje mnohonásobný přístup k přenosovému médiu. **Existují tři základní kritéria, která charakterizují různé typy lokálních sítí:**
    - metoda přístupu k přenosovému médiu
    - topologie sítě
    - přenosová rychlost

- metoda přístupu k přenosovému médiu je důležitý parametr sítí se sdíleným přenosovým médiem (kabel, radiový kanál). U tohoto typu sítí je třeba zajistit nerušené předávání dat jednotlivými uživateli. **Přístupová metoda řeší tento úkol, jedná se buď o:**
    - deterministický přístup na základě povolení
    - náhodný přístup na základě zjištění obsazenosti přenosového média.

- topologie sítě úzce souvisí s metodou přístupu. Lokální síť může mít topologii typu sběrnice, kruh, hvězda či strom. Je třeba rozlišovat mezi logickou a fyzickou topologií sítě (token-ring je logickým kruhem, ale fyzicky je to hvězda).

- přenosové rychlosti různých typů lokálních sítí se liší až o dva řády (1 – 100 Mbit/sec). Výkonnosti sítě souvisí nejen s přenosovou rychlostí a metodou přístupu, ale také s velikostí přenášených rámců. Všechny sítě dovolují proměnnou velikost rámce (podle požadavků aplikace). Čím delší rámec, tím větší propustnost (lepší poměr režie/data)

- lokální sítě jsou normalizovány jako podmnožina architektura OSI. Základní sada je z roku 1985 a má označení ISO 8802 (IEEE 802.x). **Nejdůležitější typy normalizovaných lokálních sítí jsou následující:**
    - Ethernet (802.3)
    - Token-bus (802.4)
    - Token-ring (802.5)
    - bezdrátová lokální síť (802.11)
      (v závorce je příslušná norma IEE)

ŘÍZENÍ PŘÍSTUPU K PŘENOSOVÉMU MÉDIU

- specifikace IEEE 802 pokrývá pouze fyzickou a spojovací vrstvu RM-OSI. Fyzická vrstva zahrnuje funkce kódování signálu, generování a odstraňování synchronizačních bitů (preambule) a vlastní přenos či příjem bitů. **Funkce spojové vrstvy jsou rozděleny do dvou podvrstev:**
    - řízení přístupu k přenosovým prostředkům (MAC, Medium Access Control) – závisí na topologii
    - řízení logického spoje (LLC, Logial Link Control)

řízení přístupu k přenosovému médiu

- je bezprostředně svázáno s konkrétní fyzickou strukturou sítě (závisí na typu lokální sítě). Jedinou společnou vlastností vrstvy MAC všech typů lokálních sítí je způsob adresace. Každé připojení k síti má jednu adresu. Adresy MAC mají dvě různé délky: 16 nebo 48 bitů. První bit adresy určuje, zda se jedná o adresu individuální (=0) nebo skupinovou (=1). Druhý bit udává, má-li adresa lokální platnost (=1) nebo se jedná o univerzálně spravovanou adresu (=0). Ve druhém případě je adresa na celém světě jedinečná. Toho je dosaženo tak, že každý výrobce síťových karet má přiděleno 24bitové číslo (kód výrobce). Zbytek adresy přiděluje výrobce tak, aby stejné číslo nebylo použito opakovaně. 16bitové adresy jsou pouze lokální. Speciální nulová adresa (všechny nulové) je rezervována pro zkušební, případně prázdné aplikace.

řízení logického spoje

- **specifikuje tři typy služeb:**
    - nepotvrzená služba bez spojení (nejčastější)
    - služba se spojením
    - potvrzovaná služba bez spojení (zřídka používaná)

- **protokolová datová jednotka má stejný formát pro všechny tři typy služeb. Formát PDU má čtyři pole:**
    - zdrojový bod služby (SSAP)
    - cílový bod služby (DSAP)
    - řídící pole
    - data

prvky lokálních sítí

- **základní elementy lokálních sítí jsou přenosové prostředky a síťové karty. Používají se nejrůznější typy přenosových prostředků:**
    - kabely (symetrický, koaxiální, optický)
    - rádiové vlny
    - infračervené vlny

síťové karty (NIC, Network Interface Card)

- **tvoří rozhraní uzlu sítě (datové stanice) pro připojení k síti. Na straně styku se sítí plní síťová karta čtyři funkce:**
    - uskutečňuje fyzické spojení s přenosovým médiem
    - zajišťuje dodržování pravidel přístupu k médiu
    - vysílá elektrické signály
    - přijímá elektrické signály a vybírá ty, které ji přísluší a dále je zpracovává

- uvnitř uzlu sítě zajišťuje karta přesun dat z nebo do operační paměti. Protože vnitřní sběrnice počítače pracuje paralelně a přenosové médium sériově provádí karta příslušný převod.

NORMALIZOVANÉ SÍTĚ: ETHERNET S PŘEDÁVÁNÍM TOKENU

- lokální síť ethernet je sběrnicový typ lokální sítě. Pod názvem Eternet jsou známy dva typy zcela neslučitelných sítí (Ethernet a IEEE 802.3). Používají sice stejnou přístupovou metodu a rychlost přenosu, ale zásadně se liší formátem rámce. Původní verze z roku 1980 je označována jako Ethernet II (také DIX) a byla vzata za základ standardu IEEE 802.3. V tomto standardu je jinak definována spojová vrstva (původně je jedna podvrstva, proto jiní formát rámce). Standard IEEE 802.3 je dále rozvíjen (Fast Ethernet, 100 Mb/sec).

  - lokální sítě typu Ethernet jsou nejrozšířenějším typem sítě (80%)

důvody:

- velký rozsah přenosových rychlostí
- jednoduchá implementace protokolu
- možnost používat nejrůznější přenosová média

nevýhody:

- podporuje pouze komunikaci bez spojení – nelze zajistit kvalitu složky (není zaručeno doručení rámce)
- žádný mechanizmus pro zajištění přenosů citlivých na rychlost
- v prostředí průmyslových sítí je největším nedostatkem nedeterministická přístupová metoda (nezaručení doba přístupu)

csma/cd

- je metoda mnohonásobného přístupu k sdílenému prostředku prostřednictvím naslouchání nosné a s detekcí kolizí (Carrier Sense Multiple Access with Collision Detection)

- stanice, která potřebuje vysílat, sleduje co se děje na přenosovém prostředku. Pokud je médium v klidu, stanice může začít vysílat. Může se však stát, že více stanic začne vysílat ve stejný okamžik – dojde ke kolizi, protože ani jeden signál nebude přenesen bezchybně. Vysílací stanice stále naslouchá dění na médiu a proto snadno rozpozná kolizi. Stanice, která první detekuje kolizi, okamžitě přestane vysílat a vyšle speciální krátký signál oznamující kolizi (JAM, 32 bitů, 45 µsec). Všechny stanice okamžitě přestanou vysílat a odmlčí se na náhodně stanovenou dobu. Kolize jsou způsobeny transportním zpožděním, obě stanice se mohou domnívat, že médium je volné, zatímco signál od druhé z nich se k nim teprve blíží. Největší přenosové zpoždění mají nejvzdálenější stanice. Z maximálního zpoždění 51 . 2 µsec (minimální ráme 64 oktetů = 512 bitů, pro přenosovou rychlost 10 Mb/sec) lze odvodit maximální vzdálenost mezi stanice (pro tenký a koaxiální kabel je to 185 m)

- základním principem této přístupové metody je stálé naslouchání nosné pro zjištění obsazenosti přenosového média a detekce kolizí. Dalším podstatným rysem je náhodnost zpoždění začátku vysílání. Jedná se tedy o stochastickou, nedeterministickou metodu, která je velmi efektivní v sítích s menším zatížením.

  výhody:
  - metody je jednoduchá implementace
  - správa v síti

  nevýhody:
  - rostoucí počet kolizí při zvyšování počtu připojených stanic

  - procentní využití transportního média se průměrně pohybuje do 30%, maximální asi tak do 80%.

- **rámec MAC 802.3 má následující strukturu:**
  - preambule (slouží k synchronizaci) 8 oktetů
  - cílová a zdrojová adresa 6 oktetů
  - délka rámce 2 oktety
  - data, zabezpečení 4 oktety.

implementace fyzické vrstvy

- zařízení se k síti Ethernet připojují pomocí MAU (Medium Attachment Unit) – součást síťové karty (transceiver). Základními úkoly MAU je rozpoznání přítomnosti signálu, vzniku kolize, vysílání a příjem dat. Norma specifikuje kódování signálu typu Manchester II s návratem k nule (všechny typ médií s výjimkou optického kabelu) což znamená, že pro 10 Mb/sec se využívá pásmo 10 Mhz. Signál se většinou přenáší v základním pásmu (výjimkou je 10 BROAD-36 – 3 kanály pro 6 Mhz pro vysílání, 3 kanály pro 6 Mhz pro příjem (=36), přenosové médium je koaxiální kabel 75 ohmů, délka segmentu 1800 m, rozlehlost: 3.6 km).

- **přenosový prostředek nebo délka segmentu ve stovkách metru (hrubě). Příklady:**
  - 10BASE-2 je označení pro tenký ethernet (10 Mb/sec), tenký koaxiální kabel, maximální délka segmentu 185 m)
  - 10BASE-T (10 Mb/sec, UTP kabel, segment maximálně 100 m).

topologie sítě

- **v důsledku nutnosti detekce kolizí má Ethernet omezený rozsah sítě. Omezující skutečnost (pro 10BASE-2) jsou následující:**
  - koaxiální kabel s impedancí 50 ohm
  - útlum při 10 Mhz je 8.5 dB
  - rychlost signálu 0.66 rychlosti světla = maximální délka segmentu 185 m

- segmenty lze propojit pomocí opakovačů – zařízení, která pouze regenerují signál (případně detekují kolize). Přičemž platí pravidlo 5-4-3 – propojit lze pět segmentů pomocí čtyř opakovačů tak, že pouze tři segmenty slouží k připojování stanic. (zbylé dva jsou propojovací). Maximální počet stanic v segmentu je 30, v celé síti 1,024.

fast ethernet

- pracuje s přenosovou rychlostí 100 Mb/sec
- existují dva standardy (802.3 U,Y)

- **oba mají shodné řešení fyzické vrstvy:**

  kabeláž utp kategorie 5 s použitím dvou párů vodiče (100 base-tx)
  - patrně nejrozšířenější u nových instalací
  - nosná frekvence je 125 MHz
  - data jsou kódována metodou 4B5B

  kabeláž UTP kategorie 3 (100 BASE – T2)
  - užívá speciální kódování (5 úrovní)

  kabeláž UTP kategorie 3,4,5 s využitím čtyř párů vodičů (100 BASE – T4)
  - je vhodná pro existující instalace s dostatkem párů vodičů
  - data se přenášejí po třech párech nosnou frekvencí 25 MHz
  - čtvrtý pár slouží k detekci kolizí

dvě optická vlákna (100 BASE – FX)

- při rychlosti 10 Mb/s jsou data kódována pomocí kódu Manchester II
- při rychlosti 100 Mb/s jsou data kódována pomocí kódu 4B5B

  - vysílač pro koaxiální kabel modeluje data pomocí proudového zdroje přičemž 41 mA je typický proud vysílače (-37 až 45 mA). Základní proud je modulován signálem s úrovněmi ± 28 mA. Vysílaný signál má sníženou strmost stran, z důvodu zmenšení rušení (25 ns náběh a doběh). Přijímač má vstup s vysokou impedancí a malou vstupní kapacitou. Při zesílení je vyrovnána kmitočtová charakteristika kabelu a pomocí regulace citlivostí velikost signálu

kódování bitů

- **čtyři základní typy přenosových médií:**
  - tlustý koax
  - tenký koax
  - kroucený pár
  - optický kabel (metal – symetrický/nesymetrický, optika – jednovidový/dvouvidový)

- volba kódování závisí na typu přenosového média a rychlosti přenosu

kódování manchester II

- pro rychlost 10 Mb/a metalickou kabeláž se používá kódování Manchester II

- **doba pro vysílání jednoho bitu je rozdělena na 2 poloviny:**
  - v 1. polovině se přenáší inverzní hodnota bitu
  - ve 2. polovině se přenáší přímá hodnota bitu

- tím je dosaženo toho, že v každém přenášeném bitu je hrana, která slouží k synchronizaci obvodů přijímače

- hodnota logické 1 je kódována jako 01 a logické 0 jako 10

- signál je polární (logická 1 – kladná úroveň napětí, logická 0 – záporná úroveň napětí

- **data jsou vlastně dvakrát kódována:**
  - signál je dvojúrovňový, data jsou kódovány pomocí dvou bitů (kódování Manchester II s návratem k nule)

- při vyšší rychlosti se výše uvedená kombinace nehodí – použijeme kódování NRZI
- logická 1 je reprezentována změnou napěťové úrovně, logická 0 signálu nemění

výhoda:

- frekvence signálu je poloviční

nevýhoda:

- delší sekvence 0 nemění signál ⇒dochází ke ztrátě synchronizace ⇒kódování 4B5B

kódování 4b5b

- při kódování 4B5B je sekvence 4 bitů převedena na sekvenci 5 bitů, jsou zde alespoň dvě hrany (logická 1)

- výsledek (100 Mb/s) – nejprve zakódujeme 4B5B – frekvence je 125 MHz, poté použijeme NRZI a frekvence se sníží na polovinu

další možnosti

- další možností je nároky na přenosové médium zmenšit – zakódovat signál ještě jednou, tzv. tří-úrovňové kódování (MLT 3), frekvenci signálu tím zmenšíme ještě o polovinu

gigabitový ethernet

- zatím poslední vylepšení Ethernetu (802.3z)
- použit stejný mechanizmus jako u přechodu z 10 na 100 Mb/s

- **přenosovým médiem je:**
  - optické vlákno (100BASE-SX a LX)
  - měděný vodič (100BASE – CX)
  - čtyři páry vodičů nestíněného kabelu UTP kategorie 5
  - délka segmentu pro jednotlivá média – 300 (550 případně 3 km), 25 a 100 metrů

- **pro fyzická média byly přijaty čtyři specifikace, které definují dva standardy:**
  - 802.3z (1000BASE-X) – pro optické kabely - základem pro vývoj byla technologie Fiber Channel
  - 802.3ab (1000BASE-T) – pro metalickou kabeláž
  - specifikace 1000BASE-SX je určeno pro levná mnohavidová vlákna a kratší vzdálenosti (horizontální vedení nebo páteřní rozvody)
  - specifikace 1000BASE-LX – je určena pro jednovidová vlákna na delší vzdálenosti
  - specifikace 1000BASE-CX - je určena pro krátká propojení stíněným kabelem typu twinax
  - specifikace 1000BASE-T – je určeno pro UTP kategorie 5

optické kabely:

- specifikace SX pracuje s laserem o vlnové délce 850nm
- specifikace CX pracuje s laserem o vlnové délce 1300nm

- pro jednovidové optické vlákno 9 mikrometrů je dosažitelná délka segmentu 5 km

- pro mnohovidové optické vlákno 50 mikrometrů je dosažitelná délka segmentu 550 m

- pro mnohovidové optické vlákno 62,5 mikrometrů je dosažitelná délka segmentu pro LX 550 m a pro SX 275 m

- kódování 8B10B

metalická kabeláž:

- specifikace 1000BASE - CX pracuje s dvojdrátem (STP, měděný vodič o impedanci 150 Ω

- po zakódování 8B10B se výstupní signál přenáší jako NRZI, délka 25 m

- specifikace 1000BASE – T používá UTP kabel kategorie 5e
- využívají se všechny 4 páry kabelu v plném duplexu

- data jsou kódována kódem PAM-5 a přenášena jako napěťový signál o 5 úrovních

- pro přenos jednoho oktetu dat potřebujeme jedinou změnu úrovně na každém ze čtyř párů

- modulační rychlost je 125Mbd

formát rámců

- **používají se dva typy rámců (podobná struktura):**

- preambule 8 oktetů
- cílová adresa 6 oktetů
- zdrojová adresa 6 oktetů
- zabezpečovací kód (CRC) 4 oktety

- Ethernet II
  - po zdrojové adrese následuje pole, udávající typ protokolu (2 oktety)
  - zbytek jsou data (46 – 1500 oktetů)
  - délka rámce = 8 + 6 + 6 + 2 + 1500 + 4 = 1526 oktetů

- IEEE 802.3
  - typ protokolu změněno na pole (2 oktety), které udává délku datové části rámce
  - typ protokolu: 0800-IP, 0806-ARP, 6000-DEC, 6010-3COM, 809B – Apple

BEZDRÁTOVÉ SÍTĚ

- Wi-Fi (nebo také Wi-fi, WiFi, Wifi, wifi) je standard pro lokální bezdrátové sítě (Wireless LAN, WLAN) a vychází ze specifikace IEEE 802.11. Název Wi-Fi je slovní hříčka wireless fidelity – „bezdrátová věrnost“.
- Původním cílem Wi-Fi sítí bylo zajišťovat vzájemné bezdrátové propojení přenosných zařízení a dále jejich připojování na lokalní (např. firemní) sítě LAN. S postupem času začala být využívána i k bezdrátovému připojení do sítě Internet v rámci rozsáhlejších lokalit a tzv. hotspotů. Wi-Fi zařízení jsou dnes prakticky ve všech přenosných počítačích a i v některých mobilních telefonech. Úspěch Wi-Fi přineslo využívání bezlicenčního pásma, což má negativní důsledky ve formě silného zarušení příslušného frekvenčního spektra a dále častých bezpečnostních incidentů.

- Následníkem Wi-Fi by měla být bezdrátová technologie WiMax, která se zaměřuje na zlepšení přenosu signálu na větší vzdálenosti.

- Bezdrátová síť může být vybudována různými způsoby v závislosti na požadované funkci. Ve všech případech hraje klíčovou roli identifikátor SSID (Service Set Identifier), což je řetezec až 32 ASCII znaků, kterými se jednotlivé sítě rozlišují. SSID identifikátor je v pravidelných intervalech vysílán jako broadcast, takže všichni potenciální klienti si mohou snadno zobrazit dostupné bezdrátové sítě, ke kterým je možné se připojit (tzv. asociovat se s přístupovým bodem).

- Nejjednodušším způsobem, jak bezdrátovou síť skrýt, je zamezit vysílání SSID. Připojující se klient pak musí SSID předem znát, jinak se nedokáže k druhé straně připojit. Protože je však SSID při připojování klienta přenášeno v čitelné podobě, lze ho snadno zachytit a skrytou síť odhalit.

SSID (Service Set Identifier)

- je jedinečný identifikátor každé bezdrátové (WiFi) počítačové sítě. Přístupový bod (AP) vysílá pravidelně každých několik sekund svůj identifikátor v takzvaném majákovém rámci (beacon frame) a klienti si tak mohou snadno vybrat, ke které bezdrátové síti se připojí.

- Parametr SSID se skládá z řetězce ASCII znaků dlouhého maximálně 32 znaků. Tento parametr představuje klíč, kterým dochází ke spojení jednotlivých adaptérů v rámci bezdrátové sítě. Všechna bezdrátová zařízení pokoušející se o vzájemnou komunikaci mezi sebou musí předávat ten samý SSID. Pokud klíč klientského adaptéru se neshoduje s klíčem přístupového bodu (AP), je mu odmítnut přístup, proto se musí nastavit klíč shodně na přístupovém bodu (AP) a na klientském adaptéru. Nastavením různých klíčů můžeme zajistit fungování několika bezdrátových sítí v jedné lokalitě a v rámci stejného frekvenčního rozsahu.

- **Existují 2 hlavní varianty SSID:**
    - AD-HOC – je bezdrátové připojení, které se skládá z klientských zařízení bez přístupového bodu (AP). V této bezdrátové síti při ztrátě spojení s jedním zařízením můžou ostatní zařízení dále komunikovat.
    - Infrastrukturní sítě obsahuje přístupové body (AP) (BSS (Basic Service set) nebo eventuálně ESS (Extended Service Set)). Tato konfigurace může podporovat možnosti roamingu pro mobilní práci, neboli spojíme více skupin BSS, které můžeme nakonfigurovat jako ESS. Na základě tohoto spojení uživatelé skupiny ESS mohou volně cestovat mezi BSS, přičemž je zachováno trvalé připojení.

Ad-hoc sítě

- V ad-hoc síti se navzájem spojují dva klienti, kteří jsou v rovnocenné pozici (peer-to-peer). Vzájemná identifikace probíhá pomocí SSID. Obě strany musí být v přímém rádiovém dosahu, což je typické pro malou síť nebo příležitostné spojení, kdy jsou počítače ve vzdálenosti několika metrů.

Infrastrukturní sítě

- Typická infrastrukturní bezdrátová síť obsahuje jeden nebo více přístupových bodů (AP – Access Point), které vysílají své SSID. Klient si podle názvů sítí vybere, ke které se připojí. Několik přístupových bodů může mít

stejný SSID identifikátor a je plně záležitostí klienta, ke kterému se připojí. Může se například přepojovat v závislosti na síle signálu a umožňovat tak klientovi volný pohyb ve větší síti (tzv. roaming).

Zabezpečení sítě

- Problém bezpečnosti bezdrátových sítí vyplývá zejména z toho, že jejich signál se šíří i mimo zabezpečený prostor bez ohledu na zdi budov, což si mnoho uživatelů neuvědomuje. Dalším problémem je fakt, že bezdrátová zařízení se prodávají s nastavením bez jakéhokoliv zabezpečení, aby po zakoupení fungovala ihned po zapojení do zásuvky.

- Nezvaný host se může snadno připojit i do velmi vzdálené bezdrátové sítě jen s pomocí směrové antény, i když druhá strana výkonnou anténu nemá. Navíc většina nejčastěji používaných zabezpečení bezdrátových sítí má jen omezenou účinnost a dá se snadno obejít.

Zablokování vysílání SSID

- Zablokování vysílání SSID sice porušuje standard, ale je nejjednodušším zabezpečením bezdrátové sítě pomocí jejího zdánlivého skrytí. Klienti síť nezobrazí v seznamu dostupných bezdrátových sítí, protože nepřijímají broadcasty se SSID. Bohužel při připojování klienta k přípojnému bodu je SSID přenášen v otevřené podobě a lze ho tak snadno zachytit. Při zachytávání SSID při asociaci klienta s přípojným bodem se používá i provokací, kdy útočník do bezdrátové sítě vysílá rámce, které přinutí klienty, aby se znovu asociovali.

Kontrola MAC adres

- Přípojný bod bezdrátové sítě má k dispozici seznam MAC adres klientů, kterým je dovoleno se připojit. Útočník se může vydávat za stanici, která je již do bezdrátové sítě připojena pomocí nastavení stejné MAC adresy.(pokud je na ap tato funkce aktivní)

802.1X

- Přístupový bod vyžaduje autentizaci pomocí protokolu IEEE 802.1X. Pro ověření je používán na straně klienta program, který nazýváme prosebník (suplikant), kterému přístupový bod zprostředkuje komunikaci s třetí stranou, která ověření provede (například RADIUS server). Za pomoci 802.1X lze odstranit nedostatky zabezpečení pomocí WEP klíčů.

WEP

- Šifrování komunikace pomocí statických WEP klíčů (Wired Equivalent Privacy) symetrické šifry, které jsou ručně nastaveny na obou stranách bezdrátového spojení. Díky nedostatkům v protokolu lze zachycením specifických rámců a jejich analýzou klíč relativně snadno získat. Pro získání klíčů existují specializované programy.

WPA

- Kvůli zpětné kompatibilitě využívá WPA (Wi-Fi Protected Access) WEP klíče, které jsou ale dynamicky bezpečným způsobem měněny. K tomu slouží speciální doprovodný program, který nazýváme prosebník (suplikant). Z tohoto důvodu je možné i starší zařízení WPA vybavit.
- Autentizace přístupu do WPA sítě je prováděno pomocí PSK (Pre-Shared Key – obě strany používají stejnou dostatečně dlouhou heslovou frázi) nebo RADIUS server (ověřování přihlašovacím jménem a heslem).

WPA2

- Novější WPA2 přináší kvalitnější šifrování (šifra AES), která však vyžaduje větší výpočetní výkon a proto nelze WPA2 používat na starších zařízeních.

ROZLEHLÉ SÍTĚ

opakovače:

- neplní pouze funkce regenerace signálu a přenosu mezi segmenty, ale také brání přenosu problémů (rozpojení média) z jednoho segmentu do ostatních
- používá se u všech typů ethernetu (většinou fungují trochu jinak)
- je-li přenosovým médiem kroucená dvojlinka, stává se opakovač (hub) centrem segmentu sítě
- spoj mezi hubem a datovou stanicí je tvořen dvěma páry kroucené dvojlinky (duplexní spoj)
- pracuje zcela transparentně (pro stanice je neviditelný)

mosty:

- také spojuje jednotlivé segmenty sítě, ale neopakuje mechanicky všechny rámce, které se na jeho portech objeví
- pracuje s předávací tabulkou (seznam linkový adres a příslušných síťových rozhraní) – objeví – li se rámec na některém portu mostu, zjistí most adresu příjemce, z předávací tabulky urč,í za jakým rozhraním se adresát nachází a tam zopakuje rámec

přepínače:

- v podstatě inteligentní mnohoportový most
- zjistí adresu příjemce a pošlu mu rámec
- neprovádí žádnou replikaci rámců
- pokud je jádrem segmentu, získáme tzv. bezkolizní (přepínaný) Ethernet
- umí propojit segmenty s různou rychlostí

Směrovače

- doručení paketu závisí obecně na neznámém počtu směrovačů
- každý směrovač je připojen nejméně do dvou sítí
- směrovače mají principálně stejné možnosti doručení jako uzly
- pokud je adresát v síti, která je k směrovači připojena, bezprostředně je paket doručen přímo
- v opačném případě bude použito nepřímé směrování, použije se implicitní cesta nebo má směrovač ve své tabulce odpovídající záznam
- směrovač je zařízení, které je schopné učit se a poučovat jiné
- zjistí-li směrovač, že zvolená cesta není optimální, uvědomí o tom odesílatele (obvykle směrovač)
- ten následně upraví svou směrovací tabulku
- základní myšlenka principu implicitních cest je relativně prostá – zajistit úspěšné doručení paketu i při částečné znalosti nejvýhodnější cesty
- směrovač tedy zná jen určité cesty, nezná celý internet – není to žádoucí a není to ani možné (síť je dynamická struktura)
- pro směry, které nezná, použije předurčenou cestu

- když tato cesta není optimální (a dozví se to) upraví svou směrovací tabulku
- negativním důsledkem je určitá neefektivita v doručování
- hlavní a zásadní výhodou je redukce objemů směrovacích tabulek a následně snížená režie potřebná k udržení tabulek v aktuálním a konzistentním stavu
- směrovací tabulka je soubor záznamů
- každý záznam je relace: adresa sítě – adresa odpovídajícího směrovače
- není to adresa komerčního směrovače, je to jen adresa následujícího přestupního bodu (next hop – další skok)

historické souvislosti

- internet vznikl zbytměním Arpanetu
- právě ten v počítačích představoval jakousi páteřní síť (Backbone), na kterou se ostatní lokální sítě připojovaly (mohly to být i konglomeráty lokální sítí)
- každá z lokálních sítí se k Arpanetu připojovala pomocí jedinečného směrovače, který plnil funkci směrovače implicitního
- směrovače, kterými se dílčí sítě připojovaly k páteřní síti, se nazývaly hlavní brány (Core Gate)
- hlavní brány nepoužívaly implicitní směrování, ale skutečně znaly celý Internet (důvodem byla efektivita)
- což mohlo spolehlivě fungovat, protože to spravovala jediná instituce (INOC, Internet Network Operations Center)
- explozivní růst Internetu si však vynutil změnu
- struktura sítě, daná jedinou páteřní sítí se stala značně složitou
- mechanizmus udržování směrovacích tabulek, hlavních bran se stal nákladnou a komplikovanou záležitostí
- zásadním problémem bylo, že některé dílčí sítě prostě nebylo možné připojit přímo na páteřní síť
- dalším důvodem pro změnu bylo právě připojování

směrovací protokoly

- směrovací protokoly jsou aplikační protokoly, které slouží směrovačům k automatickému naplnění směrovacích tabulek (nejsou pro uživatele) a dají se dělit podle dvou nezávislých kritérií:

podle použití

- **protokoly skupin igp (interior gateway protocol)** - jsou určeny pro činnost v rámci autonomního systému
- **protokoly skupin egp (exterior gateway protocol)** - slouží pro výměnu směrovacích dat mezi autonomními systémy

podle způsobů, jak určují optimální cestu

- **skupina rvp (routing vector protocol)** - v podstatě definuje kvalitu cesty počtem přeskoků (tedy délkou cesty – vector distance algorithm)
  - jejich nevýhodou je poměrně velká režie, která s počtem směrovačům silně roste
- **skupina lsp (link – state protocol)** - také SPF, Shortest Path First)
  - testuje v pravidelných intervalech průchodnost cest (dopravní zpoždění)
  - protokoly LS mají mnohem menší režii a jsou podstatně stabilnější i pro rozsáhlé sítě
  - bohužel mají i nevýhodu – konfigurace sítě používající LS protokol je náročná záležitost i pro zkušeného pracovníka

Dvě hlavní skupiny směrovacích protokolů:

egp

- jsou vhodné pro výměnu směrovacích dat mezi systémy

bgp

- Border Gateway Protocol, dnes verze 4
- pracuje s pevně stanovenými pravidly

igp

- jsou určeny pro směrování v rámci autonomních systémů či oblastí

rip

- Routing Information Protocol
- je jedním z nejstarších protokolů

- **je charakterizován:** používá oběžníky (všesměrové vysílání), pracuje s vektorem vzdáleností a hodí a hodí se spíše pro menší sítě
- dnes existuje ve dvou verzích

ospf

- Open Shortest Path First
- je vhodný pro střední či větší sítě
- je založen na algoritmu pracujícím s kvalitou přenosové cesty (LSA) a při rozhodování bere do úvahy:
    - šířku přenosového pásma
    - zátěž
    - spolehlivost
    - transportní zpoždění
    - velikost MTU
- **je charakterizován:** používá skupinové vysílání (224.0.0.x), spouštěné aktualizace, síťové masky proměnné délky a je schopen podporovat směrování s normovanou kvalitou služby (Qos)
- nezanedbatelnou předností je podpora ověřování – směrovače mohou výměnu dat chránit heslem
- data se vyměňují mezi autorizovanými směrovači

eigrp

- Enhanced Interior Gateway Protocol
- je příkladem hybridního protokolu
- vyvinula jej firma Cisco
- je charakterizován jako vyvážený protokol, který kombinuje výhody obou algoritmů směrování (VDA i LSA).

TECHNOLOGIE SÍTÍ NA BÁZI INTERNETU

ÚČEL A PRINCIP FUNKCE INTERNETU

- internet je informační prostor
- představuje globální síť propojující heterogenní lokální sítě pomocí souboru protokolů TCP/IP (Transmission Control Protocol/Internet Protocol)
- účelem je poskytování služeb založených na výměně zprávy a sdílení prostředků

- počátky vzniku internetu leží v USA (období studené války, rok 1963)
- původním strategickým záměrem bylo zjištění státní správy (vojenské i civilní) po nukleárním útoku
- komunikační síť je od počátku považována za nespolehlivou a je proto navržena tak, aby svou nespolehlivost dokázala překonat
- všechny uzly sítě jsou si rovnocenné (není žádné řídící centrum)
- každý uzel sítě představuje autoritu pro vytváření, předávání a přijímaní zpráv
- zpráva je rozdělena na elementární části (pakety) s tím, že každý paket je přenášen samostatně, což předpokládá, že obsahuje úplnou informaci potřebnou pro jeho doručení
- konkrétní cesta, po které se pakety dostávají k cíli, není podstatná (uzly přehazují pakety jako horký brambor – co nejrychleji odeslat) --- nová technologie – přepojování paketů
- tato technologie umožní, aby se zpráva dostala k cíli i v případě, že bude část komunikační sítě nefunkční
- idea: silně decentralizovaná, robustní síť založena na přepojování paketů (packet switching) ve variantě dataprogramové služby
- do vývoje poté zasáhla ARPA (Advanced Research Project Agency) a přidělila některým vysokým školám (UCLA, UCSB, Utah a SRI) granty na rozpracování takové sítě (nazvané ARPANET)
- síť se stala funkční v roce 1969 a účelem propojení bylo sdílení superpočítače
- vlastní uzel byl realizován počítačem, který byl naprogramován tak, že plnil funkci IMP (Interface Message Processor)
- pro vzájemnou komunikaci používaly uzly IMP pevné okruhy o rychlosti 50 kbps a přenosový protokol NCP (Network Control Protocol)
- velmi brzo se ukázalo, že původní představa o způsobu využití sítě (přístup k výpočetním kapacitám) byla idealistická - uživatelé používali síť především ke komunikaci

adresy

- internet je informační prostor
- pokud v tomto prostoru potřebujeme něco najít, musíme vědět, kde je uloženo – potřebujeme znát cílovou adresu
- adresa musí být koncipována tak, aby identifikovala cíl nějakým rozumně použitelným způsobem
- například každá síťová karta má MAC adresu (unikátní číslo, které jednoznačně identifikuje počítač), jenže neidentifikuje jej rozumně použitým způsobem
- adresa nás tedy musí nasměrovat na cíl
- v internetu slouží adresa dvěma, zcela odlišným skupinám uživatelů – lidem a strojům

adresace v internetu – adresy ip

- **IP adresa se skládá ze dvou částí:**
  - adresy sítě
  - adresy rozhraní (počítače) v této síti

- **v současnosti se aktivně užívají dva typy IP adres:**
  - **ip protokol verze 4 (IPv4)** - adresy dlouhé 32 bitů, které se většinou píší jako 4 oktety v desítkové, tečkami oddělené notaci (192.168.32.55)
  - **ip protokol verze 6 (IPv6)** - adresy dlouhé 128 bitů, které se prezentují jako hexadecimální řetězec (1080:0:0:0:8:800:200C:417A)

- **oba typy adres se přidělují stejným způsobem – pomocí delegátů:**
  - autonomní systém je vydělená (regionálně/kontinentálně) část Internetu, která má svého správce IP adres
  - koncový uživatel dostane IP adresu přidělenou poskytovatelem služby (ISP, Internet Service Provider)
  - tomu adresy, které bude poskytovat, přidělí buď místní registrátor (LIR, Local Internet Registry) nebo národní registrátor (NIR)
  - pro určitou geografickou část světa je potom určen regionální registrátor (RIR)
- **existuje pět regionálních registrátorů:**
  - AFNIC (African Network Information Centre)
  - APNIC (Asia Pacifik NIC)
  - ARIN (American Registry for Internet Numbers)
  - LACNIC (Latin American and Carribean NIC)
  - RIPE NCC (Réseaux IP Européens)

- nejvyšším orgánem je IANA (Internet Assignment Numbers Authority)
- přiděluje adresy podle potřeb RIR z volné rezervy
- komu byla určitá IP adresa přidělena je možnost zjistit na webových stránkách regionálních registrátorů (RIR.net)
- rozdělení Internetu na autonomní systémy a přidělení každému interval adres je základní podmínka směrování datagramů
- interval adres je potom možné agregovat na jednu adresu supersítě

- **adresy mohou být:**
  - individuální (unicast)
  - skupinové (multicast)
  - všeobecné (anycast)

- **kolik bitů z adresy tvoří adresu sítě určuje třída adresy, je definováno 5 tříd:**

třída a

- adresa sítě je určena prvním bajtem IP adresy:
- nejvyšší bit je 0, zbylých 7 bitu je proměnných, určují adresu sítě
- tzn., může nabyt hodnot od 1 (0000 0001) do 127 (0111 1111), vyjma 10, protože se používá pro privátní sítě

třída b

- adresa sítě je určena prvními dvěma bajty IP adresy:
- dva nejvyšší bity prvního bajtu má 10, dalších 6 v prvním a 8 v druhem bajtu je proměnných a určují adresu sítě
- tzn. může nabyt hodnot od 128.0 (1000 0000 . 0000 0000) do 191.255 (1011 1111 . 1111 1111), vyjma rozsahu 172.16 až 172.31, protože se používá pro privátní sítě

třída c

- adresa sítě je určena prvními třemi bajty IP adresy:
- tři nejvyšší bity prvního bajtu má 110, dalších 5 v prvním a 8 ve druhém a třetím bajtu je proměnných a určují adresu sítě
- tzn. může nabyt hodnot od 192.0.0 (1100 0000 . 0000 0000 . 0000 0000) do 223.255.255 (1101 1111 . 1111 1111 . 1111 1111), vyjma rozsahu 192.168.0 až192.168.255, protože se používá pro privátní sítě

třída d

- má 1110 v prvém bajtu a zbytek se dále nedělí

třída e

- rezerva

specifikace ip adresy:

- adresy tvořené samými jedničkami nebo nulami mají speciální význam, běžné se nepoužívají

- je-li adresa tvořena samými nulami, znamená to tento počítač
- je-li adresa tvořena samými jedničkami, znamená to všichni (všeobecný oběžník, broadcast)

- každý systém (počítač) má adresu programové smyčky (127.0.0.1), která se na internetu tudíž nepoužívá

- adresy třídy D slouží pro skupinovou adresaci

síťová maska

- je to čtyřbajtové číslo (v bitech určující adresu sítě má samé jedničky a v ostatních bitech samé nuly)
- slouží k získání adresy sítě, ve které je stanice o dané IP adrese
- určuje, které bity v IP adrese tvoří adresu sítě

- jednotlivé třídy sítí používají jako adresu sítě různě dlouhou část IP adresy
- třída A má pro adresu sítě vyhrazen první bajt ⇒standardní síťová maska pro adresy třídy A má v prvním bajtu samé jedničky a ve bylých třech bajtech samé nuly (255.0.0.0.

- toto jsou standardní síťové masky (jsou vždy vyrovnány na hranici oktetu)
- adresu sítě, na které leží počítač o IP adrese 170.85.255.24. určíme tak, že se nejprve podíváme do tabulky tříd a zjistíme, že maska je třídy B ⇒užijeme standardní masku a provedeme logickou operaci: adresa AND maska

- výše uvedené je obecný logický postup, jak získat z IP adresy adresu sítě
- maska však nemusí být vyrovnána na hranici oktetu a směrování nemusí být založena na třídách
- realizace beztřídního směrování je ovšem poněkud složitější - adresa musí být zadána ve tvaru například 192.168.0.0/21 (maska je 21 souvislých jedniček)

identifikace

- internet je informační prostor, který obsahuje rozmanité objekty (položky zájmu), ke kterým lze přistupovat nejrůznějším způsobem

uri (uniform resource identifier)

- je jednoduchý způsob určení objektu v informačním prostoru

- **základní vlastnosti:**
    - dovoluje určit rozdílné objekty, ke kterým se přistupuje odlišným způsobem jednotně
    - není žádné omezení pro objekt (může to být dokument, obraz, informační zdroj, služba....) a nemusí se k němu ani přistupovat přes Internet (třeba knihovna v knihovně)
    - pouze identifikuje jeden objekt, neříká nic o tom, zda je objekt skutečně přístupný (nedává garanci přístupnosti)

- **genericky je URI hierarchická sekvence komponent:**
    - scheme (schéma)
    - authority (autorita)
    - path (cesta)
    - query (dotaz)
    - fragment (fragment)

- obecně se URI skládá ze schématu a na schématu závislé, specifické části

schéma:

- představuje určitou síťovou službu
- každá služba má jistou specifikaci, která vysvětluje specifické detaily o tom, jak jsou identifikátory schématu přiděleny a jak se přidružují ke zdroji
- syntaxe URI je tak federalizovaný a názvově rozšířitelný systém, v němž každá specifikace schématu může dále omezit skladbu a sémantika identifikátorů uvnitř schématu
- schémat je v současnosti kolem padesáti, schvaluje a registruje je IANA
- architektura webu sice dovoluje definici nových schémat, ale uvedení nového schématu je drahé
- nové schéma URI vyžaduje vývoj nejen klientského softwaru, aby podporoval toto schéma, ale také vytvoření pomocných agentů (brány, proxy servery). Zabývá se tím dokument RFC 2718

autorita:

- pokud je uvedena, reprezentuje jmenný prostor (adresu), kde se zdroj nachází
- může to být provedeno jak registrovaným jménem, tak adresou serveru
- pole komponenty začíná dvojicí lomítek (//) a končí znakem lomítka (/), otazníku (?), znakem pro číslo (#) nebo ukončením URI

- **komponenta může mít až tři části:**
  - **userinfo** - informace o uživateli (je-li přítomna) musí být ukončena znakem komerčního et @
  - **host** - může zde být uvedena IP adresa ve standardní notaci, uzavřená mezi hranaté závorky nebo registrované jméno
  - **port** - je uveden dvojtečkou

- pokud je autorita uveden, pak komponenta cesta musí být buď prázdná nebo začínat znakem lomítka (/)

path

- cesta k předmětu zájmu v rámci jmenného prostoru autority
- pole komponenty začíná lomítkem (/) a je ukončeno znakem (?), znakem pro číslo (#) nebo koncem URI
- může být tvořena hierarchickou sekvencí segmentů vzájemně oddělených znakem lomítka
- může být relativní, pak musí být definovaná báze. Pokud je relativní, je možné užít pro segmenty tečkovou notaci (.) (..)
- nesmí začínat (//)

query

- dotaz jsou data, které nemají hierarchickou strukturu a slouží k identifikaci předmětu zájmu v jmenném prostoru autority (podobně jako cesta)
- pole komponenty je uvedeno znakem otazníku (?) a ukončeno znakem čísla (#) nebo koncem URI
- většinou má strukturu „klíč = hodnota“ (KEY=VALUE)
- znaky ?, a, / jsou v dotazu povoleny

fragment

- je komponenta umožňující nepřímou identifikaci sekundárního zdroje, což může být určitá část primárního zdroje nebo jiný způsob prezentace (má zvláštní význam v systémech poskytování informací)
- pole komponenty začíná znakem čísla (#) a ukončeno koncem URI
- znaky /, ? jsou v poli fragmentu povoleny

použití uri

- při praktickém použití se málo kdy používá URI reference tak, jak ji definují syntaktická pravidla
- většina internetových protokolů užívá nějakým způsobem zkrácenou verzi (většinou z důvodu úspory místa)
- URI reference je řetězec znaků reprezentující URI a tažmo předmět zájmu
- absolutní reference obsahuje schéma i specifickou část, relativní pouze specifickou, na schématu závislou část („koncová část“)
- použití relativní reference předpokládá existenci báze – složením relativní reference s bází získáme referenci absolutní
- bázi můžeme odvozovat implicitně nebo definovat explicitně

- získání předmětu zájmu předpokládá rozřešení odkazu – derefernce (URI resolution)
- definujeme řetězec znaků (podle syntaktických pravidel), který reprezentuje URI

iri

- mezinárodní verze URI (International Resource Identifier)
- pro zápis znaků reprezentujících URI je možné aplikovat pouze kódování US-ASCII
- IRI umožňuje kódování znaků v UTF-8 (unicode)
- ač důvody jsou zřejmé – URI je určeno především pro člověka a v jazycích, které nepoužívají latinku, je reprezentace málo srozumitelná, bude realizace dlouhý a problematický proces
- IRI je podporovaný iniciativou OASIS (Organization for the Advancement of Structured Information Standards, web a e-byznys)

url

- jednotný lokátor zdroje (Uniform Resource Locator)
- popisuje konkrétní umístění objetu zájmu
- obsahuje veškeré informace potřebné pro jeho získání (jakou síťovou službu použít, na který server se obrátit a co po něm chtít)
- identifikuje, kde se dotyčný objekt nachází
- lokátor, který začíná schématem je absolutní
- obsahuje úplnou informaci nutnou k získání objektu zájmu (ukazuje vždy na stejný objekt a funguje odkudkoli)
- relativní lokátor obsahuje pouze cestu, schéma a server chybí
- umístění cíle získáme kombinací relativního URL a báze
- báze je levá část absolutního URI (do prvého lomítka – schéma a autorita) v jehož kontextu byl relativní lokátor definován
- lokátor je relativní, protože týž lokátor povede k různým objektům

urn

- jednotný název zdroje (Uniform Resource Name)
- představuje mnohem obecnější mechanizmus identifikace objektu zájmu
- představuje jednoznačně jméno dotyčného objektu, podle nějž by jej bylo možno obstarat
- jeho použití předpokládá vytvoření mechanizmu překladu jména na lokátor (momentálního) místa uložení (a případně i způsobu přístupu)
- jedním ze základních nedostatků URL je, že často ukazuje do prázdna – objekt byl přemístěn nebo odstraněn

- současný internet je charakterizován jako druhá generace

- na rozdíl od první generace, založené na ručně psaných hypertextových stránkách, druhá generace již poskytuje služby generované programy

- příznačné pro obě generace je, že procesy vyhledávání v informačním prostranství, jsou stále řízený člověkem

- **jakékoli další zlepšení současného využívání Internetu vyžaduje naplnění alespoň čtyř požadavků, které spolu navzájem souvisí:**
  - inteligentnější informační služby
  - univerzální vyjadřování (umět pracovat s jakoukoli formou dat)
  - syntaktická interoperabilita softwarových komponent
  - sémantická interoperabilita zdrojů

- cílem je sémantický informační prostor (web s významem) – přidáním metadat, která specifikují sémantický obsah objektu

doménová jména

- přenosové protokoly internetu identifikují jednotlivé uzly sítě prostřednictvím IP adresy, které jsou celosvětově unikátní. Člověk si ale takové adresy (32 bitová čísla, ať jsou zapsána jakkoli), velmi špatně zapamatovává. Lidé raději používají mnemonická (symbolická) jména. Což by bylo možné, kdyby ke jménům existovala jednoznačně převodní tabulka - takovému symbolickému jménu odpovídá taková IP adresa. Internet je však celosvětová síť a je proto potřeba zavést další (v podstatě organizační) opatření, aby byl takový převod realizovatelný.

- předpokládaný systém přidělování jmen musí splnit tyto základní požadavky:
    - symbolická jména nebudou přidělována libovolně (protože jméno musí být unikátní)
    - systém jmen musí mít nějakou hierarchickou strukturu (protože jinak se smysluplné názvy rychle vyčerpají a zbudou jména nic neříkající, což je právě to, čemuž se snažíme zabránit)

- tak byl vytvořen systém doménových jmen (DNS – Domain Name System)
- doména je skupina jmen, které k sobě logicky patří (názvy uzlů sítě jedné firmy, organizace, země)
- doménové jméno reflektuje příslušnost uzlu do určité skupiny
- v rámci domény je možno vytvářet další podskupiny (subdomény)
- systém doménových jmen má hierarchickou (stromovou) strukturu – nejvyšší instancí je root

- v root doméně jsou definovány generické domény (TLD – Top Level Domains), mají označení:
    - edu (školství)
    - com (komerce)
    - net (internet)
    - org (nekomerční organizace)
    - mil (armáda)
    - int (mezinárodní organizace)
    - gov (vláda USA)

    - používají se převážně v USA (má to historické kořeny)
    - následně byly to TL domény přidány národní domény (například cz)
    - zatím poslední rozšíření je z roku 2000, kdy byly přidány domény aero, biz, coop, info, jobs, mobi, museum, name, pro, travel, eu
    - vrchním správcem je IANA
    - každá doména má minimálně jednoho registrátora (generické domény), většinou několik až mnoho (národní domény)

syntaxe

- doménové jméno se uvádí v tečkové notaci (řetězce znaků oddělené tečkou)
- první řetězec je jméno počítač, další je jméno nejnižší vnořené domény
- zcela vpravo je doména nejvyšší úrovně (přesněji, zcela vpravo je root, tečka, která se většinou vynechává)
- celé jméno může mít maximálně 255 znaků, řetězce pak 63 znaky
- povolené znaky jsou písmena, číslice a pomlčka, která nesmí být na začátku ani na konci řetězce
- pozor!!! – autonomní systémy dělí internet z hlediska směrování, domény podle jmen uzlů

překlad doménových jmen

- dokum- DNS (Domain Name System)
- je hierarchický systém doménových jmen
- realizace zabezpečují servery DNS pomocí protokolu stejného jména
- primárním úkolem a příčinou vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě
- později byly doplněny další funkce (například pro elektronickou poštu a IP telefonii)
- dnes slouží především jako distribuovaná databáze síťových informací
- základem systému je protokol, který využívá UDP komunikaci i TCP spojení vždy na známém portu 53
- DNS servery mají hierarchickou organizaci, podobně jako doménová jména
- domény umožňují lepší orientaci lidem, IP adresy jsou pro stroje
- systém DNS umožňuje efektivně udržovat decentralizované databáze doménových jmen a jejich překlad na IP adresy

- stejně tak zajišťuje zpětný překlad IP adresy na doménová jména

- používání jmenných názvů je pro člověka daleko příjemnější než posouvání složitých čísel (IP adres). Potřeba používat jiný systém adres pro člověka a jiný pro stroj, vznikla už v ranných dobách ARPAnetu. V počítačích se to provádělo tak, že na všechny počítače v síti byl distribuován soubor (většinou manuálně), obsahující tabulku pro překlad (v Unixu /etc/hosts). Tato koncepce velmi rychle přestala vyhovovat, především díky nárokům na rychlou aktualizaci. Přesto se tento soubor používá dodnes. V závislosti na konfiguraci systému je možné jej použít buď prioritně před dotazem na DNS nebo v případě, že DNS server neodpovídá. Historie: V roce 1983 vyvinul Paul Mockapetris DNS protokol, který je popsán v dokumentech RFC 882 a RFC 883. V roce 1987 byl protokol aktualizován (RFC 1034, 1035). Dnes existuje cca 30 RFC entů týkajících se DNS

struktura dns

- prostor doménových jmen má hierarchickou strukturu (strom)
- každý uzel struktury obsahuje data o své části jména, které je mu přiděleno a odkazy na své subdomény
- root je kořenová doména (zapisuje se jako tečka)
- hierarchicky níže se nacházejí domény nejvyšší úrovně (TLD)
- ty jsou buď generické (tematické) nebo národní
- celá struktura se administrativně dělí do zón
- každá zóna má svého správce a svůj (autoritativní) jmenný server
- výhodou takového organizačního uspořádání je možnost zóny dále dělit a správu nové části svěřit někomu dalšímu
- právě delegování pravomocí a distribuovaná správa jsou klíčové vlastnosti systému doménových jmen

dns servery

- **každá zóna má nejméně dva DNS servery:**

primární server

- vznikají na něm data
- pokud je třeba provést v doméně nějaké změny, musí se tak učinit na primárním serveru
- každá zóna má právě jeden primární server

sekundární server

- je automatickou kopií primárního
- průběžně si aktualizuje data podle primárního serveru
- slouží především jako záloha pro případ výpadku primárního serveru
- také zabezpečuje rozkládání zátěže u frekventovaných domén
- každá doména musí mít nejméně jeden sekundární server (sekundárních serverů může být vícero)

pomocný server (caching only)

- slouží jako vyrovnávací paměť pro snížení zátěže systému
- uchovává si odpovědi, dokud nevyprší jejich platnost a poskytuje je při opakujícím se dotazu
- odpovědi pocházejí od primárního či sekundárního serveru jsou autoritativní (konečné)
- odpovědi pomocného serveru nejsou autoritativní
- v případě nutnosti může klient požádat o autoritativní odpověď primární či sekundární server

- jmenný server může být pro jednu zónu primárním a pro jinou sekundárním serverem
- všechny root servery jsou primární

dns dotaz

- je relace přeložení jména na IP adresu (popřípadě naopak)
- klient relaci inicializuje – resolver posílá požadavek jmennému serveru
- aby tak mohl učinit, musí mít ve své konfiguraci síťových parametrů adresu lokálního DNS serveru, na který se má obracet (typicky ji získá pomocí DHCP)

- **relace probíhá následujícím způsobem:**
    - klient má požadavek na přeložení jména www.yyy.zzz

    - reslover pošle dotaz lokálnímu jmennému serveru (LNS) a očekává jednoznačnou odpověď
    - LNS zná adresy root serverů, pošle tedy některému dotaz
    - root NS dopoví seznamem NS pro doménu zzz
    - LNS odešle dotaz NS domény zzz, který odpoví seznamem NS domény yyy
    - LNS odešle dotaz NS subdomény yyy, který odešle konečnou (autoritativní) odpověď
- uvedený průběh předpokládá, že žádný z dotazovaných NS, kromě posledního požadovaného odpověď nezná (protože neřešil nedávno takový požadavek)

shrnutí

- kořenové servery mají autoritativní informace o doménách nejvyšší úrovně
- konkrétně znají všechny jejich autoritativní servery
- dotaz je tedy následně směrován na některý z autoritativních serverů nejvyšší úrovně (TLD), v níž se nachází cílové jméno
- ten je opět schopen poskytnout informace o své doméně a posunout řešení o jedno patro dolů v doménovém stromě
- relace je sekvence rekurzivních dotazů – výsledkem je autoritativní odpověď (+ nebo -)

dns komponenty

- každá komponenta představuje jednu vrstvu systému doménových jmen

- **systém doménových jmen se skládá ze tří hlavních komponent:**
    - **soustava hierarchicky uspořádaných jmenných prostorů**
        - což je realizováno jako distribuovaná databáze záznamů (záznam jméno a asociovaná informace)
    - **jmenný server**
        - program, který zná strukturu jmenných prostorů a umí manipulovat se záznamy databáze
        - DNS se mu jeví soustava místně uložených dat zvaných zóny
        - server musí periodicky aktualizovat svá data (ze souboru na disku nebo komunikaci s cizími servery, ergo jemu se to jeví jako dynamický systém) a souběžně musí odpovídat na dotaz resolverů
    - **resolver**
        - program, který komunikuje s jmennými servery, od kterých získává informace podle požadavků klienta
        - uživatel k němu přistupuje pomocí jednoduché procedury nebo služby operačního systému
        - DNS se mu jeví jako stromová struktura a on může požádat o informace, z kterékoli sekce (větve)
        - DNS vidí jako neznámý počet jmenných serverů
        - každý z nich zná (spravuje) určitou část distribuované databáze doménového stromu
        - uložená data se mu jeví víceméně statická

dns funkce

- jmenný server po svém startu přenese do paměti data pro zónu, kterou spravuje
- primární sever tak učiní načtení ze souboru na lokálním disku
- sekundární server získá data od serveru primárního (dotazem zone transfer)
- tato data se označují jako autoritativní (nezvratná, nekonečná)
- dále oba servery přenesou do paměti data, která nejsou součástí zóny, kterou spravují - především se jedná o data umožňující spojení na root servery a případné odkazy na servery spravující subdomény (delegace pravomoci) - tato data se označují jako neautoritativní

- součástí systému je paměť cache:
    - do ní se ukládají kladné (případně i záporné, negativní caching) odpovědi na dotazy, které provedly jiné jmenné servery (což šetří čas při případných opětovných dotazech)
    - tyto data jsou opět neautoritativní

- **pro přenos dotazů se používá UDP protokol (komunikace bez spojení):**
    - vyšle se datagram prvému server, a pokud odpověď nepřijde (čekání je velmi krátké), vyšle se datagram dalšímu serveru (to se cyklicky opakuje do získání odpovědi nebo vypršení časového intervalu)
    - bere se ta odpověď, která přijde jako první, byť by byla negativní
    - délka UDP datagramu je omezena na 512 B a fragmentace se nepoužívá
    - pro zónové přenosy (nebo pokud je odpověď delší) se používá TCP spojení (vždy port 53)

dns záznamy

- distribuovaná databáze je soustav místně uložených dat
- reálně je to datový soubor, který obsahuj úplné informace o příslušné zóně ve tvaru zdrojových vět (RR, Resource Records)
- všechny zdrojové věty mají stejný formát (strukturu)

- **struktura vět**
    - **doménové jméno (name)** - pro něž je záznam vytvořen
    - **typ vět (type)** - specifikuje účel věty
    - **třída věty (class)** - určuje rodinu protokolů, k níž se věta vztahuje
    - **doba platnosti, exspirace (ttl – time to live)** - 32 bitové číslo, udávající dobu v sekundách, po kterou může být věta udržována v cache serveru (hodnota 0 znemožňuje uložení věty do vyrovnávací paměti)
    - **délka datového pole (rdlegth)** - představuje 16bitové číslo určující délku datové části věty
    - **vlastní data (RDATA)** - jako řetězce znaků různé délky (formát závisí na typu a třídě věty)

- **typy zdrojových vět**

    - **některé frekventované typy zdrojových záznamů:**

- **soa (start of authority)**
    - určuje autoritativní jmenný server zóny
    - věta uvozuje data zóny
    - v datovém souboru vždy právě jedna věta
- **a**
    - přiřazení IP adresy doménovému serveru
- **ns**
    - věta definující jmenný server zóny
- **cname (canonical name for alias)**
    - uvádí synonymum (alis) k doménovému serveru
    - umožňuje přiřadit k jednomu jménu několik IP adres (služba je poskytována několika servery)
- **ptr**
    - umožňuje reverzní překlad
- **hinfo**
    - slouží k charakterizaci hostitelského počítače
    - obsahuje jak popis HW tak i SW
    - má pouze informativní charakter
- **aaaa**
    - přiřazení IP6 adresy doménovému serveru
- **wks (well know service description)**
    - popisuje ostatní služby hostitelského počítače
- **mx (mail exchange)**
    - věta pro server elektronické pošty
    - je určena pro e-maily
    - specifikují poštovní server domény (není uveden v adrese mailu, není to žádoucí)

  - věta obsahuje jednak IP adresu poštovního serveru a jednak jeho prioritu (číselná hodnota)
  - daná doména může mít poštovních serverů vícero
  - pošta se doručuje na server podle priority (nejprve serve s nejnižším číslem – nejvyšší priorita)
- **text**
  - textové pole
  - má pouze informativní charakter
- **pole name**
  - definuje doménové jméno
  - pokud není vyplněno, vezme se jeho hodnota z předcházejícího řádku
  - nemá-li na konci tečku, přidá se automaticky jméno domény uvedené ve větě SOA (má-li tečku na konci je to jméno absolutní)

dns protokol

- služba překladu doménových jmen je realizována jednoduchým protokolem
- resolver pošle dotaz a server na něj odpoví
- podobně jako u jiných aplikačních protokolů, dotazy i odpovědi jsou textové řetězce (komplikací je komprese doménového jména)
- v závislosti na účelu, protokol definuje několik typů operací
- s diverzifikací protokolu přibývá a typů operací
- pro náš účel je základní DNS dotaz

- **datový formát**
  - DNS dotaz (query) používá stejný formát paketu jak pro dotaz tak pro odpověď
  - může se skládat ze záhlaví až čtyř dalších sekcí

  - **záhlaví je povinné a skládá se ze šesti 16bitových polí:**

    - **ID**
      - identifikátor zprávy, který vkládá klient a server kopíruje do odpovědi
      - slouží k párování dotaz-odpověď
    - **pole řídících bitů**
      - zbývající čtyři pole mají stejný formát a představují kladná celá čísla
    - **qdcount**
      - udává počet položek v dotazu
    - **ancount**
      - říká, kolik zdrojových vět obsahuje odpověď
    - **nscount**
      - oznamuje počet vět definujících autoritativní jmenné servery
    - **arcount**
      - určuje počet položek v doplňující odpovědi
      - **řídící bity mají následující význam:**
        - QR – dotaz/odpověď
        - AA – autoritativní odpověď
        - TC – zkráceno (odpověď se nevešla do 512 oktetů)
        - RD – klient požaduje rekurzivní překlad
        - RA – server umožňuje rekurznivní překlad

    - **opcode**
      - čtyřbitové pole
      - specifikuje typ dotazu (standardní, inverzní, na status serveru)
    - **rcode**
      - čtyřbitové pole

- charakterizuje odpověď (bez chyby, chyba formátu dotazu, server neumí odpovědět, jméno neexistuje, server rento dotaz nepodporuje, server odmítá odpovědět)

- **sekce dotazu**
  - **skládá se ze tří polí:**
    - QNAME – doménové jméno
    - QTYPE – typ požadované odpovědi (jakou větu si RR přejí)
    - QCLASS – třída dotazu (1 = Internet)
- **sekce odpovědi**
  - obsahuje pole TTL, RDLENGTH, RDATA

- doménové jméno není zde zapsáno v tečkové notaci, ale každá část jeho jména je uvedena bajtem, který uvádí délku následujícího řetězce
- konec jména je signalizován nulovou hodnotou délky
- pro dosažení minimální délky paketu je doménové jméno komprimováno
- to se provede tak, že se jméno uvede v paktu je jednou a každý další výskyt se nahradí odkazem na prvé uvedení

- **mechanizmus je následující:**
  - maximální možná délka řetězce je 63 (00111111) pokud tedy bajt délka začíná 11....pak to znamená, že se jedná o jméno, ale o odkaz

autonomní systémy

- internet je soustavou konglomerátů dílčích sítí
- komercializace internetu přispěla ke vzniku autonomních systémů
- dnes by bylo patrně přesnější definovat internet jako soustavu dílčích sítí různých poskytovatelů připojení (Provider)
- drtivá většina poskytovatelů připojení pracuje na komerční bázi, a proto si přeje svou síť spravovat svým způsobem
- internet byl proto důsledně rozdělen na autonomní systémy (i samostatná páteřní síť se soustavou hlavních bran je autonomní systém), což následně vedlo k zjednodušení jeho struktury
- každý autonomní systém je označen zkratkou AS a dvoubajtovým číslem
- jeden poskytovatel může mít i několik autonomních systémů
- poskytovatelé se zabývají dopravou paketů v rámci své sítě, mezi sebou i jako tranzitní přepravci (propojovací sítě)
- každá autonomní oblast má svou správu, která žádá autoritu o přidělení intervalu IP adres (interval adres umožňuje agregaci dílčích sítí do jedné supersítě = jedna cílová směrovací adresa)
- internet je tedy z pohledu směrování paketů rozdělen na autonomní systémy
- pro směrování mezi AS se používá EGP nebo IGP
- takže, za každý autonomní systém plně odpovídá jeho provozovatel (není anonymní)
- dále pak existuje jednotný systém předávání směrovacích informací mezi jednotlivými autonomními systémy, který jsou povinní všichni provozovatelé dodržovat (Jinak řečeno doma si může každý postupovat podle svého, navenek musí všichni postupovat jednotně

PROTOKOLY INTERNETU PODLE VRSTEV

- **protokolová architektura TCP/IP je tvořena čtyřmi vrstvami (bohužel neodpovídá RM-OSI, protože vznikla dříve):**

rozhraní sítě (network interface)

- zajišťuje přístup ke sdílenému přenosovému médiu
- využívá všechny známé přenosové prostředí a všechny známé typy sítí (LAN, MAN, WAN) pro podporu TCP/IP

mezisíťová vrstva (internet layer)

- plní úkoly logické adresace, směrování a předávání datagramů (segmentace, sestavování). - protokol IP poskytuje síťovou službu bez spojení. Každý datagram je samostatná jednotka, která musí obsahovat všechny informace potřebné k doručení. Protokol IP nezaručuje doručení. V tom spoléhá na protokoly vyšších vrstev

transportní vrstva (transport layer)

- představuje mechanizmus pro přenos dat mezi dvěma stanicemi. Nabízí službu se spojením (TCP) nebo bez spojení (UDP, User Datagram Protocol). Na této vrstvě pracují také některé směrovací protokoly (třeba RIP, Routing Information Protocol)

aplikační vrstva (application layer)

- obsahuje protokoly dávající uživatelům konkrétní aplikace. Aplikační protokoly jsou většinou závislé na transportní službě (například HTTP, Telnet a FTP užívají TCP)

- **základní protokoly (to jest služby = aplikace) Internetu jsou následující:**

Telnet	protokol virtuálního terminálu	RFC 854	port 23
FTP	protokol přenosu souborů	RFC 959	port 20, 1
TFTP	jednoduchý protokol přenosu souborů	RFC 1350	port 69
SMTP	jednoduchý protokol transferu pošty	RFC 821	port 25
DHCP	protokol dynamické konfigurace stanice	RFC 2131	port 546,7
DNS	protokol systému doménových jmen	RFC 1035	port 53
HTTP	protokol transferu hypertextových informací	RFC 2616	80
SNMP	jednoduchý protokol správy sítě	RFC 1152	161, 2

protokol ip

- funkcí protokolu je dopravovat datagramy mezi jednotlivými sítěmi (InterNet Protocol)

- **je tvořen několika dílčími protokoly:**
  - základní je protokol IP

- služební protokoly ICMP (signalizace mimořádných stavů), IGMP (doprava adresných oběžníků)
- patří sem i protokoly ARP a RARP avšak jejich rámce nemají IP záhlaví

struktura ip datagramu

- skládá se ze záhlaví a přenášených dat
- záhlaví mívá většinou 20 bajtů, obsahuje-li volitelné položky je delší

- **jednotlivé pole záhlaví jsou následující (hodnota uvedená v závorce je délka pole v bitech):**
  - verze (4), délka záhlaví (4) je ve čtyřbajtech, typ služby (8), délka datagramu v bajtech (16), identifikace datagramu (16), příznaky (3), posunutí fragmentu (13), doba života (8), protokol vyšší vrstvy (8), kontrolní součet záhlaví (16), adresa odesílatele (32) a adresa příjemce (32)

- pole identifikace, příznaky a posunutí se používají mechanizmem fragmentace datagramů. Číslo protokolu vyšší vrstvy je identifikace protokolu, který používá IP datagram ke svému transportu (1-ICMP, 2-IGMP, 4-IP/IP, 6-TCP, 17-UDP, 97 – ETHERNET, 111 – IPX). Pole typ služby se neužívá.

protokol icmp (internet conrol message protocol)

- protokol řídících hlášení
- slouží k signalizaci mimořádných stavů
- účelem protokolu není pouze informování o chybách v datagramech a směrování, ale také slouží k získávání doplňujících údajů a k ověření komunikace mezi stanicemi (PING)
- záhlaví datagramu je vždy dlouhé 8 bajtů. Prvé čtyři bajty obsahují typ zprávy (8), kód zprávy (8) a kontrolní součet (16). Pole typ je hrubé dělení zpráv (0-echo, 3-chyba, 4-žádost), kód pak upřesňuje, o co se jedná (0-nedosažitelná síť, 1-nedosažitelný uzel, 2-protokol, 3-port)

protocol igmp (internet group management protocol)

- protokol správy skupin
- slouží k šíření adresných oběžníků (multicast)
- skupinové vysílání se používá s cílem redukce nebo optimalizace provozu

- **existují tři varianty aplikací založených na skupinovém vysílání:**
  - od jednoho k mnoha (koncerty, přednášky, aktualizace databází)
  - od mnoha k jednomu (sběr dat, zjišťování prostředků)
  - od mnoha k mnoha uživatelům (konference, hry, výuka na dálku)

- stanice, která se chce připojit k určité skupině, nejprve naslouchá datagramům zasílaným na skupinovou adresu 224.0.0.1. Pak oznámí skupině, že se připojuje, sleduje příslušný provoz a případně odpovídá na dotazy. Skupinu lze opustit prostým ukončením naslouchání (IGMP verze 1) nebo vysláním zprávy leave (verze 2)

protokol arp (address resolution protocol)

- slouží k zjištění linkové adresy (fyzická adresa rozhraní) ze známé IP adresy
- provede se to vysláním linkového oběžníku (ARP funguje jen v lokální síti). ARP paket je zabalen přímo do paketu linkové protokolu a nemá tedy žádné IP záhlaví
- **formát paketu ARP je následující:**
  - typ přenosového média (16), typ protokolu (16), délka adresy MAC (8), délka síťové adresy (8), kód zprávy (16), zdrojová adresa MAC (16 nebo 48), zdrojová síťová adresa (32), cílová adresa MAC (16 nebo 48), cílová síťová adresa (32)

- paket oběhne lokální síť a příslušné stanice (nebo směrovač) odpoví odesílateli paketem (kód zprávy=2) s vyplněným polem linkové adresy

protokol rarp (reverse address resolution protocol)

- používá se v případě znalosti vlastní fyzické adresy pro zjištění adresy síťové

- používá se u stanic bez pevných disků

protokol tcp (transmission control protocol)

- je protokolem transportní vrstvy a představuje spolehlivý protokol (transportní služby se spojením)
- zajišťuje spojení mezi jakýmkoli datovými stanicemi Internetu a protokol TCP pak spojení mezi příslušnými aplikacemi běžícími na těchto stanicích

- **formát segmentu TCP je následující:**
  - zdrojový port (16), cílová port (16), pořadové číslo odesílaného bajtu (32), pořadové číslo přijatého bajtu (32), délka záhlaví (4), rezerva (6), příznaky (6), délka okna (16), kontrolní suma (16), ukazatel naléhavých dat (16)

- **protokol TCP plní tyto funkce:**
  - asociuje porty se spojeními
  - navazuje a ukončuje spojení
  - řídí tok dat (segmentuje a čísluje data, potvrzuje příjem), reguluje tok dat, signalizuje urgentní data

- čísla portu mohou nabývat hodnot 0 až 65535). Hodnoty menší než 1024 jsou porty privilegované a používají se pouze privilegovanými uživateli (servery). Hodnoty větší může použít kdokoli, avšak pouze tehdy, je-li port volný. Pořadová čísla přenášených bajtů obecně nezačínají od nuly, ale od hodnoty nastavení při navázání spojení. Položka velikost okna určuje kolik bajtů je možné přenést bez potvrzení (potvrzení přijetí každého segmentu by bylo dosti neefektivní)

protokol udp (use datagram protocol)

- poskytuje nespolehlivou transportní službu. Existují aplikace nepotřebující zabezpečení v takovém rozsahu, jak to provádí TCP nebo jsou transakčně orientovány (dotaz-odpověď) a navazování spojení je na ně příliš zdlouhavé

- **formát protokolu je tento:**
  - zdrojový port (16), cílový port (16), délka (16) a kontrolní součet (16). Protokol dovoluje vysílat na všeobecnou IP adresu (255. 255. 255. 255)

SLUŽBY SÍTĚ

telnet

- dovoluje uživateli připojit se ke vzdálenému síťovému uzlu prostřednictvím protokolu TCP/IP

- **dokument RFC 855 definuje 3 základní služby telnetu:**

  - síťový virtuální terminál (NVT – Network Virtual Terminal)
    - představuje standardní rozhraní vůči vzdáleným systémům
  - schopnost vyjednávání o nastavení určitých voleb
  - syntaktické zobrazení terminálu a procesu

- služba pracuje se spojením TCP, běžně na portu 23
- mezi terminálem a vzdáleným počítačem přenáší osmibitové znaky
- velmi jednoduchá služba – je to pouze otevřené připojení na program pro interpretaci příkazové řádky
- je nezávislý na platformě (nezávisí na OS)

princip činnosti:
- po spuštění se aplikace na počítači uživatele stává klientem
- ten naváže spojení TCP se vzdáleným počítačem (serverem Telnetu) pomocí standardní sekvence
- klient komunikuje pomocí klávesnice a na svém monitoru vidí reakce vzdáleného systému
- server používá normální terminálové rozhraní pro přístup k programu Telnet, který přenáší data do jiného systému

závěr:
- komunikace není hospodárná – každý stisk klávesy generuje znak, který jako datagram putuje systémy (objem komunikace je však minimální)

- Telnet představuje málo propracovanou službu (nulové možnosti programování a nízká úroveň konfigurovatelnosti), stále se používá ke správě vzdálených systémů a jako nástroj k odstraňování vad

síťový virtuální terminál:
- zajišťuje transparentnost všech operací prováděných uživatelem
- je to imaginární zařízení, které převede konkrétní zařízení, na kterém uživatel pracuje na standardní typ (emuluje určitý terminál – VT – 100 nebo IBM 3270)
- formát NVT používá sedmibitový ASCII kód pro znaky a zobrazovací zařízení (128 kódu)
- přenášejí se jako osmibitové položky (nejvýznamnější je bit nastaven na nulu) – právě proto může Telnet pracovat pod různými OS (potlačuje heterogenitu zařízení a systémů)
- řídící příkazy NVT upravují interakci mezi klientem a serverem a jsou začleněny do proudu dat
- příkaz je sekvence dvou nebo tří oktetů

vyjednávání o volbách:
- přestože jsou obě strany reprezentovány jako NVT, probíhá nejdříve výměna dat, ve které se dohodnou parametry a volby komunikace
- obě strany jsou si rovnoprávné – každá může žádat o požití nepovinného prostředku
- pokud strana opačná nemůže či neumí požadavek splnit, zamítne jej

- **požadavky jsou čtyři:**

  - **WIL 251** - odesílatel chce danou volbu zapnout
  - **DO 253** - odesílatel chce, aby příjemce danou volbu zapnul

- **WONT 252** - odesílatel chce danou volbu vypnout
- **DONT 254** - odesílatel chce, aby příjemce danou volbu vypnul

- komunikační strany mohou v kterémkoli okamžiku přenosu vyjednat požití různých voleb a změnit konfiguraci spojení

volby telnetu:

- dostupných voleb je přes 40, záleží na implementaci
- v OS Windows je Telnet program příkazového řádku a obsluhuje se klasickým způsobem – po spuštění lze zadávat příkazy (dají se zkracovat na jedno písmeno – o (open)....
- volby se nastavují slovně příkazem SET

hodnota	název	RFC	význam
0	binární přenos	856	osmibitový binární přenos
1	echo	857	opakovat přijatá data
5	stav	859	dotaz na stav přenosu
6	časová značka	860	vložení časové značky
24	typ terminálu	1091	změnit použitý terminál
31	velikost okna	1073	řízení velikosti okna
32	rychlost terminálu	1079	výměna dat o rychlosti
33	řízení toku dat	1372	povolit, zakázat, regulovat
34	režim řádku	1116	odesílání úplných řádků
36	stavové proměnné	1408	předání dat o konfiguraci

služba přenosu souborů

- **je podporována dvěma základními protokoly:**
  - FTP (File Transfer Protocol)
  - TFTP (Trivial File Transfer Protocol)

ftp

- jeden z nejstarších internetových protokolů (počátky v roce 1971)
- nynější podoba z roku 1985 – dokument RFC 959
- poskytuje služby pro přenos souborů mezi vzdálenými systémy
- používá spolehlivé TCP spojení
- pro přístup ke vzdálenému systému je nutné přihlášení (uživatelský účet)
- služba funguje i mezi systémy, které mají různý způsob ukládání dat nebo kódování znaků

- **od ostatních služeb se FTP liší tím, že otevírá dvě oddělená TCP spojení:**
  - jedno na portu 21 (slouží pro řízení přenosu souborů)
  - druhé na portu 20 (slouží k přenosu dat)

relace ftp

- je akt přenesení souboru z jednoho systému na druhý

- **předpokládá spolupráci pěti softwarových komponent:**
  - uživatelské rozhraní klienta (řídí interpret příkazů protokolu)
  - interpreta protokolu na straně klienta (předává příkazy serveru a řídí přenos na straně klienta)

- interpreta protokolu na straně serveru (reaguje na příkazy klienta a řídí přenos na straně serveru)
- klientských proces přenosu dat (spojka mezi vzdáleným serverem a místním souborovým systémem)
- serverový proces přenosu dat (spojka mezi klientem a serverovým souborovým systémem)

- interpret serveru naslouchá na portu 21 na řídící příkazy (čeká až klient zahájí komunikaci)
- když je řídící spojení navázáno, zůstává aktivní po celou dobu přenosu

- datové spojení na portu 20 se vytváří při přenosu souboru, iniciuje se klientem předáním adresy dat a je aktivní pouze po dobu přenosu souboru

- následně musí komunikující strany vyjednat vyjádření předávaných dat (4+3 způsoby), jejich uložení (struktura souboru, 3 způsoby) a režim přenosu (3 způsoby)

- celkem je možných 72 variant přenosu, které umožňují přenos libovolného druhu dat a souborů bez ohledu na typ souborového systému zdroje a cíle

- předpokládá se, že soubory sdílejí několik základních vlastností a ty jsou podporovány

reprezentace dat

- je dána způsobem kódování znaků a formátem
- formátem se rozumí vertikální formát (stránkování)
- řízení formátu má smysl jen u textových souborů (kódovaných ASCII a EBCDIC)

- **máme tři možnosti:**
  - neurčeno pro tisk (výchozí volba)
  - formát Telnetu
  - formát Fortran

- **přenášená data mohou být kódována čtyřmi možnými způsoby:**
  - ASCII (výchozí volba)
  - EBCDIC (sálové počítače)
  - image (pro systémy stejného typu)
  - místní typ (data o různé délce bajtu, která musí být zachována)

- všechno užité představuje prosté vyjádření přenášených dat

datové soubory

- přenášené soubory mají jistou vnitřní strukturu, která se přenosem nesmí změnit
- procesy přenosu (na obou stranách) zodpovídají za správné mapování mezi přenášenými a místními strukturami (i za situace, že se jedná o různé OS)

- **jsou možné tři možné struktury:**
  - soubor, který není interně nějak členěn (přenáší se jako proud bajtů – výchozí volba)
  - záznamy (pouze textové soubory)
  - stránky (nezávislé číslované stránky – obskurní)

- první struktura je běžná, ostatní mají jen historický význam

režimy přenosu

- určují, jakým způsobem jsou data přenášena

- **celkem tři možné volby:**
  - proud bajtů (výchozí volba, konec přenosu je dán znakem EOF)

- blok (série hlavičkou uvedených bloků)
- komprimovaný režim (používá se jen velmi zřídka)

příkazy ftp

- příkazy jsou řetězce ASCII znaků, které jsou zasílány po řídícím kanálu

- **je jich zhruba 30 a dělíme je do tří skupin:**
  - řízení přístupu
  - nastavení parametrů přenosu
  - služby

- řídící řetězce jsou uvedeny 3 nebo 4mi velkými písmeny a ukončeny znaky CLRF
- příkazy pro řízení přístupu určují, kdo může přistupovat k čemu

- **nejfrekventovanější příkazy:**
  - USER – název uživatele
  - PASS – přístupové heslo
  - CDUP – přechod do nadřazeného adresáře
  - CWD – změna pracovního adresáře

- **příkazy nastavující parametry přenosu:**
  - MODE – režim přenosu (S,B,C – proud (stream,) – blok, komprimovnáno)
  - STRU – struktura dat (F,R,P – soubor, záznam, stránka)
  - TYPE – reprezentace přenášených dat (ASCII, EBCDIC, image, local)
  - PORT – číslo portu klienta, na kterém má proces přenosu dat naslouchat
  - LIST – odešle seznam souborů v adresáři
  - RETR – načte soubor
  - STORE – uloží soubor na server
  - DELE – smaže soubor na serveru
  - MKD – vytvoří adresář na serveru
  - RMD – odstraní adresář na serveru
  - NLST – vypíše obsah adresáře
  - ABOR – zruší provádění příkazu
  - QUIT – ukončí relaci
  - HELP – pomoc

odpovědi na příkazy

- na každý příkaz je nutno odpovědět, zabezpečuje to jednak synchronizace požadavku a prováděných akcí a jednak to informuje uživatele o stavu dění

- příkazy mohou být generovány ve skupině, dojde-li k chybě, pak je třeba opakovat celou skupinu příkazů
- odpovědi jsou posílány jako trojmístné číslo, za kterým může (ale nemusí) následovat textová část. Tím je dosaženo toho, že odpovědi jsou srozumitelné jak softwaru, tak člověku.

formát odpovědí:

- trojmístné číslo, mezera, textová část, CLRF

- **číslice trojmístného čísla jsou kódovány takto:**

  - První číslice: kladná nebo záporná odpověď
    1. kladná, předběžná odpověď (čeká se na další příkaz)
    2. kladná, konečná odpověď
    3. kladná, dočasná odpověď (nutný další příkaz)
    4. záporná dočasná odpověď (příkaz nebyl dokončen, opakujte)

5. záporná, trvalá odpověď

- Druhá číslice: specifikuje, čeho se odpověď týká
  0. chyba syntaxe
  1. informace
  2. stav spojení
  3. ověřování a evidence
  4. zadáno
  5. stav souborového systému

- Třetí číslice: specifikuje stav
  200 = příkaz OK
  331 = uživatel potvrzen (vyčkává na zadání hesla)
  500 = chyba syntaxe

tftp

- původní verze protokolu pracuje s bloky pevné délky (512B)
- používá se UDP (nespojované) spojení na portu 69 (RFC 1350) -> chybí jakékoli záruky doručení či zabezpečení

popis přenosu:

- přenos je iniciován klientem, která otevře port proměnného čísla a zašle žádost na známý port 69
- odeslaný datagram definuje požadavek a specifikaci souboru
- server požadavku přiřadí nový UDP port a zahájí přenos
- soubor je přenesen jako kontinuální proud datagramů o pevné délce
- závěrečný blok je kratší, což signalizuje ukončení přenosu
- příjemce potvrzuje přijetí každého bloku (ACK) – další blok nebude odeslán, dokud nebude doručení předcházejícího potvrzeno (lock-step).
- bloky jsou číslovány vzestupně, počínaje od jedničky
- dojde-li při přenosu k jakékoli chybě, je přenos ukončen a musí se celý zopakovat
- chybu signalizuje chybový datagram, který specifikuje chybu a ukončuje spojení

- **protokol rozlišuje tři typy chybových stavů:**
  - požadavek nelze obsloužit
  - zpožděný nebo duplikovaný datagram
  - během přenosu došlo ke ztrátě přístupu k některému zdroji
  - (celkem 7 chybových kódů)

- **později byl protokol rozšířen:**
  - v úvodní fázi komunikace je možné zapnutí nebo vypnutí určitých voleb (v současnosti je volba jen jedna – velikost bloku)
  - rozšíření je však natolik pružné (je definováno pouze vyjednávání o volbách), že výrobce může zavádět volby vlastní

mime

- Multipurpose Internet Mail Extensions
- víceúčelové ujednání jak ke zprávám elektronické počty přibalovat nerůznější přílohy, jak psát texty zpráv i v jiných jazycích (znakových sadách), než je v čistém (sedmibitovém) ASCII kódu

- jednotlivé dílčí konvence, které byly původně vyvinuty pouze pro potřeby elektronické pošty, se ale dnes úspěšně využívají i mnoha jinými webovými službami (například hypertextem)

- základním důvodem pro existenci MIME je skutečnost, že mechanizmus přenosu zpráv, dodnes používaný v Internetu a označovaný jako elektronická pošta (SMTP), garantuje pouze přenos zpráv tvořených sedmibitovými znaky (čistý ASCII kód)
- je-li zapotřebí přenést elektronickou poštou cokoli jiného, co nemá podobu sedmibitových ASCII znaků, je nutné to do takovéto podoby nejprve přenést
- možnosti jak zakódovat osmibitová data do podoby čistých ASCII znaků je ovšem několik
- úkolem konvence MIME pak je přesně vymezit (a pojmenovat) několik takovýchto možností, a ty zavést jako přípustné (v tom smyslu, že odesílat si může mezi nimi sám vybírat)
- navíc je pak nutné v rámci konvence MIME učinit taková opatření, aby o použité volbě kódování byl informován i příjemce a věděl jak příslušnou část zprávy transformovat do původní formy
- po úspěšném nabytí původní přílohy musí příjemce následně zjistit co vlastně získaná data představují a jak je má případně zpracovat (například, že se jedná o archiv ZIP, obrázek, spustitelný program)
- možností je celá řady a nejsou nijak zásadně omezeny – nové datové formáty vznikají průběžně
- další část konvence MIME tedy umožňuje definovat, co předaná data představují (typ přenášených dat)

- **konvence MIME řeší tedy dvě základní otázky:**
    - jak vytvořit ze zprávy, která obsahuje binární data, zprávu přepravitelnou používanými přenosovými protokoly
    - jak rozlišit jednotlivé druhy zpráv (zavádí klasifikaci zpráv) užitečná i pro jiné webové služby

    - konvence je popsána v dokumentech RFC2045 AŽ RFC2049
    - základním pojmem jsou řádkové hlavičky MIME
    - právě přítomnost (nebo nepřítomnost) hlavičky určuje, zda přenášená zpráva vyhovuje (nevyhovuje) konvenci – tak je zajištěna zpětná kompatibilita
    - hlavičky se přenášejí jako čistý ASCII text (existují i výjimky)

    - **existují dva základní typy hlaviček:**

        - version (verze)
            - musí být vždy uvedena, je to potvrzení, že zpráva vyhovuje konvenci MIME
            - existuje z důvodu zachování kompatibility
        - content (obsah)

hlavičky

- **content-type**
    - hlavička, která definuje typ a podtyp dat obsažených v těle zprávy
    - typ určuje o jaká data se jedná a podtyp, pak udává konkrétní formát

    - **typy dat jsou dvojího druhu:**
        - **jednoduché**
            - text
            - application
            - image
            - audio
            - video
            - model

        - **- složené (kompozitní)**
            - message
            - multipart
            - report

- **formát:** Conent-Type:typ/podtyp;parametry

- **content-transfer-encoding**
  - je hlavička, která specifikuje mechanizmus převodu obecných dat na čistý ASCII kód (transformační algoritmus a kódování)
  - **dva převodní mechanizmy:**
    - Quoted-Printable
    - Base64
  - **celkem máme 6 možností:**
    - quoted-printable
    - base64
    - 7bit
    - 8bit
    - binary
    - x-rozšíření

- **content id**
  - většinou volitelná hlavička, která obsahuje popisné informace o přenášené zprávě (například to může být název obrázku, který je přenášen jako tělo zprávy)

- **content – disposition**
  - určuje, zda jsou přenášená data určena k automatickému zobrazení (inline) nebo je má příjemce zpracovat ručně (attachement)
  - **může obsahovat i další parametry:** jméno souboru, datum vytvoření, čtení a modifikace, velikost souboru

standardní kódovací mechanizmus

- **quoted-printable**
  - poskytuje výsledný text člověku vcelku srozumitelný
  - způsob transformace
  - ASCII znaky zůstanou beze změny, ostatní se nahradí znakem = a hexadecimální hodnotou nahrazovaného znaku
  - zakódovaný řádek musí mít maximálně 72 znaků – pokud není, vloží se = CRLF (měkký konec řádku)
  - nevýhodou této transformace je prodloužení zprávy na trojnásobek v případě, že všechny znaky nebudou US-ASCII

- **- base64**
  - je nečitelné
  - používá kódovací tabulku o 64 znacích (6 bitů)
  - zpráva je zpracována jako proud bitů, který se rozdělí po šesti bitech a pode tabulky Base64 zakóduje (tabulka je jednoduchá: 0=A, 1=B....)
  - postup je trochu komplikovanější, protože se vždy zpracovávají tři bajty, pokud na konci zbude méně než 24 bity, dorovná se to výplňovým znakem = (což znamená, že délka výchozího souboru v bajtech, musí být vždy dělitelná třemi nebo se dorovná znakem =)

elektronická pošta

- jedna z nejstarších webových služeb
- dnešní podoba – dokument RFC-821, RFC-822
- základním nedostatkem původní specifikace byl přenos zpráv ve formátu US-ASCII (později odstraněno konvencí MIME)

architektura služby

poštovní klient

- je komponenta, která komunikuje s uživatelem

- **v podstatě je to pouze specializovaný textový editor, který umí:**
    - manipulovat se zprávami ve formátu elektronické pošty (zobrazit obsah zprávy z poštovní schránky)
    - umí manipulovat se zprávami ve schránce
    - umí pořídit zprávu a předat ji k odeslání

- odesláním se nerozumí nějaká síťová komunikace, ale její uložení do fronty zpráv SMTP

klient

- pravidelně obchází frontu zpráv a navazuje spojení se vzdálenými SMTP servery, kterým zprávu následně předá
- SMTP server přijme zprávu a zjišťuje, je-li určena pro jeho lokální uživatele
- v případě, že tomu tak není, uloží zprávu do fronty, kterou obsluhuje jeho poštovní klient
- ten se pokusí zprávu doručit směrem k adresátovi – to se postupně opakuje
- při zjištění, že adresát je lokální uživatelem systému, SMTP server uloží přijatou zprávu do poštovní schránky adresáta
- uživatel má v systému zpravidla jednu poštovní schránku na serveru (nazvanou inbox nebo podle uživatele), kam SMTP server ukládá jeho příchozí poštu
- mimo to si uživatel může zřídit i privátní poštovní schránku, kam si ukládá ze schránky na serveru došlou poštu
- privátní schránky nejsou obsluhovány SMTP serverem a bývají zřizovány v domovském adresáři uživatele - důvod je prostý – cílem takového řešení je přimět uživatele k tomu, aby v serverové schránce příchozí poštu nearchivovali
- některé poštovní klienti pracují tak, že pokud příchozí poštu uživateli zobrazí, tak mu ji automaticky (transparentně) přenesou do privátní poštovní schránky
- elektronická (webová) pošta má díky ukládání odchozích zpráv do front a ukládání příchozí pošty do schránek jednu unikátní vlastnost – uživatel může odeslat mail, který si příjemce může vyzvednout ze své schránky, až bude chtít (není tedy nutné bezprostředně navazovat spojení na příjemcův systém v době odesílání, systém příjemce může být i vypnut)
- nepodaří-li se klientovi SMTP zprávu odeslat, pak ji ve frontě ponechá a bude se cyklicky snažit ji doručit
- to však nemůže trvat věčně – po uplynutí správcem systému definované doby (většinou 2-7 dnů) bude zpráva označena jako nedoručitelná a poslána zpět odesílateli

smtp

- Simple Mail Transfer Protocol
- jednoduchá posloupnost textových příkazů
- zajišťuje přenos zpráv elektronické pošty
- komunikační kanál k transportu zpráv využívá TCP protokol a známý port 25
- přenos zprávy představuje sérii transakcí žádost/odpověď (architektura klient/server)
- klient navazuje spojení na portu 25 a vkládá do takto vytvořeného komunikačního kanálu textové příkazy
- server odpovídá třímístným číselným kódem, případně následovaným textovým popisem stavu
- příkazy klienta mají délku čtyři znaky (nerozlišují se velká a malá písmena) a vždy jsou zakončena znaky CRLF
- příkaz může mít parametry, které se oddělují od úvodní části mezerou

- **základní příkazy:**
    - relace je uvedena příkazem HELO
    - odesílatel je definován příkazem MAIL FROM
    - příjemce je uveden příkazem RCPT TO
    - vlastní zpráva je uvedena příkazem DATA
    - zpráva je ukončena znakem . (tečka)

- RSET je abnormální ukončení relace
- VRFY je dotaz, za server zná uvedenou adresu
- QUIT – je ukončení spojení
- HELP – získání seznamu serverem podporovaných příkazů

- **relace posílání elektronické zprávy pobíhá následujícím způsobem:**
  - po navázání spojení (HELO) se server představí (220 a identifikace serveru)
  - následuje příkaz MAIL a RCPT (server odpovídá 250, je-li všechno správně)
  - následuje příkaz DATA, na který server odpovídá 354
  - zpráva je ukončena znakem tečka (server odpoví 250)
  - spojení se serverem je ukončeno příkazem QUIT (server odpoví 221)

esmtp

- Extensions SMTP
- rozšíření jednoduché poštovního protokolu
- dokument RFC-1869
- zpětná kompatibilita s SMTP je zajištěna velmi elegantním způsobem
- klient začne komunikaci příkazem EHLO
- server buď odpoví, že se klient spletl v příkazu (pak je jasné, že jde o SMTP server)
- nebo 250 (OK) a pak je zřejmé, že se jedná o ESMTP server (který hned pošle seznam příkazů, kterým rozumí)

- **podporované příkazy:**
  - 8BITMIME – server podporuje konvenci MIME
  - SIZE – maximální délka zprávy, kterou je schopen server akceptovat
  - VERB – server bude vypisovat podrobný protokol o komunikai
  - ONEX
  - ETRN

- podle původní specifikace (oba protokoly) se kdokoli může připojit na poštovní server a odeslat mail
- dnes je však nutné se po otevření spojení serveru identifikovat

- formát zprávy

- elektronická poštovní zpráva se skládá z obálky a z vlastní zprávy
- obálka je tvořena elektronickými adresami odesílatele a příjemce
- zpráva má záhlaví a tělo
- záhlaví je od těla zprávy odděleno jedním prázdným řádkem a skládá se z jednotlivých hlaviček
- hlavička je uvedena dvojtečkou, za kterou mohou následovat parametry
- tělo zprávy obsahuje jen US-ASCII text a řádek nesmí být delší než 1000 znaků
- maximální velikost zpráva je také dána

- **některé znaky mají zvláštní význam:**
  - středník a dvojtečka jsou oddělovače v seznamu
  - špičaté závorky <> použité v adrese znamenají, že se použije pouze to, co je mezi nimi (ostatní se ignoruje)
  - do kulatých závorek se uzavírá komentář
  - hranaté závorky ve jméně počítače mají význam, že jméno nepotřebuje překlad

  - **existuje asi 16 hlaviček:**

- **received**
  - má zvláštní postavení
  - do záhlaví zprávy ji připisuje každý server, kterým zpráva prošla --- tolik hlaviček RECIEVED, kolik servery prošla

- **recent**
  - přidá se na začátek záhlaví při automatickém poslání zprávy (například při vrácení nedoručitelné zprávy)

- FROM (od)
- SENDER (odesílatel)
- DATE (datum)
- TO (adresát)
- COMMENTS (komentář)
- REPLY-TO (odpovídejte na adresu)
- SUBJECT (věc)
- MESSAGE-ID (identifikace zprávy)
- IN-REPLY-TO (identifikace původní zprávy)
- KEYWORDS (klíčová slova charakterizující obsah)
- CC (na vědomí)
- BCC (tajná kopie)

pop3

- Post Office Protocol verze 3
- jednoduchý protokol pro manipulaci s obsahem lokální poštovní schránky
- dokument RFC-1939
- klient otevře komunikační kanál vytvořením TCP spojení na známý port 110
- následuje autentizace klienta jménem a heslem (na rozdíl od SMTP je vyžadována vždy)
- pokud tato akce skončí úspěšně, přejde se do transakční fáze, kdy klient může pracovat se zprávami ve své poštovní schránce
- protokol je jednoduchý, příkazy se zadávají jako US-ASCII kódovaná klíčová slova
- komunikace je koncipována tak, že v daný moment může se schránkou pracovat pouze jeden klient
- důvod je prostý: při přihlášení uživatele se schránka zkopíruje a klient pracuje s kopií
- původní schránka zůstane zachována (mohou do ní přicházet další zprávy) a obě kopie se slijí dohromady po ukončení práce (po UPDATE)
- při přihlášení ke schránce se nejprve testuje, zda neexistuje kopie, pokud ano, další přihlášení není možné

imap

- Internet Message Access Protocol
- je vylepšený protokol pro práci s poštovními schránkami elektronické pošty
- RFC-1730
- poslední verze 4 (RFC-2060)
- umožňuje současný přístup ke schránce z více aplikací (i během práce se schránkou bude nově příchozí zpráva do ní přidána, tento stav bude signalizován uživateli)
- nově otevřená schránka je přístupná pro čtení a zápis
- pokud další aplikace otevře schránku, bude mít přístup pro čtení i zápis, ale prvé aplikaci bude status schránky změněn na pouze pro čtení (a tento stav jí bude signalizován, bude-li prvá aplikace potřebovat do schránky psát, bude muset jí znovu otevřít)
- nekorektní jednání libovolné aplikace způsobí, že server spojení ukončí

- protokol využívá TCP spojení na známém portu 143 a používá jednoduché textové příkazy kódované v US-ASCII
- příkazy jsou však odlišné od příkazů POP3
- nejde jen o to, že příkazy jsou jiné, mají i jinou filosofii
- může být najednou (nebo postupně) zadána celá řada příkazů a odpovědi na ně mohou přicházet v libovolném pořadí
- proto klient příkazy označuje a server v odpovědi uvádí i označení příkazu, na který odpovídá (jak je označuje věcí klienta, nemusí to být číslo)

- **server posílá dva druhy odpovědí:**
  - **neoznačené odpovědi**
    - mají hvězdičku na místě označení odpovědí
    - nesou požadované odpovědi
  - **označené odpovědi**
    - začínají označením příkazu
    - sdělují (číselným kódem a případně textovým doplněním) jak plnění příkazu dopadlo

- **průběh komunikace je zhruba následující:**
  - po navázání TCP spojení na portu 143 nastane neautentizovaný stav a klient se musí k serveru přihlásit (příkazy LOGOUT, NOOP a CAPABILITY)
  - autentizaci klienta je možné provést pomocí jména a hesla (příkazem LOGIN) nebo použít jinou metodu (příkaz AUTENTICATE)
  - pokud server metodu podporuje, odpoví znakem plus
  - autentizační data jsou kódovány Base64

- **po identifikaci klienta je možné posílat příkazy:**
  - CREATE – vytvoř schránku
  - DELETE – zruš schránku
  - RENAME – přejmenuj schránku
  - LIST – seznam adresáře
  - STATUS – informace o schránce
  - SELECT – otevře schránku
  - EXAMINE – otevře schránku pouze pro čtení
  - COPY – zkopíruje zprávy z jedné schránky do druhé
  - SEARCH – hledání ve schránce, cca 30 možností
  - FETCH – stažení zprávy nebo její části
  - STORE – změna atributů zprávy
  - EXPUNGE – zrušení označených zpráv
  - CLOSE – uzavře otevřenou poštovní schránku

hypertext

- **pojem označující seskupení textů, které jsou mezi sebou propojeny pomocí odkazů (linků)**
  - to umožňuje čtenáři procházet textem **nelineárně**, bez přesně definovaného počátku, konce i posloupnosti textu
  - princip poprvé představen v roce 1945 (by Vannevar Bush v článku As We May Think, v němž popisuje zařízení MeMex (Memory Extender), umožňující ukládání informací a jejich vzájemné propojování
  - samotný termín hypertext vznikl až v roce 1965, zavedl jej Theodor Nelson, který je rovněž autorem jeho první počítačové realizace (informační katalog Xanadu)

- vedle pojmu hypertext existuje i termín **hypermédium**, kterým označujeme **hypertext obohacený o vizuální, sluchové či jiné informace**
- **hypertext je základem** populární **služby World Wide Web**, což je relativně nová záležitost, která využívá (realizuje) skoro 50 let starý princip

- **podstata**
  - **lidé obvykle nemyslí přímočaře**, nedomyslí určitou myšlenku až do konce, aby teprve poté přešli na myšlenku jinou
  - zkušenost ukazuje, že přemýšlíme jakýmisi přeskoky z myšlenky na myšlenku
  - důvodem k tomu jsou patrně asociace s jinými myšlenkami, které výchozí myšlenky vyvolávají (možná, že právě v těchto asociačních mechanismech spočívá unikátnost tvůrčího lidského myšlení)
  - s tím ostře kontrastuje lineární uspořádání většiny existujících textů (například knihy)
  - poprvé na tuto zajímavou disproporci upozornil právě pan Vannevar Bush
  - právě on pak přišel s nápadem přizpůsobit psané texty nelineárnosti lidského myšlení (za to je mu přisuzováno autorství myšlenky hypertextu)
  - pánové Bush i Nelson neuspěli – doba zřejmě nedozrála pro takové inovace
  - jejich nápady však nebyly zapomenuty
  - vzpomněl si na ně pan Tim Berners-Lee, když měl ve švýcarském CERNu vymyslet způsob, jak by mohlo společenství fyziků (zabývajících se vysokými energiemi) mezi sebou jednoduše a efektivně sdílet informace, které měly (tehdy) převážně psanou (tedy textovou) podobu
  - vyšel z myšlenky hypertextu: **původně lineární psané texty rozdělil na menší celky, nazývané uzly** (nodes či dokumenty; v prostředí služby WWW pak stránky – pages) **a vzájemně je provázal odkazy, tedy vazbami, které začínají na určitém místě výchozího dokumentu a směřují na místo jiné téhož dokumentu nebo i dokonce do dokumentu jiného**
  - právě tyto odkazy pak jsou obdobami asociací, které vyvolávají u lidí přeskoky z myšlenky na myšlenku
    - **v rámci WWW služby jsou odkazy implementovány tak, aby představovaly aktivní volbu**
    - **po navolení (kliknutím) je čtenář přenesen na místo, kam odkaz ukazuje**

news

- Protokol NNTP (Network News Transfer Protocol) je přenosový protokol pro síťové diskusní skupiny v Internetu. Služba umožňuje uživatelům číst a psát na diskusní skupiny, které jsou známé jako news.

SLUŽBY DOMÉNOVÝCH JMEN A PŘIDĚLOVÁNÍ ADRES

DOMÉNOVÁ JMÉNA

- přenosové protokoly internetu identifikují jednotlivé uzly sítě prostřednictvím IP adresy, které jsou celosvětově unikátní. Člověk si ale takové adresy (32 bitová čísla, ať jsou zapsána jakkoli), velmi špatně zapamatovává. Lidé raději používají mnemonická (symbolická) jména. Což by bylo možné, kdyby ke jménům existovala jednoznačně převodní tabulka - takovému symbolickému jménu odpovídá taková IP adresa. Internet je však celosvětová síť a je proto potřeba zavést další (v podstatě organizační) opatření, aby byl takový převod realizovatelný.

- předpokládaný systém přidělování jmen musí splnit tyto základní požadavky:
    - symbolická jména nebudou přidělována libovolně (protože jméno musí být unikátní)
    - systém jmen musí mít nějakou hierarchickou strukturu (protože jinak se smysluplné názvy rychle vyčerpají a zbudou jména nic neříkající, což je právě to, čemuž se snažíme zabránit)

- tak byl vytvořen systém doménových jmen (DNS – Domain Name System)
- doména je skupina jmen, které k sobě logicky patří (názvy uzlů sítě jedné firmy, organizace, země)
- doménové jméno reflektuje příslušnost uzlu do určité skupiny
- v rámci domény je možno vytvářet další podskupiny (subdomény)
- systém doménových jmen má hierarchickou (stromovou) strukturu – nejvyšší instancí je root

- v root doméně jsou definovány generické domény (TLD – Top Level Domains), mají označení:
    - edu (školství)
    - com (komerce)
    - net (internet)
    - org (nekomerční organizace)
    - mil (armáda)
    - int (mezinárodní organizace)
    - gov (vláda USA)

    - používají se převážně v USA (má to historické kořeny)
    - následně byly to TL domény přidány národní domény (například cz)
    - zatím poslední rozšíření je z roku 2000, kdy byly přidány domény aero, biz, coop, info, jobs, mobi, museum, name, pro, travel, eu
    - vrchním správcem je IANA
    - každá doména má minimálně jednoho registrátora (generické domény), většinou několik až mnoho (národní domény)

syntaxe
- doménové jméno se uvádí v tečkové notaci (řetězce znaků oddělené tečkou)
- první řetězec je jméno počítač, další je jméno nejnižší vnořené domény
- zcela vpravo je doména nejvyšší úrovně (přesněji, zcela vpravo je root, tečka, která se většinou vynechává)
- celé jméno může mít maximálně 255 znaků, řetězce pak 63 znaky
- povolené znaky jsou písmena, číslice a pomlčka, která nesmí být na začátku ani na konci řetězce
- pozor!!! – autonomní systémy dělí internet z hlediska směrování, domény podle jmen uzlů

překlad doménových jmen
- dokum- DNS (Domain Name System)
- je hierarchický systém doménových jmen
- realizace zabezpečují servery DNS pomocí protokolu stejného jména
- primárním úkolem a příčinou vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě
- později byly doplněny další funkce (například pro elektronickou poštu a IP telefonii)

- dnes slouží především jako distribuovaná databáze síťových informací
- základem systému je protokol, který využívá UDP komunikaci i TCP spojení vždy na známém portu 53
- DNS servery mají hierarchickou organizaci, podobně jako doménová jména
- domény umožňují lepší orientaci lidem, IP adresy jsou pro stroje
- systém DNS umožňuje efektivně udržovat decentralizované databáze doménových jmen a jejich překlad na IP adresy
- stejně tak zajišťuje zpětný překlad IP adresy na doménová jména

- používání jmenných názvů je pro člověka daleko příjemnější než posouvání složitých čísel (IP adres). Potřeba používat jiný systém adres pro člověka a jiný pro stroj, vznikla už v ranných dobách ARPAnetu. V počítačích se to provádělo tak, že na všechny počítače v síti byl distribuován soubor (většinou manuálně), obsahující tabulku pro překlad (v Unixu /etc/hosts). Tato koncepce velmi rychle přestala vyhovovat, především díky nárokům na rychlou aktualizaci. Přesto se tento soubor používá dodnes. V závislosti na konfiguraci systému je možné jej použít buď prioritně před dotazem na DNS nebo v případě, že DNS server neodpovídá. Historie: V roce 1983 vyvinul Paul Mockapetris DNS protokol, který je popsán v dokumentech RFC 882 a RFC 883. V roce 1987 byl protokol aktualizován (RFC 1034, 1035). Dnes existuje cca 30 RFC entů týkajících se DNS

struktura dns

- prostor doménových jmen má hierarchickou strukturu (strom)
- každý uzel struktury obsahuje data o své části jména, které je mu přiděleno a odkazy na své subdomény
- root je kořenová doména (zapisuje se jako tečka)
- hierarchicky níže se nacházejí domény nejvyšší úrovně (TLD)
- ty jsou buď generické (tematické) nebo národní
- celá struktura se administrativně dělí do zón
- každá zóna má svého správce a svůj (autoritativní) jmenný server
- výhodou takového organizačního uspořádání je možnost zóny dále dělit a správu nové části svěřit někomu dalšímu
- právě delegování pravomocí a distribuovaná správa jsou klíčové vlastnosti systému doménových jmen

dns servery

- **každá zóna má nejméně dva DNS servery:**

primární server

- vznikají na něm data
- pokud je třeba provést v doméně nějaké změny, musí se tak učinit na primárním serveru
- každá zóna má právě jeden primární server

sekundární server

- je automatickou kopií primárního
- průběžně si aktualizuje data podle primárního serveru
- slouží především jako záloha pro případ výpadku primárního serveru
- také zabezpečuje rozkládání zátěže u frekventovaných domén
- každá doména musí mít nejméně jeden sekundární server (sekundárních serverů může být vícero)

pomocný server (caching only)

- slouží jako vyrovnávací paměť pro snížení zátěže systému
- uchovává si odpovědi, dokud nevyprší jejich platnost a poskytuje je při opakujícím se dotazu
- odpovědi pocházejí od primárního či sekundárního serveru jsou autoritativní (konečné)
- odpovědi pomocného serveru nejsou autoritativní
- v případě nutnosti může klient požádat o autoritativní odpověď primární či sekundární server

- jmenný server může být pro jednu zónu primárním a pro jinou sekundárním serverem
- všechny root servery jsou primární

dns dotaz

- je relace přeložení jména na IP adresu (popřípadě naopak)

- klient relaci inicializuje – resolver posílá požadavek jmennému serveru
- aby tak mohl učinit, musí mít ve své konfiguraci síťových parametrů adresu lokálního DNS serveru, na který se má obracet (typicky ji získá pomocí DHCP)

- **relace probíhá následujícím způsobem:**
  - klient má požadavek na přeložení jména www.yyy.zzz
  - reslover pošle dotaz lokálnímu jmennému serveru (LNS) a očekává jednoznačnou odpověď
  - LNS zná adresy root serverů, pošle tedy některému dotaz
  - root NS dopoví seznamem NS pro doménu zzz
  - LNS odešle dotaz NS domény zzz, který odpoví seznamem NS domény yyy
  - LNS odešle dotaz NS subdomény yyy, který odešle konečnou (autoritativní) odpověď
- uvedený průběh předpokládá, že žádný z dotazovaných NS, kromě posledního požadovaného odpověď nezná (protože neřešil nedávno takový požadavek)

shrnutí

- kořenové servery mají autoritativní informace o doménách nejvyšší úrovně
- konkrétně znají všechny jejich autoritativní servery
- dotaz je tedy následně směrován na některý z autoritativních serverů nejvyšší úrovně (TLD), v níž se nachází cílové jméno
- ten je opět schopen poskytnout informace o své doméně a posunout řešení o jedno patro dolů v doménovém stromě
- relace je sekvence rekurzivních dotazů – výsledkem je autoritativní odpověď (+ nebo -)

dns komponenty

- každá komponenta představuje jednu vrstvu systému doménových jmen

- **systém doménových jmen se skládá ze tří hlavních komponent:**
  - **soustava hierarchicky uspořádaných jmenných prostorů**
    - což je realizováno jako distribuovaná databáze záznamů (záznam jméno a asociovaná informace)
  - **jmenný server**
    - program, který zná strukturu jmenných prostorů a umí manipulovat se záznamy databáze
    - DNS se mu jeví soustava místně uložených dat zvaných zóny
    - server musí periodicky aktualizovat svá data (ze souboru na disku nebo komunikaci s cizími servery, ergo jemu se to jeví jako dynamický systém) a souběžně musí odpovídat na dotaz resolverů
  - **resolver**
    - program, který komunikuje s jmennými servery, od kterých získává informace podle požadavků klienta
    - uživatel k němu přistupuje pomocí jednoduché procedury nebo služby operačního systému
    - DNS se mu jeví jako stromová struktura a on může požádat o informace, z kterékoli sekce (větve)
    - DNS vidí jako neznámý počet jmenných serverů
    - každý z nich zná (spravuje) určitou část distribuované databáze doménového stromu
    - uložená data se mu jeví víceméně statická

dns funkce

- jmenný server po svém startu přenese do paměti data pro zónu, kterou spravuje
- primární sever tak učiní načtení ze souboru na lokálním disku
- sekundární server získá data od serveru primárního (dotazem zone transfer)
- tato data se označují jako autoritativní (nezvratná, nekonečná)
- dále oba servery přenesou do paměti data, která nejsou součástí zóny, kterou spravují - především se jedná o data umožňující spojení na root servery a případné odkazy na servery spravující subdomény (delegace pravomoci) - tato data se označují jako neautoritativní

- součástí systému je paměť cache:
    - do ní se ukládají kladné (případně i záporné, negativní caching) odpovědi na dotazy, které provedly jiné jmenné servery (což šetří čas při případných opětovných dotazech)
    - tyto data jsou opět neautoritativní

- **pro přenos dotazů se používá UDP protokol (komunikace bez spojení):**
    - vyšle se datagram prvému server, a pokud odpověď nepřijde (čekání je velmi krátké), vyšle se datagram dalšímu serveru (to se cyklicky opakuje do získání odpovědi nebo vypršení časového intervalu)
    - bere se ta odpověď, která přijde jako první, byť by byla negativní
    - délka UDP datagramu je omezena na 512 B a fragmentace se nepoužívá
    - pro zónové přenosy (nebo pokud je odpověď delší) se používá TCP spojení (vždy port 53)

dns záznamy

- distribuovaná databáze je soustav místně uložených dat
- reálně je to datový soubor, který obsahuj úplné informace o příslušné zóně ve tvaru zdrojových vět (RR, Resource Records)
- všechny zdrojové věty mají stejný formát (strukturu)

- **struktura vět**
    - **doménové jméno (name)** - pro něž je záznam vytvořen
    - **typ vět (type)** - specifikuje účel věty
    - **třída věty (class)** - určuje rodinu protokolů, k níž se věta vztahuje
    - **doba platnosti, exspirace (ttl – time to live)** - 32 bitové číslo, udávající dobu v sekundách, po kterou může být věta udržována v cache serveru (hodnota 0 znemožňuje uložení věty do vyrovnávací paměti)
    - **délka datového pole (rdlegth)** - představuje 16bitové číslo určující délku datové části věty
    - **vlastní data (RDATA)** - jako řetězce znaků různé délky (formát závisí na typu a třídě věty)

- **typy zdrojových vět**

    - **některé frekventované typy zdrojových záznamů:**

- **soa (start of authority)**
    - určuje autoritativní jmenný server zóny
    - věta uvozuje data zóny
    - v datovém souboru vždy právě jedna věta
- **a**
    - přiřazení IP adresy doménovému serveru
- **ns**
    - věta definující jmenný server zóny
- **cname (canonical name for alias)**
    - uvádí synonymum (alis) k doménovému serveru
    - umožňuje přiřadit k jednomu jménu několik IP adres (služba je poskytována několika servery)
- **ptr**
    - umožňuje reverzní překlad
- **hinfo**
    - slouží k charakterizaci hostitelského počítače
    - obsahuje jak popis HW tak i SW
    - má pouze informativní charakter
- **aaaa**
    - přiřazení IP6 adresy doménovému serveru

- **wks (well know service description)**
  - popisuje ostatní služby hostitelského počítače
- **mx (mail exchange)**
  - věta pro server elektronické pošty
  - je určena pro e-maily
  - specifikují poštovní server domény (není uveden v adrese mailu, není to žádoucí)
  - věta obsahuje jednak IP adresu poštovního serveru a jednak jeho prioritu (číselná hodnota)
  - daná doména může mít poštovních serverů vícero
  - pošta se doručuje na server podle priority (nejprve serve s nejnižším číslem – nejvyšší priorita)
- **text**
  - textové pole
  - má pouze informativní charakter
- **pole name**
  - definuje doménové jméno
  - pokud není vyplněno, vezme se jeho hodnota z předcházejícího řádku
  - nemá-li na konci tečku, přidá se automaticky jméno domény uvedené ve větě SOA (má-li tečku na konci je to jméno absolutní)

dns protokol

- služba překladu doménových jmen je realizována jednoduchým protokolem
- resolver pošle dotaz a server na něj odpoví
- podobně jako u jiných aplikačních protokolů, dotazy i odpovědi jsou textové řetězce (komplikací je komprese doménového jména)
- v závislosti na účelu, protokol definuje několik typů operací
- s diverzifikací protokolu přibývá a typů operací
- pro náš účel je základní DNS dotaz

- **datový formát**
  - DNS dotaz (query) používá stejný formát paketu jak pro dotaz tak pro odpověď
  - může se skládat ze záhlaví až čtyř dalších sekcí

  - **záhlaví je povinné a skládá se ze šesti 16bitových polí:**

    - **ID**
      - identifikátor zprávy, který vkládá klient a server kopíruje do odpovědi
      - slouží k párování dotaz-odpověď
    - **pole řídících bitů**
      - zbývající čtyři pole mají stejný formát a představují kladná celá čísla
    - **qdcount**
      - udává počet položek v dotazu
    - **ancount**
      - říká, kolik zdrojových vět obsahuje odpověď
    - **nscount**
      - oznamuje počet vět definujících autoritativní jmenné servery
    - **arcount**
      - určuje počet položek v doplňující odpovědi
      - **řídící bity mají následující význam:**
        - QR – dotaz/odpověď
        - AA – autoritativní odpověď
        - TC – zkráceno (odpověď se nevešla do 512 oktetů)
        - RD – klient požaduje rekurzivní překlad
        - RA – server umožňuje rekurznivní překlad

- **opcode**
  - čtyřbitové pole
  - specifikuje typ dotazu (standardní, inverzní, na status serveru)
- **rcode**
  - čtyřbitové pole
  - charakterizuje odpověď (bez chyby, chyba formátu dotazu, server neumí odpovědět, jméno neexistuje, server rento dotaz nepodporuje, server odmítá odpovědět)
- **sekce dotazu**
  - **skládá se ze tří polí:**
    - QNAME – doménové jméno
    - QTYPE – typ požadované odpovědi (jakou větu si RR přejí)
    - QCLASS – třída dotazu (1 = Internet)
- **sekce odpovědi**
  - obsahuje pole TTL, RDLENGTH, RDATA

- doménové jméno není zde zapsáno v tečkové notaci, ale každá část jeho jména je uvedena bajtem, který uvádí délku následujícího řetězce
- konec jména je signalizován nulovou hodnotou délky
- pro dosažení minimální délky paketu je doménové jméno komprimováno
- to se provede tak, že se jméno uvede v paktu je jednou a každý další výskyt se nahradí odkazem na prvé uvedení

- **mechanizmus je následující:**
  - maximální možná délka řetězce je 63 (00111111) pokud tedy bajt délka začíná 11....pak to znamená, že se jedná o jméno, ale o odkaz

PŘIDĚLOVÁNÍ ADRES

adresy

- internet je informační prostor
- pokud v tomto prostoru potřebujeme něco najít, musíme vědět, kde je uloženo – potřebujeme znát cílovou adresu
- adresa musí být koncipována tak, aby identifikovala cíl nějakým rozumně použitelným způsobem
- například každá síťová karta má MAC adresu (unikátní číslo, které jednoznačně identifikuje počítač), jenže neidentifikuje jej rozumně použitým způsobem
- adresa nás tedy musí nasměrovat na cíl
- v internetu slouží adresa dvěma, zcela odlišným skupinám uživatelů – lidem a strojům

adresace v internetu – adresy ip

- **IP adresa se skládá ze dvou částí:**
  - adresy sítě
  - adresy rozhraní (počítače) v této síti

- **v současnosti se aktivně užívají dva typy IP adres:**
  - **ip protokol verze 4 (IPv4)** - adresy dlouhé 32 bitů, které se většinou píší jako 4 oktety v desítkové, tečkami oddělené notaci (192.168.32.55)
  - **ip protokol verze 6 (IPv6)** - adresy dlouhé 128 bitů, které se prezentují jako hexadecimální řetězec (1080:0:0:0:8:800:200C:417A)

- **oba typy adres se přidělují stejným způsobem – pomocí delegátů:**
  - autonomní systém je vydělená (regionálně/kontinentálně) část Internetu, která má svého správce IP adres

  - koncový uživatel dostane IP adresu přidělenou poskytovatelem služby (ISP, Internet Service Provider)
  - tomu adresy, které bude poskytovat, přidělí buď místní registrátor (LIR, Local Internet Registry) nebo národní registrátor (NIR)
  - pro určitou geografickou část světa je potom určen regionální registrátor (RIR)
- **existuje pět regionálních registrátorů:**
  - AFNIC (African Network Information Centre)
  - APNIC (Asia Pacifik NIC)
  - ARIN (American Registry for Internet Numbers)
  - LACNIC (Latin American and Carribean NIC)
  - RIPE NCC (Réseaux IP Européens)

- nejvyšším orgánem je IANA (Internet Assignment Numbers Authority)
- přiděluje adresy podle potřeb RIR z volné rezervy
- komu byla určitá IP adresa přidělena je možnost zjistit na webových stránkách regionálních registrátorů (RIR.net)
- rozdělení Internetu na autonomní systémy a přidělení každému interval adres je základní podmínka směrování datagramů
- interval adres je potom možné agregovat na jednu adresu supersítě

- **adresy mohou být:**
  - individuální (unicast)
  - skupinové (multicast)
  - všeobecné (anycast)

- **kolik bitů z adresy tvoří adresu sítě určuje třída adresy, je definováno 5 tříd:**

třída a

- adresa sítě je určena prvním bajtem IP adresy:
- nejvyšší bit je 0, zbylých 7 bitu je proměnných, určují adresu sítě
- tzn., může nabyt hodnot od 1 (0000 0001) do 127 (0111 1111), vyjma 10, protože se používá pro privátní sítě

třída b

- adresa sítě je určena prvními dvěma bajty IP adresy:
- dva nejvyšší bity prvního bajtu má 10, dalších 6 v prvním a 8 v druhem bajtu je proměnných a určují adresu sítě
- tzn. může nabyt hodnot od 128.0 (1000 0000 . 0000 0000) do 191.255 (1011 1111 . 1111 1111), vyjma rozsahu 172.16 až 172.31, protože se používá pro privátní sítě

třída c

- adresa sítě je určena prvními třemi bajty IP adresy:
- tři nejvyšší bity prvního bajtu má 110, dalších 5 v prvním a 8 ve druhém a třetím bajtu je proměnných a určují adresu sítě
- tzn. může nabyt hodnot od 192.0.0 (1100 0000 . 0000 0000 . 0000 0000) do 223.255.255 (1101 1111 . 1111 1111 . 1111 1111), vyjma rozsahu 192.168.0 až192.168.255, protože se používá pro privátní sítě

třída d

- má 1110 v prvém bajtu a zbytek se dále nedělí

třída e

- rezerva

specifikace ip adresy:

- adresy tvořené samými jedničkami nebo nulami mají speciální význam, běžné se nepoužívají

- je-li adresa tvořena samými nulami, znamená to tento počítač
- je-li adresa tvořena samými jedničkami, znamená to všichni (všeobecný oběžník, broadcast)

- každý systém (počítač) má adresu programové smyčky (127.0.0.1), která se na internetu tudíž nepoužívá

- adresy třídy D slouží pro skupinovou adresaci

síťová maska

- je to čtyřbajtové číslo (v bitech určující adresu sítě má samé jedničky a v ostatních bitech samé nuly)
- slouží k získání adresy sítě, ve které je stanice o dané IP adrese
- určuje, které bity v IP adrese tvoří adresu sítě

- jednotlivé třídy sítí používají jako adresu sítě různě dlouhou část IP adresy
- třída A má pro adresu sítě vyhrazen první bajt ⇒standardní síťová maska pro adresy třídy A má v prvním bajtu samé jedničky a ve bylých třech bajtech samé nuly (255.0.0.0.

- toto jsou standardní síťové masky (jsou vždy vyrovnány na hranici oktetu)
- adresu sítě, na které leží počítač o IP adrese 170.85.255.24. určíme tak, že se nejprve podíváme do tabulky tříd a zjistíme, že maska je třídy B ⇒užijeme standardní masku a provedeme logickou operaci: adresa AND maska

- výše uvedené je obecný logický postup, jak získat z IP adresy adresu sítě
- maska však nemusí být vyrovnána na hranici oktetu a směrování nemusí být založena na třídách
- realizace beztřídního směrování je ovšem poněkud složitější - adresa musí být zadána ve tvaru například 192.168.0.0/21 (maska je 21 souvislých jedniček)

ÚČEL A PRINCIPY SMĚROVÁNÍ, ZÁKLADNÍ SMĚROVACÍ PROTOKOLY

- IP adresy představují jednotný abstraktní mechanizmus umožňující protokolu mezisíťové vrstvy (internet) nepracovat s fyzickými adresami (abstrahovat od fyzických adres)
- IP adresa (verze 4) je 32bitové číslo, které jednoznačně identifikuje uzel v lineárním adresovém prostoru sítě

- **logicky je IP adresa dvousložková, tvoří ji:**
  - adresa sítě
  - adresa uzlu v síti

- neboli, internet je členěn na jednotlivé lokální sítě (což ostatně říká i sám název Inter - Net), která spolu určitým způsobem komunikuje
- základní důvod proč je internet strukturovaná síť, je prostý – příčinou je právě směrování

- účelem směrování je předávání datagramů mezi lokálními sítěmi do místa určení
- směrování lze tedy definovat jako rozhodování, kudy dále poslat datový paket

- **v zásadě máme dva základní mechanizmy:**
  - **přímé směrování**
    - direct routing
    - kdy se oba počítače (zdrojový a cílový) nacházejí ve stejné lokální síti
  - **nepřímé směrování**
    - indirect routing
    - kdy se nacházejí v obecně různých sítích

- dále mám opět dvě možnosti, buď si uzel vyřeší problém směrování sám, nebo o to požádá jiný subjekt
- historicky převládlo řešení, kdy přímé směrování provádí uzel sám (je to relativně jednoduché) a nepřímé směrování řeší směrovač, což je specializovaný uzel sítě (neběží na něm žádné jiné aplikace)

příklad 1 - přímé směrování

- uzel A chce odesílat paket uzlu B
- rozloží proto IP adresu cíle pomocí masky a tak získá adresu cílové sítě
- následně zjistí, že je stejná jako jeho vlastní – jedná se přímé směrování – může tedy poslat paket bezprostředně adresátovi

příklad 2 – nepřímé směrování

- uzel A chce poslat paket uzlu B
- rozloží IP adresu pomocí masky a tak získá adresu cílové sítě
- porovnáním zjistí, že je jiná než jeho vlastní – cíl leží v jiné lokální síti
- pro takovou situaci zná uzel adresu směrovače jeho lokální sítě (brány)
- pošle proto inkriminovaný paket svému směrovači
- tím jeho úloha končí, o ostatní se stará směrovač
- IP adresu, masku a adresu implicitního směrovače získá uzel při inicializaci operačního systému většinou pomocí DHCP

- každá lokální síť, která je připojena do Internetu, musí mít alespoň jeden směrovač (gate, bránu)
- otázkou je co se stanu, kdy je směrovač v lokální síti víc než jeden
- každý uzel má jednu a právě jednu adresu implicitního směrovače

- pokud trasa přes implicitní směrovač není optimální, pošle (ten implicitní) směrovač uzlu zprávu (datagram ICMP), že je pro tuto adresu cílové sítě výhodnější použít směrovač Sx (tím definuje uzlu explicitní směrovač)

- doručení paketu závisí obecně na neznámém počtu směrovačů
- každý směrovač je připojen nejméně do dvou sítí
- směrovače mají principálně stejné možnosti doručení jako uzly
- pokud je adresát v síti, která je k směrovači připojena bezprostředně, je paket doručen přímo
- v opačném případě bude použito nepřímé směrování, použije se implicitní cesta nebo má směrovač ve své tabulce odpovídající záznam
- směrovač je zařízení, které je schopné učit se a poučovat jiné
- zjistí-li směrovač, že zvolená cesta není optimální, uvědomí o tom odesílatele (obvykle směrovač)
- ten následně upraví svou směrovací tabulku
- základní myšlenka principu implicitních cest je relativně prostá – zajistit úspěšné doručení paketu i při částečné znalosti nejvýhodnější cesty
- směrovač tedy zná jen určité cesty, nezná celý internet – není to žádoucí a není to ani možné (síť je dynamická struktura)
- pro směry, které nezná, použije předurčenou cestu

- když tato cesta není optimální (a dozví se to) upraví svou směrovací tabulku
- negativním důsledkem je určitá neefektivita v doručování
- hlavní a zásadní výhodou je redukce objemů směrovacích tabulek a následně snížená režie potřebná k udržení tabulek v aktuálním a konzistentním stavu
- směrovací tabulka je soubor záznamů
- každý záznam je relace: adresa sítě – adresa odpovídajícího směrovače
- není to adresa komerčního směrovače, je to jen adresa následujícího přestupního bodu (next hop – další skok)

Historické souvislosti

- internet vznikl zbytměním Arpanetu
- právě ten v počítačích představoval jakousi páteřní síť (Backbone), na kterou se ostatní lokální sítě připojovaly (mohly to být i konglomeráty lokální sítí)
- každá z lokálních sítí se k Arpanetu připojovala pomocí jedinečného směrovače, který plnil funkci směrovače implicitního
- směrovače, kterými se dílčí sítě připojovaly k páteřní síti, se nazývaly hlavní brány (Core Gate)
- hlavní brány nepoužívaly implicitní směrování, ale skutečně znaly celý Internet (důvodem byla efektivita)
- což mohlo spolehlivě fungovat, protože to spravovala jediná instituce (INOC, Internet Network Operations Center)
- explozivní růst Internetu si však vynutil změnu
- struktura sítě, daná jedinou páteřní sítí se stala značně složitou
- mechanizmus udržování směrovacích tabulek, hlavních bran se stal nákladnou a komplikovanou záležitostí
- zásadním problémem bylo, že některé dílčí sítě prostě nebylo možné připojit přímo na páteřní síť
- dalším důvodem pro změnu bylo právě připojování

máme dvě hlavní skupiny směrovacích protokolů:

egp

- jsou vhodné pro výměnu směrovacích dat mezi systémy

bgp

- Border Gateway Protocol, dnes verze 4
- pracuje s pevně stanovenými pravidly

igp

- jsou určeny pro směrování v rámci autonomních systémů či oblastí

rip

- Routing Information Protocol
- je jedním z nejstarších protokolů
- **je charakterizován:** používá oběžníky (všesměrové vysílání), pracuje s vektorem vzdáleností a hodí a hodí se spíše pro menší sítě
- dnes existuje ve dvou verzích

ospf

- Open Shortest Path First
- je vhodný pro střední či větší sítě
- je založen na algoritmu pracujícím s kvalitou přenosové cesty (LSA) a při rozhodování bere do úvahy:
    - šířku přenosového pásma
    - zátěž
    - spolehlivost
    - transportní zpoždění
    - velikost MTU
- **je charakterizován:** používá skupinové vysílání (224.0.0.x), spouštěné aktualizace, síťové masky proměnné délky a je schopen podporovat směrování s normovanou kvalitou služby (Qos)
- nezanedbatelnou předností je podpora ověřování – směrovače mohou výměnu dat chránit heslem
- data se vyměňují mezi autorizovanými směrovači

eigrp

- Enhanced Interior Gateway Protocol
- je příkladem hybridního protokolu
- vyvinula jej firma Cisco
- je charakterizován jako vyvážený protokol, který kombinuje výhody obou algoritmů směrování (VDA i LSA).

RIP

- je patrně jeden z nejstarších a nejrozšířenějších směrovacích protokolů
- dnes existuje ve dvou verzích: verze 1 (RFC 1058, rok 1988) a verze 2 (RFC 2453, rok 1998)
- původní návrh představuje implementaci směrovacího algoritmu Bellman-Ford (také Ford-Fulkerson) a realizovala jej firma Xerox (prapůvod je patrně routed z univerzity v Berkeley).
- jmenovaní tři pánové položili teoretické základy algoritmů směrování s vektorem vzdálenosti (kolem roku 1962, Princeton Univerzity).

výhody:

- největší výhodou protokolu RIP je jeho jednoduchost nastavení a uvedení do provozu
- směrovače RIP si udržují své lokální databáze tras, které se označují jako směrovací tabulky. Ve výchozím stavu obsahuje směrovací tabulka pouze sítě, ke kterým je daný směrovač fyzicky připojen. Všechny směrovače RIP pak v pravidelných intervalech odesílají zprávy s obsahem svých směrovacích tabulek. Tak informují ostatní směrovače RIP o dostupných sítích.

nevýhody:

- nevýhodou protokolu RIP je dlouhá doba zotavení. Když dojde ke změně v topologií propojených sítí, může proces automatické úpravy konfigurace směrovačů RIP trvat i několik minut. Navíc, v rámci automatických úprav mohou vznikat uzavřené směrovací smyčky, jejichž existence může způsobit ztrátu nebo nedoručitelnost dat. Nejlepšího výkonu tedy dosahují RIP směrovače v meších sítí.
- největší nevýhodou je neschopnost pracovat v rozsáhlých a velmi rozsáhlých síťových strukturách
- nejvyšší počet úrovní směrování (přeskoků) používaný směrovači RIP je 15. Sítě, které jsou vzdálené 16 a více přeskoků, jsou považovány za nedostupné

- se zvětšováním síťově struktury může pravidelná výměna dat o trasách mezi směrovači RIP výrazně zatížit komunikační cesty

- protokol RIP verze 1 používá k odesílání zpráv pouze oběžníky (pakety všesměrového vysílání IP)
- protokol RIP verze 2 používá jak oběžníky, tak pakety skupinového vysílání (na IP adresu 224.0.0.0)
- všechna zařízení, která pracují s protokolem RIP, naslouchají na portu 520 UDP a podle došlých dat aktualizují svoje směrovací tabulky
- v každé aktualizaci může být nejvýše 25 položek
- pokud je směrovací tabulka rozsáhlejší je nutné provést několik aktualizací (odeslat tabulku po částech)
- směrovací data se odesílají každých 30 vteřin, bez ohledu na to, došlo-li k nějaké změně
- takže oběžníky, interval 30 vteřin, opakované aktualizace a bezdůvodné aktualizace (bez ohledu na existenci změn) představují značnou transportní zátěž sítě. Což je nepříjemné

RIP 1 - formát datagramu

- protokol používá několik druhů zpráv pro výměnu směrovacích dat
- typ zprávy (také příkaz) určuje prvý oktet datagramu

- **existuje zhruba osm typů zpráv, významné jsou dnes jen dva:**
    1. žádost odeslaná při inicializaci směrovače
        - všechny sousední směrovače jsou žádány o odeslání svých směrovacích tabulek
    2. odpověď na žádost nebo pravidelná aktualizace

    - ostatní příkazy buď zastaraly (zapnutí/vypnutí trasování) nebo se používají zřídka (žádost/odpověď pro okruhy na vyžádání)

- **všechny zprávy používají stejný formát datagramu, ten se skládá z hlavičky pevné délky a směrovacích dat:**
    - **hlavička**
        - má délku 8 oktetů:
            - typ zprávy (1)
            - verze protokolu (1)
                - definuje verzi RIP protokolu, podle kterého byl datagram vygenerován
                - v síti mohou pracovat směrovače, které pracují s různými verzemi RIP
            - prázdné pole (2)
            - identifikátor rodiny adresy AFI (2)
                - má prakticky vždy hodnotu 2, tedy IP protokol (teoreticky se může jednat i o jiný protokol a jiný typ adresy)
            - prázdné pole (2)
    - **pole směrovacích dat:**
        - IP adresa cílové sítě (4)
        - prázdné pole (2x4 oktety)
        - metrika (4)

    - jedna položka směrovací tabulky se tedy přenáší jako 16 oktetů
    - protokol není právě úsporný – pole s nulovou hodnotou jsou pozůstatky minulosti (pole, které ztratily význam a musely zůstat zachovány z důvodu dílčí kompatibility).
    - v jednom datagramu může být maximálně 25 položek směrovací tabulky
    - význam IP adresy prvé položky je různý v závislosti na typu zprávy
    - je-li to žádost, pak je to IP adresa odesílatele datagramu
    - v datagramu odpovědi je to jedna z IP adres ze směrovací tabulky odesílatele

RIP 2 - formát datagramu

- z důvodu zpětné kompatibility se změnil jen málo
- od prázdného pole (rozumí se prázdné ve verzi 1) se za IP adresou zapisuje síťová maska (4) a IP adresa počátku (prvý hop) cesty (4)
- pole AFI má nyní dvojí význam

- pokud je v něm zapsána hodnota FFFFh pak prvá položka nejsou směrovací, ale autorizační data odesílatele datagramu (celkem 16 oktetů)

- typ prováděné autorizace upřesňuje pole za AFI (dříve prázdné)
  - zatím je definován jediný způsob prosté nezašifrované heslo o délce 16 bajtů (indikuje hodnota 2 zapsána v poli typu autorizace)
  - toto pole má nyní také dvojí význam – u směrovacích dat slouží k zápisu značky cesty (route tag)
  - značka je součástí mechanizmu pro rozlišení interní a externí cesty
  - interní cestu zjistil RIP-2 v rámci své oblasti, u externí cesty se dozvěděl od jiných směrovacích protokolů (jako třeba BGP)

- **topologie sítě se v čase dynamicky mění**
  - pro směrovače to prakticky znamená, že některý sousední směrovač přestane pracovat
  - neposílá pravidelné aktualizace účinkem čehož je cesta prohlášena za neplatnou
  - změna množiny sousedních uzlů následně vede k přepočtu metriky některých cest
  - v důsledku postupného šíření směrovacích dat je systém v nekonzistentním stavu
  - potřebuje určitý čas, aby se všechny směrovače v dílčí síti shodly na nové topologii
  - proces dosažení jednotného pohledu na topologii sítě se nazývá konvergence
  - během konvergence dochází k problémům
  - příčin je několik
  - pokud k cíli vede několik tras, může vzniknout směrovací smyčka – na cestě k cíli se datagram znovu dostane k uzlu, kterým již prošel (mrtvá cesta)
  - dále, i když směrovač zjistí, že určitá cesta je neplatná, nemůže o tom uvědomit ostatní ihned, musí čekat na vypršení intervalu časovače aktualizací
  - de facto, směrovač zatajuje skutečný stav sítě, číž situaci zhoršuje

- **řešení problému směrovacích smyček je vícero**
  - **pravidlo rozdělení horizontu**
    - split horizont
    - určuje, že směrovač nesmí informovat svého souseda o cestách, které vedou přes něj samotný
    - důsledek je ten, že se neodesílají celé směrovací tabulky, ale jen položky o cestách, které přes přijímací směrovač nevedou
    - **příklad:** směrovač A neodešle směrovači tu položku své směrovací tabulky, která má jako první hop uvedenou adresu směrovače B
    - toto pravidlo má zabránit vzniku smyček mezi sousedními směrovači

  - **pravidlo otrávení zpětných dat**
    - poison reverse
    - má zabránit vzniku rozsáhlých smyček
    - příznakem rozsáhlé smyčky je postupné navyšování metriky cesty
    - tento předpis umožňuje směrovači porušit pravidlo pro rozdělení horizontu
    - u předávaných položek (ty, které pravidlo rozdělení horizontu porušují) je však metrika nastavena na 16
    - pokud příjemce ve své tabulce má stejnou cestu s lepší metrikou, bude položku ignorovat

  - **spouštěné aktualizace (trigger update)**
    - když směrovač zjistí změny v topologii sítě, nečeká na pravidelnou aktualizaci a ihned odešle dílčí data (co se změnilo)

  - **zadržovací časovač**
    - když je cesta označena jako nedostupná trvá ještě 90 vteřin než je z tabulky vymazána (časovač garbage – collector)
    - v tomto časovém úseku směrovač nereaguje na změny v dané cestě, akceptovatelná je pouze zpráva, že se cesta vrátila do původního stavu

- uvedený postup má zabránit v šíření nepravdivých směrovacích dat během procesu konvergence sítě na novou topologii

RIP 2

- jak bylo uvedeno, RIP vznikl v ranných počátcích Internetu
- je to jeden z nejstarších a nejúspěšnějších protokolů
- tehdy bylo směrování založeno na IP adresách rozdělených do tříd
- síťové masky měla standardizovanou délku, kterou definovala třída IP adresy
- postupem času se přešlo k beztřídnímu směrování (CIDR, Classless Inter Domain Routing a VLSM, Variable Length Subnet Mask) a zákonitě musela vzniknout i nová verze protokolu RIP
- následující skutečnosti jsou základními nedostatky protokolu RIP verze 1:
    - velmi omezená podpora pro tvorbu podsítí
    - žádná autentizace vysílajícího uzlu
    - metrika omezená na 15 přeskoků
    - všesměrné vysílání zpráv

- RIP verze 2 je definován v dokumentu RFC 1723
- vylepšení (proti verzi 1) bylo dosaženo dodáním nových funkcí při zachování zpětné kompatibility
- přídavné funkce jednak využívají pole s nulovým obsahem (MBZ, must be zero) původního paketu nebo určitým způsobem (drobně) modifikují pole původní
- všechny zásadní nedostatky s výjimkou omezení počtu přeskoků byly odstraněny
- zrušení omezení metriky na 15 přeskoků by vedlo ke ztrátě kompatibility s verzí 1, proto bohužel muselo zůstat zachováno
- provedená vylepšení výrazně zdokonalily tento směrovací protokol při zachování všech jeho kladných vlastností – snadné implementaci, ovladatelnosti a použitelnosti
- dále byly také přidány rozšíření, které sice pro provozní funkčnost protokolů nemají zásadní význam, ale dále zlepšují jeho vlastnosti, např.:značkování externích cest a informační bloku

- **autentizace**
    - důvěryhodnost průvodce směrovacích dat má zásadní bezpečnostní význam
    - šíření nepravdivých dat z nepravých zdrojů způsobí minimálně poškození směrovacích tabulek a v konečném důsledku to může vést až ke kolapsu síťového provozu
    - mechanizmus autentizace pracuje takto:
        1. struktura hlavička je stejná
        2. pouze prvá položka směrovacích dat je použita pro přenos ověřovacích údajů (maximální počet směrovacích položek ve zprávě se zmenší na 24)
        3. pole AFI má hodnotu FFFFH
        4. pole se značkou cesty se změní na pole vyjadřující typ autentizace

    - dokument RFC 1723 definuje pouze jeden způsob autentizace původní zprávy – prosté, textové (nezašifrované) heslo ao délce nejvýše šestnáct znaků (v poli typ autentizace je 2)
    - obecně právě to, že protokol nešifruje žádnou část datové zátěže, je jeden z největších nedostatků protokolu RIP 2
    - protokol tudíž není příliš robustní
    - lze pouze zabránit podvržení dat osobami, které jsou mimo síť

IGRP

- používá kompozitní metriku vypočítanou z několika položek, jako jsou zpoždění sítě (delay), šířka pásma (bandwidth), spolehlivost (reliability) a zatížení (load). Správce sítě může nastavovat závažnost faktorů pro každou z těchto metrik, ačkoli zásah do těchto nastavení se musí dělat s velkou opatrností. IGRP poskytuje širokou škálu nastavení metriky. Spolehlivost a vytížení může například dosahovat jakékoliv hodnoty mezi 1 a 255; šířka pásma dosahuje hodnot odrážejících rychlosti od 1200 bps až do 10 Gbps, přičemž zpoždění dosahuje hodnot mezi 1 a 224. Tohle velké rozmezí metriky jsou dále doplněny sérií uživatelsky definovanými konstantami, které umožňují správci sítě ovlivňovat výběr směru cesty. Tyto nastavení jsou hashovány pro případ

metriky, a jiných algoritmů, které poskytuje jednoduchá kompozitní metrika. Správce sítě může ovlivnit výběr cesty nastavením vyšší či nižší váhy specifické metriky. Tato přizpůsobenost umožňuje správci doladit IGRP automatickou volbu cesty.

- Poskytovaním dodatečné přizpůsobivosti, dovoluje IGRP vícecestné směrování. Na zdvojené šířce pásma může proudit jednoduchý tok provozu způsobem round-robin, s automatickým pčepínáním, pokud jedna linka vypadne. Vícenásobné cesty můžou mít nerovnoměrné metriky, přesto jsou stále platné. Např. pokud je jedna cesta 3x lepší než druhá (její metrika je 3x nižší), bude tato cesta 3x častěji využita. Jen cesty s metrikou uvnitř určitého rozsahu nebo rozdílných nejlepších cest můžou být použity jako mnohonásobné cesty. Rozdílnost je další významný faktor, který může být zajištěn správcem sítě.
-

Stabilita protokolu

- IGRP zajišťuje několik vlastností, která jsou navrženy pro zvýšení stability. To zahrnuje holddown, split horizon a poison-reverse updaty.

- Holddowny předcházejí obnovám vadných směrovacích cest v pravidelných updatech. Pokud směrovač "spadne", sousedícím směrovačům chybějí pravidelné updaty. Tyto směrovače pak vypočítají novou cestu a informují "sousedy" o změnách v síti. O nových updatech není okamžitě informován každý síťový prvek, tak je možné pro zařízení, které bylo právě informováno o poruše, poslat klasickou update message, která propaguje problematickou cestu jako platnou, tomu zařízení, které bylo právě informováno o změnách. V tomto případě může druhé zařízení obsahovat nekorektní směrovací informaci. Holddowny zajišťují směrovači udržení jakékoliv změny, která může ovlivnit směrování po jistý čas. Holddown perioda je vypočítavána tak aby byla o něco větší než je doba pro aktualizaci změn v směrování celé sítě.

- Split horizon pravidlo je odvozeno z předpokladu, že není užitečné posílat zpět informace o cestách, směrem odkud přišly.

ORGANIZACE INTERNETU

adresace v internetu – adresy ip

- **IP adresa se skládá ze dvou částí:**
  - adresy sítě
  - adresy rozhraní (počítače) v této síti

- **v současnosti se aktivně užívají dva typy IP adres:**
  - **ip protokol verze 4 (IPv4)** - adresy dlouhé 32 bitů, které se většinou píší jako 4 oktety v desítkové, tečkami oddělené notaci (192.168.32.55)
  - **ip protokol verze 6 (IPv6)** - adresy dlouhé 128 bitů, které se prezentují jako hexadecimální řetězec (1080:0:0:0:8:800:200C:417A)

- **oba typy adres se přidělují stejným způsobem – pomocí delegátů:**
  - autonomní systém je vydělená (regionálně/kontinentálně) část Internetu, která má svého správce IP adres
  - koncový uživatel dostane IP adresu přidělenou poskytovatelem služby (ISP, Internet Service Provider)
  - tomu adresy, které bude poskytovat, přidělí buď místní registrátor (LIR, Local Internet Registry) nebo národní registrátor (NIR)
  - pro určitou geografickou část světa je potom určen regionální registrátor (RIR)
- **existuje pět regionálních registrátorů:**
  - AFNIC (African Network Information Centre)
  - APNIC (Asia Pacifik NIC)
  - ARIN (American Registry for Internet Numbers)
  - LACNIC (Latin American and Carribean NIC)
  - RIPE NCC (Réseaux IP Européens)

- nejvyšším orgánem je IANA (Internet Assignment Numbers Authority)
- přiděluje adresy podle potřeb RIR z volné rezervy
- komu byla určitá IP adresa přidělena je možnost zjistit na webových stránkách regionálních registrátorů (RIR.net)
- rozdělení Internetu na autonomní systémy a přidělení každému interval adres je základní podmínka směrování datagramů
- interval adres je potom možné agregovat na jednu adresu supersítě

IANA

- Internet Assigned Numbers Authority. Organizace formálně odpovědná za systém doménových jmen Internetu a za používání a přidělování nejrůznějších technických parametrů adres a jiných číselných identifikátorů, které musí mít stejný význam v rámci celého Internetu nebo musí být v celém Internetu unikátní.
-

IAB

- Internet Architecture Board. Technický orgán zabývající se celkovou architekturou Internetu. Po vzniku ISOC se stal jeho součástí, předtím byl fakticky podřízen vládním orgánům USA pod názvem Internet Activities Board. Fakticky se zabývá vývojovými trendy v Internetu a zastřešuje celý proces tvorby standardů, je také formálním vydavatelem dokumentů RFC. Má dvě větve: IRTF (Internet Research Task Force) a IETF (Internet Engineering Task Force).

IETF

- Internet Engineering Task Force. Standardizační organizace (podřízená orgánu IAB) tvořená volným sdružením odborníků, sestávající z více jak 80 pracovních skupin, ve kterých vzniká faktická technická příprava standardů Internetu. Pro technická řešení vzniklá mimo IETF pak půda tohoto orgánu slouží k jejich schvalování.

autonomní systémy

- internet je soustavou konglomerátů dílčích sítí
- komercializace internetu přispěla ke vzniku autonomních systémů
- dnes by bylo patrně přesnější definovat internet jako soustavu dílčích sítí různých poskytovatelů připojení (Provider)
- drtivá většina poskytovatelů připojení pracuje na komerční bázi, a proto si přeje svou síť spravovat svým způsobem
- internet byl proto důsledně rozdělen na autonomní systémy (i samostatná páteřní síť se soustavou hlavních bran je autonomní systém), což následně vedlo k zjednodušení jeho struktury
- každý autonomní systém je označen zkratkou AS a dvoubajtovým číslem
- jeden poskytovatel může mít i několik autonomních systémů
- poskytovatelé se zabývají dopravou paketů v rámci své sítě, mezi sebou i jako tranzitní přepravci (propojovací sítě)
- každá autonomní oblast má svou správu, která žádá autoritu o přidělení intervalu IP adres (interval adres umožňuje agregaci dílčích sítí do jedné supersítě = jedna cílová směrovací adresa)
- internet je tedy z pohledu směrování paketů rozdělen na autonomní systémy
- pro směrování mezi AS se používá EGP nebo IGP
- takže, za každý autonomní systém plně odpovídá jeho provozovatel (není anonymní)
- dále pak existuje jednotný systém předávání směrovacích informací mezi jednotlivými autonomními systémy, který jsou povinní všichni provozovatelé dodržovat (Jinak řečeno doma si může každý postupovat podle svého, navenek musí všichni postupovat jednotně

Telnet	protokol virtuálního terminálu	RFC 854	port 23
FTP	protokol přenosu souborů	RFC 959	port 20, 1
TFTP	jednoduchý protokol přenosu souborů	RFC 1350	port 69
SMTP	jednoduchý protokol transferu pošty	RFC 821	port 25
DHCP	protokol dynamické konfigurace stanice	RFC 2131	port 546,7
DNS	protokol systému doménových jmen	RFC 1035	port 53
HTTP	protokol transferu hypertextových informací	RFC 2616	80
SNMP	jednoduchý protokol správy sítě	RFC 1152	161, 2

BESPEČNOST VÝPOČETNÍCH SYSTÉMŮ

INFORMAČNÍ BEZPEČNOST

- informace mají tržní hodnotu. Představují majetek. Lze je teda vlastnit, kupovat a prodávat.
- lze je ale také možnos krást, ničit a falzifikovat
- hodnota informace je dána především její výlučností, vím něco, co jiný neví a také přesností a vypovídající schopností

informační bezpečnost

- je definována jako souhrn ochranných opatření zajišťujících principy důvěrnosti, integrity, dostupnosti a prokazatelné zodpovědnosti při činnosti informačně-technologického systému (Information Security, také počítačová bezpečnost – Computer Security)
- informačně-technologický systém tvoří technické prostředky, programové prostředky, paměťová média, data a osoby nějakým způsobem zaineresované na procesech tohoto systému
- informační bezpečnost je zaměřená na ochranu všech prostředků informačních technologií a zpracovávaných, uchovávaných a distribuovaných dat
- v zásadě se jedná o ochranu informací během jejich vzniku, zpracování, ukládání, přenosů a likvidace
- zahrnuje bezpečnost administrativní, komunikací, počítačovou, personální a fyzickou
- je to komplex logických, technických, fyzických a organizačních opatření, která zabraňují ztrátě důvěrností, integrity a dostupností obhospodařovaných dat
- cílem bezpečnostích opatření je zabránit kompromitaci nebo nedovolené modifikaci dat či destrukci systému nebo jeho částí. Což by následně umožnilo zneužití citlivých informací, použití klamných dat, ze kterých budou vyvozeny chybné výsledky či závěry, nesprávná interpretace hodnot.
- všechny tyto záporné skutečnosti označujeme souborným názvem hrozby

- **výše uvedené v zásadě říká, že musíme informačnísystém chránit tak, aby nedošlo ke ztrátě, ta může mít různé podoby:**
    - přímá finanční ztráta nebo fyzické poškození
    - ztráta dostupnosti, kdy informace je nedostupná všem nebo někomu, kdo k ní má mít dostup
    - ztráta důvěrnosti, kdy je informace dostupná i těm, kteří k ní neměli mít dostup
    - ztráta integrity, kdy data byly modifikovány a informace je v důsledku toho zničena nebo změněná, úmyslně či neúmyslně
    - ztráta autentičnosti, kdy není možné určit původce či zdroj informace

- data jako taková, představují největší hodnotu informačního systému
- existují tři základní důvody pro cennost dat:
    - data jsou utajována
    - existence dat je utajována
    - důvod utajování dat je utajován

- počítačová kriminalita je jakákoli trestná činnost páchána pomocí výpočetní techniky nebo ohrožující informační systémy či data v nich uložená (Computer Crime – počítačový zločin obecně)

historické souvislosti

- bezpečnostní informační systém je realitvně mladý obor
- v říjnu roku 1967 byli pod vedením Defense Science Board osloveni všichni výrobci pracující na ochraně počítačoých systémů, kteří by se chtěli podílet na vývoji standardů pro klasifikované informace
- zpráva publikovaná v roce 1970, stanovila politiku a technická doporučení
- později v roce 1972 vydalo DoD direktivu 5200.28, jenž definovala jednotnou politiku ochrany informací zpracovávaných počítačovými systémy tohoto ministerstva USA
- následně se do práce vložil NBS (National Bureau Standards) a MITRE Corporation
- výsledkem snažení je dokument, dnes známý jako oranžová kniha – přesná specifikace je CSC-STD-001-83 „Trusted Computer Systém Evaluation Criteria (kriteria zhodnocení zabezpečených počítačových systémů)
- v roce 1981 vzniká při DoD National Compute Security Center (NCSC)
- patrně nejvýznamnějším počinem této instituce je vydávání tak zvané „duhové série“, posloupnosti knih a dokumentů, které představují základní specifikace pro zabezpečené systémy. Knih je víc než 25
- s nástupem internetu se situace dále komplikuje
- v roce 1994 šestnáctiletý student Londýnské hudební konzervatoře R. Pryce, pronikl do sítě maerického letectva a firmy Lockheed. Použilo k tomu počítač za 750 liber. Tajná služba zaznamenala zhruba 200 neoprávněných přístupů. CIA byla přesvědčena, že je to práce vyspělé východoevropské špionážní skupiny. Chycen byl náhdou, když si vyměňoval na webu s ostatními hackery, přístupová telefonní čísla do vládních sítí. Soud mu uložil pokutu 1200 liber – Pryce se hájil tím, že šlo jen o zábavu. Prý se připravoval na zkoušku z informačních technologií. Tento případ je zvláštní tím, že se jednalo o vysoce „zajištěné“ systémy.
- dne 2. listopadu 1988 byl Internet napaden počítačovým červem. Internet měl v té době zhruba 60.000 počítačů. Během zhruba 12 hodin bylo infikováno asi 6.000 počítačů. Červ napadal stroje SUN a VAX, na kterých byl nainstalován UNIX (Berkeley Standard Distribution, BSD 4.3). K šíření využíval bezpečnostních děr, především v programu sendmail a démonu fingerd. Po napadení a rozmnožení paralyzoval počítač, ale nezpůsoboval žádnou škodu. Výsledná ztráta byla později odhadnuta na 1 – 100 milionů dolarů. Pachatelem byl R.T.Morris, student na Cornell University. Trest 5 let natrvrdo, $250.00% + náhrada škody (později trest zmírněn)

základní pojmy

- jako každý jiný obor lidské činnosti má i bezpečnost informačních systémů své odborné názvosloví
- většinou vychází z anglické (americké) terminologie

bezpečnost (security)

- je vlastnost nebo stav ochrany proti nevyhnutelným ztrátám
- bezpečnost informačních systémů zahrnuje ochranu činnosti zpracování, úschovy, distribuce a prezentace informací
- je to kombinace důvěrnosti, integrity, dostupnosti a zodpovědnosti
- jinými slovy, je to ochrana dat proti neautorizovanému přístupu, modifikaci nebo destrukci

hrozba (thread)

- je akce nebo událost, která může ohrozit bezpečnosti, je to zneužití zranitelnosti
- také lze definovat jako pravděpodobnost útoku daná atraktivitou systému pro útočníka

zranitelnost (vulnebarility)

- je slabé místo v systému, které může být zneužito k narušení zamýšleného chování v systému

aktivum (asset)

- je cokoli, co má cenu
- můžeme dělit na:
  - **hmotné** - počítače, komunikační technika
  - **nehmotné** - data, programové vybavení

hodnota aktiva (asset value)

- představuje oceněnní důležitosti a významu objektu pro vlastníka
- nemusí být vyjádřitelné v penězích

citlivá data (sensitive data)

- jsou data, která vyžadují zvláštní ochranu, protože existuje určitá pravděpodobnost působení hrozeb (přesnější termín je citlivé informace)

průnik (intrusion)

- je akt neautorizovaného použití informačního systému

důvěrnost (confientiality)

- je chrakteristika informace, která znemožňuje její odhalení neautorizovaným subjektům

integrita (integrity)

- je vlastnost, která určuje, že daný objekt byl změněn pouze specifikovaným a autorizovaným způsobem
- také celistvost, neporušenost
- **příklady:** integrita dat – fakt, že daná data jsou stále platnou reprezentací určité informace. Data, která mají dostatečnou ochranu integrity, nemůžou být neautorizovaně změněna v rozsahu a sémantické obsahu a také v konzistenci jejich reprezentace
- **možné hrozby:** modifikace, přidání, smazání
- integrita systému – možnost vykonávat určené funkce neměnným způsobem, bez jakékoli skryté nebo úmyslé neautorizované manipulace se systémem

riziko (risk)

- je pravděpodobnost, zničení nebo poškození určitého aktiva konkrétní hrozbou – míra ohrožení příslušného aktiva
- také pravděpodobnost úspěšného útoku

dostupnost (availability)

- je vlastnost systému, která zabraňuje neautorizovanému zadržení zdrojů, které jsou pak autorizovaným subjektům dostupné pouze se zdržením
- také zabránění odmítnutí služby (Denial of Service)

BEZPEČNOSTNÍ POLITIKA

- definuje cíle, požadavky, principy, omezení a postupy, které určují způsoby správy, ochrany a distribuce citlivých informací
- je to soubor kritérií pro aplikace služeb bezpečnosti
- představuje základní východisko pro řízení bezpečnosti informačního systému
- vždy se vztahuje k určité organizační jednotce
- bezpečnostní politika může být například státní, podniková, oddělení
- cílem bezpečnostní politiky je minimalizace rizik
- není to pojistka mezi omezováním uživatelů a chráněným zájmem organizace
- bezpečnostní politiku by měly vypracovávat skupiny odborníků, ze všech struktur organizace
- charakter bezpečnostní politiky je dán charakterem organizace

bezpečnostní politika státu

- představuje základní strukturu ochrany informací
- **tvoří ji zákony a z nich odvozené předpisy, nařízení a směrnice, jedná se o následující normativy:**
    1. zákon č. 412/2005 Sb. O ochraně utajovaných informací a o bezpečnostní způsobilosti
    2. zákon č. 140/1961 Sb. Trestní zákon
    3. zákon 101/2000 Sb. O ochraně osobních údajů
    4. zákon 106/1999 Sb. O právu na svobodný přístup k informacím

- **státem utajovaná informace (dříve státní, hospodářské či služební tajemství) je definováno takto:**
    - utajovaná informace je informace v jakékoli podobě, zaznamenaná na jakékoli nosiči a označená v souladu s tímto zákonem, jejíž vyzrazení nebo zneužití může způsobit újmu zájmu České republiky nebo by to mohlo být pro tyto zájmy nevhodné

- nutnou podmínkou je, že tato skutečnost je uvedena v seznamu utajovaných skutečností
- obecný rámec vytváří příslušný zákon
- utajovanou informací mohou být fakta politického, vojenského či hospodářského významu
- klasifikací chráněných informací na základě požadavků vládních institucí provádí Národní bezpečnostní úřad (NBÚ), který také kompiluje příslušný seznam
- únik chráněných informací řeší trestní zákon
- zákon o ochraně utajovaných informací musí být, mimo jiné, v souladu s předpisy vyšší právní síly, zejména článkem 17 Listiny základních práv a svobod, který upravuje právo na informace a přípustné výjimky z něj
- státní normativy musí jednoznačně definovat všeobecně platné prinicipy
- musí však být natolik obecné, aby nesvazovaly ruce bezpečnostním politikám organizací
- každá organizace má vždy své specifické požadavky, na které musí být brán zřetel
- legislativní politika státu ptoom představuje právní rámec, ze kterého musí bezpečnostní politiky jednotlivých organizací vycházet

bezpečnostní politika organizace

- je zásadní dokument definující základní principy vedoucí k prosazení bezpečnostních funkcí organizace
- v dokumentu je uvedeno, jakými mechanismy logického, fyzického a organizačního zabezpečení budou bezpečnostní funkce prosazeny

- **dokument je sestavován na základě analýzy, jež se skládá z následujících bodů:**
    - **analýza aktiv**
        - vymezuje aktiva, která je nutno chránit
        - mohou to být data (jejich dostupnost, důvěrnost a integrita), technické a programové prostředky zajišťující podporu obchodním procesům, a podobně

- **analýza hrozeb**
  - definuje, proti čemu chceme aktiva chránit
  - například: vyzrazení, zničení, znepřístupnění informací, podvržení nebo modifikaci dat, zneužívání technických prostředků, narušování chodu informačního systému
- **analýza rizik**
  - vymezuje, jaké rizika hrozí informačním aktivům, jak velká tato rizika jsou, proti kterým rizikům chce organizace chránit svá aktiva a následně kolik prostředků je ochotna investovat do eliminace těchto rizik
- **výsledkem vyhodnocení těchto analýz je návrh protiopatření**

- bezpečnostní politika může mít charakter povinných nařízení měnitelných pouze velmi omezenou skupinou lidí nebo je to souhrn bezpečnostních doporučení, jejichž následné uplatnění prosazují vlastníci jednotlivých aktiv
- obecně pro vypracování bezpečnostní politiky existuje několik různých přístupů
- u nás se většinou vychází z britského standardu BS7799
- za východiska považujeme především: ČSN ISO/IEC 17799 Informační technologie – Soubor postupů pro řízení informační bezpečnosti, ČSN ISO/IEC TR 13335 Informační technologie – směrnice pro řízení informačních bezpečnostni IT, Information Security Management
- překlad a interpretace standardu BS 7799 pro české prostředí
- zpracovávaná bezpečnostní politika by měla obsahovat následující okruhy:
  - popis informačního systému, cíle bezpečnostní politiky, legislativní rámec, definice aktiv, definice hrozeb, bezpečnostní funkce (bezpečnostní opatření), zásady personální politiky, zásady organizační politiky, technicko-provozní opatření, politika zálohování, plán obnovy pro havárii, metodika řešení krizových stavů

- bezpečnostní politika odráží přístup organizace k řešení problému bezpečnostni informací
- definuje organizační a systémová opatření, nástroje a technologie, jež mají za úkol příslušná bezpečnostní opatření realizovat
- za bezpečnostní politiku je zodpovědný management organizace

popis informačního systému
- definuje účel informačního systému, jaké v něm probíhají informační toky
- jaká je návaznost na ostatní struktury organizace
- stanovuje co všechno je a není informační systém, hranice a vliv okolí

cíle bezpečnostní politiky
- vycházejí z obchodních dílů organizace, legislativy, smluv a interních požadavků
- jasné cíle umožní vytyčit strategii a rámcové postupy pro jejich dosažení

legislativní rámec
- je vymezen právními předpisy, které musí informační systém respektovat a bezpečnostími normativy, které musí splňovat, aby mohl být certifikován

definice aktiv
- aktiva organizace mají svojí hodnotu, která je ve většině případů pro organizaci z hlediska jejího fungování kritická
- v případě ztráty nebo závažného poškození některých aktiv může dojít případně i k ukončení činnosti organizace
- klasifikace a srpáva aktiv pojednává o přidělení hodnoty určitým infomacím podle síly dopadu na organizaci při ztrátě nebo nedostupnosti

definice hrozeb
- uvádí, proti jakým hrozbám hodláme aktiva chránit
- to je výchozím bodem pro ohodnocení rizik

bezpečnostní funkce
- představují postupy vedoucí k minimalizaci rizik

- formulace bezpečnostní funkce probíhá v posloupnosti určení hodnoty aktiva ⇒odpovídající riziko ⇒ bezpečnostní funkce, která eliminuje nebo omezí riziko na požadovanou míru
- formulací bezpečnostní funkce lze provést výběrem z katalogu nebo jako výstup z automatizovaného expertního systému

personální bezpečnost
- obecně omezuje rizika od chyb lidí
- má dvě roviny:
  - jednak řeší problematiku přístupu osob k chráněným aktivům, prověřování osob, definování stupně důvěry
  - jednak definuje požadavky na kvalifikaci

zásady organizační politiky
- zahrnují organizační opatření – tedy, kdo je zodpovědný za co
- koordinuje jednotlivé bezpečnostní složky organizace – informační, majetkové a osobní

technicko-provozní zabezpečení
- řeší finační a materiální požadavky potřebené pro zavedení požadovaných bezpečnostních opatření

plán obnovy po havárii
- je definovaná posloupnost úkonů, které umožní obnovu činnosti informačního systému po havárii

metodika řešení klíčových stavů
- slouží k řešení bezpečnostních incidentů

britský standard 7799
- je nejznámější, celosvětový rozšířený bezpečnostní standard
- vznikl kolem roku 1995 a byl několikrát revidován
- v prosinci roku 2000 byl přijat jako mezinárodní norma **ISO/IEC 17799**
- představuje standard komplexní ochrany informací
- pokrývá všechny oblasti bezpečnosti, popisuje rozhsáhlý počet bezpečnostních opatření a je použitelná pro různé typy organizací
- skládá se ze tří částí:
  - **BS7799-1:1999**
    - Code of Practice for Information Security Management
    - katalog bezpečnostních funkcí a bezpečnostních opatření
    - **také ISO/IEC 17799:2000**
      - představuje podrobné shrnutí praktických zkušeností s řešením informační bezpečnosti
      - definuje 120 bezpečnostních funkcí rozložených do 10 zón
  - **BS7799-2:1999**
    - Specification for Information Security Management Systems (označuje se zkratkou ISMS)
    - představuje návod k výstavbě systému řízení bezpečnostních informací
    - v podstatě říká, jak aplikovat prvoučást normy
  - **BS7799-2:2002**
    - je inovace druhé části
    - doporučuje, jak používat prvou část normy a jak provozova, udržovat a vylepšovat existující ISMS

české normy
- vznikají obvykle adopcí norem mezinárodních
- v říjnu 2006 byla zrušena norma ČSN BS 7799-2 a byla nahrazena nově vydanou normou ČSN ISO/IEC 27001

- Informační technologie – Bezpečnostní techniky – Systémy managementu bezpečnosti informací – Požadavky
- norma je propojena a harmonizována s normami ISO/IEC 9001:2000 (kvalita) a ISO/IEC 14001:2004 (životní prostředí) tak, aby bylo podpořeno konzistentní zavedení a provoz

- **hlavní části normy:**
  - definuje požadavky na vybudování, zavedení, provoz, monitorování, přezkoumávání, udržování, zlepšování a případnou certifikaci systému managementu bezpečnosti informací
  - dále jsou zde specifikovány požadavky na výběr a zavedení bezpečnostních opatření chránících informační aktiva
  - příloha A uvádí cíle a opatření, která jsou přímo propojena s cíly a opatřeními uvedenými v ISO/IEC 17799:2005
  - v příloze B je uvádeň vztah mezi principy OECD pro bezpečnosti informačních systémů a sítí a fázemi PDCA cyklu (bude vysvětleno dále)
  - příloha C uvádí veztah mezi ISO/IEC 9001:2000, ISO/IEC 14001:2004 a ISO/IEC 27001:2005
- je plánováno, že v budoucnu bude ISO/IEC 27xxx (ze které naše norma vychází) obsahovat sedm dokumentů:
  - 27000, principy a slovník
  - 27001, požadavky na ISMS (ISO ekvivalent BS 7799-2:2004)
  - 27002, návody pro zavádění
  - 27003, analýza rizik
  - 27004, metrikya měření
  - 27005, řízení rizik
  - 27006, kontinuita podnikání a obnova po havárii

BEZPEČNOSTNÍ DEKOMPOZICE INFORMAČNÍHO SYSTÉMU

- postup vypracování takového systému je hruba následující a vychází z druhé části standardu BS7799
- základem je definování cílů vedením organizace, to je do ISMS zapojeno dvěma vazbami:

základní vazba

- základní vazbu představuje neoddělitelnost informace od řízení a firemních procesů jako takových
- informace představují v tomto případě zdroje, stejně jako peníze nebo pracovní síla
- nezajištěnost vlastních zdrojů ohrožuje produkci a tím vede k růstu rizik

marketingová souvislost

- je druhá vazba
- společnost, která se snaží zvýšit svou důvěryhodnost na trhu získáním certifikátu jakosti, musí očekávat také dotaz na úroveň zabezepečení informací
- pro státní zakázky je certifikát jakosti nutnou podmínkou

zjištění legislativního rámce

- je následný krok
- získáme podklady, jaké právní předpisy musí ISMS respektovat a jaké bezpečnostní normativy musí splňovat, aby mohl být certifikován
- normy lze chápat také jako soubor zkušeností a dobrých postupů, přijatých širokou odbornou komunitou pro tu kterou oblast lidské činnosti

identifikace a ocenění aktiv

- je dalším krokem
- je nutno rozhodnout, jaká data a informace jsou danou organizací důležitá
- provedeme klasifikaci aktiv
- určíme, jaká data jsou důležitá
- rozdělíme je na skupiny a podle síly negativního dopadu na organizaci při ztrátě nebo nedostupnosti
- následně definujeme možné hrozby

bezpečnostní dekompozice systému

- informační systém je většinou tvořen několika standardními částmi
- obecně můžeme vydělit data, programové vybavení, technické prostředky, komunikační prostředky a personál
- informační systém je složitý, komplexní celek, který je tak bezpečný, jak je bezpečná jeho nejslabší část
- bezpečnostní dekompozice je rozložení systému na logické části a jejich analýza s cílem odhalit hroby, ocenit rizika a vypracovat odpovídající bezpečnostní funkce
- vzhledem k složitosti systému je právě rozložení na logické a standardní části velmi důležité
- nejobecnějším rozdělením je vydělení vnitřní a vnější bezpečnosti:
    - **vnitřní bezpečnost systému**
        - je systém ochran, které realizuje sám informační systém svým programovým nebo technickým vybavením
    - **vnější bezpečnost**
        - jsou všechna opatření, která si nemůže informační systém zajistit svými prostředky

bezpečnostní parametr (security parameter)

- je část systému vnitřní bezpečnosti (omezená oblast), ve které jsou v činnosti a platnosti kontrolní opatření za účelem ochrany aktiv
- v zásadě lze na uvedené nahlížet jako na vrstovový model informačního systému

dekompozice systému

- **základními složkami vnitřní bezpečnosti je:**

1. ***datová bezpečnost***
   - ochrana dat proti neoprávněné změně, poškození nebo ztrátě
   - antivirová ochrana a systém řízení přístupu k datům
2. ***programová bezpečnost***
   - výběr programového vybavení podle bezpečnostních kvalit a pracovní spolehlivosti
   - kontrola přístupu k software
3. ***technická bezpečnost***
   - výběr technických prostředků pro manipulaci s daty na základě spolehlivosti
   - návrh a realizace s ohledem na spolehlivost a odolnost proti poruchám
   - ochrana proti elektromagentickému vyzařování
4. ***komunikační bezpečnost***
   - souhrn bezpečnostních opatření zajišťujících integritu dat při přenosu mezi komponentami informačního systému

- **vnější bezpečnost je tvořena následujícími složkami:**
  1. **fyzická bezpečnost**
     - zabezpečení aktiv technickými prostředky
     - ochrana budov, strážní služba
  2. **personální bezpečnost**
     - souhrn opatření pro minimalizaci hrozeb způsobených lidským faktorem
  3. **režimová bezpečnost**
     - komplex aministrativních opatření a systém kontrol, které jsou zavedeny za účelem zajištění bezpečnostního chodu systému

fyzická bezpečnost

- zahrnuje ochranu objektů včetně zařízení a organizace přístupu do nich
- dále pak umístění technologických, zabezpečovacích a monitorovacích zařízení
- především se jedná o zabezpeční budov proti neoprávněnému vniknutí (zamykání systému, poplašné zařízení, strážní služba), monitorování pohybu osob (kontrolované zóny, identifikace osob) a katastrofám (záplavy, bouře, požár, pád letadla)
- další oblastí je technické zabezpečení provozu, nepřetržitá dodávka médií (vody, el. energie – UPS), ventilace a chlazení (klimatizace)
- systém technických ochranných prostředků je tvořen mechanickými zábrannými prostředky, elektrickým signalizačním zařízením, souborem organizačních opatření a ostrahou
- účelem zábranných prostředků je vytvoření časové prodlevy mezi okamžikeme napadení objektu a časem jeho dokončení
- velikost prodlevy je kritériem bezpečnostní úrovně zábrany

- **do tétu skupiny patří:**
  - části vnějšího uzavření (ploty, vrata)
  - stavební prvky budov (zdi, podlahy, střechy)
  - otvorové výplně (okan, dveře, mříže)
  - úschovné objekty (schránky, trezory, pokladny)

- **mezi prostředky technického elektrického zabezpeční patří:**
  - požární signalizace
  - signalizace výskytu hořlavých plynů
  - zabezpečovací signalizace

- pro strážní službu musí být vypracovány směrnice ostrahy spolu s plány činností v případě mimořádných okolností

- směrnice musí deklarovat zcela jednoznačně, jaké jsou pravomoci strážných v případě nutnosti ochraňovat střežený objekt

autentizace

- prvým stupněm ochrany aktiv je zabránit neoprávněným osobám v přístupu k nim pomocí záchranných prostředků
- v soudobých informačních systémech, které jsou založeny na přístupu ke vzdáleným prostředkům a službám, to představuje značný problém
- proto se závažnou složkou fyzické bezpečnosti stává autentizace přístupu
- autentizací rozumíme ověření a tím určení identity a požadovanou mírou záruky
- ustanovení identity je platné jen v daném kontextu a pro danou relaci

- **autentizaci můžeme provést třemi základními způsoby:**
    1. *na základě znalosti něčeho (something you know)*
    2. *vlastnictví něčeho (something you have)*
    3. *osobní charakteristiky (něco jsme, something you are)*

- v praxi se většinou využívá nějaká kombinace alespoň dvou způsobů – dvoufaktorová autentizace
- klasickým způsobem autentizaci je identifikace na základě znalosti hesla
- heslem rozumíme důvěrnou autentizační informaci

- **postup je následující:**
    a. prohlášení identity (například uvedení jména)
    b. sdělení hesla, čímž se potvrdí identita

- hesla mohou být individuální nebo skupinová

- **dobré heslo musí splňovat alespoň tyto podmínky:**
    1. obsahuje malá a velká písmena, číslice a jiné znaky
    2. má dostatečnou délku
    3. nepředstavuje smysluplná slova

- slabinou je sám akt uvádění hesla (zavádění do systému)
- heslo je uváděno jako prostý text a může být odhaleno monitorováním nebo paděláním protokolu autentizace (také login attack)
- bezpečnější je použít jednorázová nebo krypto-systému typu výzva/odpověď
    - výzva je otevřená zpráva, odpověď jen kryptogram

- bezpečnější jsou systémy, které nepoužívají prostou (tedy konstantní) výzvu, ale úvodní frázi (Passphrase)
    - ta je určitým postupem transformována na virtuální heslo
    - například: „Dnes je pátek, 12 září, 08:09:120“
    - podstatný je časový údaj – správná odpověď vznikne jeho transformací

- uvedené postupy předpokládají důvěryhodnost systému, ke kterému se přihlašujeme, což nemusí být přijatelné
- pokud oba účastníci sdílejí společný šifrovací klíč je řešení jednoduché
- výzva je zašifrovaná zpráva a časová známka
- odpověď je zašifrovaná kombinace hesla a výzvy
- když účastníci nesdílejí společný šifrovací klíč, je potřebná účast centrální autority, která sdílí šifrovací klíče obou účastníků a podílí se na transakci

- výše uvedený postup je použitelný především při komunikaci mezi počítači

- poměrně často se ale nějaký typ autentizace pomocí věrného hesla (Passphrase) používá ve vztahu člověk-stroj
- heslo je celá věta, ze které se použije vždycky jenurčitá část, například prvá písmena jednotlivých vět
- varianty této metodyse používají při telefonických transakcí

- **identifikátor**
  - je předmět, který slouží k autentizaci (Identity Token)
  - tento předmět není částí uživatele
  - musí být jedinečný a nepadělatelný
  - obvykle se jedná o identifikační nebo čipovou kartu (případně kovový klíč)

karty

- **karta s čárovým kódem doplněná případně fotografií**
  - je nejprimitivnější
  - nese informaci pro člověka obtížně zapamatovatelnou, obvykle PIN (Personal Identification Number)
  - mají nízkou životnost a nelze je přepisovat
- **karty s magnetickým páskem**
  - jsou na tom podobně jako karty s čárovým kódem
  - jsou méně odolné proti opotřebování, ale je možné je přepisovat

- oba typy karet lze poměrně snadno padělat
- určitou nevýhodou je i kontaktní způsob čtení identifikační informace
- karty je nutno do čtečky zasunout
- přes všechny nevýhody se stále používají
- důvody jsou patrně finanční – je to levná technologie

rádiofrekvenční technologie (také indukční nebo RFID)

- další stále častěji používanou
- pro přenos informacemi mezi kartou a snímačem se používají rádiové vlny většinou pasivní
- jednak nemá napájecí zdroj a potřebnou energii získává usměrněním nostné frekvence, kterou vysílá snímač a jednak je identifikační informace na kartu zapsána při výrobě a nelze ji změnit (a ani padělat)
- značnou výhodou je bezkontaktní způsob přenosu informace
- tato technologie představuje do budoucna velký potenciál
- na straně jedné, lze vyrábět jednoduché a tudíž levné identifikátory, navíc nepatrných rozměrů, které je možné použít k identifikaci předmětů (zboží, knih, nářadí)
- na straně druhé je možné vyrobit aktivní identifikační karty, které obsahují jak neměnnou identifikační informaci, tak je možné na ně zapisovat data (například elektronická zdravotní knížka)

optické karty

- představují zvláštní skupiny
- informace je zaznamenávána holograficky
- technologie je poměrně drahá a dnes se už moc nepoužívá
- poslední skupinu představují karty s integrovanými obvody
- obvykle jsou kontaktní

- **v zásadě existují dva typy:**
  - **karty s postupným mazáním**
    - obsahují paměť PROM a řídící obvody
    - čtením dochází k postupné destrukci paměťového obvodu (princip telefonní karty)
    - pro naše účely jsou omezeně použitelné
  - **programovatelné (také chytré, Smart) karty**
    - představují miniaturní výpočetní systém

- ten je tvořen procesorem, pevnou pamětí, která obsahuje identifikační informaci, pamětí RAM a pamětí ROM, ve které je zapsán řídící program
- karta komunikuje s okolím pomocí šifrovaného protokolu
- karety tohoto typu neslouží jen k prosté identifikaci na základě toho, že něco mám, ale obsahují biometrické charakteristiky svého majitele – dvoufaktorová identifikace (například elektronický pas)

autentizace na základě osobních charakteristik

- biometrika je technologie založená na využití specifických osobních rysů pro ověření identity uživatele
- podstatou je snímání, rozpoznání a vyhodnocení fyzických markantů osob
- klasickým příkladem jsou otisky prstů nebo duhovka oka
- markanty jsou jedinečné biometrické prvky (také biometrické vlastnosti), které lze využít v rámci biometrického procesu
- biometrický prvek musí být univerzální – existuje u všech osob, jedinečný – musí každou osobu odlišovat, stálý – každá osoba se v průběhu času biometrický prvek trvale uchovává
- biometrie je tedy obor, který zkoumá člověka, případně jiné živé organismy, podle jedinečných měřitelných charakteristik, a to buď anatomicko-fyzikálních, nebo behaviorálních, (týkajících se chování)
- biometrické systémy jsou aplikace biometrických technologií, které umožňují automatickou identifikaci nebo autentizace /vertifikací určité osoby
- výhodou takového řešení je nevyvratitelnost potvrzení identity, kterou nelze přenést na jinou osobu, ukrást, nebo padělat
- síla biometriky není v utajení informací používaných pro autentizaci, ale v jedinečnosti těchto informací

- **vlastní identifikace osoby může probíhat v zásadě dvěma způsoby:**
  - **úvodní fáze**
    - je v obou případech stejná
    - nejprve dojde k získání biometrického vzorku
    - biometrický systém změří příslušné parametry (například snímač otisku prstu sejme otisk), což se provádí během kroku „zápis“ za použití sensoru specifického pro každý typ jednotlivce – provádí strukturovanou redukci biometrického obrazu
    - ta vytvoří biometrickou šablonu, digitálně zaznamenané, biometrické měření jednotlivce
    - dále se zpracovává pouze šablona nikoli obraz

  - **druhá fáze**
    - vyhodnocení se liší
    - buď je získaná šablona porovnána s daty v databázi šablon, nebo se k verifikaci použijí data uložená na identifikační kartě prověřované osoby
    - každý způsob má své výhody a nevýhody
    - databáze umožňuje nepoužívat identifikační karty, které jsou padělatelné, drahé a pro majitele nepohodlné (musí ji mít stále u sebe)
    - na druhé straně i databáze musí nějakým způsobem vzniknout, musí se udržovat a distribuovat
    - už její existence představuje bezpečnostní riziko – kdo ji vlastně používá a k jakým účelům
    - vždyje možné zneužití třetími stranami, jako základ pro porovnávání a výzkum v rámci jejich vlastních účelů mimo původně sledovaný záměr
    - třetí stranou mohou být i orgány prosazování práva
    - široké a nekontrolované používání biometrike vedek obavám v souvislosti s ochranou základních práv a svobod jednotlivců

- **existují dvě hlavní kategorie biometrických postupů**
  - **ty, které používají stabililní data**
    - jde o postupy založené na fyzických a fyziologických aspektech, které měří fyziologické vlastnosti osoby a zahrnují:
      1. verifikací otisku prstu
      2. analýzu obrazu prstu
      3. rozpoznání duhovky

4. analýzu sítnice
5. rozpoznání obličeje
6. geometrií ruky
7. rozpoznání tvaru ucha
8. detekci pachu těla
9. rozpoznávání hlasu
10. analýza DNA
11. analýzu potních pórů

- **ty, které pracují s daty dynamickými**
  - do této kategorie patří postupy založené na rysech chování, které měří chování osoby a zahrnují verifikaci vlastnoručního podpisu, analýzu stisku tlačítek, analýzu způsobu chůze

- **otisk prstů**
  - nejběžnější a nejlépe prostudovaná metody biometrické identifikace
  - aby současné čtečky docílili téměř stoprocentní přesností pro vzorek několika tisíc lidí, musejí mít k dispozici nejméně dva otisky prstů téhož člověka
  - otisk se však vlivem stárnutí, používání chemických prostředků či zranění mění
  - procedura snímání navíc musí být pod dohledem někoho, kdo ověří, zda je na snímač skutečně položen prst a ne nějaký odlitek
  - metoda je levná, ale poměrně pomalá
  - základy položila daktyloskopie, která vychází z faktu, že na vnitřním povrchu prstů jsou drobné, vyvýšené brázdovité útvary, které vytvářejí různé vzory
  - **vzory se pak dělí do tří základních kategorií:**
    - smyčky
    - přesleny
    - oblouky
  - **podstatná je frekvence výskytu:**
    - smyčky obsahuje 65% všech otisků
    - přesleny okolo 30% všech otisků
    - oblouky jen asi 5% všech otisků
  - v daktyloskopii se pro porovnávání otisků prstů využívají identifikační body – markanty, které se nacházejí v rýchách vzoru
  - identifikační bod je definován určitým vzorem: rozdvojení (konce dvou rýh vytvářejí vidličku), krátká rýha, ukončovací rýha, ohrazení (spojení dvou rýh vytvářející vidličku na obou koncích), izolované body, roztrojení a tak podobně
  - je zjištěno, že některé ze vzorů se vyskytují častěji než ostatní
  - například, krátké rýhy, rozdvojení a ukončovací rýhy jsou daleko frekventovanější než roztrojení, izolované body a ohrazení
  - dakteleskopie sleduje, jak přítomnost markantů, tak i jejich umístění v daném otisku
  - otisk prstu obsahuje v průměru 100-200 markantů
  - v praxi však není stanoven přesný počet markantů nutných k rozlišení mezi dvěma otisky
  - biometrické technologie verifikace pomocí otisku prstu používají různé metody: některé emulují daktyloskopie (porovnávají podle markantů), jiné pracují s přímým porovnáním otisku prstu jako celku, další používají speciální postupy (moaré)

otisk dlaně

- biometrické technologie pro identifikaci pomocí ruky (dlaně) jsou založeny na měření fyzikálních charakteristik ruky a prstů z hledisk trojrozměrné perspektivy
- v prvopočátcíh bylo základem prosté a jednoduché měření délky prstů
- pozdější, zdokonalené postupy snímají tvar ruky – zkoumá se délka a šířka dlaně a jednotlivých prstů, boční profil ruky podobně
- v principu se jedná o speciální skener, který produkuje trojrozměrné digitální fotografie, které se následně redukují na poměrně nevelkou (co do objemu dat) biometrickou šablonu
- metoda je tedy velmi vhodná pro aplikace s omezenou pamětí pro ukládání získaných dat
- geometrie ruky poskytuje poměrně dobrou vyváženost výkonnostních charakteristik a relativní snadností používání
- technologie je vhodná i pro větší databáze uživatelů nebo pro uživatele s ne příliš častým přístupem
- tací nejsou tak disciplinovaní z hlediska správného používání biometrického systému (protože chodí málo), což může vést k častějšímu zamítnutí žadatele
- právě jednoznačnost charakteristik ruky dovoluje docílit poměrně vysoké přesnosti systému
- většinou systém prvé volby

verifikace obličeje

- je dnes patrně nejvíce zkoumanou metodou, neboť potenciál identifikace osob podle tváří je velmi rozsáhlý
- rozpoznávání je založeno na sejmutí obrazu obličeje kamerou
- biometrická šablona vznikne digitalizací tvaru obličeje a polohy opticky významných míst na tváři, jako jsou oči, nos, ústa či obočí
- neuchovává se teda přesná poloha očí, nosu a rtů, ale jen vzdálenost očí, vzdálenost rtů od nosu, úhel mezi špičkou nosu a jedním okem a tak podobně
- jinou možností je matice jasových úrovní, po aplikaci nějaké diskriminační funkce, která zajistí odstranění redundantních dat z původního obrazu
- biometrické identifikace na základě snímání obličeje je dnes velmi atraktivní technologie, má totiž vlastnost, které jiné biometrické metody nemají – především umožňuje verifikaci osoby ve skupině a případně i bez jejího vědomí

verifikace duhovky

- pdoobně jako otisky prstů i duhovku oka má každý člověk jedinečnou
- existuje asi šedesát odlišných forem otisků prstů, které mohou být různě kombinovány na jednom otisku
- v případě duhovky je počet různých forem vyšší než 400
- nalezení dvou identických duhovek je tedy mnohonásobně méně pravděpodobné, než nalezení dvou identických otisků prstů
- i duhovky dvou jednovaječných dvojčat jsou rozdílné
- a i obě duhovky jednoho člověka jsou rozdílné
- neexistuje žádný jiný biometrický markant, který by byl více specifický než právě oční duhovka

verifikace sítnice

- sítnice je na světlo citlivý povrch zadní strany oka
- skládá se z většího počtu nervových buněk, které převádějí světelné paprsky na nervové signály, říká se jim tyčinky a čípky
- čípky poskytují barevné a tyčinky pouze černobílé vidění
- každá tyčinka i čípek převádí signály do mozku pomocí očního nervu
- artérie sítnice a oční nerv vystupují společně z oka v bodě, kde nejsou žádné čípky ani tyčinky
- místo se označuje jako slepá skrvna

- pro ověření sítnice se používá obraz struktury sítnice v okolí slepé skvrny. Ten lze získat pomocí zdroje světla s poměrně nízkou intenzitou a vhodného optoelektronického systému
- následně je obraz digitalizován a převeden na biometrickou šablonu v délce okolo 40 bajtů
- obraz sítnice vykazuje zhruba stejné specifické vlastnosti jako otisky prstů
- verifikace sítnice je velice přesnou, spolehlivou a obtížně padělatelnou biometrickou metodou
- vyžaduje však, aby se při snímnání uživatel díval přesně do ohraničeného prostoru a měl zaostřeno na daný bod
- což muže být subjektivně nepříjemné
- patrně právě to je důvod, proč mnoho uživatelů metodu verifikace sítnice odmítá, tím se použití této tehcniky redukuje na vrcholně zabezpečené systémy

autentizace pomocí hlasu

- lze definovat jako biometrickou metodu založenou na elektronické analýze digitálního vzorku lidské řeči
- tvar hlasivek, ústní dutiny, jazyka a zubů způsobuje, že rezonance vokálního traktu je u různých osob dostatečně specifická
- existuje vícero variant této metody – porovnání vzorků mluvy pomocí spektrální analýzy signálů nebo autentizací rozhodnutí na základě analýzy vět
- věta má více akustické informace než jednotlivá slova, která jsou často krátká a nestačí pro spolehlivé rozlišení mluvčího
- více informace úměrně zvyšuje spolehlivost srovnávacího procesu
- navíc, věty zná pouze autentický mluvčí a mohou jimi být i množiny slov (nesmyslná věta)
- neoprávnění uživatelé (podvodníci) neví, kterou větu použít, natož jakým hlasem ji vyslovit
- verifikace hlasu se dnes používá především při vzdáleném přístupu do informačních systémů prostřednictvím telefonu, například identifikace klientů call center poskytování služeb
- obecně je hlasová komunikace v každodenní činnosti lidi významná a častá
- verifikace hlasu je tudíž velmi zajímavá biometrická technika
- i díky faktu, že telefon používáme bez potíží
- digitální data vzniklá při telefonování lze přímo použít k hlasové identifikaci
- bohužel, charakteristickým příznakem současných systémů je jistá nespolehlivost v důsledku náhodných vlivů
- například nastydnutí mluvčího či šum okolí

datové nosiče

- zneužít se dá jak prázdný tak i naplněný nosič dat. Operační paměť je po každém jeho zapnutí jako nosič dat prázdná. Naproti tomu pevný disk, disketa, CD-disk, optický disk, magnetická páska mění svůj nehmotný obsah pouze jistým speciálním úkonem – zápisem, přepsáním, vymazáním, formátováním a tak podobně
- nosiče informací, stejně jako jinou část hardware je možno poškodit vnějším zásahem. Tím lze jednak pozměnit nebo zničit nehmotný obsah nosiče a jednak změnit či zcela znehodnotit jeho základní vlastnost – uchovávat data po definovanou dobu
- oba způsoby jsou z kriminalistické praxe známy. Například úmyslné pozměňování či ničení informací na magnetických nosičích cíleným magnetickým polem, nebo poškozování dat pomocí speciálních programů (virů, červů).
- všechna datová média mají jednu společnou vlastnost – dají se přenášet. Navíc na rozměrově nevelké zařízení se dá uložit značné množství dat
- ochrana datových nosičů se realizuje především jako ochrana prostorová – pomocí mechanických zábran
- v každém výpočetním systému, existují alespoň dvě místa, kde je koncentrace datových nosičů kritická
- prvým je místnost serverů. To je nejlákavější místo pro nelegální zájemce (zloděje)
- zde je největší kumulace dat, programového vybavení i hardware
- místnosti serverů je nutno věnovat pozornost (bude diskutováno dále)
- klasickou chybou je společný prostor pro servery obsahující personál („systémáky“)
- dalším kritickým místem je archiv. Data na záložních nebo archivních nosičích představují značné bezpečnostní riziko
- v archivu není přístup k datům chráněn bezpečnostními prvky systému (například kontrolou přístupu)

- proto je prostorová ochrana velmi důležitá. Existuje nepřeberné množství skříní, trezorů a schránek speciálně určených k ukládání nosičů
- chrání nejen proti odcizení, ale což je stejně důležité, také proti poškození při skladování mechanickému – v důsledku pádu z výšky, teplotnímu – požár, znehodnocení vodou či agresivními plyny
- jinou možností jsou bankovní bezpečnostní schránky, ty bohužel většinou nechrání proti extrémním vlivům (například zatopení). Poslední dva aspekty se často přehlížejí
- základním faktorem pro zajištění spolehlivých dat je spolehlivý nosič. Je dobrou praxí, nakupovat média podle předpokládaného použití a nikoli pouze podle ceny
- pro cenná data, která se budou uchovávat dlouho, spolehlivý a tudíž drahý nosič
- velmi opomíjeným faktem je likvidace dat a datových nosičů. Prosté vymazání nebo přeformátování nosiče pro citlivá data nestačí. Likvidace nosičů je většinou okrajová záležitost, což dokladuje množství případů

technické zabezpečení provozu

- postatou je zajištění spolehlivého, bezporuchového provozu technických prostředků informačního systému
- základním krokem je zajištění kvalitní dodávky elektrické energie. Statistiky ukazují, že zhruba 60-70% závad v informačních systémech je způsobeno poruchami napájení
- nestability v napájecím napětí můžeme klasifikovat jako podpětí, kdy je napětí menší o více než 15% nominální hodnoty, přepětí, kdy je napětí vyšší, napěťové a proudové špičky, kdy napětí několikanásobně převyšuje nominální hodnotu a rušení
- zdroj elektrické energie musí zajistit stabilitu napětí a frekvence. Navíc, některé kritické části informačního systému musí být napájeny stále – i krátkodobý výpadek napájení zde způsobí nezvratné poškození dat (například servery)

- **existují dvě řešení:**
  - ***motorgenerátor***
    - umožňuje překlenout delší výpadky (hodiny až dny), ale nenaběhne okamžitě (řádově vteřiny, rozběh motoru a nasazování generátoru)
    - obecně je vhodný pro rozsáhlejší systémy (například nemocnice)

  - ***záskokový zdroj (ups, uninterruptible power supply)***
    - záskokové zdroje řeší krátkodobé výpadky (minuty až hodiny)
      - **známe dva základní typy záskokových zdrojů:**
        - off-line
          - je jednodušší, ale poskytuje méně komplexní ochranu. Za normální situace jsou připojené spotřebiče přímo napájeny ze sítě, kvalitnější typy obsahují i filtr po odstranění špiček a rušení. Při výpadku dojde k odpojení od sítě a k připojení na měnič, který převádí stejnosměrné napětí z akumulátoru na střídavé. Přepnutí realizuje relé, přepínací čas je zhruba desítky milisekund, což většinou vyhovuje - on-line
        - on-line
          - on-line záskokový zdroj napájí spotřebiče z měniče trvale a stále dobíjí akumulátory. Poskytované napájení je podstatně kvalitnější. Zdroj eliminuje přepětí i podpětí, špičky i většímu rušení. Navíc při výpadku nedochází k přepnutí, napájení je plynulé. Důležitou vlastností UPS je schopnost komunikace s počítačem. Řídící modul záskokového zdroje je standardní součástí všech soudobých operačních systémů

klimatizace

- chlazení a ventilace jsou důležité složky technického zabezpečení provozu. Komponenty informačního systému jsou většinou vnímavé na teplo, vlhkost a vyžadují čistý vzduch. Navíc zhusta samy produkují značné množství

tepla, které je třeba odvádět. Platí tyto doporučení. Vzduchotechniku je třeba instalovat mimo obecné přístupné prostory a k jejímu ovládání smí mít přístup pouze obslužný personál. Řízení vzduchotechniky musí mít vztah na požární komunikaci a detekci nebezpečných plynů.

personální bezpečnost

- ze statistik vyplývá, že zhruba 60-80% bezpečnostních incidentů mají na svědomí vlastní zaměstnanci. Přitom každý bezpečnostní incident představuje v průměru zhruba 45 000 $. Mezi základní příčiny patří nedbalost, hloupost a zlý úmysl. Ochranu před nežádoucím vlivem lidského faktoru, často označovaného jako „vnitřní hrozba“, zajišťuje personální bezpečnost. Jak již bylo řečeno, má dvě roviny. Řeší problematiku přístupu osob k chráněným aktivům, prověřování osob a definování stupně důvěry. Definuje požadavky na osoby pracující v informačním systému, jejich vlastnosti a kvalifikaci, kritéria jejich výběru, výchovu a profesní vzdělávání, včetně průběžných kontrol.

- **personál představuje následující hrozby:**
    1. *úprava přístupových práv ať již svých nebo cizích*
    2. *neúmyslné nebo záměrné poškození dat*
    3. *úpravy konfigurace systému*
    4. *nelegální instalace software*
    5. *nelegální instalace hardware*
    6. *nedodržování stanovených bezpečnostních předpisů a to jak z neznalosti tak i nedbalosti*

- výběr nových zaměstnanců přestavuje základní předpoklad pro získání kvalitního personálu. Většinou se prověřují hlavně odborné a jazykové znalosti, organizační a řídící schopnosti uchazeče.
- neméně důležitým kritériem jsou morální, osobní a pracovní vlastnosti: pracovní spolehlivost, poctivost, psychická odolnost a dobré rodinné zázemí.
- tyto vlastnosti se však obecně prokazují mnohem obtížněji než odborné schopnosti. Určitou váhu mají reference z minulých pracovišť nebo doporučení vlastních, důvěryhodných zaměstnanců. Používají se také nejrůznější psychotesty nebo pohovor s psychologem. Bohužel, morální kvality člověka se zjišťují jen velmi obtížně. Často bývá formální podmínkou výpis z rejstříku trestů, případně testy na používání drog nebo alkoholu. Přijatý zaměstnanec musí absolvovat úvodní zaškolení, jehož neoddělitelnou součástí je i seznámení s bezpečnostními předpisy a směrnicemi organizace. Absolvování školení je potřeba potvrdit písemným dokladem. Jen tak lze zabránit pozdějším námitkám a výmluvám. Nedílnou součástí vstupního školení je zapracování do konkrétní funkce (většinou pod dohledem zkušeného zaměstnance). Cílem je minimalizace škod v důsledku nekvalifikovaného úkonu nového zaměstnance. Získání a udržení zaměstnanců je stálý proces. Průběžné zvyšování kvality zaměstnanců z hlediska bezpečnosti se dosahuje tréninkem, školením a pečlivou kontrolou dodržování požadavků a směrnic.

režimová bezpečnost

- je soubor administrativních opatření a nařízení spolu se systémem kontrol
- také je nazývána bezpečnost organizační nebo administrativní
- režim je obecně souhrn opatření nutných pro nějaký účel, takže název režimová bezpečnost je patrně výstižnější
- režim je administrativní, organizační a věcné uspořádání vztahů mezi lidmi, jejich činnostmi a vlastními procesy v oblasti výkonu i řízení
- účelem je dosažení harmonického stavu
- režim souvisí se všemi druhy ochranných opatření, v podstatě je završuje
- například, pokud máme sofistikované mechanické zábrany, směrnice pro pohyb cizích osob v objektu, ale nikdo nebude nic kontrolovat je efekt bezpečnostních funkcí prakticky nulový

- **mezi základní procedury režimové bezpečnosti patří zejména následující:**
    - opatření upravující vstup do objektů a místností, pohyb cizích osob a způsoby kontroly
    - práva, povinnosti, odpovědnost a kompetence jednotlivých pracovníků
    - řízení vlastního provozu (manažerské funkce)

- organizace práce a provozní režim
- definice oprávněných a zakázaných činností
- způsoby zálohování dat a uložení médií
- vztahy s dodavatelskými organizacemi

- respektování režimové bezpečnosti je významné z hlediska trestně právních následků při narušení bezpečnosti informačního systému
- v takovém případě představuje základní kriterium při posuzování charakteru vzniklých škod a určování viníků a míry jejich zavinění

softwarová bezpečnost

- zajišťuje ochranu hodnot informačního systému pomocí programových prostředků
- informační systém poskytuje oprávněným uživatelům odpovídající zdroje
- potřebujeme definovat vztahy mezi objekty a subjekty (nepřesně, mezi zdroji a uživateli)
- přístup subjektů k objektům není definován klasicky (například, povoleno čtení, zápis, mazání, atd), ale přidělením práva, požívat určité procedury, kterými se objekt zpřístupňuje
- základem je řekněme, centrální databáze. Každý objekt v systému je v ní zaregistrován a je mu přidělen vlastník. Ten má právo přidělovat přístupová práva k objektu ostatním subjektům
- každý objekt v systému je v ní zaregistrován a je mu přidělen vlastník. Ten má právo přidělovat přístupová práva k objektu ostatním subjektům
- každý subjekt má přidělenu jistou strukturu, seznam. Ten obsahuje odkazy (a popisy),* na všechny objekty, ke kterým má daný subjekt alespoň jedno oprávnění. Platí zásada, že žádný subjekt nesmí modifikovat svůj vlastní seznam (definovat sám sobě přístupová práva). Výše uvedené schéma není pro rozsáhlé systémy vhodné. Existuje mnoho variant výše uvedeného (přístupová matice)
- spolehlivé programové vybavení představuje základ bezpečného informačního systému

- **nejvýznamnější hrozby spojené s programovým vybavením jsou v podstatě dvě:**
  - provádění neoprávněných operací s daty
  - chybné operace v důsledku chyb v programech

- teoreticky můžeme za funkčně správný prohlásit takový program, který vykonává všechny požadované funkce a neprovádí žádné jiné

- spolehlivý je potom takový program (Trusted Software), kterému věříme, že je funkčně správný a tuto vlastnost vynucuje i software, který sám spouští.

- **programové vybavení určitého informačního systému získáme obvykle ze tří zdrojů:**
  1. *nákupem hotového produktu*
  2. *nákupem a úpravou určitých funkcionalit*
  3. *vývojem vlastními silami nebo pomocí subdodavatelů*

- ve všech případech je patrně největším problémem prověření (testování), že program zachovává správnost dat a to i tehdy, když dostává nesprávné nebo nelegální příkazy, jinými slovy, že zaručuje ochranu integrity

- druhou podstatou vlastností je řízení přístupu. Pokud program přistupuje k citlivým datům, nesmí povolit nebo umožnit přístup ostatnímu nespolehlivému programovému vybavení

- v reálných podmínkách nelze dosáhnout stavu, aby informační systém používal pouze spolehlivé programové vybavení

- **k vyloučení hrozeb se využívají tři základní metody:**
    - vzájemné podezírání programů (Mutual Suspicion)
    - omezení programových zdrojů
    - parcelace datového a paměťového provozu

- prvá metoda pracuje na základě skutečnosti, že pracující spolehlivý program považuje všechen ostatní software za vadný. Nevěří ničemu a s okolím komunikuje pouze prostřednictvím přesně definovaného a dobře chráněného rozhraní

- druhou metodu používají většinou spolehlivé operační systémy. Spouštěným programům (potencionálně nespolehlivým) přesně vymezí, jaké zdroje mohou používat a vše ostatní zablokují pomocí ochranných mechanizmů operačního systému

- parcelací se pak rozumí, že veškerá data a programy jsou v systému rozděleny tak, aby každá informace (nepřesně SW a data) ležela pouze a jedině v jedné oblasti. K ostatním oblastem nemá aplikace přístup

- pro vývoj bezpečného software se používají tyto základní metodiky. Aplikace je rozdělena na logické celky. Každou část vyvíjí oddělený tým. Testování provádí jiná, nezávislá skupina (metoda Peer Reviews). Metoda zapouzdření je podobná. Jednotlivé moduly jsou zcela samostatné a komunikují spolu přes přesně definovaná rozhraní. Každý modul je pro ostatní moduly černá skříňka.

LEGISLATIVA

britský standard 7799

- je nejznámější, celosvětový rozšířený bezpečnostní standard
- vznikl kolem roku 1995 a byl několikrát revidován
- v prosinci roku 2000 byl přijat jako mezinárodní norma **ISO/IEC 17799**

- představuje standard komplexní ochrany informací
- pokrývá všechny oblasti bezpečnosti, popisuje rozhsáhlý počet bezpečnostních opatření a je použitelná pro různé typy organizací
- skládá se ze tří částí:
  - **BS7799-1:1999**
    - Code of Practice for Information Security Management
    - katalog bezpečnostních funkcí a bezpečnostních opatření
    - **také ISO/IEC 17799:2000**
      - představuje podrobné shrnutí praktických zkušeností s řešením informační bezpečnosti
      - definuje 120 bezpečnostních funkcí rozložených do 10 zón
  - **BS7799-2:1999**
    - Specification for Information Security Management Systems (označuje se zkratkou ISMS)
    - představuje návod k výstavbě systému řízení bezpečnostních informací
    - v podstatě říká, jak aplikovat prvoučást normy
  - **BS7799-2:2002**
    - je inovace druhé části
    - doporučuje, jak používat prvou část normy a jak provozova, udržovat a vylepšovat existující ISMS

české normy

- vznikají obvykle adopcí norem mezinárodních
- v říjnu 2006 byla zrušena norma ČSN BS 7799-2 a byla nahrazena nově vydanou normou ČSN ISO/IEC 27001
- Informační technologie – Bezpečnostní techniky – Systémy managementu bezpečnosti informací – Požadavky
- norma je propojena a harmonizována s normami ISO/IEC 9001:2000 (kvalita) a ISO/IEC 14001:2004 (životní prostředí) tak, aby bylo podpořeno konzistentní zavedení a provoz

- **hlavní části normy:**
  - definuje požadavky na vybudování, zavedení, provoz, monitorování, přezkoumávání, udržování, zlepšování a případnou certifikaci systému managementu bezpečnosti informací
  - dále jsou zde specifikovány požadavky na výběr a zavedení bezpečnostních opatření chránících informační aktiva
  - příloha A uvádí cíle a opatření, která jsou přímo propojena s cíly a opatřeními uvedenými v ISO/IEC 17799:2005
  - v příloze B je uvádeň vztah mezi principy OECD pro bezpečnosti informačních systémů a sítí a fázemi PDCA cyklu (bude vysvětleno dále)
  - příloha C uvádí veztah mezi ISO/IEC 9001:2000, ISO/IEC 14001:2004 a ISO/IEC 27001:2005
- je plánováno, že v budoucnu bude ISO/IEC 27xxx (ze které naše norma vychází) obsahovat sedm dokumentů:
  - 27000, principy a slovník
  - 27001, požadavky na ISMS (ISO ekvivalent BS 7799-2:2004)
  - 27002, návody pro zavádění
  - 27003, analýza rizik
  - 27004, metrikya měření

- 27005, řízení rizik
- 27006, kontinuita podnikání a obnova po havárii

KRYPTOGRAFIE V INFORMAČNÍCH SYSTÉMECH

- Kryptografie neboli šifrování je nauka o metodách utajování smyslu zpráv převodem do podoby, která je čitelná jen se speciální znalostí. Kryptologie zahrnuje kryptografii a kryptoanalýzu, neboli luštění zašifrovaných zpráv.
- Kryptografické metody obecně využívají tzv. "klíč", pomocí kterého tajná data zakódují a posléze opět rozkódují. Současně některé metody umožňují nebo i vynucují použití více klíčů různých pro zakódování a rozkódování.
- Utajení dokumentu se skládá z dvou částí. Utajení šifrovací metody a utajení klíče. Zásadní je zejména utajení klíče, jelikož metod není takové množství, aby nemohlo dojít k jejímu odhalení. Často se tedy ani k utajení vlastní metody nepřistupuje a utajení zajišťuje jen klíč.
- Většina moderních algoritmů je založena na matematické teorii čísel. Tzv. kryptografická transformace T je libovolné prosté zobrazení množiny celých čísel na množinu celých čísel. Kryptografický systém je pak parametrický systém kryptografických transformací TK = (Tk : k ∈ K), kde k je klíč a K je prostor klíčů.

- **Podle použití způsobu práce s klíči se kryptografické metody dělí na:**
    - Symetrické
    - asymetrické

Symetrické metody kryptografie

- používá stejný klíč jak pro zakódování, tak pro rozkódování zprávy.
- Vstupem je tedy nějaký tajný text ze stanovené abecedy, a klíč. Šifrovací funkcí se za pomoci klíče tajný text převede na kód, který může být odeslán příjemci zprávy. Příjemce pak použije dešifrovací funkci se stejným klíčem a tím získá původní tajný text. Důležité je, že pro dešifrování musí mít příjemce k dispozici stejný klíč, jakým byl text zakódován. Je tedy třeba zajistit bezpečný způsob doručení klíče, aby se tento klíč nedostal do nepovolaných rukou.
- Výhodou symetrických metod je jejich rychlost. Dají se velmi dobře využít pro šifrování dat, která se nikam neposílají (zašifrují se dokumenty na počítači, aby je nikdo nemohl číst).
- Největší nevýhodou je, že pokud chceme s někým tajně komunikovat, musíme si předem bezpečným kanálem předat klíč. To někdy může být obrovský problém.
- Druhá nevýhoda je počet klíčů. Chceme-li zajistit, aby mohli tajně spolu komunikovat 2 osoby, je zapotřebí 1 klíče. Pro 3 osoby jsou to již 3 klíče, pro 4 osoby 6 klíčů, obecně počet klíčů = n*(n-1)/2, kde n je počet osob. Při vyšším počtu osob tak začíná být správa klíčů problémem.

Asymetrické metody kryptografie

- Asymetrická proto, že využívá jiného klíče pro zakódování a jiného pro rozkódování. Dohromady se oba klíče nazývají párem klíčů ("kaypair"). Šifruje se pomocí tzv. veřejného klíče ("public key") a dešifruje pomocí soukromého klíče ("private key").
- Veřejný klíč je skutečně veřejný, tj. pokud uživatel chce, aby mu někdo mohl poslat zakódovanou zprávu, musí nejprve dát k dispozici tento svůj veřejný klíč. Ten použije kolega pro zakódování tajné zprávy a kód odešle. Pro rozkódování pak potřebuje příjemce mít druhý klíč z páru, soukromý klíč, který jediný lze použít pro rozkódování. Klíčový pár se většinou tvoří zároveň. Algoritmus uživateli vygeneruje oba klíče, veřejný klíč uživatel dá k dispozici kolegům a soukromý klíč si dobře uschová.
- S délkou klíče asymetrických metod je to kapku jinak, než u symetrických šifer. Asymetrické šifry většinou pracují se specifickým druhem čísel, např. s prvočísly. Při záškodníkových pokusech o rozkódování se pak stačí zabývat jen tímto oborem čísel a tedy i počet bitů klíče je třeba oproti symetrickým metodám patřičně navýšit, aby byla zachována požadovaná míra bezpečnosti. V dnešní době se tak běžně pracuje s 1024 bitovými či 2048 bitovými klíči.
- Hlavní výhodou je to, že není třeba nikam posílat soukromý klíč a tak nemůže dojít k jejímu vyzrazení. Naproti tomu veřejný klíč je možné dát k dispozici všem.
- Je třeba méně klíčů než u symetrických metod – pro komunikaci několika osob postačí pro každou osobu jen jeden pár klíčů.
- Nevýhodou asymetrických metod je však rychlost. Tyto metody jsou až 1000 x pomalejší než metody symetrické.

- Další nevýhodou asymetrické kryptografie je nutnost ověření pravosti klíče, tj. stoprocentní identifikace majitele veřejného klíče. Pro tyto účely existují např. certifikační úřady, které zjednodušeně řečeno udržují databázi osob s ověřenou totožností a jejich veřejných klíčů. V teoretickém případě nabourání takového úřadu však může záškodník např. zaměnit klíče u různých registrovaných osob a tak nic netušící uživatel zakóduje tajnou zprávu veřejným klíčem záškodníka místo klíčem skutečného adresáta.

- Vzhledem k pomalosti asymetrických metod kódování se často využívá kombinace obou metod, kdy se z každé metody využívají její přednosti. Tajný klíč symetrické metody je např. zakódován veřejným klíčem asymetrické metody a tak je zajištěno jeho bezpečné předání adresátovi. Tajným klíčem pak lze kódovat vlastní tajnou zprávu.

Charakteristika dobré šifry

- Šifra by neměla být prolomitelná v reálném čase a s použitím „rozumných“ výdajů.
- Šifrování by mělo proběhnout rychle.
- Množství práce vynaložené na šifrování a dešifrování by mělo být úměrné požadovanému stupni utajení.
- Šifrovací algoritmus by neměl obsahovat zbytečná omezení.
- Implementace algoritmu by měla být co nejjednodušší.
- Chyby při šifrování by se neměly příliš šířit.
- Zprávy by se šifrováním neměly zvětšovat. ¨

Metody šifrování

Jednosměrné šifrování

- Při této metodě není možné dešifrování zašifrovaných dat. Aplikuje se nejčastěji např. při ukládání uživatelských hesel do databáze. Nově zadané heslo se zašifruje (např. pomocí funkce MD5) a uloží se bezpečně do databáze. Při opětovném přihlášení uživatele se zadané heslo opět zašifruje a tato šifrovaná hodnota se porovná s dříve uloženou hodnotou v db.

Steganografie (tajnopis)

- Slouží k ukrývání tajných zpráv tak, aby i samotná informace o předání tajné zprávy nebyla známa. V historii se používaly např. tajné neviditelné inkousty, nepatrné vpichy ve vybraných znacích, mřížky zakrývající většinu zprávy s výjimkou několika písmen apod.
- Dnes se používá např. ukrývání zprávy do tištěných obrazů s využitím více barevných odstínů, než kolik může lidské oko rozlišit. Do obrazu 1024 x 1024 s různými stupni šedi lze ukrýt až 64 kB zprávu.

Digitální podpis (elektronický podpis nebo též e-podpis)

- Zajímavou aplikací asymetrických metod kryptografie je tzv. digitální podpis. Pro použití digitálního podpisu potřebujeme nejprve nějakou známou hashovaní funkci (např. MD5 nebo SHA-1). Známou v tom smyslu, aby všichni adresáti, kteří budou chtít ověřit pravost naší zprávy, tuto funkci znali. Hash funkce udělá z naší zprávy tzv. otisk nebo se výsledek také dá nazvat jakýmsi kontrolním součtem zprávy. Tento otisk má vždy stejnou délku bez ohledu na délku vstupní zprávy. Jednou z vlastností této hashovací funkce je fakt, že zaprvé prakticky není možné z otisku zpětné získání původní zprávy, a zadruhé je i velmi nepravděpodobné nalezení jiné zprávy, která by použitím hashovaní funkce dala stejný otisk.
- Jestliže takto vzniklý otisk zakódujeme svým soukromým klíčem, vznikne nám kýžený digitální podpis. Podpis pak přiložíme k původní zprávě, kterou podepisujeme, a zprávu i s touto přílohou odešleme. Příjemce zprávu otevře, a pomocí stejné hashovaní funkce zakóduje její obsah. Pomocí veřejného klíče odesílatele dále rozkóduje obsah digitálního podpisu. Je-li tento rozkódovaný obsah totožný s otiskem přijaté zprávy, je identita odesílatele potvrzena, jelikož nikdo jiný, než vlastník soukromého klíče nemohl digitální podpis s touto vlastností vytvořit.
- Hashovaní funkce se používá z důvodu, aby přikládaný digitální podpis nebyl příliš velký. Pokud by odesílatel svým soukromým klíček kódoval celou zprávu, digitální podpis by byl minimálně jednou tak velký a tedy finální zpráva s podpisem by zvětšila objem minimálně na dvojnásobek. V případě použití hashovaní funkce je zaručena stejná funkčnost, avšak s minimální datovou přítěží k původní zprávě.

PGP

- kombinuje oba druhy metod kryptografie. Nejprve vygeneruje náhodný tajný klíč, který pak použije pro symetrické kódování vlastní tajné zprávy. Tento tajný klíč zakóduje pomocí veřejného klíče asymetrické metody

(RSA či DH). Oba kódy (zakódovaný klíč i zakódovaný text zprávy) odešle příjemci. Příjemce pak vezme nejprve zakódovaný tajný klíč. Ten pomocí svého soukromého klíče dekóduje a následně jej použije pro rozkódování vlastní zprávy.

- Tento princip tak šikovně kombinuje vlastnosti obou typu kryptografických algoritmů. Tj. rychlou funkčnost symetrického kódování a vysokou bezpečnost předání tajného klíče pomocí asymetrické metody kódování.

SSL

- Další aplikací kryptografických metod je protokol SSL, který vytvořila firma Netscape. Vrstva SSL (Secure Sockets Layer) řeší zabezpečení přenášených dat mezi klientem a serverem a je vložena mezi aplikační protokol a protokol TCP. Přenášená data se pak tedy mezi WWW serverem a browserem přenášejí kódovaně pomocí šifrování veřejným a soukromým klíčem (asymetrické šífrování). Klíče mohou navíc obsahovat autentifikační informaci od certifikační autority.
- Klient i server si vygenerují dvojici klíčů. Každá strana jeden ze svých klíčů prohlásí za soukromý a druhý za veřejný klíč. Na začátku spojení si oba vymění veřejné klíče. Pokud odesílatel zašifruje data veřejným klíčem "protější" strany, má jistotu, že je přečte jenom příjemce s odpovídajícím tajným klíčem. Klíče bez podpisu certifikační autority slouží pouze ke kódování přenosu, nikoliv ke zprostředkování možnosti ověření autenticity serveru.

BEZPEČNOST POČÍTAČOVÝCH SÍTÍ

Firewall

- síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a/nebo zabezpečení. Zjednodušeně se dá říct, že slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje. Tato pravidla historicky vždy zahrnovala identifikaci zdroje a cíle dat (zdrojovou a cílovou IP adresu) a zdrojový a cílový port, což je však pro dnešní firewally už poměrně nedostatečné – modernější firewally se opírají přinejmenším o informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS. Firewally se během svého vývoje řadily zhruba do následujících kategorií:
  - Paketové filtry
  - Aplikační brány
  - Stavové paketové filtry
  - Stavové paketové filtry s kontrolou známých protokolů a popř. kombinované s IDS

Paketové filtry

- Nejjednodušší a nejstarší forma firewallování, která spočívá v tom, že pravidla přesně uvádějí, z jaké adresy a portu, na jakou adresu a port může být doručen procházející paket, tj. kontrola se provádí na třetí a čtvrté vrstvě modelu síťové komunikace OSI.
- Výhodou tohoto řešení je vysoká rychlost zpracování, proto se ještě i dnes používají na místech, kde není potřebná přesnost nebo důkladnější analýza procházejících dat, ale spíš jde o vysokorychlostní přenosy velkých množství dat.
- Nevýhodou je nízká úroveň kontroly procházejících spojení, která zejména u složitějších protokolů (např. FTP, video/audio streaming, RPC apod.) nejen nedostačuje ke kontrole vlastního spojení, ale pro umožnění takového spojení vyžaduje otevřít i porty a směry spojení, které mohou být využity jinými protokoly, než bezpečnostní správce zamýšlel povolit.

Aplikační brány

- Jen o málo později, než jednoduché paketové filtry, byly postaveny firewally, které na rozdíl od paketových filtrů zcela oddělily sítě, mezi které byly postaveny. Říká se jim většinou Aplikační brány, někdy také Proxy firewally. Veškerá komunikace přes aplikační bránu probíhá formou dvou spojení – klient (iniciátor spojení) se připojí na aplikační bránu (proxy), ta příchozí spojení zpracuje a na základě požadavku klienta otevře nové spojení k serveru, kde klientem je aplikační brána. Data, která aplikační brána dostane od serveru, pak zase v původním spojení předá klientovi. Kontrola se provádí na sedmé (aplikační) vrstvě síťového modelu OSI (proto se těmto firewallům říká aplikační brány).
- Jedním vedlejším efektem použití aplikační brány je, že server nevidí zdrojovou adresu klienta, který je původcem požadavku, ale jako zdroj požadavku je uvedena vnější adresa aplikační brány. Aplikační brány díky tomu automaticky působí jako nástroje pro překlad adres (NAT), nicméně tuto funkcionalitu má i většina paketových filtrů.
- Výhodou tohoto řešení je poměrně vysoké zabezpečení známých protokolů.
- Nevýhodou je zejména vysoká náročnost na použitý HW – aplikační brány jsou schopny zpracovat mnohonásobně nižší množství spojení a rychlosti, než paketové filtry a mají mnohem vyšší latenci. Každý protokol vyžaduje napsání specializované proxy, nebo využití tzv. generické proxy, která ale není o nic bezpečnější, než využití paketového filtru. Většina aplikačních bran proto uměla kontrolovat jen několik málo protokolů (obyčejně kolem deseti). Původní aplikační brány navíc vyžadovaly, aby klient uměl s aplikační branou komunikovat a neuměly dost dobře chránit svůj vlastní operační systém; tyto nedostatky se postupně odstraňovaly, ale po nástupu stavových paketových filtrů se vývoj většiny aplikačních bran postupně zastavil a ty přeživší se dnes používají už jen ve velmi specializovaných nasazeních.

Stavové paketové filtry

- Stavové paketové filtry provádějí kontrolu stejně jako jednoduché paketové filtry, navíc si však ukládají informace o povolených spojeních, které pak mohou využít při rozhodování, zda procházející pakety patří do již

povoleného spojení a mohou být propuštěny, nebo zda musí znovu projít rozhodovacím procesem. To má dvě výhody – jednak se tak urychluje zpracování paketů již povolených spojení, jednak lze v pravidlech pro firewall uvádět jen směr navázání spojení a firewall bude samostatně schopen povolit i odpovědní pakety a u známých protokolů i další spojení, která daný protokol používá. Například pro FTP tedy stačí nastavit pravidlo, ve kterém povolíte klientu připojení na server pomocí FTP a protože se jedná o známý protokol, firewall sám povolí navázání řídícího spojení z klienta na port 21 serveru, odpovědi z portu 21 serveru na klientem použitý zdrojový port a po příkazu, který vyžaduje přenos dat, povolí navázání datového spojení z portu 20 serveru na klienta na port, který si klient se serverem dohodli v rámci řídícího spojení a pochopitelně i odpovědní pakety z klienta zpět na port 20 serveru. Zásadním vylepšením je i možnost vytváření tzv. virtuálního stavu spojení pro bezstavové protokoly, jako např. UDP a ICMP.

- K největším výhodám stavových paketových filtrů patří jejich vysoká rychlost, poměrně slušná úroveň zabezpečení a ve srovnání s výše zmíněnými aplikačními branami a jednoduchými paketovými filtry řádově mnohonásobně snazší konfigurace – a díky zjednodušení konfigurace i nižší pravděpodobnost chybného nastavení pravidel obsluhou.
- Nevýhodou je obecně nižší bezpečnost, než poskytují aplikační brány.

Stavové paketové filtry s kontrolou protokolů a IDS

- Moderní stavové paketové filtry kromě informací o stavu spojení a schopnosti dynamicky otevírat porty pro různá řídící a datová spojení složitějších známých protokolů implementují něco, co se v marketingové terminologii různých společností nazývá nejčastěji Deep Inspection nebo Application Intelligence. Znamená to, že firewally jsou schopny kontrolovat procházející spojení až na úroveň korektnosti procházejících dat známých protokolů i aplikací. Mohou tak například zakázat průchod http spojení, v němž objeví indikátory, že se nejedná o požadavek na WWW server, ale tunelování jiného protokolu, což často využívají klienti P2P sítí (ICQ, gnutella, napster, apod.), nebo když data v hlavičce e-mailu nesplňují požadavky RFC apod.
- Nejnověji se do firewallů integrují tzv. in-line IDS (Intrusion Detection Systems – systémy pro detekci útoků). Tyto systémy pracují podobně jako antiviry a pomocí databáze signatur a heuristické analýzy jsou schopny odhalit vzorce útoků i ve zdánlivě nesouvisejících pokusech o spojení, např. skenování adresního rozsahu, rozsahu portů, známé signatury útoků uvnitř povolených spojení apod.
- Výhodou těchto systémů je vysoká úroveň bezpečnosti kontroly procházejících protokolů při zachování relativně snadné konfigurace, poměrně vysoká rychlost kontroly ve srovnání s aplikačními branami, nicméně je znát významné zpomalení (zhruba o třetinu až polovinu) proti stavovým paketovým filtrům.
- Nevýhodou je zejména to, že z hlediska bezpečnosti designu je základním pravidlem bezpečnosti udržovat bezpečnostní systémy co nejjednodušší a nejmenší. Tyto typy firewallů integrují obrovské množství funkcionality a zvyšují tak pravděpodobnost, že v některé části jejich kódu bude zneužitelná chyba, která povede ke kompromitování celého systému.

Antivirová ochrana

- počítačový software, který slouží k identifikaci, odstraňování a eliminaci počítačových virů a jiného záškodného software (malware). K zajištění této úlohy používají dvě techniky:
    - Prohlížení souborů na lokálním disku, které má za cíl nalézt sekvenci odpovídající definici některého počítačového viru v databázi.
    - Detekcí podezřelé aktivity nějakého počítačového programu, který může značit infekci. Tato technika zahrnuje analýzu zachytávaných dat, sledování aktivit na jednotlivých portech či jiné techniky.

- Úspěšnost závisí na schopnostech antivirového programu a aktuálnosti databáze počítačových virů, které si programy pravidelně, nejčastěji z Internetu, pravidelně stahují.

Virové slovníky/databáze

- V momentě, kdy antivirový program kontroluje nějaký soubor, kontroluje, jestli se nějáká jeho část kódu neshoduje s některým s virů. Ty jsou zapsány ve slovníku (databázi) virů. Pokud k tomu dojde, má program tyto možnosti:
    - Pokusí se opravit/vyléčit soubor odstraněním viru ze souboru. (pokud je to technicky možné)

- Soubor umístí do karantény (virus se dále nešíří, není povoleno jeho spouštění - jakoby zmrazený)
- Smaže infikovaný sobour

- K dosažení trvalého úspěchu ve středním a dlouhém období vyžaduje virová databáze pravidelné aktualizace, které obsahují informace o nových virech (vetšinou online stahování). Pokud je antivirový program neaktualizovaný, představují viry přinejmenším stejné nebezpečí, jako kdyby antivir v počítači vůbec nebyl. Uživatelé mohou také sami zaslat svůj infikovaný sobour výrobcům antivirových programů, kteří do databáze virů informaci o novém viru začlení.

Nebezpečné chování

- Metoda zjištění nebezpečného chování se oproti virovým databázím nesnaží najít známé viry, namísto toho sleduje chování všech programů. Pokud se některý pokusí zapsat data do spustitelného programu, tak antivirus například označí toto nebezpečné chování a upozorní uživatele, kterého se většinou zeptá, jak chce postupovat dále.
- Výhodu má tento postup zjištění nových virů v tom, že ačkoli je virus zcela nový, neznámý ve virových databázích, může ho snadno odhalit. Nicméně má tato metoda nevýhodu v tom, že hlásí spoustu falešných "nálezů" viru. To může mít za výsledek, že uživatel přestane vnímat ty "pravé" varování. Pokud tedy uživatel automaticky povolí pokračování programu. Je jasné, že v takovém případě antivirus neplní dále svoji funkci, tedy varovat uživatele před možným nebezpečím. S tohoto důvodu tento postup stále více moderních antivirových programů využívá méně a méně.

Další metody

- Určité antivirové programy používají další typy heuristických analýz. Například se může pokusit napodobit začátek kódu každého nového spustitelného souboru tak, že ho systém vyvolá ještě před přenosem do tohoto souboru. Pokud se program chová tak, že použije "samo-modifikační" kód nebo se jeví jako virus (pokud například začne hledat další spustitelné soubory), můžeme předpokládat, že virus nakazil další spustitelné soubory. Nicméně tato metoda může opět hlásit falešné pozitivní nálezy.

- Další metoda detekce virů se týká užití tzv. sandboxu. Sandbox, neboli pískoviště, napodobuje systém a spouští .exe soubory v jakési simulaci. Po ukončení programu software analyzuje sandbox, aby zjistil nějaké změny, ty mohou ukázat právě viry. Tato metoda může taky selhat a to pokud jsou viry nedeterministické a výsledek nastane za různých akcí nebo akce nenastanou při běhu - to způsobí, že je nemožné detekovat virus pouze z jednoho spuštění.

- Existují také antiviry, které varují uživatele před viry na základě toho, jakého typu soubor je.

- Perspektivní metoda, která si obvykle poradí s malware je tzv. "whitelisting". Spíše než vyhledávání jen známého zákeřného softwaru tato technika předchází spouštění všech kódů kromě těch, které byly již dříve označeny jako důvěryhodný administrátorem (uživatelem). Navíc aplikace v počítači, které jsou označeny jako malware, mají automaticky zakázáno spouštění, jakmile nejsou na "whitelist", tedy seznamu povolených programů. Dnes již existuje velké množství aplikací vytvořených velkými organizacemi, které jsou široce používané a "whitelist" je tedy tvořen především administrátory, kteří software rozpoznávají. Možné provedení této techniky zahrnuje nástroje pro automatické zálohy a whitelist procesy údržby.

Blokování nevyžádané pošty

- V posledních letech se stává stále větším problémem nevyžádaná pošta (spam), jejíž objem dosahuje až 70% všech zpráv elektronické pošty. Kromě obtěžování uživatelů začíná navíc být spojena i s bezpečnostními riziky plynoucími z vložených odkazů a s technickými problémy vyplývajícími z velkého zatížení serverů.

Blacklisting

- Blacklisting rozhoduje, zda dopis je nebo není spam, podle adresy odesílatele (která může být zfalšována), nebo lépe podle IP adresy, ze které dopis přišel na cílový SMTP server. Blacklisty obsahující IP adresy, ze kterých bylo zaznamenáno rozesílání spamu, bývají zveřejňovány nejčastěji pomocí systému DNS. Výskyt adresy v

blacklistu může mít za následek buď přímé odmítnutí (nepřevzetí) dopisu ještě během SMTP relace, nebo může být informace z blacklistu použita jako dodatečná informace při následné filtraci podle obsahu.

Greylisting

- Greylisting rozhoduje také podle IP adresy a emailové adresy odesílatele a adresáta, ale dělá to dynamicky. SMTP server, který provozuje greylisting, udržuje databázi, kde pro trojici (IP adresa, odesílatel příjemce) je uvedeno, zda dopis s těmito atributy má být převzat k dopravě, nebo zda jeho převzetí má být dočasně odmítnuto. První dopis je odmítnut a je zaznamenán čas, kdy k tomu došlo. Po určitou dobu (typicky několik desítek minut) pak jsou dopisy s týmiž atributy odmítány. Po uplynutí této doby, pokud se původní SMTP server stále pokouší o odeslání dopisu, je záznam v databázi potvrzen a dopisy jsou naopak přijímány a dopravovány bez zdržení. Po další době (typicky několik málo týdnů) je záznam z databáze odstraněn, takže příští dopis bude opět pozdržen. K odstranění záznamu z databáze dojde také v případě, že v příslušném intervalu, kdy byly dopisy odmítány, se nepokusí původní SMTP server o znovudoručení.

Filtrace podle obsahu dopisu

- Automatické rozpoznávání nemůže z principu fungovat dokonale, protože názor, zda konkrétní dopis je spam je individuální. Přesto filtrování podle obsahu dává použitelné výsledky a hojně se používá. Existují dvě základní metody, některé antispamové programy (např. SpamAssassin) je kombinují.

Filtry založené na pravidlech

- Filtry založené na pravidlech vyhledávají v dopisech rysy, které jsou pro spam typické. Jde jednak o některá slova (např. viagra) a slovní spojení, jednak jsou vyhledávány chyby pro spam typické. Příkladem je třeba datum odeslání v budoucnosti, nedovolené znaky v hlavičce, chybně označený MIME-typ zprávy apod. Za každý rozpoznaný rys je dopisu přiděleno bodové ohodnocení, body se zpravidla sesčítají a pokud součet přesáhne hranici, je dopis pokládán za spam. Rozpoznávané rysy jsou definovány pomocí pravidel, která je třeba pravidelně aktualizovat a přizpůsobovat praktikám spammerů. K vytváření a údržbě souboru pravidel je třeba mít znalosti, není to práce pro běžného uživatele, laika.

Filtry založené na učení (bayesovské)

- Filtry založené na učení (často nazývané bayesovské) využívají triky z oblasti umělé inteligence. V režimu učení se filtru předkládají dopisy explicitně označené jako spam a ham (ne spam), filtr z předložených dopisů extrahuje informace, které si ukládá do databáze. Nejčastěji je dopis rozkládán na slova (popř. jiné úseky textu) a pro jednotlivá slova se statisticky zjišťuje pravděpodobnost, že dopis, který toto slovo obsahuje, je spam. V režimu rozpoznávání pak filtr využívá nashromážděné informace a testovanému dopisu přiřadí pravděpodobnost, že je to spam. Nejčastěji se pro výpočet pravděpodobnosti používá vzorec, který navrhl matematik Bayes. Velkou výhodou je, že filtr může učit i uživatel – laik. Učící se filtry jsou nejúčinnější, učí-li je přímo sami koncoví uživatelé podle svého individuálního názoru, co je spam a co ne. Přesto se bayesovské filtry používají i na serverech, kde učení probíhá pro všechny uživatele serveru společně.

IDS

- Sebelepší firewall a antivirová ochrana však nemůže ochránit počítačovou síť proti všem nebezpečím. Kromě útoků, které přicházejí přes neošetřené zranitelnosti legálních aplikací např. v podobě automaticky se šířících červů, se jedné i o různý spyware a o útoky vedené z vnitřní sítě. Nebezpečí představují i přenosné počítače, s kterými se uživatelé připojují k Internetu i mimo chráněné prostředí počítačové sítě. Ke kontrole se používají systémy pro detekci narušení.

IPsec

- Potřeba bezpečnějšího protokolu v síti Internet, než je protokol IP (Internet Protocol), byla známa už dávno a nový "bezpečný protokol", zvaný IPsec ([RFC 1825], [RFC 1826], [RFC 1827]), je vyvíjen už delší dobu. Jeho implementace se však objevily až v poslední době. IPsec se skládá ze dvou protokolů. Jeden z nich zajišťuje integritu dat a druhý zajišťuje důvěrnost. Některé rysy IPsec (především zajištění integrity) jsou také požadovány v nové verzi IP protokolu IPv6.

- V protokolu IPsec jsou použity dva nové typy paketů. Zajištění integrity je provedeno pomocí autentizační hlavičky (Authentication Header, AH). Při použití AH každý paket obsahuje zvláštní hlavičku, která obsahuje autentizační informace, následované daty samotného protokolu. Autentizační informace se skládají z výsledku klíčovaného kryptografického kontrolního součtu (používá se algoritmus SHA nebo MD5), z bezpečnostních parametrů (Security Parameter Index, SPI) a z ukazatele na hlavičku samotného protokolu vyšší úrovně. Položky, které se v hlavičce protokolu vyšší úrovně během přenosu paketu mění, jako je např. položka TTL, jsou při výpočtu autentizačních informací ignorovány.
- Důvěrnost přenášených dat je zajišťována pomocí protokolu ESP (Encapsulated Security Payload). Stejně jako u AH je k paketu protokolu IP připojena dodatečná hlavička, která obsahuje bezpečnostní parametry, a pak následují zašifrovaná data. ESP používá dva režimy činnosti. V transportním režimu obsahuje ESP paket data některého z vyšších protokolů, jako je například TCP protokol nebo UDP protokol. V tunelovacím režimu obsahuje ESP paket pouze datagram na úrovni protokolu IP.
- Stanice, používající IPsec, si musí spravovat bezpečnostní kontext, který je identifikován hodnotami SPI. Bezpečnostní kontext obsahuje informace o použitém kryptografickém algoritmu, inicializační vektory a kryptografické klíče. Pro IPsec je doporučováno, aby bezpečnostní kontext obsahoval také doby života klíčů a adresy komunikujících stran.
- Protokol IPsec má několik nevýhod a problémů. IPsec neobsahuje žádné automatizované prostředky pro správu kryptografických klíčů. Kryptografické klíče jsou obvykle distribuovány manuálně, což nelze považovat za vyhovující. Při praktickém používání protokolu je třeba použít jak AH, tak ESP. Pokud je použit pouze jeden z těchto protokolů, je IPsec náchylný k některým typům kryptografických útoků. Přes tyto nedostatky je však tento protokol možno při zachování jistých bezpečnostních zásad bezpečně používat.

Secure Socket Layer (SSL)

- Protokol SSL (Secure Socket Layer) se snaží řešit bezpečnost rodiny protokolů TCP/IP na transportní úrovni. Jeho cílem bylo poskytnout bezpečný komunikační kanál mezi dvěma stanicemi sítě Internet na úrovni spojení TCP/IP, který umožní bezpečnou implementaci všech běžných protokolů (např. telnet, ftp, http atd.). SSL může použít pro ustanovení klíče relace nejrůznější algoritmy s veřejným klíčem. Po ustanovení klíče relace je další komunikace zabezpečena zašifrováním některým z mnoha volitelných algoritmů s tajným klíčem. Asi nejznámějším protokolem, který byl implementován pomocí SSL, je protokol HTTPS, používaný například v produktu Netscape.
- Protokol je navržen tak, aby zajišťoval vysokou míru bezpečnosti. Používá kombinaci kryptografie veřejným klíčem s kryptografií tajným klíčem. Kryptografie veřejným klíčem se používá pro ustavení klíčů relace. Těmito klíči relace je pak zabezpečena samotná komunikace pomocí některého blokového nebo proudového kryptografického algoritmu, jako je 3-DES nebo IDEA. Protokol také zajišťuje vzájemnou kryptografickou autentizaci obou komunikujících stran. Při návrhu SSL byla jako jeden z cílů brána v úvahu i interoperabilita a snadnost rozšiřování protokolu o nové kryptografické algoritmy.
- SSL je protokol založený na záznamech, který obsahuje mnoho různých typů zpráv. Základním pojmem SSL je relace, která představuje alespoň jedno spojení mezi klientem a serverem na transportní úrovni. Při ustavení relace se může provádět autentizace uživatele a v rámci jedné relace může být současně otevřeno několik zabezpečených spojení mezi klientem a serverem (například HTTP spojení nebo FTP spojení). Každá relace obsahuje své stavové informace, včetně identifikátoru relace, informací o použité kompresi dat, informací o kryptografických algoritmech a o algoritmech použitých pro zajištění integrity dat. Je zde také uložen klíč relace, sekvenční čítače a inicializační vektory pro kryptografické algoritmy.
- Pokud chce klient použít SSL, nejdříve kontaktuje server a dohodne si s ním parametry relace, jako jsou identifikace relace, verze protokolu a dostupné kompresní a kryptografické algoritmy. Každá ze stran také předá protistraně jedno náhodně vygenerované číslo. Obě strany si vzájemně vymění certifikáty svých veřejných klíčů, které jsou podepsány elektronickým podpisem. Klient vygeneruje klíč relace, zašifruje jej veřejným klíčem serveru a pošle jej serveru spolu s náhodnou výzvou. Server dešifruje klíč relace pomocí svého soukromého klíče a autentizuje se klientovi tím, že mu vrátí jeho náhodnou výzvu zašifrovanou klíčem relace. V současné době používá SSL certifikáty veřejných klíčů podle doporučení ITU X.509. Vzhledem k naprosto nedostatečnému rozšíření infrastruktury adresářových služeb podle X.500 to prozatím přináší jisté obtíže. Pro masové rozšíření SSL je třeba, aby existovalo celosvětové schéma distribuce klíčů. Jedna z možností, jak toto schéma implementovat v prostředí Internetu, je DNS (Domain Name Service). DNS se jeví být vhodnou možností, protože jde o jedinou globální adresářovou službu, která je v současné době masově využívána.

POČÍTAČ TYPU IBM PC A OSTATNÍ HARDWARE

ZÁKLADNÍ ČÁSTI

ZÁKLADNÍ DESKA

- Hlavním účelem základní desky je propojit jednotlivé součástky počítače do fungujícího celku a poskytnout jim elektrické napájení. Postupem času se funkce základní desky rozšiřovala v tom, že sama začínala obsahovat některé součástky počítače, které se dříve musely do ní zapojovat zvlášť.

- Typická základní deska umožňuje zapojení procesoru, operační paměti. Další komponenty (např. grafické karty, zvukové karty, pevné disky, mechaniky) se připojují pomocí rozšiřujících slotů nebo kabelů, které se zastrkávají do příslušných konektorů. Na základní desce je dále umístěna energeticky nezávislá paměť ROM, ve které je uložen systém BIOS, který slouží k oživení počítače hned po spuštění.

- Nejdůležitější integrované obvody jsou zabudovány v čipu, který se označuje jako čipset. Fyzicky může být tento čip buď jenom jeden, nebo dva (v tom případě se označují jako northbridge a southbridge). Ten je určující pro věci typu, jaký procesor a operační paměť je možné k základní desce připojit.

Rozšiřující sloty

- umožňují připojit k počítači další zařízení. Postupem času se jich vyvinul velký počet. Odlišují se zejména přenosovými rychlostmi a schopnostmi napájet připojená zařízení.
    - ISA - dnes již nepoužívané
    - EISA - dnes již nepoužívané
    - VESA - dnes již nepoužívané
    - PCI - běžně používaná
    - AGP - navržená speciálně pro grafické karty. Je výrazně rychlejší než PCI.
    - PCI Express - nástupce PCI. Dosahuje mnohem vyšších přenosových rychlostí. Zařízení určená pro PCI Express nejsou zpětně kompatibilní s PCI

Konektory pro připojení dalších zařízení

- **je možné je dělit na:**
    - interní - nachází se na ploše základní desky a připojovaná zařízení obvykle uvnitř počítačové skříně
        - IDE
        - SATA
        - FLOPPY
        - napájecí konektory
        - konektory pro připojení ventilátorů
        - konektory zvukové karty
        - rozšiřující konektory USB a Fireware
    - externí - nachází se na zadním panelu základní desky
        - USB
        - PS/2
        - Fireware
        - eSATA
        - COM
        - LPT
        - D-SUB = VGA
        - DVI
        - HDMI
        - konektory zvukové karty
        - LAN

Zařízení, která se běžně integrují do základních desek

- Zvuková karta
- Grafická karta - zejména u kancelářských počítačů
- Síťová karta
- Input/Output čip
- řadiče pevných disků

Existuje několik typů, např.:

- ATX - vytvořen firmou Intel v roce 1995. Dnes patří k nejpoužívanějším.
- microATX - zmenšená verze ATX. O 25% kratší. Obsahuje méně rozšiřujících slotů. Dnes patří k nejpoužívanějším zejména v kancelářskcýh počítačích.
- PC/XT - vytvořen firmou IBM. První deska pro domácí počítače. Vzhledem k tomu, že měla otevřenou specifikaci, tak bylo vyráběno mnoho jejích klonů a stala se defacto stadartem.
- AT form factor (Advanced Technology) - vytvořen firmou IBM. Následovník PC/XT a předchůdceATX. Velmi populární za éry procesorů Intel 80386.
- Baby AT - zmenšená varianta AT.
- ETX - používán v embedded počítačích.
- FlexATX
- LPX
- NLX - nízkoprofilová základní deska. Vytvořena v roce 1997.
- BTX (Balanced Technology Extended) - vytvořen firmou Intel. Měl nahradit ATX. Lepší chlazení a napájení. Příliš se neujal.
- Mini-ITX - velmi malé. Malá rozšiřitelnost. Od firmy Via

Northbridge (Severní most)

- je také znám jako systémový řadič. Je jedním ze dvou základních čipů na základní desce. Druhý, se nazývá jižní most a společně je označujeme jako tzv. čipset. Dělení čipsetu na severní most a jižní most je běžné, ačkoli existují i čipy, které obsahují oba najednou za cenu vyšší složitosti při výrobě.

- Severní most zajišťuje komunikaci mezi CPU, pamětí RAM (řadič paměti), AGP portem nebo PCI Express sběrnicí a také zajišťuje spojení s jižním mostem. Některé severní mosty obsahují integrované grafické karty. Protože různé procesory a paměti vyžadují rozdílnou signalizaci, pracuje severní most pouze s jedním nebo se dvěma typy procesorů a zpravidla pouze s jedním typem paměti RAM. Existují čipsety, které podporují dva druhy paměti RAM, které jsou dostupné při přechodu na nový standard.

Význam

- Severní most je na základních deskách základním prvkem, který určuje rychlost, druh procesorů, jejich množství a druh paměti RAM, který bude použit. Ostatní faktory jako jsou regulace napětí a počet konektorů také hrají roli. Prakticky všechny čipsety podporují pouze jednu procesorovou sadu s maximálním množstvím paměti RAM měnící se podle druhu procesoru a typu základní desky. Éra prvních Pentií měla často omezení na 128 MB. Architektura procesoru Pentium Pro umožňovala adresovat více, než 4 GB paměti (36 bitů, což umožňuje adresovat až 64 GB paměti), avšak základní desky takové množství fyzické paměti RAM obvykle nepodporovaly.

- Severní most je obvykle schopen propojení s jedním nebo se dvěma různými jižními mosty, což ovlivňuje výsledné možnosti a nabízené technologie základní desky.

Southbridge (Jižní most)

- je také znám jako vstupně-výstupní řadič (I/O Controller Hub). Čip realizuje pomalejší funkce základní desky v počítačové architektuře se severním a jižním mostem. Jižní most odlišíme od severního snadno tak, že není přímo spojen s procesorem. Severní most realizuje spojení jižního mostu a procesoru.

- Protože jižní most je z hlediska architektury více vzdálen od procesoru, má v typickém počítači na starosti obsluhu pomalejších zařízení. Jižní most je obvykle schopen spolupracovat s několika různými severními mosty, avšak oba čipy musí být pro vzájemnou kompatibilitu navrženy. Průmyslový standard pro komunikaci mezi severním a jižním mostem neexistuje. Tradičně byla pro komunikaci mezi severním a jižním mostem využívána PCI sběrnice, protože však toto řešení vytvářelo z hlediska výkonu úzké místo, většina současných čipsetů využívá pro vzájemnou komunikaci vlastní proprietární rozhraní s vyšším výkonem.

BIOS

- zkratka anglického názvu Basic Input-Output System, který označuje základní programové vybavení osobního počítače. BIOS vytváří základní vrstvu abstrakce pro vyšší programy. Vznikl převážně proto, aby sjednotil rozhraní různých počítačů a zjednodušil psaní operačních systémů. Programový kód BIOSu je obvykle uložen v paměti (integrované na základní desce) typu ROM nebo EEPROM (či modernější flash paměť) s možností přepisu (upgrade). BIOSu je předáno řízení při (re)startu počítače. Po inicializaci systému pak BIOS nahraje zaváděcí část operačního systému do paměti a předá mu řízení. Starší operační systémy jako DOS spoléhaly na BIOS, že obstará většinu vstupně/výstupních úloh v počítači. V současnosti, BIOS ovládá více komplexních funkcí, jako jsou: power management, hot swapping (výměna modulů za provozu) a thermal management (řízení teploty).

Služby BIOSu

- nastavení taktu procesoru a operační paměti, napájecí napětí procesoru
- nastavení cache
- detekce harddisků, CD-ROM, DVD-ROM
- nastavení periferií (integr. zvuková, síťová karta, modem)
- bootovací sekvence (HDD, CD-ROM, USB, LAN, FDD)
- hardware monitor - zobrazuje informace o teplotě procesoru, napětí zdroje, otáčky ventilátorů
- power management - nastavení možností napájení
- další služby - u notebooků např. kalibrace baterií
- služby obsluhy klávesnice
- služby obsluhy řadiče disků (disketa, pevný disk)
- základní služby obsluhy grafické karty (většinou jen pro textový režim)
- výstup na tiskárnu
- řízení komunikačních portů

PAMĚŤ

- V počítači existuje vícero druhů pamětí, které se liší svými vlastnostmi, účelem použití i způsobem připojení k počítači.
  - **Operační paměť** – Rychlá a drahá paměť, převážně typu RWM-RAM (zápis i čtení, s libovolným přístupem), např. DDR RAM. Slouží pro dočasné uchovávání zpracovávaných dat a kódu vykonávaných programů. Její obsah je závislý na elektrickém napájení počítače.
  - **Cache** - Menší rychlá vyrovnávací paměť, jedná se o kompromis mezi rychlostí, kapacitou a cenou. Slouží k uchování nejčastěji používaných datových položek, což k nim značně zkracuje přístupovou dobu. Existuje procesorová cache paměť (nejblíže k procesoru, někdy také přímo jeho součástí), disková cache paměť (buď jako část operační paměti vyrovnávající pomalé a opakované čtení z disku nebo také v dnešní době přímo paměť mikropočítače ovládajícího samotný disk), atd. (Odvozené české sloveso pak zní „kešování“.)
  - **Vnější (externí) paměť** – Bývá nejlevnější, pomalejší a nezávislá na napájení. Používají se typy ROM (jen čtení, např. CD) i RWM (zápis i čtení, např. pevný disk, disketa, magnetická páska), flash paměť a paměťové karty. Obvykle má mnohem vyšší kapacitu než základní operační paměť. Z hlediska přístupu se používají paměti s libovolným přístupem RAM (disk) i se sekvenčním přístupem (páska).
  - **Permanentní pamět, firmware** – Paměti typu RWM, ROM nebo EPROM, které obsahují software a data nutné pro funkci hardware. Zajišťují např. zavedení operačního systému, realizaci síťových protokolů a podobně. Tato pamět obsahuje základní programové vybavení počítače (např. systém BIOS u PC).

Některé videokarty, síťové karty a další hardware pro své řízení poskytují vlastní systémy podobné BIOSu. Některé přenosné (mobilní), jednoúčelové nebo specializované počítače mohou mít celé programové vybavení umístěné v paměti ROM (viz Embedded systém). Z pohledu zbytku počítače bývá obvykle připojena tak, aby počítač po svém zapnutí přečetl první instrukci z této paměti.

- **Virtuální paměť** – Specializovaný termín z oblasti správy operační paměti, kde příslušné moduly operačního systému počítače vytváří tzv. virtuální paměťový prostor, který je několikanásobně větší, než je skutečná velikost operační paměti zařízení. Technicky je tato záležitost obvykle řešena odkládáním dat (tzv. „swapováním“) z operační paměti na pevný disk, který je ale výrazně pomalejší než operační paměť. Proto využití virtuální paměti vede ke zpomalování výpočetních procesů, neboť operace odkládaní dat na disk a jejich zpětné čtení jsou relativně velice pomalé

Dělení OP podle technologie

- Rozlišujeme dvě základní technologie výroby pamětí, zvané SRAM (Static RAM) a DRAM (Dynamic RAM). (Nezaměňujme je s rozdílem SIMM a DIMM.)

- SRAM je od toho, že v ní informaci uchovávají dva vhodně spojené tranzistory jako bistabilní klopný obvod. Tato paměť si informaci uchovává, dokud jí nevypneme napájení, při použití technologie CMOS má minimální příkon a má krátkou přístupovou dobu.

- Podstatou DRAM je kondenzátor (nabitý/nenabitý). Základní paměťová buňka je založena na parazitní (Müllerově) kapacitě řídícího tranzistoru. Je levnější, snadnější na výrobu, ale má nevýhody: musí se čas od času obnovovat (zajišťuje řadič paměti), po přečtení se vymaže, musí se tedy obnovit po každém čtení (proto je 1,5x delší než zápis).

Dělení DRAM modulů do počítače

- DIPP,DILL,SIPP
- SIMM - (72pin, 30pin) - (Single Inline Memory Module)
- DIMM - 3,3V a 5V - (Dual Inline Memory Module) - Jedná se defacto o dva moduly SIMM integrované na jedné desce. Důvodem je obsazení celé šířky sběrnice.
  - SDR – (Single Data Rate) někdy mylně označovány jako SDRAM, starší typ pamětí typu DIMM (3,3, nebo 5 V), 168 pinů, kapacity od 32 MB do 512 MB, rychlost od 66 MHz do 133 MHz, dva zářezy jako pojistka.
  - DDR – (Double Data Rate) novější typ pamětí typu SDR, 3,3 V, 184pinů (ale jiné umístění zářezů, místo dvou jen jeden), kapacity od 128 do 2048 MB. Vylepšení je v tom, že přenáší data na náběžné i koncové hraně taktovacího impulsu.
  - DDR2 – "nejnovější" typ pamětí, podobné jako DDR, mají vyšší frekvence, stávají se v současné době standardem. Nevýhodou DDR2 jsou vyšší cas latence, než u DDR.
  - DDR3 - momentálně absolutně nové paměti, kvůli vysoké ceně a malé podpoře u motherboardů zatím příliš nerozšířené.
- SO-DIMM – paměti používané pro notebooky, 72pin, nebo 144 a 200 pin,
- RIMM – Rambus DRAM. Oproti DDR DIMMu má jen 16 bitů přenosové šířky, ale zato je výrazně rychleji taktován.

PROCESOR

- Procesor (CPU – Central Processing Unit) je ústřední výkonnou jednotkou počítače, která čte z paměti instrukce a na jejich základě vykonává program. Pokud bychom přirovnali počítač např. k automobilu, postavení procesoru by odpovídalo motoru. Protože procesor, který by vykonával program zapsaný v nějakém vyšším programovacím jazyku, by byl příliš složitý, má každý procesor svůj vlastní jazyk - tzv. strojový kód, který se podle typu procesoru skládá z jednodušších nebo složitějších instrukcí. Pod pojmem procesor se dnes téměř vždy skrývá elektronický integrovaný obvod, i když na samých počátcích počítačové éry byly realizovány procesory i elektromechanicky.

ŃAPÁJECÍ ZDROJ

- Počítačové skříně AT měly zapínací tlačítko napojené přímo do silové části napájecího zdroje pomocí 4 žilového kabelu a uzemnění kovové kostry skříně. Byl to klasický dvoupólový vypínač spínající fázi a střední pracovní vodič napájecího napětí (v Evropě 230 V). U počítačové skříně formátu ATX není zdroj spojen přímo se zapínacím tlačítkem. To umožňuje zapínání počítače i jinými způsoby (Wake on LAN, Wake on RING, klávesnicí nebo myší). Přesto má mnoho napájecích zdrojů ATX klasický vypínač na své zadní straně. Tím se počítač skutečně vypne a softwarové zapnutí pak není možné. Pokud je tento vypínač zapnutý, počítač stále spotřebovává energii, i když vypadá jako vypnutý. Tento režim se nazývá "soft-off" nebo "stand-by" a slouží pro vzdálené zapnutí přes Wake on RING nebo Wake on LAN. Běžně se ale počítač zapíná tlačítkem umístěným na přední stěně skříně.

- Konektor napájecího zdroje zapojovaný do základní desky se změnil. Starý AT zdroj měl dva podobné konektory, které se daly lehce zaměnit a bylo tak možné základní desku zničit. ATX používá jeden velký konektor, takže je instalace snadnější. Nový konektor obsahuje napětí 3.3 voltů, takže si toto napětí nemusí základní deska stabilizovat sama. Některé poslední typy základních desek AT obsahovaly jak konektory pro zdroj AT, tak i pro nový ATX. S nástupem procesoru Pentium 4 byl přidán pomocný dvanáctivoltový 4pinový konektor (dva žluté vodiče a dva černé vodiče). Později ho začaly používat i základné desky s procesory Athlon XP a Athlon 64. Některé počítačové systémy vyšší kategorie používají ještě další konektory pro větší výkon. Některé velmi výkonné grafické karty vyžadují extra napájení; to bývá řešeno 4pinovým konektorem molex, podobným jako u napájení pevných disků. Moderní grafické karty na sběrnici PCI Express využívají jeden nebo více 6pinových konektorů pro dostatečné napájení.

ZOBRAZOVACÍ SUBSYSTÉM

GRAFICKÁ KARTA

- Grafická karta nebo také videoadaptér je součást počítače, která se stará o grafický výstup na monitor, TV obrazovku či jinou zobrazovací jednotku. V případě, že grafická karta obsahuje tzv. VIVO (video - in a video-out), umožňuje naopak i analogový vstup videosignálu např. při ukládání videosouborů z kamer, videopřehrávačů apod. Dříve byla grafická karta nedílnou součástí základní desky, dnes jsou grafické karty oddělené a připojené do počítače pomocí některého typu sběrnice. Grafická karta samozřejmě může být i integrovaná na základní desce počítače, v tomto případě se však jedná o tzv. low-end desky nebo desky nižší střední třídy. Pokud je grafická karta integrovaná na základní desce, lze ji vypnout a nahradit grafickou kartou, která se zasune do příslušné pozice na desce. Grafické karty jsou rok od roku složitější a výkonnější, a jelikož již dlouhou dobu obsahují vlastní mikroprocesor (GPU – graphics processing unit), paměti i sběrnice, daly by se označit za „počítače v počítači“.

Součástky v grafické kartě

- GPU - grafický procesor. Výpočetní jádro grafické karty.
- Paměť - zde jsou ukládány informace nutné pro grafické výpočty. Pokud je grafická karta integrovaná na základní desce, tak používá Operační paměť celého počítače. Pokud ne, tak má vlastní paměť, nejčastěji nějaký typ GDDR (GDDR 1, 2, 3, 4).
- Firmware - čip, který ho obsahuje.
- RAMDAC - převodník signálu mezi digitálním, se kterým pracuje grafická karta na analogový, kterému rozumí zobrazovací zařízení.

Výstupy

- VGA - Analogový grafický výstup (používán starými monitory CRT a kompatibilními zařízeními). Možno převést redukcí z digitálního výstupu DVI
- DVI - Digitální grafický výstup (používáno většinou LCD panelů, projektory a novějšími zobrazovacími zařízeními)
- S-video
- Component video
- Composite Video
- HDMI - Výstup na zobrazovací zařízení (nejčastěji televize) s vysokým rozlišením. Konektor HDMI získáte většinou připojením redukce do konektoru DVI.

Grafické módy (PC):

1981

- MDA
- CGA 4 barvy, 320×200´

1982

- Hercules 2 barvy, 720×348

1984

- EGA 16 barev, 640×350
- EGC 16 barev, 640×400
- MCGA
- VGA 256 barev, 320×200 nebo 16 barev, 640×480
- SVGA až 24bitová barevná hloubka, až 4096×4096

Novější

- Veškeré současné grafické adaptéry (pro obyčejné spotřebitele) podporují 32 bitovou barevnou hloubku.

MONITOR

- Monitor je výstupní elektronické zařízení sloužící k zobrazování textových a grafických informací.
- Na rozdíl od televizoru není obvykle vybaven vysokofrekvenčním vstupním obvodem (tunerem). Signál je do něj přenášen buď analogově, nebo digitálně. Monitory můžeme podle používaných technologií rozdělit na několik skupin:
    - CRT (klasická lampová obrazovka),
    - LCD (tekuté krystaly),
    - plazmová obrazovka,
    - a další, méně obvyklé typy (LED, projektory).

- Monitor je většinou propojen s grafickou kartou (v případě použití s počítačem), ovšem může být připojen i k dalším zařízením nebo do nich přímo integrován (PDA). Monitor může být také součástí odděleného počítačového terminálu.

CRT

- Obraz se vytváří pomocí svazku 3 elektronových paprsků (všechny paprsky stejné, neexistují žádné barevné elektrony)
- Barevné body (RGB) vznikají po dopadu elektronového paprsku na daný fosforový bod (luminofor)
- Barevné CRT obrazovky potřebují tzv. masku (delta, trinitron, štěrbinová)
- Při výrobě se pro nanášení fosforu příslušné barvy (luminoforů) využívá fotografická cesta - nanese se všude, rozsvítí se patřičný paprsek a projde se celá obrazovka (paprskem). Poté se vypláchne, neosvícená místa se vyplaví. Proces se opakuje pro každou barvu.

LCD

- Displej z tekutých krystalů (anglicky Liquid crystal display, zkratkou LCD) je tenké a ploché zobrazovací zařízení skládající se z neomezeného počtu barevných nebo monochromatických pixelů seřazených před zdrojem světla nebo reflektorem. Vyžaduje poměrně malé množství elektrické energie; je proto vhodné pro použití v přístrojích běžících na baterie.

- Každý pixel LCD displeje se skládá z molekul tekutých krystalů uložených mezi dvěma průhlednými elektrodami a mezi dvěma polarizačními filtry, přičemž osy polarizace jsou na sebe kolmé. Bez krystalů mezi filtry by bylo světlo procházející jedním filtrem blokováno filtrem druhým. Molekuly tekutých krystalů jsou bez elektrického proudu v chaotickém stavu. Elektrický proud způsobí, že se molekuly srovnají s mikroskopickými drážkami na elektrodách. Drážky na elektrodách jsou vzájemně kolmé, takže molekuly se srovnají do spirálové struktury (onen krystal). Světlo procházející filtrem je při průchodu tekutým krystalem rotováno, což mu umožňuje projít i druhým filtrem. Polovina světla je absorbována prvním polarizačním filtrem, kromě toho je ale celá sestava průhledná.

- V okamžiku vpuštění elektrického proudu do elektrod jsou molekuly tekutých krystalů taženy rovnoběžně s elektrickým polem, což snižuje rotaci vstupujícího světla. Pokud nejsou tekuté krystaly vůbec stočené, procházející světlo bude polarizováno kolmě k druhému filtru, a tudíž bude úplně blokováno a pixel se bude jevit jako nerosvícený. Pomocí kontroly stočení krystalů v pixelu lze kontrolovat množství procházejícího světla, a tudíž i celkovou svítivost pixelu.

- Je obvyklé srovnat polarizační filtry tak, že bez přívodu elektrické energie jsou pixely průhledné a až při průchodu elektrického proudu se stanou neprůhlednými. Někdy je ovšem pro dosažení speciálních efektů uspořádání opačné.

- Elektrické pole potřebné pro rychlé srovnání molekul tekutých krystalů je ale také dostatečné pro jejich úplné „vystrčení“ z pozice, což poškozuje displej. Tento problém je vyřešen použitím střídavého proudu.

- Pro finanční úsporu v elektronice jsou LCD displeje často multiplexovány. V multiplexovaném displeji jsou elektrody na jedné straně displeje seskupeny (typicky po sloupcích) a každá skupina má svůj zdroj napětí. Na druhé straně jsou elektrody také seskupeny (typicku po řádcích), přičemž každá tato skupina má svůj spotřebič

napětí. Skupiny jsou navrženy tak, aby každý pixel měl unikátní kombinaci zdroje a spotřebiče. Elektronika pak řídí zapínání zdrojů a spotřebičů.

- Důležité faktory pro hodnocení LCD monitoru jsou rozlišení, rozměry zobrazované plochy, doba odezvy, typ mřížky (pasivní nebo aktivní), pozorovací úhel, podpora barev, jas, kontrast, poměr stran a vstupní porty (DVI nebo VGA).

Barevné displeje

- V barevných LCD displejích je každý pixel rozdělený do tří subpixelů, a to červeného, zeleného a modrého (tedy RGB). Svítivost každého pixelu je možné kontrolovat nezávisle na ostatních, díky tranzistorům; jejich kombinací lze pak dosáhnout milionů barev. Starší CRT monitory používaly podobnou metodu.

- Barevné složky (subpixely) je možné sestavit v různých geometriích, v závislosti na použití monitoru. V případě, že software zná geometrii monitoru, je možné zvýšit viditelné rozlišení pomocí metody subpixel rendering. Tato metoda je obzvláště praktická pro vyhlazování písma.

- LCD disleje rozdělujeme na pasivní STN (Supertwist Nematic) a aktivní TFT (Thin-Film Transistors).

DISKOVÝ SUBSYSTÉM

PEVNÝ DISK

- Pevný disk (anglicky hard disk drive, 'HDD') je zařízení, které se používá v počítači k trvalému uchování většího množství dat.
- Hlavním důvodem velkého rozšíření pevných disků je velmi výhodný poměr kapacity a ceny disku, doprovázený relativně vysokou rychlostí blokového čtení. Data se při odpojení disku od napájení neztrácí a počet přepsání uložených dat jinými je prakticky neomezena.
- Dnes se pevné disky kromě počítačů běžně používají i ve spotřební elektronice – MP3 přehrávače, videorekordéry apod.

Diskové plotny

- Data jsou na pevném disku uložena pomocí magnetického záznamu. Disk obsahuje kovové nebo skleněné desky - tzv. plotny pokryté tenkou magneticky měkkou vrstvou (viz hysterezní křivka). Hustota datového záznamu se udává jako počet bitů na měrnou jednotku plochy disku. Plotny jsou neohebné (odtud pevný disk), narozdíl od ohebných ploten v disketách - floppy disk. Ploten bývá v dnešních discích často několik (1 – 3, výjimečně až 5). Disk se otáčí na tzv. vřetenu poháněném elektromotorem.

- Plotny se rychle otáčejí (to je obvykle uváděná „rychlost disku“, udává se v otáčkách za minutu). V běžných starších disích plotny rotují rychlostí 5 400 ot/min, rychlejší mají pak rychlostmi 7 200, 10 000 a u některých špičkových disků i 15 000 ot/min. Při 7 200 ot/min je obvodová rychlost plotny zhruba 30 km/h. Disky v noteboocích mají většinou 4 200 otáček/min (občas jen 3 600 ot/min ale někdy mají i 5400 ot/min). Otáčky disku společně s hustotou záznamu a rychlostí vystavovacího mechanismu určují celkový výkon disku.

- V současné době mají skoro všechny disky plotny o průměru 3,5 palce (tj. 8,9 cm), v noteboocích jsou menší varianty 2,5", které mají nižší otáčky nižší (zřejmě kvůli kroutivému momentu a nižšímu množství energie nutnému k roztočení disku). Malý disk Microdrive vyvinutý firmou IBM a používaní ve spotřební elektronice využívá 1" plotny. Ve starších typech počítačů PC XT byly disky s plotnami o průměru 5,25".

Hlavy

- Čtení a zápis dat na magnetickou vrstvu zajišťuje čtecí a zápisová hlava (vpravo). Dříve se na čtení používaly magnetodynamické hlavy, nyní se používá krystal měnící vodivost podle mag. pole. Na jednu plotnu jsou dvě hlavy, protože jsou data z obou stran, strana plotny, na které je magnetický záznam se nazývá povrch. Hlava „plave“ ve vzduchoprázdnu těsně nad povrchem, ve vzdálenosti řádově mikrometrů (10-6m).

- Zařízení, které vystavuje čtecí hlavy na správnou pozici nad povrchem, se nazývá vystavovací mechanismus. Ve starších discích se pro vystavování hlav používá přesný krokový motor. Ten se „odvaluje“ za pomocí ocelového pásku po „patce“, která je spojena s hlavami. V novějších discích se používá rychlejšího lineárního motoru (elektromagnetu), hlavy se vystavují v závislosti na el. proudu, který protéká elektromagnetem s nimi spojeným a uloženým v silném magnetickém poli jiného permanentního magnetu, díky tomu je samoparkovací. Z pevných disků se tedy dají demontovat velmi silné a křehké magnety ze slitin gadolinia.

- Operace nutné pro čtení nebo zápisu dat
  1. vystavit čtecí hlavu na správnou pozici
  2. vyčkat na utlumení rozkmitu způsobeném setrvačností hlav (vystavení trvá řádově milisekundy [ms])
  3. vyčkat na pootočení disku na místo, od kterého se začne čtení nebo zápis (tzv. latence)

- Průměrný (střední) čas, za který je disk připraven číst nebo zapisovat data se označuje jako přístupová doba. V současné době je okolo 8,5 ms, u disků s 15 000 ot./min je to pod 4 ms.

- Při vystavení hlav na požadovanou pozici je možné číst a zapisovat data ze všech povrchů bez pohybu hlav.

Řadič

- Řadič najdeme přišroubovaný na spodní straně disku. Vysílá informace mechanice pohybující hlavami, kde jsou dané data. Podle toho se mechanika pohybuje.

Deska rozhraní

- Deska rozhraní zajišťuje připojení disku k základní desce (ta není umístěna v pouzdře disku, ale zpravidla ji najdeme integrovanou na základní desce.

Organizace dat

- Data jsou na povrchu pevného disku organizována do soustředných kružnic zvaných stopy, každá stopa obsahuje pevný anebo proměnný počet sektorů z důvodu efektivnějšího využití povrchu - povrch je většinou rozdělen do několika zón, každá zóna má různý počet sektorů na stopu. Sektor je nejmenší adresovatelnou jednotkou disku, má pevnou délku (donedávna 512 byte na sektor, nyní by se již po domluvě výrobců měly vyrábět disky s 4 KB na sektor). Pokud disk obsahuje více povrchů, všechny stopy, které jsou přístupné bez pohybu čtecí hlavičky, se nazývají cylinder (válec). Uspořádání stop, povrchů a sektorů se nazývá geometrie disku.
- Adresa fyzického sektoru na disku se skládá z čísla stopy (cylindru), čísla povrchu a čísla sektoru.
- Stejným stopám na různých površích se říká cylindr

- Pro přístup k datům disku se používá starší metoda adresace disku CHS, která disk adresuje podle jeho geometrie (odtud název CHS - cylinder, head, sector). Hlavní nevýhodou je u osobních počítačů IBM PC omezená kapacita takto adresovaného disku (8GB) a nutnost znát geometrii disku. U disků vyšších kapacit na rozhraní ATA, již neodpovídá zdánlivá geometrie disku skutečné fyzické implementaci (viz CHS).
- Novější metoda pro adresaci disku je (u rozhraní ATA) LBA, sektory se číslují lineárně. Není třeba znát geometrii disku, max. kapacita disku je až 144 PiB (144 miliónů GiB). Rozhraní SCSI používá lineární číslování sektorů disku již od své první verze. Ostatní novější rozhraní již převážně metodu jako je LBA používají.

Teplotní kalibrace (therminal calibration)

- U velkokapacitních disků s velkou hustotou stop je nutné umístit hlavy nad stopy s velkou přesností. Během práce se však disk ohřeje a vystavování hlaviček by vlivem teplotních výkyvů nebylo přesné. Proto disk pravidelně kontroluje polohu hlavičky nad stopou a provádí případné korekce její polohy.

Přístupová doba (access time)

- Vyjadřuje rychlost, s níž disk vyhledává data. Je součtem dvou časů: doby vystavení a doby čekání. Její hodnota se pohybuje okolo 10ms.

Doba vystavení (seek time)

- Je časem nutným k pohybu hlav nad určitou stopou. Hlavy většinou "přelétávají" pouze několik stop (málokdy celý disk), a tak je doba vystavení definována jako jedna třetina času potřebného pro pohyb přes celý disk.

Doba čekání (rotary latency period)

- I přesto, že hlava se dostane nad správnou stopu, nemůže ještě začít se čtením. Musí totiž počkat, až se pod ni dotočí ten sektor, v němž má začít číst data. Doba čekání záleží na náhodě, ale jako technická hodnota se uvažuje jedna polovina otáčky disku.

ŘADIČE

- Pro připojení pevných disků k počítači jsou používána různá rozhraní. V osobních počítačích je dnes nejrozšířenější ATA (Advanced Technology Attachment, což je v podstatě synonymum názvu IDE Integrated Drive Electronics), které se někdy pro odlišení od SATA nazývá PATA - „paralelní ATA“. ATA rozhraní je relativně jednoduché a tedy i levné. ATA rozhraní má max. teoretickou přenosovou rychlost okolo 1Gb/s = 133MB/s, což je při jednom připojeném disku dostačující, protože pevný disk obvykle dokáže vysílat data pouze rychlostí 640Mb/s = 80MB/s. Na jeden ATA kabel se ovšem dají připojit disky dva a pak se již přenosová rychlost ATA stává úzkým hrdlem.

Přenosové módy ATA (paralelní ATA)

Přenosový mód	Standard	Přenosová rychlost
PIO 0	ATA (IDE)	3.3 MB/s
PIO 1	ATA (IDE)	5.2 MB/s
PIO 2	ATA (IDE)	8.3 MB/s
PIO 3	ATA2 (EIDE)	11.1 MB/s
PIO 4	ATA2 (EIDE)	16.7 MB/s
UltraDMA 33	ATAPI-4 (UltraATA-33)	33 MB/s
UltraDMA 66	ATAPI-5 (UltraATA-66)	66 MB/s
UltraDMA 100	ATAPI-6 (UltraATA-100)	100 MB/s
UltraDMA 133	ATAPI-7 (UltraATA-133)	133 MB/s

- Nově se prosazuje sériová verze Serial ATA (SATA). Výhodou SATA je o něco vyšší rychlost; vyšší inteligence řadiče, umožňující optimalizaci datových přenosů (NCQ); možnost připojování disků za chodu systému a menší rozměry kabelů, které nebrání toku vzduchu ve skříni a tedy zlepšují chlazení počítačů. Z hlediska operačního systému je řízení disků pomocí tohoto rozhraní shodné s paralelní ATA.

Přenosové módy SATA

Přenosový mód	Standard	Přenosová rychlost
SATA 1	SATA (SATA/150)	150 MB/s
SATA 2	SATA II (SATA/300)	300 MB/s

- Pro dosažení vyššího výkonu (především počtu operací za sekundu) používá rozhraní SCSI (čti [skazi], zkratka Small Computer System Interface) nebo novější rozhraní Fibre Channel. Na jedno rozhraní (resp. kabel) je možné připojit více periférií. SCSI navíc podporuje periférie různých typů. Max. délka propojujícího kabelu je u SCSI obecně větší něž u standardu ATA/IDE. SCSI rozhraní je mnohem sofistikovanější než ATA/IDE, což samozřejmě znamená vyšší cenu jak radičů v počítači tak i samotných pevných disků a proto je používáno zejména u serverů a pracovních stanic.

Standardy

Rozhraní	Sběrnice	Přenosová rychlost
SCSI	8bit	5 MB/s
Fast SCSI	8bit	10 MB/s
Wide SCSI	16bit	10 MB/s
Ultra SCSI	8bit	20 MB/s
Ultra Wide SCSI	16bit	40 MB/s
Ultra 2 SCSI	8bit	40 MB/s
Ultra 2 Wide SCSI	16bit	80 MB/s
Ultra 3 SCSI	16bit	160 MB/s
Ultra 320 SCSI	16bit	320 MB/s
SAS SCSI	32bit	375 MB/s (v každém směru)

- Pro externí disky (umístěné mimo skříň počítače) se používají rozhraní USB (Universal Serial Bus) či FireWire (IEEE 1394)

VSTUP DAT

KLÁVESNICE

Klávesy s galvanickou vazbou

- U tohoto typu je použit místo speciální pružinky sloupek pěnové gumy. Stlačená guma působí proti štítku z plastu, na jehož spodní straně je kovová ploška spojená s elektronickou deskou. Na elektronické desce je obdobný kovový plíšek. Toto spojení obou kovů vyvolá průchod signálu při stlačené klávese. Po uvolnění se pěnová guma vypruží do své původní velikosti a kontakt obou kovových plošek je rozpojen.

Klávesy s kapacitní vazbou

- U klávesy s kapacitní vazbou tvoří prostředníka mezi plastovým krytem klávesy a elektronickou deskou speciální pružinka, která se po stlačení přiblíží k dvěma podložkám ze směsi cínu, niklu a mědi. V těchto ploškách je rozdílná polarita a napětí stejné velikosti. Při stlačení se pružinka klávesy zachová jako kondenzátor a vznikne malý, ale zjistitelný proud. Po uvolnění klávesy se pružinka opět vrátí na své místo a proud zaniká.

Klávesy s plošným spojem

- Po celé ploše pod klávesami je rozprostřen gumový plát s vylisovanými výstupky pro jednotlivé klávesy. Po stisku klávesy se prohne vylisované místo na gumovém plátu a dotkne se elektronické desky. Tím se zaručí propojení obvodu a předání informace počítači. Zpět klávesu vypruží onen zmiňovaný gumový výlisek.

Kódy kláves

- Bez ohledu na to, jakým nápisem je opatřena horní plocha klávesy, způsobí její stisknutí proudovou změnu v obvodech příslušejících této klávese. Klávesy jsou zapojeny v matci.
- Mikroprocesor vestavěný do klávesnice, jako např. Intel 8048, neustále sleduje obvody vedoucí ke klávesám. Zajímá ho zvětšení nebo zmenšení proudu v obvodu stisknuté klávesy. Zjištěním změny proudu může procesor poznat jednak to, kdy byla klávesa stisknuta a pak, kdy byla opětovně uvolněna. Každá klávesa má jednoznačně stanovený kód, což platí i u kláves, které mohou uživatelům připadat identické. Procesor umí například rozlišit mezi levou a pravou klávesou přeřaďovače. Aby mohl být rozlišen skutečný signál od náhodného proudového kmitu, opakují se vyhledávací cykly mnohosetkrát za sekundu. Pouze signály zjištěné ve dvou nebo více cyklech po sobě jsou procesorem zpracovány.
- V závislosti na tom, ze kterého obvodu přijde do mikroprocesoru signál, vygeneruje mikroprocesor číslo, kterému říkáme kód klávesy. Pro každou klávesu existují dva kódy. Jeden pro okamžik, kdy je klávesa stisknuta a druhý, když je klávesa opět uvolněna. Procesor uloží číslo do vlastní paměti klávesnice a ta je zapíše do spojovacího portu, který může přečíst BIOS počítače. Procesor pak kabelem klávesnice vyšle signál přerušení, aby tak informoval počítač, že pro něj má kód klávesy. Přerušení sdělí počítači, aby nechal všeho, co právě dělá, a obrátil svoji pozornost na požadovanou službu.
- Systém BIOS přečte kód klávesy z portu klávesnice a do klávesnice odešle signál, který znamená, že může ze své paměti zpracovaný kód klávesy odstranit.
- Jestliže kód klávesy patří jedné z rozšiřujících kláves nebo některé z kláves, které jsou pokládány za zvláštní rozšiřující klávesy - Ctrl, Alt, Num Lock, Caps Lock, Scroll Lock nebo Insert - změní BIOS obsah dvou byte ve zvláštní oblasti paměti, ve které si uchovává informaci o stisku těchto kláves.
- Pro všechny ostatní klávesy BIOS tyto dva byte testuje, aby zjistil polohu rozšiřovacích a přepínacích kláves. V závislosti na jejich stavu BIOS přetransformuje kód klávesy do kódu ASCII, používaného na osobních počítačích, který představuje buď nějaký znak nebo zvláštní kód funkční klávesy nebo klávesu pro pohyb kurzoru. Velká a malá písmena mají různé kódy ASCII. Ve všech případech umístí BIOS ASCII nebo zvláštní kód do své vlastní pracovní paměti, kde si jej ihned po ukončení operace může přečíst operační systém nebo aplikační software.

MYŠ

V současnosti se používá několik typů počítačových myší:

- Kuličková – (starší typ) vespodu myši je vložena kulička, která se pohybem myši odvaluje a přenáší tak svůj pohyb na dvě hřídele (vertikální a horizontální pohyb). Narozdíl od optických myší není tak přesná a spolehlivě funguje pouze na některých površích a její mechanismus se rychle zanáší prachem, a tudíž vyžaduje pravidelné čištění.

- Optická – pracuje na principu optického snímání povrchu pod myší. V myši je umístěn optický snímač (CCD či CMOS prvek s maticí o velikosti několik desítek bodů), který snímá obraz v podobném rozlišení, jaké má například ikona programu. Rychlost snímání je zhruba 1000 – 6000 vyhodnocených obrazů za sekundu. Vyhodnocení polohy provádí zabudovaný procesor. K osvětlení plošky snímané senzorem se využívá červená LED dioda, jejíž použití je nejlevnější. Principiálně však není vyloučena ani jiná barva. Optická myš pracuje spolehlivě na téměř každém povrchu kromě zrcadla. Problém se správným vyhodnocením polohy může nastat také při rychlém sledu pohybů (na rozdíl od kuličkové myši).

- Laserová - Jedná se o typ optické myši s velmi přesným snímačem. Nepoužívá běžné světlo jako klasické optické myši, ale laserový paprsek. Rozlišení dosahuje běžně 2000dpi. Tuto myš používají grafici a hráči PC her. Je dražší než klasická optická myš. Vyrábí se přibližně od roku 2006.

Způsoby připojení k počítači

- Počítačovou myš je možné k počítači připojit pomocí šestikolíkového konektoru mini-DIN, známého jako PS/2, nebo přes USB - což je modernější a rychlejší. Zejména v minulosti se pak používaly myši pro sériový port. V poslední době se také stále více rozšiřují bezdrátové myši.

TRACKBALL

- je vstupní zařízení podobné myši.
- Jde jednoduše o kuličku umístěnou v podložce, jíž se dá pohybem prstů pohybovat - kulička je navrchu, nikoliv zespodu jako v případě myši. Bývá buď samostatně nebo zabudován v notebooku.
- Trackball je nasazován v případě, kdy standardní myš není vhodná (průmyslové použití, veřejné informační stánky), nebo pro odvětví, kde je potřeba velmi přesné polohování kurzoru. Například pro použití v počítačové grafice, aplikacích typu CAD, nebo DTP. Naopak se příliš nehodí pro rychlý pohyb s vysokou přesností, který je požadován například v počítačových hrách.

TRACKPOINT

- polohovací zařízení u některých přenosných počítačů. Poprvé se na světle světa objevil s notebookem IBM ThinkPad 700 (rok 1992), díky němuž se povedlo ušetřit místo na klávesnici. Nyní u některých notebooků stále přetrvává, ale bývá doplněn novějším touchpadem. V podstatě se jedná o malý joystick umístěný mezi klávesami G, H a B. Jeho nakláněním do stran pohybujete kurzorem myši po pracovní ploše. Samotný trackpoint nemá funkci tlačítek. Ta jsou umístěna pod klávesnicí. Spolu s touchpadem nahradil trackball.

TOUCHPAD

- vstupní zařízení běžně používané u laptopů. Jeho účelem je pohybovat kurzorem po obrazovce podle pohybů uživatelova prstu. Touchpad je náhrada za počítačovou myš. Touchpady se vyrábějí v různých velikostech, ale jen zřídka větší než 50cm2.
- Touchpady většinou pracují na principu snímání elektrické kapacity prstu nebo kapacity mezi senzory. Kapacitní senzory obvykle leží podél horizontální a vertikální osy touchpadu. Poloha prstu je pak zjištěna ze vzorků kapacity z těchto senzorů. To je důvod, proč touchpad nereaguje na špičku tužky nebo dokonce na prst s rukavicí. Vlhký prst může být touchpad problematický, protože se nelze spolehnout na výsledky měření.
- U touchpadu se obvykle nacházejí tlačítka podobně jako na počítačové myši. U některých touchpadů (v závislosti na modelu a ovladači) je možné také kliknout klepnutím prstu na touchpad a přesouvat objekty klepnutím následovaným plynulým pohybem.

- Některé touchpady také mají „hotspoty“, tedy místa, která mohou mít jiný účel než kliknutí. Například posouvání u pravé strany může kontrolovat posuvník a rolovat aktivní okno vertikálné. Pohyb v dolní části touchpadu pak může rolovat okno horizontálně. Některé touchpady také mohou emulovat více myších tlačítek klikáním do rohů nebo klikáním více prsty naráz.

SKENER

Nasvícení

- Dokument je nutné ve skeneru nejdříve nasnímat. Základním požadavkem je dobré a rovnoměrné osvícení předlohy po celé její ploše. To zajišťovala u plošných skenerů donedávna tzv. "Chladná katodová lampa", neboli zářivka. Výhodou tohoto řešení je vysoká intenzita produkovaného světla, nevýhodou pak nerovnoměrné osvětlení (nejvíce světla je vyzařováno uprostřed). Aby byl tento nedostatek v co možná největší míře odstraněn, je zářivka obvykle doplněna systémem zrcadel, které vrací odražené světlo na místo, kde je ho potřeba. Novější řešení u tzv. CIS technologie využívá řadu luminiscenčních LED diod. Všechny použité diody jsou přirozeně stejné a to zaručuje maximální možnou stejnoměrnost osvětlení po celé šíři snímaného dokumentu. Osvětlovací a snímací mechanizmus se postupně posouvá po předloze a snímá jeden řádek za druhým.

CCD

- Kombinace zářivka - optická soustava - snímací prvek CCD je klasická technologie, nazývaná CCD. Skenery vybavené tímto způsobem snímání jsou trochu dražší, choulostivější na poškození, ale mají lepší barevnou citlivost.

CIS

- Nejedná se o nic jiného, než o dvě řady diod, jednu vysoce svítivých LED diod a řadu diod snímacích. Kladem jsou nižší výrobní náklady a tudíž nižší cena "CIS" skenerů, menší rozměry a větší odolnost. Nevýhodou je naopak nižší svítivost a citlivost (to se projeví například při snímání jemných barevných odstínů nebo třeba u silnější rozevřené knihy ve hřbetu).

Převod obrazové informace na elektronickou

- Snímač (CCD nebo CIS) pracuje tak, že intenzita světla, které dopadá na jeho jednotlivé buňky je přeměněna na elektrický náboj o různé síle. Každý bod elektronické podoby obrazu je složen ze tří informací - intenzity tří základních barev - R (červená), G (zelená) a B (modrá). Každý bod snímané předlohy je tedy měřen třemi buňkami snímače - každá buňka pomocí speciálních filtrů vyhodnocuje jednu z uvedených barevných složek bodu. V plošných skenerech jsou použity tzv. řádkové CCD nebo CIS snímače, použitý snímač tedy určuje maximální možné optické rozlišení skeneru.
- Kvalita skeneru je přímo závislá na kvalitě použitého snímače a počtu jeho buněk. v současné době většina plošných skenerů používá snímače s rozlišením 300 nebo 600 dpi. Označení dpi (dot per inch) udává, kolik bodů je snímač schopen změřit na vzdálenosti jednoho palce (asi 2,5 cm). CCD snímač s rozlišením 600 dpi má tedy 1800 buněk (každý bod je snímán třikrát) na každých přibližně 2,5 cm. Plošný skener určený pro formáty A4 má přibližně 15 000 buněk.
- Skenery s udávaným rozlišením 1200 dpi mají obvykle snímací prvek s rozlišením 600 dpi. Pohybující se snímací mechanismus je schopen na dráze dlouhé jeden palec změřit 1200 řádek předlohy, takže výsledné optické rozlišení elektronické podoby obrázku z takového skeneru je oněch 600 x 1200 dpi. Obdobně skenery označené rozlišením 600 dpi mají většinou snímač s rozlišením 300 dpi, který snímá předlohu v 600 krocích (řádkách) na palec.

ROZHRANÍ

SÉRIOVÉ ROZHRANÍ

COM

- Standard RS-232, resp. jeho poslední varianta RS-232C z roku 1969, (také sériový port nebo sériová linka) se používá jako komunikační rozhraní osobních počítačů a další elektroniky. RS-232 umožňuje propojení a vzájemnou sériovou komunikaci dvou zařízení, tznamená že jednotlivé bity přenášených dat jsou vysílány postupně za sebou (v sérii) po jediném vodiči podobně jako u síťové technologie Ethernet nebo rozhranní USB.

- V současné době (2006) se v oblasti osobních počítačů od používání sériového rozhranní RS-232 již téměř definitivně ustoupilo a to bylo nahrazeno výkonějším Univerzálním Sériovým Rozhranním (USB). Nicméně v průmyslu je tento standard, především jeho modifikace - standardy RS-422 a RS-485, velice rozšířen a pro své specifické rysy tomu tak bude i nadále. Na rozdíl od komplexnějšího USB, standard RS-232 pouze definuje, jak přenést určitou sekvenci bitů a nezabývá se už vyššími vrstvami komunikace. V referenčním modelu ISO/OSI tak představuje pouze fyzickou vrstvu.

Základní technický popis

- Standard definuje asynchronní sériovou komunikaci pro přenos dat. Pořadí přenosu datových bitů je od nejméně významného bitu (LSB) po bit nejvýznamnější (MSB). Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se také setkat se 7 nebo 9 bity. Logický stav „0“/„1“ přenášených dat je reprezentován pomocí dvou možných úrovní napětí, které jsou bipolární a dle zařízení mohou nabývat hodnot ±5 V, ±10 V, ±12 V nebo ±15 V (logická 1 = záporná hodnota, logická 0 = kladná hodnota). Nejčastěji se používá varianta při které logické hodnotě 1 odpovídá napětí -12 V a logické hodnotě 0 pak +12 V. Základní tři vodiče rozhranní (příjem RxD, vysílání TxD a společná zem GND) jsou doplněny ještě dalšími sloužícími k řízení přenosu (vstupy DCD, DSR, CTS, RI, výstupy DTR, RTS). Ty mohou a nemusí být používány (zapojeny), nebo mohou být použity pro napájení elektronických obvodů v zařízení, jako je například počítačová myš. Výstupní elektronika je vybavena ochranou proti zkratu, kdy po překročení proudu 20 mA proud již dále neroste.

Asynchronní komunikace

- I když komunikující zařízení znají rychlost, jakou se data přenášejí, musí přijímač začít přijímat ve správný okamžik, tedy musí proběhnout synchronizace. V případě synchronní komunikace souběžně s datovým vodičem existuje i synchronizační vodič, na kterém vysílač oznamuje přijímači „teď jsem poslal data“, viz LPT a signál STROBE. Naopak u asynchronní komunikace se synchronizační vodič nepoužívá, pouze vysílač pošle nějaká definovaná data po datovém vodiči, po jejichž přijetí se přijímač zasynchronizuje. V případě RS232 každé sekvenci datových bitů předchází jeden start bit, kterým se logická hodnota na lince přepne (původně v klidovém stavu) do opačného stavu. Po datových bitech následuje paritní bit a za ním jeden nebo více stop bitů, během kterých je linka opět v klidovém stavu. Je tak možné pro komunikaci použít méně vodičů na úkor určitého snížení rychlosti způsobeného synchronizací. K podobné synchronizaci dochází i u Ethernetu, kde na začátku každého rámce vyšle vysílač několik bajtů, ve kterých se střídají bity 0 a 1.

USB

- USB (Universal Serial Bus) je univerzální sériová sběrnice. Moderní způsob připojení periférií k počítači. Nahrazuje dříve používané způsoby připojení (sériový a paralelní port, PS/2, GamePort apod.) pro běžné druhy periférií - tiskárny, myši, klávesnice, joysticky, fotoaparáty, modemy atd., ale i pro přenos dat z videokamer, čteček paměťových karet, MP3 přehrávačů, externích disků a externích vypalovacích mechanik.

Plug and Play

- Výhodou je možnost připojování Plug & Play bez nutnosti restartování počítače nebo instalování ovladačů. Zařízení lze připojit za chodu k počítači a během několika sekund je přístupné. Další zařízení lze připojit také do výstupního portu již připojeného zařízení, nebo pomocí rozbočovače (USB Hub).

- Existují dvě hlavní verze, USB 1.1 (max. přenosová rychlost 12 Mbit/s) a USB 2.0 (480 Mbit/s, pokud je zařízení high-speed). USB 2.0 je zpětně kompatibilní s USB 1.1.
- Dnešní počítače obvykle disponují alespoň dvěma konektory USB. Pro funkci USB je třeba jeho podpora na straně operačního systému.
- Maximální délka kabelu mezi dvěma zařízeními je 5 metrů.
- USB dovoluje připojit až 127 zařízení pomocí jednoho typu konektoru.

4piny:

- +5V
- data+
- data-
- zem

Napájení

- Připojeným zařízením USB zároveň poskytuje i stejnosměrné napájecí napětí 5 V. Připojené zařízení tak může po sběrnici odebírat proud až 100 mA, v případě potřeby může zařízení požádat o větší proud, maximálně však o 500 mA. U osobních počítačů občas bývají napájecí vodiče sběrnice vyvedeny přímo ze zdroje počítače a USB zařízení připojené k počítači tak může odebírat i mnohem vyšší proud. Tohoto triku zneužívají například některé externí USB pevné disky, jejichž odběr je vyšší než požadovaných 500 mA a které po připojení k jinému počítači nemusí fungovat.

PARALELNÍ ROZHRANÍ

- Paralelní port je původní název rozhraní, které je kompatibilní s počítači IBM. Byl navržen pro komunikaci s tiskárnou, která užívá 8bitovou prodlouženou ASCII sadu znaků. Název odvozený od řádkové tiskárny byl běžný všeobecný termín pro různé druhy tiskáren. Grafické tiskárny se spoustou jiných zařízení byly navrženy pro práci se systémem. Ve skutečnosti to byl průmyslový standard po dlouhou dobu až do roku 1990, kdy byl normován jako IEEE 1284. Dnes je používání paralelního portu v útlumu a to příchodem USB (Universal serial bus) a FireWire (IEEE 1394).

- **Většina počítačů v 80. a 90. letech měla jeden nebo dva porty.**
  - LPT1 : I/O port 0x378, IRQ 7
  - LPT2 : I/O port 0x278, IRQ 5

- Časem byla rozšířena zařízení navržená ke zpracování na paralelním portu. Nejvíce bylo jednosměrných zařízení, pouze informace poslané z počítače. Nicméně některá zařízení jako Zip byla schopna pracovat v obousměrném režimu. Tiskárny také časem začaly pracovat v obousměrném režimu dovolujícím posílat různé stavové informace.
- LPT port má 8bitovou paralelní datovou sběrnici + 4 piny pro ovládání výstupu (Strobe, Linefeed, Initialize, and Select In) a 5 pinů pro ovládání vstupu (ACK, Busy, Select, Error a Paper Out). Přenosová rychlost je 12000 kbit/s.
- Původní význam "LPT" byl "Line Print Terminal". Podobně znějící název se domluvil pro užívání na systémech ITS, DEC a CP/M.

- Ve většině případů nahrazuje paralelní port USB rozhraní. Nejnovější tiskárny jsou propojeny přes USB a nemívají paralelní port. Na spoustě nových počítačů je paralelní port vynechán kvůli úspoře nákladů, a protože jsou považovány za zastaralé. V laptopech je paralelní port obvykle dostupný kvůli rozšiřujícím stanicím.

Použití paralelních portů

- Tiskáren,
- Zip mechanik,
- Starších skenerů,

- Starších webkamer
- Některých prvních zvukových karet
- SCSI zařízení přesm paralelní port do SCSI přípojky

Definice signálů

- Výstupní signály paralelního portu jsou definovány klasickou TTL logickou úrovní signálů, tzn. log. 1. odpovídá hladině +3.5V až +5V a log. 0 hladině 0V až +0.4V (viz obrázek 2.).

- Rozdílné hodnoty maximálních zátěžových proudů paralelního portu, které se liší v závislosti na fyzické realizaci paralelního portu. Maximální hodnoty odebíraného proudu se mohou pohybovat od 4mA až po 20mA. Proto je vždy nejvýhodnější na paralelním rozhraní s PC komunikujícího zařízení využít oddělovací buffer.

TISKÁRNY

- Tiskárna je výstupní zařízení počítače, převádějící různým způsobem informace v elektronické podobě (text či obrázky) na papír.

Jehličková tiskárna

- U jehličkové tiskárny se k tisku využívá tisková hlava, která obsahuje sadu pod sebou umístěných jehliček. V závislosti na počtu těchto jehliček se dále jehličkové tiskárny rozlišují na:
    - 1 jehličkové a 2 jehličkové: technické rarity vyráběné svého času v ČSSR
    - 7 jehličkové: poskytují tisk s velmi nízkou kvalitou a jsou používány pouze ve speciálních případech, jako jsou např. pokladny v prodejně, kde na kvalitu tisku nejsou kladeny téměř žádné nároky.
    - 9 jehličkové: dovolují tisk v tzv. NLQ (Near Letter Quality - blízký dopisní kvalitě) režimu. Tento režim svou kvalitou tisku odpovídá přibližně kvalitě elektrického psacího stroje. Tyto tiskárny jsou vhodné pro tisky výpisů programů a pro tisk dokumentů, na jejichž kvalitě příliš nezáleží.
    - 24 jehličkové: umožňují kvalitnější tzv. LQ (Letter Quality - dopisní kvalita) režim tisku. Oproti 9 jehličkovým tiskárnám poskytují také větší rychlost tisku. Jsou používány opět zejména pro dokumenty, na jejichž kvalitu jsou kladeny nižší nároky.
- Jednotlivé jehličky jsou připojeny k elektromagnetům, které je při práci (tisku) vystřelují proti barvící pásce. Tato barvící páska dopadne v daném bodě pak na papír, kde způsobí malý barevný bod. Obecně platí, že jehličkové tiskárny jsou poměrně hlučná zařízení, která nejsou vhodná pro tisk grafických dokumentů a neposkytují příliš velkou rychlost tisku (řádově 100 zn/s). Barevný tisk je u jehličkových tiskáren možný použitím vícebarevné pásky. Vzhledem k výše popsanému principu tisku nevyžadují tyto tiskárny žádný speciální papír. Jejich pořizovací cena i cena za vytištěnou stránku jsou poměrně nízké.

Tepelná tiskárna

- Tepelné tiskárny tisknou na podobném principu jako tiskárny jehličkové. Jsou opět vybaveny tiskovou hlavu, která obsahuje sadu jehliček připevněných k elektromagnetům. Jednotlivé jehličky jsou však na rozdíl od jehličkové tiskárny zahřáty na vyšší teplotu, která poté, co se jehlička přiblíží ke speciálnímu papíru citlivému na teplo, způsobí jeho zabarvení.

- Tepelné tiskárny poskytují podobnou kvalitu a rychlost tisku jako tiskárny jehličkové. Jejich velkou nevýhodou je nutnost použít speciální papír a tím i vyšší cena za vytištěnou stránku. V dnešní době se tyto tiskárny používají jen výjimečně.

Inkoustová tiskárna

- Inkoustová tiskárna tiskne pomocí inkoustu, který je stříkán na papír. Inkoust bývá umístěn v malé nádržce, jež se pohybuje společně s tiskovou hlavou.

- Kvalita tisku inkoustové tiskárny je silně závislá na použitém papíru. V případě kvalitního papíru je možné dosáhnout velmi kvalitního tisku (za cenu vyšších nákladů na tuto vytištěnou stránku). Barevný tisk bývá prováděn pomocí různobarevných inkoustů.

- Inkoustové tiskárny poskytují vyšší rychlost tisku než tiskárny jehličkové. Jedná se o zařízení vhodná pro tisk běžných textových i grafických dokumentů. Jejich pořizovací cena dnes již není příliš vysoká. Jejich nevýhodou je však poměrně vysoká cena za vytištěnou stránku, která je dána cenou inkoustu a vyšší cenou kvalitního papíru.

- **Rozdíl je ve vystřelování kapek, který může být řešen několika způsoby:**

    termická technologie

    - Tehdy se inkoust v komůrce tiskové hlavy rychle ohřeje na vysokou teplotu (až sto tisíc °C), tím změní svůj objem a skupenství (na páru), kapka se tím vystřelí. Vzniklým podtlakem se nasaje další kapka

inkoustu, celá procedura se opakuje mnohotisíckrát za vteřinu. Tuto technologii používá např. Canon, HP, Lexmark, Xerox.

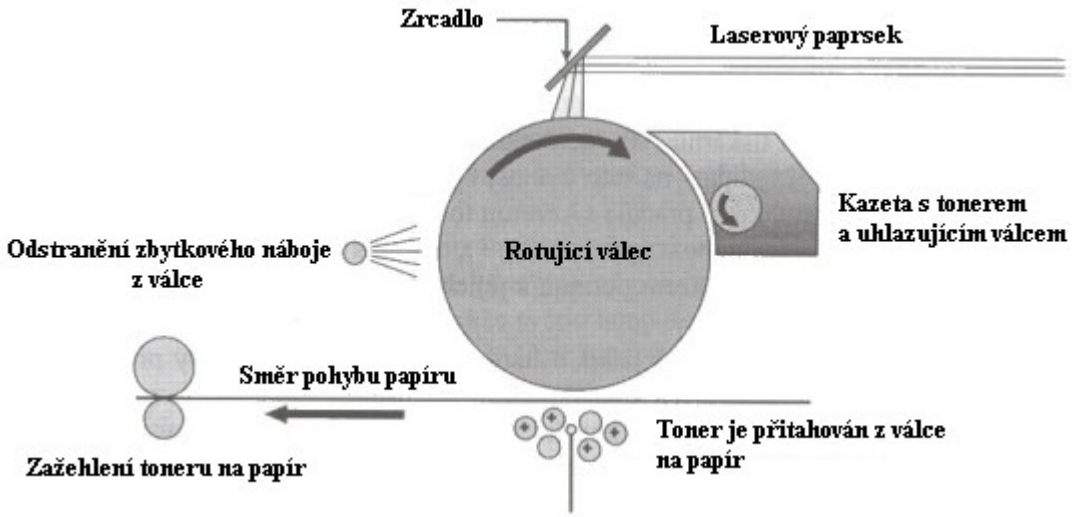
piezoelektrická metoda

- Zde již nedochází k zahřívání inkoustu, ale k změně tvaru piezoelektrického krystalu v tiskové hlavě působením elektrického proudu. Vznikne tlak a kapka opustí tiskovou hlavu. Tato metoda má výhodu ve vyšší životnosti tiskové hlavy a také rychlosti tisku. Přestože je toto řešení objemnější a musí být opatřeno menším počtem trysek, než je tomu u metody termické, kvalita je lepší (anebo by aspoň měla být), protože mechanické řízení krystalem je přesnější a umožňuje tisk menšími kapkami inkoustu. Ten je vystřikován 20.000 - 30.000 tisíckrát za vteřinu 64-180 tryskami (pro černobílý inkoust) a 48-96 tryskami pro každý ze tří až pěti barevných

- V případě barevného tisku je nutné pracovat se subtraktivním modelem mísení barev (na rozdíl od obrazovky, kde pracujeme s aditivním mísením). Tento model označovaný také jako CMYK používá pro tisk tří až čtyř základních barev, jejichž mísením se dostávají barvy ostatní:
  - Cyan - indigově modrá
  - Magenta - fialová
  - Yellow - žlutá
  - Black - černá

Laserová tiskárna

- Laserová tiskárna je zařízení určené zejména pro profesionální použití. Poskytuje velmi vysokou kvalitu (300 dpi - 1200 dpi) i rychlost tisku (desítky stránek za minutu). Jedná se o poměrně drahé zařízení - ale cena za vytištěnou stránku bývá většinou nižší než u inkoustových tiskáren.



Princip tisku laserové tiskárny

- Při tisku laserové tiskárny jsou nejdříve znaková data zasílaná počítačem převáděna řadičem na videodata. Ta jsou zasílána na vstup polovodičovému laseru. Polovodičový laser vysílá laserový paprsek, který je vychylován soustavou zrcadel na rotující válec. V místech, kam tento paprsek na válec dopadne, dojde k jeho nabití statickou elektřinou, na potenciál řádově 1000 V. Rotující válec dále prochází kolem kazety s barvícím práškem (tonerem), který je vlivem statické elektřiny přitažen k nabitým místům na povrchu válce. Papír, který vstoupí do tiskárny ze vstupního podavače, je nejdříve nabit statickou elektřinou na potenciál vyšší, než jsou nabitá místa na válci (cca 2000 V). V okamžiku, kdy tento papír prochází kolem válce, dojde k přitažení toneru z nabitých míst válce na papír. Toner je do papíru dále zažehlen (v peci) a celý papír je na závěr zbaven elektrostatického náboje a umístěn na výstupní zásobník. Rotující válec po otištění na papír prochází dále kolem sběrače elektrostatického náboje a čističe od toneru.

- Barevný tisk je u laserových tiskáren možné docílit použitím různobarevných tonerů a opakováním popsaného způsobu.

OSTATNÍ ZÁZNAMOVÁ MÉDIA

CD-ROM

- Médium CD-ROM vznikalo původně jako audio nosič a jeho autory byly firmy Philips a Sony. Jedná se o médium, které je určeno pouze ke čtení informací. Dovoluje uložení až 650 MB programů a dat.

- Na rozdíl od dříve uvedených diskových zařízení (pružné disky, pevné disky, ZIP disky, Magnetooptické disky apod.) nejsou data ukládána do soustředných kružnic, ale do jedné dlouhé spirály podobně jako na gramofonové desce. Spirála začíná u středu média a rozvíjí se postupně až k jeho okraji. Záznam (spirála dat) je pouze na spodní straně disku, tj. záznam na CD-ROM disku je jednostranný. Délka celé spirály je zhruba 6 km a hustota dat v ní uložených je konstantní. Podle rychlosti, kterou je CD-ROM mechanika schopna číst tato data, se mechaniky rozlišují na:
    - single speed: rychlost čtení dat je 150 kB/s, dostačuje pouze pro přenos souborů
    - double speed: data je schopna číst rychlostí 300 kB/s, což poskytuje plynulou rychlost pro práci s datovými soubory. Nedostačuje pro přehrávání videa
    - triple speed: dovoluje číst data rychlostí až 450 kB/s
    - quadruple speed: mechanika dovolující čtení dat rychlostí 600 kB/s
    - 6x: rychlost čtení: 900 kB/s
    - 8x: rychlost čtení: 1200 kB/s
    - 12x: rychlost čtení: 1800 kB/s
    - 16x: rychlosot čtení: 2400 kB/s
    - 24x: rychlost čtení: 3600 kB/s
- Rychlost čtení spirály je v single speed mechanice asi 1,3 m/s. Rychlost otáčení CD-ROM disku není konstantní, ale je kontinuálně přizpůsobována podle toho, zda se čtení provádí blíže kraji nebo středu disku. U středu disku je rychlost otáčení vyšší (asi 500 otáček za minutu) a u kraje naopak nižší (asi 200 otáček za minutu). Toto přizpůsobování otáček disku zaručuje, že data jsou čtena ze spirály konstantní rychlostí.

- Přístupová doba u datových CD-ROM disků je potom závislá na čase nutném k regulaci otáček. Je tedy velmi nevhodné číst data uložená v různých částech disku, protože je neustále nutné přizpůsobovat rychlost otáčení. Tento problém plně neodstraňují ani mechaniky s vyšší přístupovou rychlostí, i když samozřejmě mechaniky s vyšší rychlostí čtení mají i nižší přístupovou dobu. Přístupová doba se u CD-ROM mechanik pohybuje od 100 ms do 300 ms.

- Protože šířka stopy spirály je velmi malá, data jsou uložena s poměrně velkou hustotou a vlastní CD-ROM nosič není ničím chráněn, je velká pravděpodobnost, že i při běžné manipulaci s CD-ROM diskem může dojít ke špatnému přečtení některých uložených bitů. Proto informace uložené na médiu CD-ROM jsou silně redundantní (nadbytečné) a mechanika má obvody realizující na základě těchto nadbytečných informací poměrně složité algoritmy pro korekturu chyb vzniklých při čtení.

- Sektor se skládá z 98 rámců. Každý rámec začíná 24 synchronizačními bity a pak následují 2 bloky s 12ti uživatelskými byty a 4mi byty CIRC- korekce. Každý bajt začíná trojicí spojovacích nulových bitů a 14ti bitovým kódem bajtu. Při zakódování nesmí být dvě jedničky vedle sebe. Aby toto nenastalo při spojování bajtů, jsou proloženy 3mi spojovacími nulovými bity. Každá sekunda záznamu obsahuje 75 sektorů.
- Čtení a zápis se provádí laserem, který přes zrcátko svítí na disk.
- Data jsou na CD-ROM uloženy v drážkách,
- které jsou široké 1,6 mikrometrů (tisícina milimetru). Samotný bit je pak reprezentován "dolíkem" v této drážce. Ve chvíli, kdy na tento "dolík" posvítí laser mechaniky, světlo se buď odrazí, nebo neodrazí - v závislosti na tom, zda je na daném místě umístěna jednička nebo nula.

DVD

- DVD média jsou plastové disky, navenek stejná jako CD média. DVD média mají průměr 120 mm a jsou 1,2 mm silná. Data se ukládají pod povrch do jedné nebo dvou vrstev ve stopě tvaru spirály (jako CD). Pro čtení dat se

používá laserové světlo s vlnovou délkou 660 nm, tedy kratší než v případě CD; to je jeden z důvodů jejich vyšší kapacity. Stejně tak příčný odstup stop je menší - 0,74 µm oproti 1,6 µm u CD.
- Označení „+“ (plus) a „–“ (pomlčka) představuje dva různé technické standardy, které jsou do určité míry kompatibilní.

Médium může být typu:

- DVD-ROM (read only, jen pro čtení, vyrábí se lisováním) je potenciální nástupce formátu CD ROM, tedy víceúčelový formát pro přehrávání počítačových dat a multimediálních aplikací. Čtení DVD je možné ve všech PC (a ostatních platforem) vybavených jednotkou DVD s podporou logického formátu UDF.
- DVD+R/RW (R = Recordable, jen pro jeden zápis, RW = ReWritable, pro přepisování)
- Formát DVD+R je mezi široce rozšířenými formáty nejmladší, kupodivu mladší, nežli formát DVD+RW. Disky DVD+R lze v současnosti běžně zapisovat osminásobnou rychlostí oproti standardní rychlosti DVD, tedy 10800kB za vteřinu. Touto rychlostí trvá zápis na disk přibližně 10 minut. DVD+RW jde o přepisovatelnou verzi formátu DVD+. Standardní rychlost pro zápis na toto médium je čtyřnásobná oproti základní rychlosti čtení DVD.

- Médium umožňuje zápis na jednu nebo obě dvě strany, v jedné nebo dvou vrstvách na každou stranu. Na počtu stran a vrstev závisí kapacita média.
    - DVD-5: jedna strana, jedna vrstva, kapacita 4,7 GB (4,38 GiB)
    - DVD-9: jedna strana, dvě vrstvy, 8,5 GB (7,92 GiB)
    - DVD-10: dvě strany, jedna vrstva na každé straně, 9,4 GB (8,75 GiB)
    - DVD-14: dvě strany, dvě vrstvy na jedné straně, jedna vrstva na druhé, 13,2 GB (12,3 GiB)
    - DVD-18: dvě strany, dvě vrstvy na každé straně, 17,1 GB (15,9 GiB)

PÁSKOVÉ PAMĚTI

- Páskové paměti jsou typickým sekvenčním zařízením, to znamená, že pokud je potřeba zpřístupnit libovolnou informaci na pásce, je nutné, aby nejdříve byly přečteny všechny informace předcházející. Mezi první páskové paměti patří devítistopá páska o šířce 1/2". Hustota záznamu na těchto páskách dosahovala až 6250 bpi (bits per inch = bitů na palec). Tyto páskové paměti se používaly zejména u velkých sálových počítačů a vyžadovaly poměrně náročnou obsluhu, protože páska byla navinuta pouze na cívce (nikoliv umístěna v kazetě) a tudíž se musela pracně zavádět do čtecího zařízení.

- Páskové paměti jsou vhodné zejména pro zálohování velkého objemu dat a jeho případné obnovy. Jsou naprosto nevhodné pro časté zpřístupňování určitých částí dat. Toto je dáno jejich sekvenčním přístupem k datům, který může způsobit, že přístupová doba k datům uloženým na konci pásky může dosáhnout až několika hodin.

- Připojování pásky se provádělo přes rozhraní SCSI, záznam byl prováděn magneticky a životnost pásky byla odhadována na 25 let.

DISKETOVÁ MECHANIKA

- Mechaniky pružných disků jsou zařízení pro čtení a zapisování na pružné disky. Je možné je rozdělit podobně jako pružné disky podle velikosti (51/4", 31/2") a podle hustoty záznamu (DD, HD).

- První počítače PC/XT měly většinou osazeny dvě mechaniky pružných disků 51/4" DD. Jedna se používala pro zavedení operačního systému a druhá pro spouštění aplikačních programů a čtení (ukládání) dat. Později se začaly objevovat první pevné disky. Počítače PC/AT byly zpočátku vybaveny jednou mechanikou 51/4" HD a pevným diskem. Později se začaly více prosazovat mechaniky 31/2", takže počítače byly osazovány jednou mechanikou 51/4" HD a jednou mechanikou 31/2" HD. Dnes se u počítačů PC používají zejména 31/2" HD mechaniky.

- Nízká kapacitě - 3.5" disketa má kapacitu 1,44MB

ZIP DISKY

- ZIP disky jsou média vyrobená firmou Iomega a jedná se disk o průměru 31/2", na který je možné uložit 100 MB dat. Princip práce ZIP disku je podobný jako u disketové mechaniky. Provádí se na magnetickou vrstvu pomocí čtecích (zapisovacích) hlav, které při práci přímo dosedají na povrch média. Mechaniky pro ZIP disky se vyrábějí v interním i externím provedení.