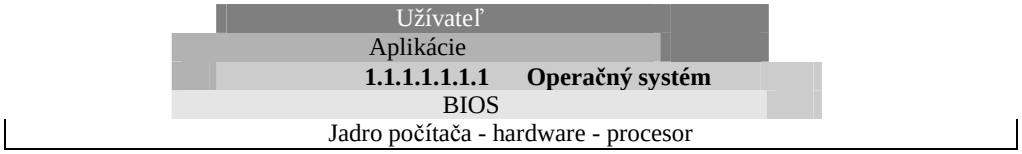
1 Operačné systémy

1.1 Úvod

1.1.1 História programového vybavenia počítačov

Prvé počítače boli konštruované ako veľmi úzko špecializované na riešenie úzkeho okruhu úloh, ktoré si boli principiálne veľmi blízke. Programové vybavenie bolo vyvíjané taktiež ako jednoúčelové, na riešenie určitej konkrétnej úlohy a pre určitý konkrétny počítač. Každý konkrétny program bol úzko zviazaný s hardwarom počítača, s prenositeľnosťou programov na iný počítač sa neuvažovalo. Hlavným nástrojom na tvorbu programov v tomto období bol strojový kód.
S nárastom počtu počítačov a s komercionalizáciou trhu s programovým vybavením nastupuje potreba oddelenia užívateľských programov od konkrétnej hardwarovej platformy. V prvom kroku sa programátorom poskytuje aspoň základná sada a procedúr a programov na ovládania technických prostriedkov počítača a jeho periférií – Basic Input and Output System – BIOS.

Hierarchické postavenie operačného systému v softwarovej štruktúre:



Technológia s využitím BIOS-u však nebola všeobecne prijatá, systémy Intel sú viazané na BIOS prakticky od prvých 16- bitových procesorov, kým iné systémy – napr. Apple vrstvu BIOS-u vôbec nepoznajú.

Nad BIOS-om (prípadne priamo nad HW vrstvou) bol vybudovaný systém programov, ktoré poskytujú rozhranie medzi prekladačmi štandardných programovacích jazykov a konkrétnym hardwarom určitého počítača – operačný systém.
Búrlivý rozvoj operačných systémov však prichádza najmä v období nástupu mikroprocesorov.

1.1.2 Základné úlohy operačných systémov

Teraz sa skúsme spoločne zamyslieť nad tým, s akými úlohami sa musí potýkať programátor aplikácie, ak **nemá k dispozícii operačný systém**. Na tieto úvahy je však potrebné oprášiť si základné vedomosti o programovaní v assembléri a programovaní v nejakom vyššom programovacom jazyku – napríklad v Pascale alebo jazyku „C“.

Skúste si sami – bez čítania nasledujúcich riadkov - zodpovedať otázku, na čo všetko musíte ako programátori myslieť pri programovaní jednoduchej úlohy v assembléri. Porovnajte to so situáciou, kedy tú istú úlohu programujete vo vyššom programovacom jazyku a na obsluhu základných systémových prostriedkov môžete využívať operačný systém.
Ak ste si rozmysleli odpoveď, skúste svoje odpovede porovnať s nasledovným odstavcom. Zrejme ste prišli na to, že pri programovaní v strojovom kóde (alebo pri programovaní v jazyku symbolických adries – assembléri, čo je principiálne to isté) musíte

- veľmi presne definovať spôsob práce s operačnou pamäťou a priamo riadiť jej využívanie; u rozsiahlejších programov musí riešiť aj úlohy spojené s jej segmentáciou či stránkovaním
- do programu zabudovať aj všetky mechanizmy obsluhy vstupných, výstupných, vstupnovýstupných zariadení a vonkajších pamäťových zariadení
- podrobne sa zaoberať aj otázkou obsluhy vonkajšej pamäte a podrobne riešiť jej adresovanie aj transfer dát
- vychádzať presne z konkrétneho typu procesora, na ktorom má byť program prevádzkovaný a presne sa riadiť inštrukčnou sadou daného procesora

Je zrejmé, že napísať rozsiahlejší či náročnejší program v strojovom kóde je práca veľmi komplikovaná a náročná na dôslednosť, kombinačné schopnosti, trpezlivosť a v zásade predstavuje obrovské množstvo hodín práce vysokokvalifikovaného odborníka. A to sme sa nezaoberali problémami spojenými so súčasným vykonávaním viacerých úloh (viacúlohové systémy, multitasking), s paralelným behom viacerých procesorov v rámci riešenia jedinej úlohy (multiprocesing), s prideľovaním pamäťového priestoru jednotlivým úlohám a s ochranou dát

v operačnej pamäti tak, aby jedna úloha nemohla ani omylom zasiahnúť do dát v tej oblasti pamäti, ktorá je vyhradená pre inú úlohu, s ochranou dát pred neoprávneným prístupom neautorizovaného užívateľa, atď.

Na tomto mieste nesmieme zabudnúť poznamenať, že programovanie úloh priamo v strojovom kóde ponúka aj značné výhody: keďže aplikácia sa nemusí obracať na žiadneho sprostredkovateľa (okrem BIOSu), je vykonávanie programu veľmi rýchle v porovnaní s programami napísanými vo vyššom programovacom jazyku a spúšťanými prostredníctvom operačného systému.

Nič to však nemení na skutočnosti, že programovanie priamo v strojovom kóde bez využitia služieb operačného systému a bez použitia vyšších programovacích jazykov je príliš nepraktické na to, aby bolo využívané na programovanie bežných aplikácií.

1.1.3 Komponenty systému, úzko spolupracujúce s operačným systémom

V súvislosti s operačným systémom sa veľmi často stretávame s pojmami Firmware, BIOS a SETUP BIOS. Rozdiel medzi Firmware a Operačný systém.

1.1.3.1 Firmware

Pod pojmom Firmware rozumieme program, ktorý je napevno uložený v neprepisovateľnej pamäti (prípadne v pamäti typu Flash, non-volatile memory) zariadenia a ktorého úlohou je umožniť, prípadne zjednodušiť ovládanie a využívanie príslušného zariadenia. Firmware je napevno zviazaný so zariadením, spúšťa sa vždy súčasne so spustením zariadenia. Zdrojový kód firmware obsahuje zvyčajne desiatky až stovky kB. Upgrade firmware za novšiu verziu (prepísaním programu vo Flash pamäti) umožňuje zlepšenie výkonu, zlepšenie kompatibility s rôznymi verziami OS, pridanie nových funkcií, alebo odstránenie existujúcich chýb. Umožňuje realizáciu tzv. *power on-self* testu. V špeciálnych prípadoch (u tzv. embeded zariadení) môže umožňovať spúšťanie jednoduchých aplikácií priamo, bez účasti operačného systému, spravidla však firmware s aplikáciami nepracuje.

Rozdielov oproti operačnému systému je veľa:

- Je napevno zviazaný s konkrétnym zariadením
- Je uložený v neprepisovateľnej pamäti zariadenia
- Nie je určený na komunikáciu s užívateľom
- Nie je určený na spúšťanie nadstavbových aplikácií.

S operačným systémom spolupracuje pomocou ovladačov (drivers).

1.1.3.2 BIOS

(**Basic Input Output System**) predstavuje základnú knižnicu obslužných programov ku základnej doske počítača. Uľahčuje prácu operačného systému tým, že ak operačný systém vyžaduje vykonanie nejakej štandardnej úlohy (načítanie konkrétneho sektora z disku, zobrazenie písmena na zobrazovacom zariadení), zavolá si príslušnú službu BIOSU, a tá vykoná požadované činnosti na úrovni strojového kódu. Operačný systém sa nemusí zaťažovať štandardnými rutinnými úlohami, obstará to BIOS.
BIOS je uložený v neprepisovateľnej pamäti, prípadne v pamäti typu Flash na základnej doske počítača. Výmena BIOSu za novšiu verziu dokáže zlepšiť výkon počítača, prípadne umožniť kompatibilitu s novými verziami rozhraní či periférnych zariadení.
BIOS môžeme v zásade považovať za firmware základnej dosky počítača.
Pozn: Základná doska vôbec nemôže bez funkčného BIOS pracovať, takže akýkoľvek zásah do BIOSu (napr. upgrade BIOS) je značne riziková operácia, do ktorej by sa mal púšťať iba skúsený technik. Určitým riešením, ktoré minimalizuje riziko zničenia dosky neúspešným upgradom BIOSu je používanie dosiek s dual- BIOS modulmi. Niektoré systémy, napríklad počítače s operačným systémom Apple, vôbec BIOS nepoužívajú a všetky úlohy súvisiace s obsluhou zariadení sú delegované priamo na operačný systém.

1.1.3.3 SETUP BIOS

je konfiguračný súbor základnej dosky. Je uložený v pamäti typu CMOS, táto pamäť je energeticky závislá a preto si vyžaduje zdroj napájania aj pre stav pri vypnutom PC. Kapacita tejto pamäte je rádovo desiatky kB a obsahuje základné údaje o HW konfigurácii základnej dosky. Okrem údajov a dátume a čase uchováva údaje o inštalovaných rozhraniach a ich nastavení, o konfigurácii periférnych zariadení a o stave on-board zariadení. Obsahuje údaje o konfigurácii zbernicového systému, nastavenie prostriedkov, používaných jednotlivými zariadeniami pri

komunikácii so zbernicou (IRQ, I/O , DMA parametre). Definuje parametre použitých RAM modulov (taktovanie, timing).
SETUP BIOS je užívateľovi dostupný pri štarte počítača, stlačením určitej klávesy (o ktorú klávesu ide je užívateľ informovaný výpisom na monitore pri štarte počítača, pre rôzne systémy sa metóda prístupu do SETUP BIOS líši). Väčšina nastavení v SETUP BIOS býva realizovaná systémom automaticky. Manuálna zmena niektorých hodnôt môže v určitých prípadoch zlepšiť výkon zostavy, ale nesprávna konfigurácia môže zapríčiniť znefunkčnenie celého systému. Ku manuálnym úpravám SETUPu je treba pristupovať s maximálnou opatrnosťou a jednotlivé kroky mať dobre premyslené. Dobrým zvykom je zabezpečiť prístup do SETUP BIOS heslom.

1.1.4 Charakteristika súčasných operačných systémov

V súčasnosti považujeme operačný systém za základné programové vybavenie počítača, bez ktorého je efektívne používanie počítača nemožné. Operačný systém sa štandardne spúšťa vždy okamžite po spustení počítača plní viacero základných úloh:

- *zabezpečuje nezávislosť vyšších programových vrstiev od konkrétneho hardware*
- *zabezpečuje komunikáciu počítača s užívateľom prostredníctvom užívateľského rozhrania*
- *prostredníctvom tohto rozhrania priebežne informuje používateľa o aktuálnom stave výpočtového systému a o stave vykonávaných úloh*
- *vytvára vhodnú platformu pre spúšťanie aplikácií a správu bežiacich aplikácií v rámci celého výpočtového systému*
- *prideľuje výpočtové prostriedky jednotlivým aplikáciám (operačnú pamäť, procesory)*
- *zabezpečuje správu operačnej pamäti vo vzťahu k bežiacim procesom a aplikáciám, tzn. prideľuje im adresný priestor, spravuje adresný priestor zdieľaný viacerými aplikáciami či procesmi a zabezpečuje privátny adresný priestor jednotlivých aplikácií pred narušením integrity uložených dát neoprávneným zásahom iného procesu*
- *je zodpovedný za efektívne a bezchybné využívanie operačnej pamäte, teda obsluhuje segmentovanie či stránkovanie pamäte, jej defragmentáciu a za vytváranie a používanie virtuálnej operačnej pamäte*
- *prideľuje čas výpočtového systému aplikáciám alebo užívateľom*
- *obsluhuje vstupno-výstupné zariadenia vo vzťahu ku výpočtovému systému*
- *prostredníctvom súborového systému organizuje adresovanie, uloženie a správu údajov na vonkajších pamäťových médiách*
- *poskytuje základné nástroje na elementárnu správu súborov, ako je ich organizovanie do prehľadných hierarchických štruktúr (priečinkov), prezeranie, presúvanie, kopírovanie, mazanie, triedenie, úprava základných atribútov*
- *u viacužívateľských operačných systémov zabezpečuje ochranu súborov a dát pred neoprávneným prístupom*
- *zabezpečuje diagnostické funkcie, ako je kontrola bezchybného chodu výpočtového systému, ošetruje neštandardné stavy a generuje chybové hlásenia*
- *autodetekcia a automatické odstraňovanie chýb, ktorých správa je v kompetencii operačného systému*
- *u sieťových operačných systémov poskytuje základné nástroje na komunikáciu so sieťovými prostriedkami či klientmi*

Okrem uvedených hlavných funkcií OS bývajú súčasné komerčné operačné systémy vybavované množstvom ďalších funkcií, ktoré z programátorského hľadiska predstavujú už čisto aplikačnú vrstvu, z komerčného hľadiska ich firmy však ponúkajú ako súčasť dodávaného OS. Sú to rôzne súborové managery, nástroje na údržbu a správu SW počítača a uložených dát, knižnice štandardných programov a linkovacie programy na spájanie užívateľských programov s knižnicami, prekladače assembléru a prekladače rôznych vyšších programovacích jazykov, emulátory iných systémov, rôzne komunikačné nástroje a exploatačné nástroje na využívanie sieťových služieb, jednoduchšie verzie textových a grafických editorov, nástroje na prehrávanie a prípadne aj editáciu multimediálnych súborov, jednoduché hry a mnohé ďalšie, súhrnne označované ako „príslušenstvo“.

Niekedy je problematické rozhodnúť, ktoré moduly ešte považovať za moduly operačného systému a čo sú už aplikácie, u moderných operačných systémov je táto hranica stále menej zreteľná.

Už z uvedeného výpočtu hlavných funkcií operačného systému (ktorý navyše z pochopiteľných dôvodov nemôže byť úplný) je zrejmé, že vývoj súčasného operačného systému je úloha pre rozsiahle tímy špičkových špecialistov.

V prípade komerčných operačných systémov ide o vývojárov platených softwarovými firmami (Microsoft, Apple, Novell...), špecialistov ale aj študentov rôznych vysokých škôl (najmä u rôznych verzií UNIXu), ale v rámci projektu Open Source, pod ktorým beží vývoj operačného systému LINUX sa na vývoji tohto produktu môže podieľať skutočne každý, kto má potrebné vedomosti a trúfa si verejne sa prezentovať so svojimi nápadmi a vylepšeniami uvedeného OS. Mimochodom – najpočetnejšou skupinou, ktorá sa zaoberá vývojom programov na platforme Open Source sú vysokoškolskí študenti vo veku medzi 19 a 26 rokov.

1.1.4.1 Vlastnosti operačných systémov, typy operačných systémov

Skôr, kým budete pokračovať ďalej si skúste sami zodpovedať nasledovné otázky:

- aké konkrétne operačné systémy poznáte
- s ktorými systémami ste už niekedy pracovali alebo sa aspoň v rýchlosti oboznámili?
- ak môžete porovnať aspoň dva systémy, prípadne viaceré, skúste ich porovnať z hľadiska ich vlastností!
- skúste porovnať systémy typu DOS a Windows na základe Vašich doterajších skúseností z nasledovných hľadísk:
  - prostredia, komfortu práce
  - počtu programov, ktoré môžu byť súčasne spustené
  - nárokov na technické prostriedky počítača
- ak máte aspoň základné skúsenosti s prácou v operačnom systéme Linux či Unix, skúste urobiť obdobné porovnanie napríklad medzi DOSom a Linuxom

Ak ste sa zamysleli nad otázkami v predchádzajúcom odstavci a skúsili ste ich zodpovedať, určite ste došli k záveru, že operačné systémy môžeme rozdeliť podľa rôznych vlastností do niekoľkých skupín

1.1.4.1.1 podľa účelu použitia

- *špecializované*
- *univerzálne*

Špecializované operačné systémy bývajú určené na konkrétny účel - vývojové, riadiace, komunikačné, diagnostické a pod.

Univerzálne – na univerzálne použitie. Sú to vlastne všetky známe a bežne používané OS. V ďalšom texte sa budeme zaoberať výhradne univerzálnymi OS

1.1.4.1.2 podľa spôsobu uloženia v pamäti

- *pamäťovo rezidentné*
- *diskovo orientované*

Pamäťovo rezidentný OS je taký, **kde je celá exekutíva pevne umiestnená v pamäti ROM, používa sa pre jednoduché mikropočítače**, diskovo orientovaný načíta do operačnej pamäte iba základnú časť (jadro) a programové moduly potrebné pre ďalšie služby si dodatočne načíta z disku až keď sú potrebné

1.1.4.1.3 podľa typu používateľského rozhrania

- *pracujúce v režime príkazového riadku (CLI)*
- *využívajúce grafické používateľské rozhranie (GUI)*
- *využívajúce grafické používateľské rozhranie s podporou dotykového displeja*

Pozn.: Tu je potrebné sa zastaviť u častého omylu mnohých používateľov, ktorí sa domnievajú, že rozdiel medzi grafickým a textovým operačným systémom spočíva iba v implementácii grafického rozhrania. Napríklad v prípade Linuxu sa používa grafické rozhranie KDE, Gnome alebo X-Window, u operačných systémov typu DOS bol velmi

obľúbený Norton Commander, resp. obdobné semigrafické rozhrania. Uvedené grafické či semigrafické rozhrania sú však iba aplikáciami, spustenými nad operačným systémom.
Z pohľadu posúdenia vlastností operačného systému je rozhodujúce, či grafické užívateľské rozhranie je natívne *implementované ako súčasť jadra systému.*
Moderné operačné systémy, určené na prácu so zariadeniami typu tablet *alebo* smartphone*, disponujú ďalšou funkciou – podporou dotykového displeja, ktorý prináša ďalšiu úroveň komunikácie medzi zariadením a užívateľom.*

1.1.4.1.4 podľa stupňa ochrany súborov jednotlivých užívateľov

- *jednopoužívateľské*
- *viacpoužívateľské*

Mnohí menej skúsení používatelia sa dopúšťajú omylu tým, že za viacpoúžívateľský OS považujú napríklad Windows 98. Je síce pravda, že pri spúšťaní týchto Windows je používateľ vyzvaný zalogovať sa – prihlásiť sa svojím používateľským menom a heslom, ale v prípade ignorovania tejto výzvy sa nestane vôbec nič a používateľ môže so systémom normálne pracovať, čo viac, má ničím neobmedzený prístup ku všetkým súborom bez ohľadu na to, kto dané súbory vytvoril a má plný prístup aj ku systémovým súborom, ktoré môže neobmedzene nielen používať, ale aj modifikovať či dokonca mazať.
Skutočný viacpoužívateľský systém nedovolí vôbec pracovať so systémom nikomu, kto sa neprihlási relevantným loginom, a po prihlásení striktne vymedzuje prístupové práva ku súborom a jednotlivým prostriedkom systému tak, ako boli zadefinované správcom systému (administrátorom, rootom). Takýmito systémami sú napríklad UNIX alebo Windows skupiny 2000/XP pracujúci na súborovom systéme NTFS.

1.1.4.1.5 Podľa schopnosti súbežne spracovávať viacero úloh

- *jednoúlohové - umožňujú spustiť jedinú aplikáciu*
- *viacúlohové - môže sa spústiť viac programov súčasne, tieto programy môžu vzájomne spolupracovať a vymieňať si dáta.*

Viacúlohové operačné systémy potom môžeme rozdeliť podľa použitej metódy multitaskingu na systémy, ktoré používajú multitasking

- *kooperatívny*
- *preemptívny*

Množstvo súčasne spustiteľných programov u viacúlohového systému je obmedzené najmä kapacitou RAM.

Toto rozdelenie vyjadruje schopnosť programu vykonávať súčasne viacero úloh a prideľovať týmto úlohám systémové prostriedky (čas procesora, adresný priestor v operačnej pamäti, ochranu pamäťového priestoru pred neoprávnenými zásahmi iných úloh). Úlohou rozumieme buď samostatný aplikačný program (typickým príkladom je súbežné spustenie rezidentnej antivírusovej ochrany na počítači pri súbežnom spustení aplikačného programu), alebo môže ísť o viacero súbežných procesov v rámci jediného aplikačného programu.

Predstavme si bežnú prácu napríklad v tabuľkovom kalkulátore (Excel, Calc):
- *monitorovanie klávesnice a vkladanie údajov z klávesnice na aktuálnu pozíciu kurzora*
- *monitorovanie myši a nastavenie kurzora na aktuálnu polohu*
- *kontrola dát v bunkách a priebežné prerátavanie hodnôt v závislých bunkách*
- *generovanie aktuálneho zobrazenia buniek podľa nastaveného formátu aktuálnej bunky*
- *generovanie formátu bunky na základe výslednej hodnoty pri podmienenom formátovaní*
- *atď.*

Schopnosť paralelne spracovávať viacero úloh potom úzko súvisí so schopnosťou operačného systému efektívne rozdeliť úlohy medzi viacero procesorov – teda s podporou multiprocesingu.

1.1.4.1.6 Podľa podpory práce v sieti

- *bez podpory siete – sieťové funkcie nie sú priamo implementované do operačného systému*
- *sieťové – sieťové funkcie sú priamo súčasťou operačného systému*

1.1.4.1.7 Podľa prístupnosti zdrojového kódu

- *Closed Source, uzatvorený systém: Zdrojový kód operačného systému je obchodným tajomstvom poskytovateľa (SW firmy) a užívateľ, ani nik iný nemá ku zdrojovému kódu prístup. Používanie systému podlieha podmienkam komerčného licenčného ujednania medzi poskytovateľom a koncovým používateľom.*
- *Open Source, otvorený systém: Zdrojový kód operačného systému je verejne prístupný, na odladovaní systému a jeho vylepšovaní sa môže podielať široká verejnosť, v rámci licencie priradenej ku produktu. Spravidla ide o licenciu typu GPU/GPL.*

1.1.4.2 Hlavné moduly operačných systémov

Na základe Vašich doterajších znalostí o operačných systémoch si skúste samostatne rozmyslieť, či je vhodné a potrebné, aby bol samotný operačný systém rozdelený do viacerých vrstiev, prípadne z akých častí sa operačný skladá.

Zoberme si operačný systém Windows XP

- *sú časti ako kalkulačka alebo WordPad súčasťou operačného systému?*
- *sú súčasťou operačného systému ovládače hardwarových zariadení?*
- *ktorá časť operačného systému sa stará o využívanie pamäti?*
- *už sa vám niekedy stalo , že operačný systém „zamrzol"? čím to bolo spôsobené?*
- *už sa vám niekedy stalo, že operačný systém vypísal oznam, že „program vykonal neplatnú operáciu a bude ukončený"? čím to bolo spôsobené?*
- *aké prostriedky používa systém Windows na správu operačnej pamäte? Koľko pamäte dokáže obsluhovať DOS a koľko Windows? čím suú dané rozdiely medzi nimi v schopnosti obhospodarovať operačnú pamäť?*
- *aké prostriedky používa Windows na správu súborov na diskoch? Aké sú rozdiely v správe súborov pod DOSom a Windows? A aké prostriedky poskytuje Linux?*

Operačný systém má niekoľko „vrstiev". Najnižšiu - fyzickú tvorí hardware počítača, nad ním operujú programy označované ako „firmware" - BIOSu (Basic Input Output System), na nich sú postavené vyššie „logické vrstvy" operačného systému.

1.1.4.2.1 Vo všeobecnosti rozoznávame tieto súčasti operačného systému:

Jadro (exekutíva = výkonná časť) operačného systému - táto časť je rezidentne umiestnená v pamäti; podľa potreby sa inicializuje (spúšťa) alebo nahráva do pamäte ostatné dôležité časti operačného systému. Ovláda riadenie prostriedkov, ktorými je tento systém vybavený, t.j. procesor, resp. procesory, operačná pamäť, zariadenia vonkajšej pamäte, vstupno – výstupné zariadenia a súbory dát.
V prípade, ak viaceré bežiace úlohy súčasne požadujú využívanie toho istého prostriedku je úlohou jadra OS prideľovať HW prostriedky systému optimálne tak, aby sa maximalizoval výkon a jednotlivé komponenty systému boli efektívne využívané.
Zavádzací modul – spúšťa sa ako celkom prvá služba. Jej úlohou je zaviesť OS do pamäte počítača a odovzdať mu riadenie. Určenie lokalít, z ktorých môže byť OS zavedený, a poradie prehľadávania lokalít, určuje SETUP BIOS.
Monitor operačného systému (od slova monitorovať = sledovať) - tiež nazývaný interpreter príkazov; zabezpečuje komunikáciu systému s užívateľom. Prijíma a analyzuje impulzy z klávesnice, zisťuje význam systémových príkazov, vypisuje príslušné odozvy na zobrazovacie (výstupné) zariadenie,…
Ovládače (drivery) - obslužné programy vstupno/výstupných zariadení.
Nadstavba – nie je nevyhnutná jako súčasť OS z technologického hľadiska , ale z marketingového hradiska sa implementuje jako súčasť instalačního balíka operačního systému. Ide o rôzne prehliadače, editory a ďalšie nástroje, ktoré skôr spadajú do oblasti aplikačních programov.

1.1.5 Hlavné úlohy operačného systému

- správa procesov
- správa pamäte
- správa súborov
- správa vstupných a výstupných zariadení
- komunikácia aplikácií s jadrom výpočtového systému
- komunikácia užívateľa s jadrom počítača
- základné služby na správu systému a správu súborov
- nadstavbové služby systému (textový editor atď)

Medzi operačným systémom a samotným jadrom počítača (procesor, pamäť,...) pracuje program na najnižšej úrovni – BIOS – (basic input-output system)t.j. vstupno výstupný systém.
BIOS je uložený napevno na základnej doske ako jej súčasť a bežný užívateľ ho nemá možnosť zmeniť.

1.1.6 Správa procesov

Z hľadiska úloh operačného systému bola situácia u starších typov počítačov, ktoré dokázali v reálnom čase pracovať iba na jedinej úlohe pomerne jednoduchá. Úloha naplno využívala všetky systémové zdroje aj schopnosti operačného systému. Systémy boli vybavené jediným procesorom a procesor disponoval jedinou výpočtovou jednotkou.
Tlak zo strany používateľov, ktorí žiadali riešiť stále náročnejšie úlohy pozostávajúce s množstva čiastkových „podúloh“ si vyžiadal radikálne zmeny v konštrukcii technických prostriedkov počítača, ale aj v štruktúre a schopnostiach operačných systémov.
Súčasné procesory bývajú vybavené množstvom paralelných výpočtových jednotiek, umožňujú vzájomnú spoluprácu viacerých procesorov (multiprocesing) a umožňujú paralelný beh viacerých programov v rámci jediného systému. Technické prostriedky nemôžu byť pridelované spusteným programom súbežne, ale ich prideľovanie musí byť v reálnom čase postupné – z pohľadu výkonnej jednotky procesora musí nastať „prepínanie“ medzi jednotlivými úlohami – ***multitasking***.
V západe existujú dva typy multitaskingu: Kooperatívny – spustený proces preberá kontrolu nad systémovými prostriedkami a sám rozhoduje o tom, kedy ukončí svoj beh a odovzdá systémové prostriedky naspäť operačnému systému
Preemptívny – o pridelení prostriedkov systému jednotlivým procesom rozhoduje operačný systém; ten rozhoduje o tom, ktorý proces a ako dlho bude využívať systémové prostriedky.

Pri behu jediného programu často nastávajú situácie, kedy sa beh programu zastaví, pretože sa čaká na výsledok nejakej udalosti (ošetrenie prerušenia vyvolaného hardwarom, načítanie dát z vonkajšieho pamäťového média atď.). V tomto čase je procesor nevyužitý a využitie multitaskingu dokáže významne zvýšiť využitie procesora.
Je zrejmé, že požiadavky kladené na moderný operačný systém (OS) sú veľmi vysoké. Na tento účel je program ako taký príliš veľkou jednotkou, aby s ním mohol operačný systém na uvedenej úrovni pracovať a je nevyhnutné, aby každý spustený program bol partikularizovaný na elementárne vykonávané jednotky – procesy.

Činnosť súčasného počítača pozostáva spravidla z veľkého množstva súbežne spustených procesov:

- *procesov operačného systému*
- *procesov aplikačných programov*

1.1.6.1 Vlastnosti procesov

- *všetky procesy bežia zdanlivo súčasne, CPU ich striedavo obsluhuje – „prepína“ medzi nimi; systém ako celok sa tým stáva produktívnejší*
- *proces nie je totožný s programom – a dokonca proces nie je ani časť programu ako taká, ale časť programu vo stave vykonávania*
- *proces nie je tvorený iba inštrukciami (kódom programu), ale obsahuje v sebe aj všetky aktivity vyvolané procesom – napríklad obsahy registrov procesora, ktoré boli naplnené určitými hodnotami v súvislosti s činnosťou procesu, stavom zásobníkov, kde sa nachádzajú dočasne uložené dáta ako napríklad parametre podprogramov, návratové adresy, dočasné premenné, ale aj hodnotami reprezentujúcimi aktuálny stav vykonávania programu a jeho adresy (obsahy segmentových registrov a index pointera)*
- *s jediným programom môže byť zviazaných viacero procesov: typickým príkladom je ak užívateľ má otvorených viacero rôznych internetových stránok na svojom prehliadači: program je ten istý,*

ale dátová časť je u každého procesu iná, tvorená náplňou a prípadnými aktivitami tej-ktorej stránky

- *na riadení procesu sa podielajú komponenty operačného systému nazývané dispečer a plánovač*
- *dispečer má za úlohu strážiť stav využitia technických prostriedkov počítača a zabezpečuje prideľovanie procesov tým technickým prostriedkom, ktoré momentálne nie sú používané*
- *plánovač musí z jednotlivých procesov, pripravených na spracovanie vybrať ten proces, ktorý má byť vykonávaný. Pritom musí mať na zreteli nielen efektivitu využitia technických prostriedkov počítača, ale aj prioritu jednotlivých čakajúcich procesov a efektivitu priebehu celej úlohy*

Úloha (Job) je súhrn činností potrebných na vykonanie zákazky zadanej užívateľom, spravidla sa vykonáva po jednotlivých krokoch
Kroky úlohy – činnosti, ktoré musia byť vykonané v určitom poradí (napr. preklad programu, zavedenie programu, jeho spustenie atď.
Proces – je to elementárna jednotka vykonávania úlohy. Samotné vykonávanie úlohy v sebe môže zahŕňať množstvo elementárnych úloh, ktoré môžu vzájomne od seba závisieť, ale môžu byť aj nezávislé a môžu byť vykonávané súbežne. Vykonávanie procesov si riadi operačný systém v spolupráci s procesorom. Súčasné vykonávanie viacerých procesov umožňujú multiprocesorové systémy. Súčasné procesory bývajú vybavené viacerými výkonnými jednotkami v rámci jediného procesora a tiež umožňujú paralelný beh viacerých procesov. Zdanlivo paralelný beh viacerých procesor umožňuje technika multitaskingu – prepínanie medzi jednotlivými procesmi a prideľovanie technických prostriedkov procesora určitému procesu. Organizáciu práce, prideľovanie prostriedkov a prepínanie medzi procesmi vykonávajú vo vzájomnej spolupráci **scheduler** a **dispečer**.
Prerušenie (interrupt) – udalosť, ktorá si vynúti prerušenie vykonávania procesu. Prerušenie sa dá zakázať maskovaním prerušenia.

Prostriedky prerušenia (interrupt hardware) – dovoľujú operačnému systému prerušiť práve prebiehajúci proces, napríklad na žiadosť HW komponenty. Systém prerušení umožňuje operačnému systému koordinovať paralelne prebiehajúce operácie a vykonávať paralelne bežiace programy.

1.1.6.2 Stavy procesu

Každý proces sa vzhľadom na svoju momentálnu aktivitu môže nachádzať v niektorom z nasledovných stavov:

- ***Nový (New)*** *- proces bol práve vytvorený*
- ***Prebiehajúci (Running)*** *- inštrukcia procesu začala byť vykonávaná*
- ***Čakajúci (Waiting)*** *- proces čaká na udalosť (napr. na dokoncenie I/O operácie alebo prijatie signálu)*
- ***Pripravený (Ready)*** *- proces čaká na pridelenie procesoru*
- ***Ukončený (Terminated)*** *- proces dokončil svoju činnost*

V rámci jednoho výpočtového systému (jediný CPU) môže byť vždy iba jediný proces vo fáze „prebiehajúci“, ale vo zvyšných fázach sa môže nachádzať teoreticky ľubovoľný počet ďalších procesov.
Názvy jednotlivých stavov procesu sa môžu odlišovať podľa terminológie zaužívanej vo „svete“ rôznych systémov, ale všetky operačné systémy musia poznať a obsluhovať všetky uvedené stavy.

Úlohy schedulera a dispečera pri organizovaní vykonávania procesov.

1.1.6.3 Mechanizmus obsluhy prerušení

Vykonávanie služieb OS a prepínanie medzi prebiehajúcimi procesmi je realizované systémom prerušení. Prerušenia sú definované ako kolekcia služieb BIOS-u. Mechanizmus obsluhy prerušení prebieha nasledovne:
Prerušenie môže byť inicializované

- hardwarovo (napríklad stlačením klávesy počítača, požadavkou sieťovej karty na komunikáciu so systémom); signál požiadavku prerušenia je prostredníctvom prerušovacieho vektoru IRQ odovzdaný operačnému systému, ktorý spustí obslužný podprogram na vyplnenie žiadosti
- softwarovo (požiadavkou aplikácie alebo operačného systému); požiadavka je inicializovaná príslušnou hodnotou kľúčových registrov.

OS zavolá podprogram na obsluhu žiadosti

1.1.6.3.1 Príklady obsluhy niektorých softwarových prerušení v systéme MS DOS:

Žiadosť na určenie verzie DOSu: Realizuje sa volaním služby ***30h***:
Do registra AH je zavedená hodnota 30h. Služba BIOSu 30h vykoná nasledovné: vynuluje obsahy registrov BX a CX, do AL zavedie číslo verzie OS a do AH číslo revízie OS.

Žiadosť na zobrazenie systémového dátumu:
Do registra AH je zavedená hodnota 2Ah. Služba BIOSu 2Ah vykoná nasledovné:
Do registra AL uloží hodnotu zodpovedajúcu označeniu dňa v týždni (0= nedela, 1= pondelok, ..., 6= sobota); do CX uloží rok, do DH uloží mesiac a do DL deň.

Žiadosť na nastavenie systémového dátumu:
Do registra AH je zavedená hodnota 2Bh. Služba BIOSu 2Bh vykoná nasledovné:
Aktivuje zobrazenie výzvy na zadanie systémového času a zabezpečí ukladanie hodnôt z klávesnice do registrov: do CX uloží rok, do DH mesiac a do DL deň. Návratová hodnota sa ukladá v AL. Ak boli zadané hodnoty dátumu korektné, v registri AL bude hodnota 00h a služba zabezpečí uloženie hodnôt z registrov CX, DH a DL do CMOS pamäti a prestaví systémový čas; ak boli hodnoty nesprávne (kombinácia čísla dňa a mesiaca bola nereálna, napríklad 31.2., alebo 2. 16., príp. rok bol zadaný mimo rozsah prípustných hodnôt), do AL sa zavedie hodnota FFh a ta zabezpečí, že hodnoty dátumu v registroch budú ignorované a monitor zobrazí chybové hlásenie „nesprávny dátum“.

Žiadosť na premenovanie súboru:
Do registra AH je zavedená hodnota 56h. Služba BIOSu 56h vykoná nasledovné:
V jednom pamäťovom segmente v OS musí byť uložený existujúci názov súboru, v ďalšom pamäťovom segmente bude uložený požadovaný názov nového súboru (obidva názvy súborov musia byť uložené vrátane ich absolútnej cesty). Názvy súborov sú v pamäti zapísané ako ASCII reťazce.
Dvojica registrov DS:DX (sú interpretované v režime segment:offset) obsahuje adresu začiatku segmentu, v ktorom je uložený existujúci názov súboru. Registre ES:DI predstavujú ukazovateľ na začiatok segmentu v operačnej pamäti, kde sa nachádza nové – požadované meno súboru a nová cesta.
Ak je operácia vykonaná úspešne a nové hodnoty sú systémom akceptované, nastaví sa CF (carry Flag) na hodnotu 0. Ak došlo ku chybe, bude hodnota CF = 1 a do registra AX bude službou 56h uložená hodnota kódu chyby („čo sa stalo“)

Zobrazenie znaku na monitore (textový mód)
Do registra AH sa zavedie hodnota 02h. Do registra DL sa uloží požadovaný znak. Služba 02h zabezpečí odoslanie znku z DL na štandardné výstupné zariadenie (monitor).

Načítanie znaku zo sériového portu (AUX, COM)
Do registra AH sa zavedie hodnota 02h. Služba zabezpečí načítanie znaku z príslušného rozhrania do refistra AL. Služba musí spolupracovať so službou 14h Sekvenčné čítanie.

Toto bol len výber niekoľkých služieb BIOS-u. Podrobný zoznam dostupných prerušení je možné nájsť na adresách:

http://en.wikipedia.org/wiki/BIOS_interrupt_call
http://www.ctyme.com/rbrown.htm
http://spike.scu.edu.au/~barry/interrupts.html

Obsluha prerušení v systémoch Windows, Unix a ďalších je už výrazne komplikovanejšia a vyžaduje si hlbšie znalosti z oblasti systémovéo programovania.

1.1.7 Správa pamäte

1.1.7.1 Prideľovanie pamäte

Funkcie modulu prideľovania pamäte:

- *Monitorovanie stavu operačnej pamäte, jej naplnenie, sledovanie každej adresovateľnej jednotky – či boľa pridelená, alebo či je voľná*
- *Stratégia prideľovania pamäte – plánovanie vybavovania požiadaviek na prideľovanie pamäte, špecifikovanie rozsahu a výber konkrétnej oblasti, ktorá bude procesu pridelená, plánovanie a príprava zdieľania, určenie pravidiel prístupu jednotlivých procesov ku zdielaným oblastiam pamäte*
- *Prideľovanie pamäťového miesta procesu*
- *Udržiavanie tabuľky informácií o pridelení príslušného pamäťového miesta danému procesu*
- *Odoberanie pamäťového priestoru procesu. Proces sám môže pamäť uvoľniť, alebo o odobratí pamäte procesu rozhodne modul prideľovania pamäte*

Prideľovanie operačnej pamäti je pomerne komplikovaný proces, ktorý musí zohľadňovať rôzne aspekty:

- *náročnosť jednotlivých procesov na pamäť*
- *kapacitu dostupnej fyzickej pamäte*
- *schopnosť technických prostriedkov adresovať pamäť (šírka adresnej zbernice, metódy adresovania podporované procesorom)*
- *schopnosť operačného systému obslúžiť celú inštalovanú pamäť*
- *schopnosť operačného systému obsluhovať viacero súbežne spustených procesov a zabezpečiť ochranu inštrukcií a dát před neoprávneným prístupom iných procesov*

Existuje viacero techník prideľovania pamäte

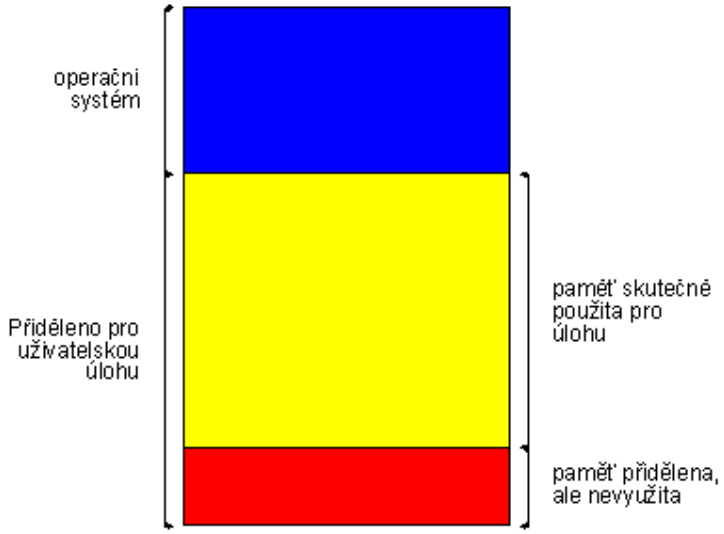
- *prideľovanie jediného súvislého bloku pamäte v operačnej pamäti*
- *prideľovanie pamäte po sekciách*
- *stránkovanie*
- *virtuálna pamäť so stránkovaním*
- *segmentácia*
- *virtuálna pamäť so segmentovaním*
- *stránkovanie kombinované so segmentovaním a s využívaním virtuálnej pamäte*

1.1.7.2 Hlavné metódy správy pamäte operačným systémom

1.1.7.2.1 Prideľovanie jediného súvislého bloku pamäte

Táto technika je použiteľná iba u jednoduchých systémov a jednoduchých programovacích techník. Použitie tejto techniky nepodporuje viacúlohové a viacužívateľské režimy. Užívateľ zadá úlohu, tá sa vykonáva postupne po krokoch, pričom z hľadiska systému v danom okamihu v reálnom čase krok úlohy predstavuje samotnú úlohu, úloha generuje jediný proces ktorý je postupne vykonávaný, ak proces vygeneruje iný proces, musí byť rodičovský proces pozastavený do ukončenia synovského procesu.

Vlastnosti:



obrázok – prideľovanie jediného súvislého bloku pamäte

- Pamäť pozostáva z troch súvislých oblastí
  - časť obsadená operačným systémom
  - časť obsadená úlohou
  - nevyužitá oblasť pamäte

Výhody

Hlavnou výhodou je jej jednoduchosť .Táto technika nevyžaduje žiadne špeciálne technické prostriedky, dokáže pracovať aj s malou pamäťou a nekladie žiadne zvláštne nároky na schopnosti programátora.

Nevýhody

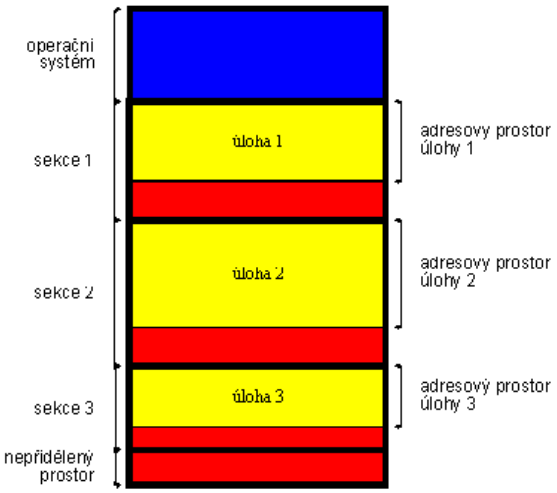
Časť pamäte ostáva nevyužitá
Ak je úloha v stave čakajúca, ostávajú nevyužité ako pamäť , tak aj procesor (stav čakajúca môže predstavovať bežne okolo 70% z celkového času
Do pamäti sú zavedené aj tie časti programu, ktoré nebudú vykonávané (vetvenie).
Ak sa ukáže, že nároky úlohy na pamäť sú väčšie, ako je dostupný adresný priestor, úloha nemôže byť vykonaná

1.1.7.2.2 Prideľovanie pamäte po sekciách

Dostupný pamäťový priestor neobsadený operačným systémom sa rozdelí na samostatné úseky – sekcie (memory partions). Každá sekcia má konštantnú veľkosť, je samostatne obsluhovaná plánovačom a každej sekcii môže byť pridelená iba jediná úloha
V rámci tejto techniky existujú dve metódy :

- *statické prideľovanie sekcií – pamäť sa rozdelí na pevne definované sekcie už pri štarte operačného systému*
- *dynamické prideľovanie sekcií – jednotlivé sekcie vytvára operačný systém za behu tak, aby jednotlivé sekcie svojou veľkosťou a svojimi vlastnosťami čo najlepšie zodpovedali potrebám jednotlivých úloh.*

Technika prideľovania pamäte po sekciách sa používa u najjednoduchších multiprogramových systémov. Oproti predchádzajúcej technike pribúda iba správa viacerých sekcií pamäte a potreba zabezpečiť tieto sekcie pred neoprávaným prístupom inej úlohy, u dynamického prideľovania sekcií potom úlohy vyhľadať v pamäti a pripraviť čo najoptimálnejší pamäťový priestor pre sekciu.



Efektivita systému je nízka. Požiadavky na technické prostriedky sú minimálne, najnáročnejšou úlohou je ochrana sekcií pamäte pred neoprávevým prístupom a komplikované ošetrovanie prechodu mezi dvoma procesmi . ďalšou nevýhodou je vysoký stupeň fragmentácie dát v pamäti.

obráz k – prideľovanie pamäte po sekciách

1.1.7.2.3 Stránkovanie pamäte

Najväčšou nevýhodou techniky sekcií je, že sekcia musí tvoriť v pamäti súvislú oblasť. Túto požiadavku je možné obísť používaním techniky stránkovania.
Princíp stránkovania:
Adresný priestor každej úlohy sa rozdelí na rovnaké úseky – stránky.
Priestor operačnej pamäte sa rozdelí na rovnako veľké úseky – tzv . fyzické stránky, stránkové rámy.
Pomocou technických prostriedkov na transformáciu adries je potom možné každú stránku úlohy vložiť do ľubovoľného stránkového rámu.
Jednotlivé bloky operčnej pamäte – fyzické stránky – nemusia na seba nadväzovať, nemusia v pamäti vytvárať súvislú oblasť, ale vo vzťahu ku užívateľskému programu môžu viaceré stránky vytvárať jediný logický súvisle adresovaný celok.
Týmto princípom sa automaticky rieši aj problematika fragmentovania pamäti – hoc vo fyzickej pamäti je úloha fragmentovaná do viacerých oblastí pamäti, z pohľadu úlohy žiadna fragmentácia nie je zrejmá – logické adresovanie vytvára súvislý adresný priestor.
Rovnako ako u sekcií nemajú transformácie adries žiadny vplyv na užívateľskú úlohu.
Kľúčovú úlohu pri stránkovaní pamäte zohráva tabuľka priradenia stránok pamäťovým blokom – Page Map Tables. Tieto sú realizované prostredníctvom špeciálnych registrov obsluhovaných priamo procesorom (napríklad u procesorov Intel 80386 a všetkých nasledujúcich túto úlohu zabezpečuje Page Unit), alebo túto úlohu môže vykonávať vyhradená časť operačnej pamäte
Dôležité pre efektivitu práce systému je zvoliť optimálnu veľkosť stránky. Príliš veľká stránka je príčinou neúmernej fragmentácie, príliš malá stránka si vyžaduje množstvo registrov na obsluhu stránok , spotrebováva príliš veľa systémových zdrojov a spomaľuje beh systému.

1.1.7.2.4 Principiálna schéma správy pamäte stránkovaním:

	pamäť úlohy	Page of Task
úloha 3	16384	4
	12288	3
	8192	2
	4096	1
	0	0
úloha 2	8192	2
	4096	1
	0	0
úloha 1	12288	3
	8192	2
	4096	1
	0	0

Page unit		
Task	Page	Frame
T3	4	10
	3	9
	2	8
	1	7
	0	4
T2	2	11
	1	3
	0	2
T1	3	6
	2	5
	1	1
	0	0

Frame of RAM	RAM
16	65536
15	61440
14	57344
13	53248
12	49152
11	45056
10	40960
9	36864
8	32768
7	28672
6	24576
5	20480
4	16384
3	12288
2	8192
1	4096
0	0

Komentár ku obrázku:

Stĺpec vľavo obsahuje jednotlivé úlohy a stránky úlohy. Prvá úloha zaberá 4 stránky (stránky č. 0, 1, 2, 3), druhá úloha 3 stránky a tretia úloha 5 stránok. Každá stránka má veľkosť 4 kB.

Stĺpec vpravo znázorňuje operačnú pamäť RAM rozdelenú na rámce (Frame of RAM) a obsadenie rámcov jednotlivými stránkami spustených úloh. Rámce sú zadefinované s veľkosťou 4 kB.

Rovnaká veľkosť stránok úlohy a rámcov operačnej pamäte je nutnou podmienkou pre správne fungovanie systému.

Stredný stĺpec predstavuje stránkovaciu jednotku procesora. Táto jednotka je súčasťou HW CPU, konkrétne jeho adresovacej jednotky.
Úlohou stránkovacej jednotky je priradiť jednotlivým stránkam konkrétnej úlohy príslušné rámce v RAM a udržiavať prehľad priradenia stránok úloh ku rámcom v RAM.

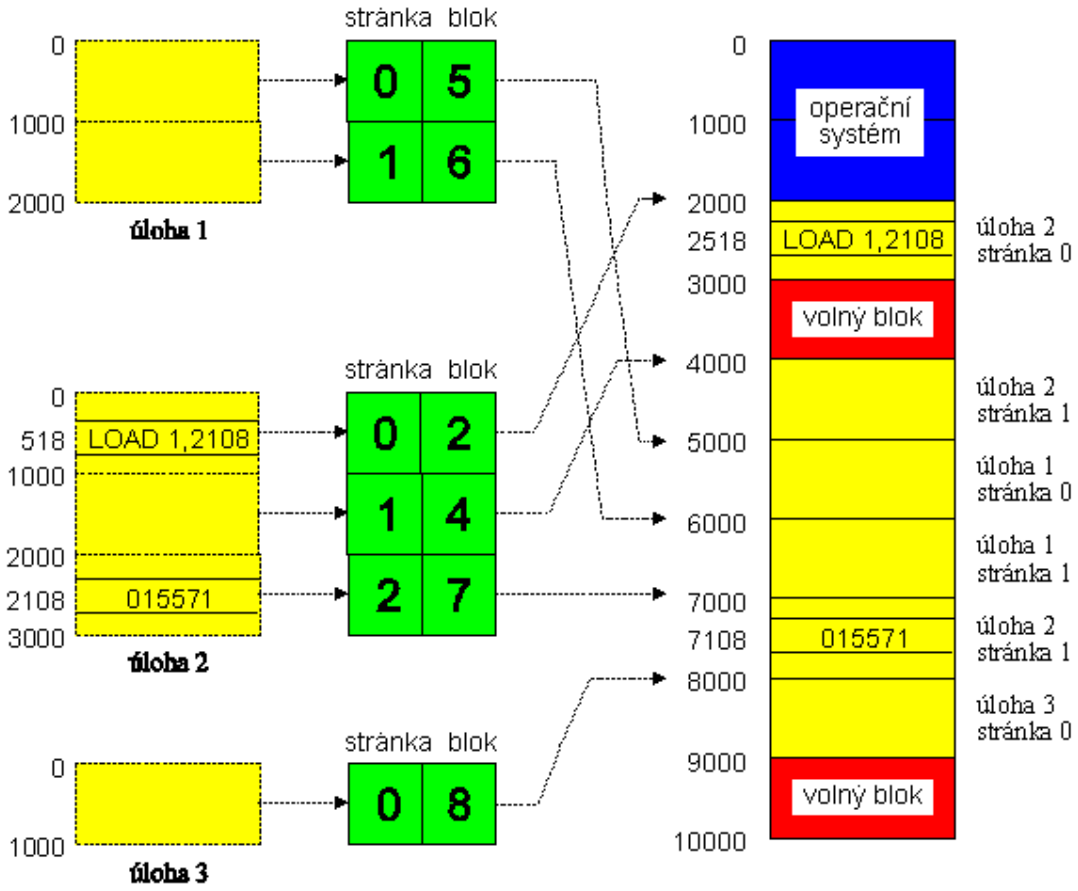
Príklad: Na akej fyzickej adrese sa nachádza 8201-tá inštrukcia úlohy 1?

Najskôr vyriešime relatívnu adresu: prvá inštrukcia sa nachádza na nultej adrese, 8201-vá inštrukcia sa nachádza na relatívnej adrese 8200. Táto adresa sa nachádza na stránke č. 2 prvej úlohy. Presne na adrese 8200 – 8192 = 8, teda ide o adresu 0008 (dec) stánky č. 2 úlohy č. 1.
Pohľadom do stránkovacej jednotky zistíme, že stránka č. 2 úlohy č. 1 je umiestnená v rámci č. 5 fyzickj pamäte.
Adresa rámca č. 5 fyzickej pamäte, teda fyzická adresa stránky č. 2 úlohy 1, je od 20480 do 24575.
Adresa 0008 v rámci č. 5 fyzickej pamäte má fyzickú adresu 20488. Hľadaná inštrukcia má vo fyzickej pamäti RAM adresu 20488 (dec).

Problematika fragmentácie:

Stránky úlohy nie sú spravidla v RAM usporiadané kontinuálne, nezaberajú súvislý priestor. Sú fragmentované. Ku fragmentácii dochádza pri dynamickom spúšťaní úloh a ich ukončení a spúšťaní ďalších úloh. Stránky práve spúšťanej úlohy sú priradené prázdnym rámcom, ktoré ostali nevyužité po ukončení behu niektorých úloh. Prázdne rámce však nevytvárajú súvislý priestor, preto aj stránky novej úlohy sú ukladané nespojito, fragmentovane.

Operačný systém v tomto prípade komunikuje s operačnou pamäťou prostredníctvom stránkovacej jednotky, nie priamo s operačnou pamäťou. Z pohľadu operačného systému sa teda stránky javia ako spojitý, na seba nadväzujúci priestor – operačný systém žiadnu fragmentáciu „nevidí“.



obrázok – prideľovanie pamäte po stránkach

Komentár k obrázku:

Veľkosť stránky je 1000 B.

Celková veľkosť pamäte je 10 000 B.

Adresy 0 až 1999 (fyzicky) obsadzuje operačný systém.

úloha 1 má potrebu pamäte 2000B

obsadzuje dve stránky označené 0 a 1. PMT im priraďuje bloky 5 a 6, tzn. fyzický adresný priestor 5000 – 6999 B.

úloha 2 má potrebu 3000B

obsadzuje tri stránky – 0, 1 a 2. PMT im priraďuje bloky 2 (fyzicky 2000 – 2999), 4 (fyzicky 4000 – 4999) a 7 (fyzicky 7000 – 7999). Fyzicky sú bloky adresované ako nesúvislá oblasť (nastáva fragmentácia), ale z pohľadu úlohy žiadna fragmentácia nenastáva – úloha je adresovaná súvisle od adresy 0 po adresu 2999 (logické adresovanie).

úloha 3 potrebuje 1000 B

využíva jedinú stránku s č. 0

tej je priradený blok 8, teda fyzicky oblasť 8000 až 8999 B

ostávajú dva voľné bloky – 3000 až 3999 a 9000 až 9999, ktoré môžu byť priradené ďalšej úlohe ako logicky súvislá oblasť s kapacitou 2000 B.

Modul prideľovania pamäte plní nasledovné úlohy:

Monitorovanie stavu obsadenia pamäti sa realizuje prostredníctvom stránok:

a) Pre každú úlohu je vygenerovaná jedna tabuľka stránok, každej stránke zodpovedá jeden záznam
b) Systémová tabuľka bloku pamäti (Memory Block Table) priraďuje každému bloku pamäti hodnotu „voľný“ alebo „použitý“.
c) Rozhodovanie o prideľovaní pamäti vykonáva plánovač úloh – pridelí sa množina tých voľných blokov pamäti, ktoré sa nájdu najskôr
d) Mechanizmus samotného pridelenia pamäti spočíva v zavedení stránok do priradených blokov a aktualizujú sa záznamy v tabuľke stránok a tabuľke blokov.
e) Uvoľnenie pamäti sa vykoná jednoducho zapísaním hodnôt „voľný“ do príslušných záznamov v tabuľke blokov.

Nevýhody metódy stránkovania

1. Výpočtový systém musí disponovať prostriedkami na stránkovanie
2. Uchovanie tabuliek vyžaduje časť operačnej pamäte
3. Časť výkonu procesora musí byť obetovaná na obsluhu stránkovacieho mechanizmu

   Poznámka: *Moderné procesory bývajú priamo vybavené jednotkami na obsluhu stránkovacieho mechanizmu (Page Unit), ktoré pracujú autonómne – samotné výpočtové jednotky nie sú týmito úlohami zaťažované.*
4. Fragmentácia pamäte je síce eliminovaná, ale vzniká tzv. vnútorná fragmentácia na základe neúplného zaplnenia stránok dátami (čím väčšie stránky, tým viac kapacity pridelenej stránke ostáva nevyužitého; čím menšie stránky, tým náročnejšia réžia na ich údržbu)
5. Nerieši sa ani problém, keď voľný priestor v pamäti je menší, ako sú nároky čakajúcej úlohy.
6. Nedá sa adresovať viac pamäte, ako je fyzicky inštalovaná operačná pamäť
7. Priestor v pamäti často zaberajú údaje, ktoré sa používajú zriedka alebo dokonca nikdy.

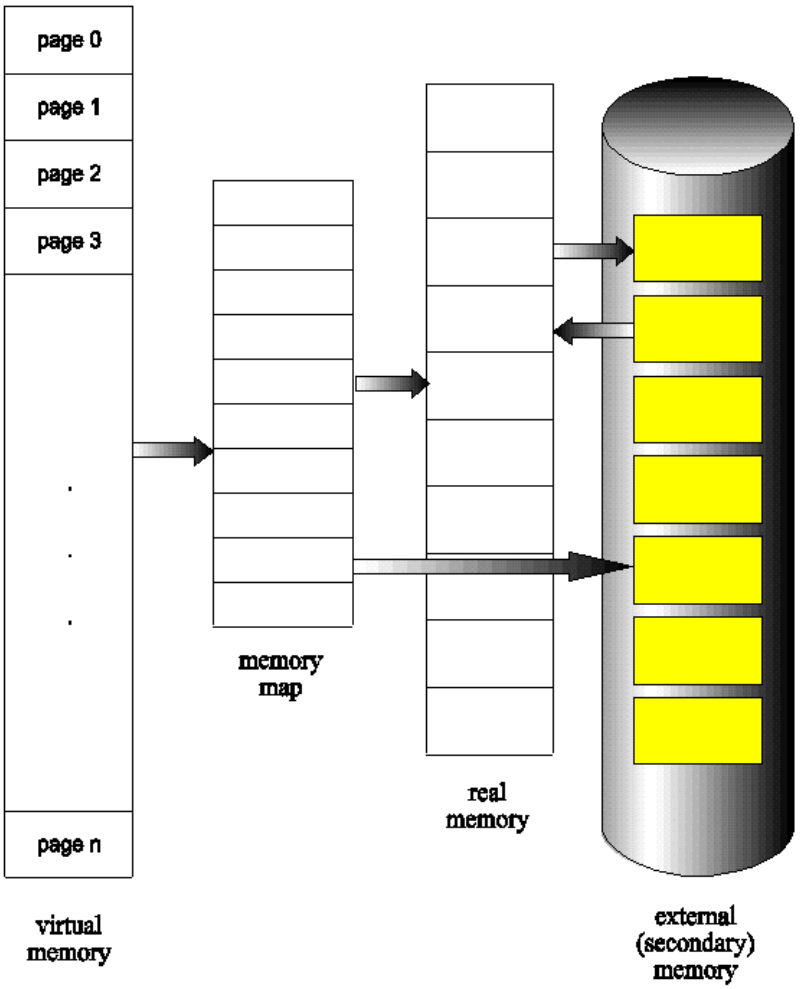
Stránkovanie - zhrnutie

- OS sa stará o tri základné tabuľky:
  - tabuľka úloh (Job Table)
  - tabuľka blokov (Memory block table)
  - tabuľka stránok (Page Memory Table)
- V tabuľke úloh je každej úlohe priradená položka obsahujúca údaje o umiestnení a veľkosti jej tabuľky stránok a stavové informácie týkajúce sa týchto stránok
- Tabuľka bloku udáva stav každého bloku pamäte (voľný – použitý)

Ak niektorú stránku používa viacero procesov, použije sa technika prekrývania stránok

1.1.7.2.5 Virtuálna pamäť – stránkovanie na žiadosť

Rieši problém nedostatku inštalovanej fyzickej pamäte.
Okrem fyzickej pamäte je simulovaný ďalší adresný priestor.
Princíp je v tom, že vo fyzickej operačnej pamäti sa nachádzajú iba tie stránky, ktoré obsahujú aktuálne potrebné dáta. Momentálne nevyužívané stránky sa odkladajú na pevný disk vo forme swapovacieho súboru.
Potrebné stránky sú opätovne zavedené do fyzickej operačnej pamäte na žiadosť systému; namiesto zavedenej stránky sa musí obetovať iná stránka, ktorá sa odloží do swapovacieho súboru na disk. Načítanie odloženej stránky do pamäte je pomerne zdĺhavý proces, preto je treba predchádzať situáciám, aby práve odložená stránka bola vzápätí načítaná opäť do fyzickej operačnej pamäte – technika kladie vysoké nároky na plánovač pamäte.
Keďže pri tejto technike bývajú celé úlohy odložené vo forme stránok na disku a potrebné stránky sú zavádzané do pamäte iba na žiadosť systému, nazýva sa táto technika stránkovanie na žiadosť.



obrázok – stránkovanie pamäte s využitím virtuálnej pamäte

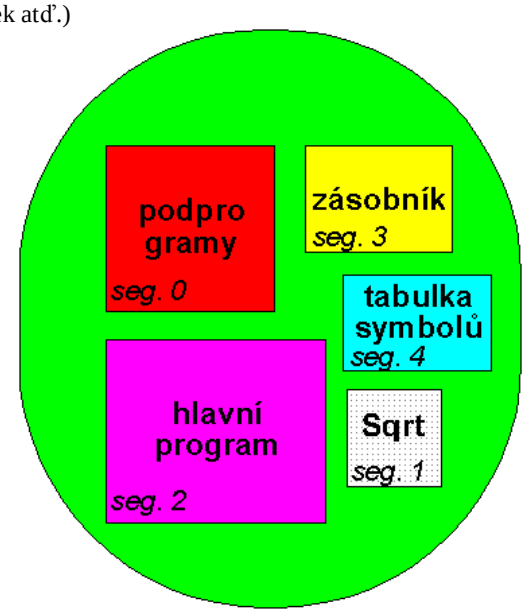
Technické prostriedky

V prvom rade musí byť technické prostriedky schopné adresovať dostatočne veľký adresný priestor (napríklad procesor i286 dokáže adresovať 16 MB fyzického adresného priestoru (24 bitová adresa), ale s využitím segmentovania a virtuálnej pamäte dokáže adresovať až 1 GB. Procesor i386 (32 bitová adresa) dokáže fyzicky adresovať 4 GB, ale virtuálne dokáže adresovať až 64 TB).
Oproti jednoduchému stránkovaniu je nutné zabezpečiť pri správe pamäte ďalšie dôležité funkcie:
V tabuľke stránok zaviesť bit, ktorý bude rozlíšovať, či daná stránka je práve prítomná v operačnej pamäti
Mechanizmy prerušenia musia byť rozšírené o ošetrenie situácie, keď proces žiada prístup ku dátam, ktoré sú práve odložené na disku (OS musí zabezpečiť odloženie procesu do stavu „čakajúci“, pretože proces načítania stránky z disku je pomerne zdĺhavý)
Je potrebné vytvoriť a sledovať štatistiku používania jednotlivých stránok a zvoliť vhodnú stratégiu rozhodovania, ktoré stránky udržiavať v operanej pamäti a ktoré v prípade potreby odložiť na disk.

1.1.7.2.6 Segmentácia pamäte

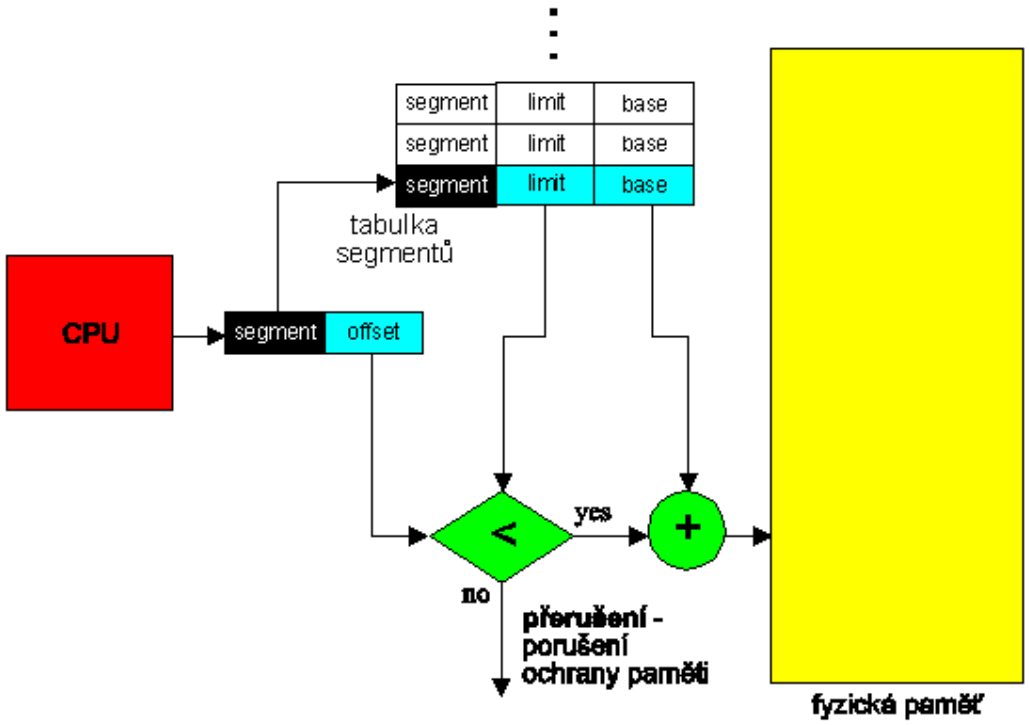
Princípom segmentácie pamäti z pohľadu technických prostriedkov je vytváranie adresy prostredníctvom vzájomnej kombinácie údajov z dvoch adresných registrov: segmentu a offsetu.
Na úspešné používanie tejto techniky je nevyhnutné, aby procesor podporoval túto adresovaciu techniku a dokázal z dvojrozmernej adresy typu segment – offset vypočítať fyzickú adresu.
Princíp vytvárania adresy zo segmentu a offsetu podporujú napríklad všetky procesory Intel radu 80x86 (vrátane všetkých modelov Pentií).

Z pohľadu úlohy je segmentácia prostriedkom, ktorý umožňuje vytvoriť rôzne veľké bloky pamäte, ktoré budú logicky zoskupené podľa typu používaných dát (segment s hlavným programom, segment zásobníka, segment podprogramov, segment tabuliek atď.)



obrázok – segmentácia pamäte

obrázok –mechanizmus prideľovania segmentovanej pamäte



Adresný priestor každej úlohy je tvorený viacerými segmentami rôznej veľkosti.
Segment je možné definovať ako logické zoskupenie informácií rovnakého druhu, napr. hlavný program, podprogramy, dátová oblasť apod.

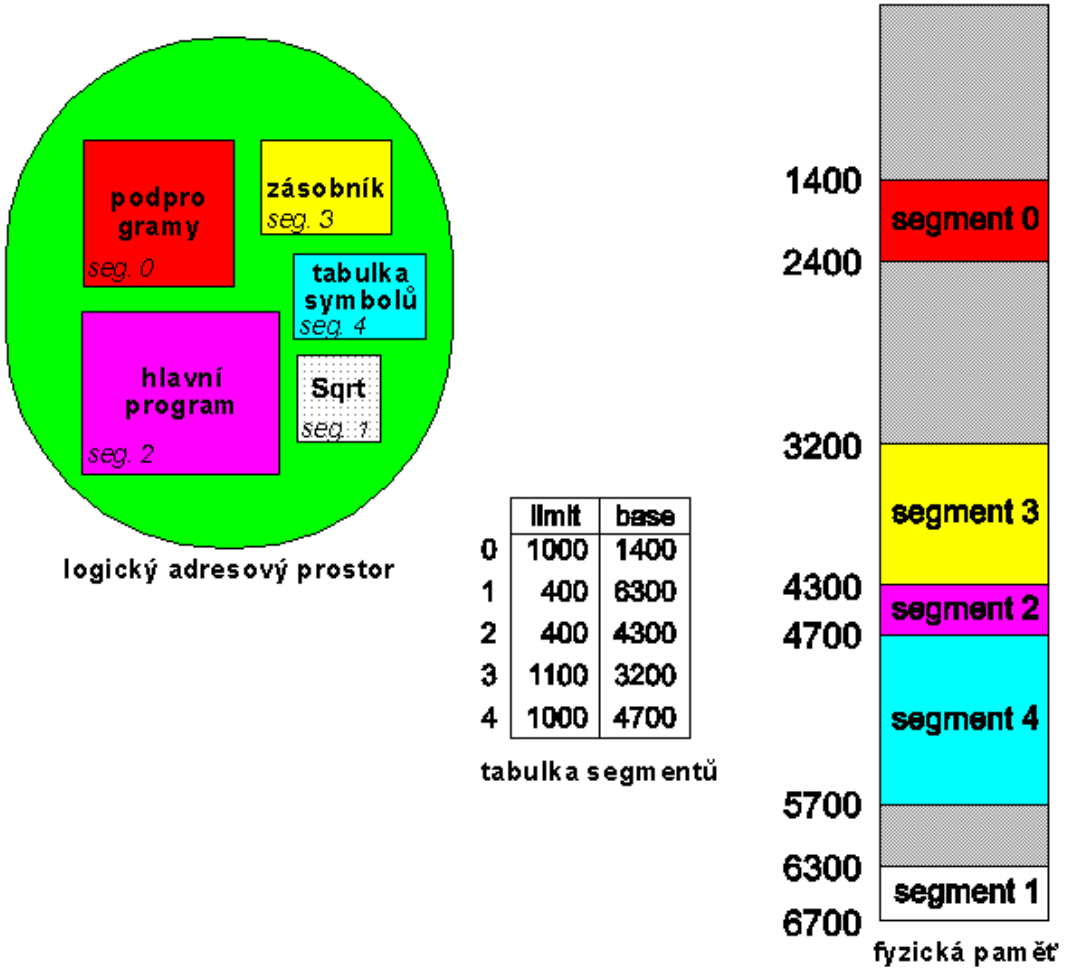
Logická adresa sa skladá z adresy segmentu a adresy offsetu. Offset nemôže byť väčší, ako je dĺžka segmentu (v opačnom prípade je vyvolaná chyba poruchy ochrany pamäte). Súčtom adries posunutého segmentu a offsetu získa systém reálnu fyzickú adresu.
Základná veľkosť segmentu je daná technickými prostriedkami – konštrukciou konkrétneho procesora.
Táto technika je podrobne rozoberaná v predmete výpočtová technika v treťom ročníku pri analýze technických prostriedkov procesorov triedy Intel 8086.

Ochrana a zdieľanie segmentu

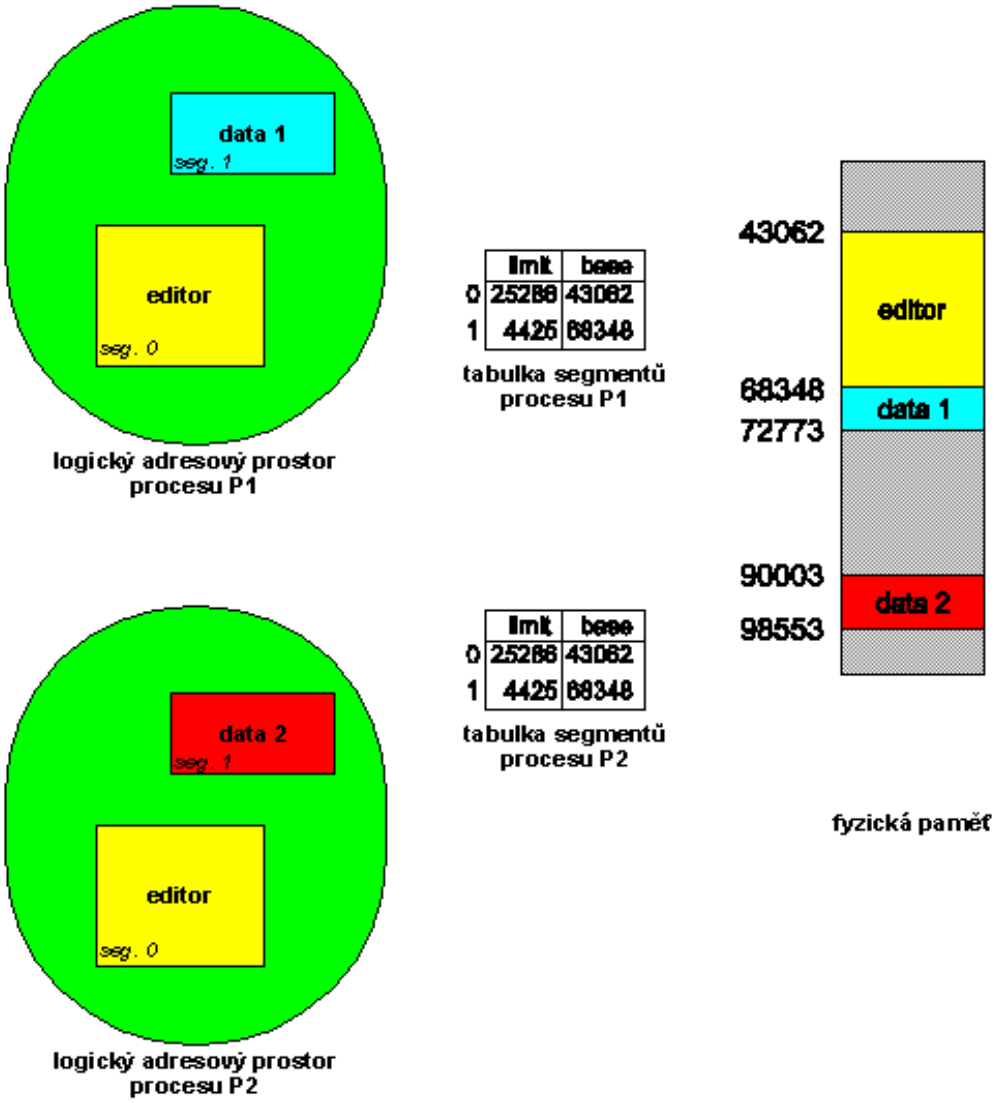
Segmenty sú systémom chápané ako samostatné jednotky a podľa toho sa k nim pristupuje
Sú definované segmenty obsahujúce programový kód, segmenty obsahujúce dáta a operačný systém k nim podľa toho pristupuje
Segmenty môžu mať rôzne príznaky, ako read-only alebo execute-only, prípadne execute-only. Systém kontroluje tieto príznaky skôr, kým ku segmentu pristúpi
memory-mapping hardware automaticky kontroluje, či nebola prekročená dĺžka segmentu a zabráni neoprávnenej manipulácii mimo hranice segmentu. Tieto kontroly sa najčastejšie uplatňujú pri práci s poliami.
Segmenty môžu byť zdielané viacerými úlohami

Problémy a nevýhody segmentovania

Rovnako ako u stránkovania musí aj pri segmentovaní plánovač úloh nájsť a alokovať dostatočný pamäťový priestor pre všetky segmenty úlohy. Na rozdiel od stránkovania je dĺžka segmentu rôzna.
Pri fragmentácii pamäťového priestoru (existuje veľmi veľa veľmi malých voľných fragmentov pamäte) je možné použiť techniku zhusťovania. Keďže segmenty predstavujú dynamicky adresované jednotky, je možné ich v pamäti presúvať a eliminovať voľné miesta. Defragmentáciu môž esystém vykonávať ako samostatný proces s malou prioritou – teda vždy, keď CPU je nevyužitý a čaká na pridelenie nejakého procesu.
Rovnako ako u stránkovania platí, že čím väčšie segmenty, tým je problém fragmentácie výraznejší.



obrázok – mechanizmus prideľovania segmentovanej pamäte



obrázok – prideľovanie segmentovanej pamäte pri multiprocesingu

1.1.7.2.7 Virtuálna pamäť vytvorená systémom so segmentovaním pamäte

Mechanizmus segmentácie je možné využiť rovnako ako stránkovanie na vytváranie virtuálnej pamäte v podobe swapovacieho súboru na pevnom disku.
Túto techniku využívali napríklad niektoré operačné systémy na procesore Intel 80286, ktorý mal pomerne silnú podporu segmentácie, ale nepodporoval stránkovanie.
Mechanizmy vytvárania a správy virtuálnej pamäte sú obdobné, ako v prípade stránkovania.

1.1.7.2.8 Segmentácia so stránkovaním

Segmentácia aj stránkovanie majú určité výhody aj nevýhody. Mnohé procesory (napr. radu Motorola 68000 a Intel 80x86 od modelu 386) podporuje kombináciu obidvoch techník.
V takom prípade prebieha vždy najskôr segmentácia a segmenty sa potom spracujú ešte stránkovaním.

1.1.8 Správa súborov

1.1.8.1 Úvod

Pre väčšinu používateľov je systém súborov najviditeľnejšou súčasťou operačného systému. Sprístupňuje všetky programy aj dáta, uložené v pamäti počítača.

1.1.8.2 Súborové systémy

Systém súborov sa skladá z

- množiny súborov – súbory obsahujú dáta
- adresárovej štruktúry – organizuje všetky súbory v systéme a sprístupňuje informácie o súboroch
- partitions – používajú sa na logické či fyzické oddelenie veľkých adresárových štruktúr (napr. diskov)

Súbory sú základné jednotky na správu dát uložených na vonkajších pamätiach, s ktorými pracuje operačný systém.
Adresárová štruktúra predstavuje logický systém organizácie súborov
Partitions využívajú niektoré operačné systémy na veľké celky nadradené adresárovým štruktúram

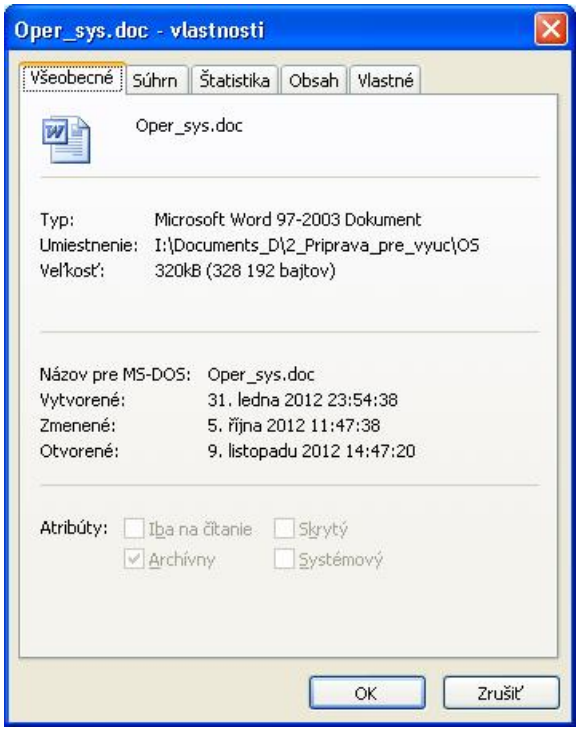
1.1.8.3 Koncepcia súboru

Počítače môžu uchovávať informácie na rôznych médiách, ako sú magnetické disky, magnetické pásky, optické disky, magnetooptické disky apod. Úlohou operačného systému je abstrahovať od konkrétneho fyzického média a udržiavať jediný logický prehľad o uložených informáciách.
Súbor je pomenovaná množina príbuzných informácií, ktorá je uložená na pamäťovom médiu. Všetky uložené dáta musia byť organizované v súboroch – systém nedokáže pracovať s dátami mimo súbory. Dátové súbory môžu byť binárne, alfanumerické, môžu mať voľnú formu ako napr. textové súbory, či môžu mať špeciálne usporiadanie ako napríklad databázové súbory či obrazové a multimediálne súbory.
Pomenovaný súbor je nezávislý na systéme, ktorý ho vytvoril, na autorovi (užívateľovi), aj na procese, ktorým bol vytvorený. Stáva sa autonómnou jednotkou.
Z iného pohľadu je súbor postupnosť bitov, pričom ich význam je daný programátorom a spôsobom interpretácie tejto bitovej postupnosti príslušným programom. Koncepcia súboru je tak veľmi obecná.
Štruktúra súboru je daná jeho typom. Textový súbor je postupnosť znakov, organizovaných pomocou formátovacích značiek, objektový súbor je postupnosť bytov do štruktúry, ktorú dokáže interpretovať linker, zdrojový súbor je postupnosť príkazov, funkcií a procedúr, ktorým „rozumie“ kompilátor príslušného jazyka a spustiteľný súbor je sekvencia príkazov, ktoré po zavedení do operačnej pamäti dokáže operačný systém interpretovať ako sekvenciu príkazov pre procesor počítača.

1.1.8.4 Atribúty souboru

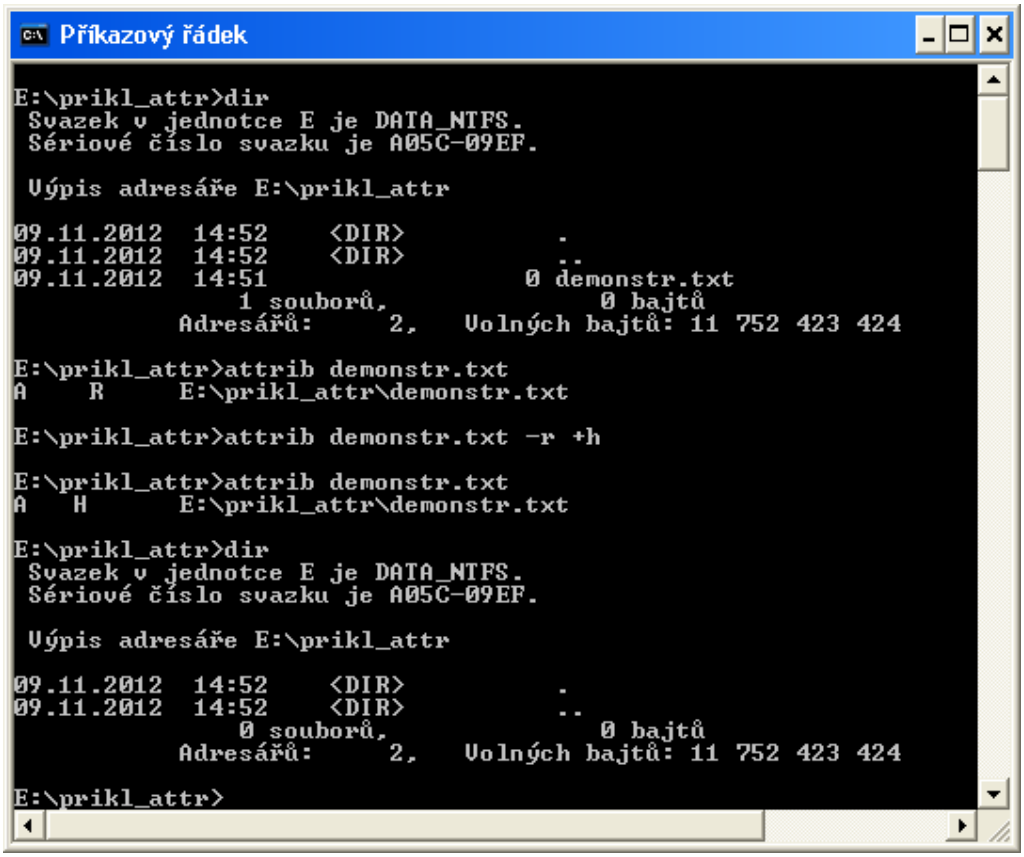
Atribúty súboru slúžia na bližšiu špecifikáciu ich vlastností. V rôznych operačných systémoch sa používajú rôzne atribúty, medzi typické však patrí:

- meno
- typ
- veľkosť
- umiestnenie v adresárovom systéme (cesta ku súboru)
- veľkosť
- čas vytvorenia, príp. čas poslednej úpravy či čas posledného prístupu ku súboru
- vlastník
- ochrana a prístupové práva



Zobrazenie atribútov súboru v OS Windows

Príklad zmeny atribútov súboru pomocou príkazov CMD OS Windows:



Príklad nastavenia prístupových práv pod OS Linux:

- rw-r-x--x 1 student operator 3775 Jun 5 15:45 XF86Config.old

```
-rw-r-x--x   1 student     operator    3775 Jun  5 15:45 XF86Config.old
```

	Owner	Group	Every (world)
D	r w x	r w x	r w x
	1 1 0	1 0 1	0 0 1
	6	5	1

D – directory
R – read only
W – write
X –eXecute

1.1.8.5 Operácie so súbormi

Principiálne existuje šesť rôznych operácií pre prácu so súbormi.

- vytvorenie súboru
- zápis do súboru
- čítanie zo súboru
- premenovanie, resp. premiestnenie súboru
- vyprázdnenie súboru
- zmazanie súboru

Pre každú z uvedených operácií má operačný systém priradené príslušné systémové volanie. Každá z uvedených operácií predstavuje niekoľko čiastkových úloh.

- **vytvorenie súboru** predpokladá dva kroky: vyhľadanie dostatočne veľkého priestoru na pamäťovom médiu a vytvorenie záznamu o novom súbore v adresárovom systéme. Záznam musí obsahovať predovšetkým meno súboru a jeho umiestnenie.
- **zápis do súboru** je podmienený uvedením mena súboru a špecifikovaním miesta v súbore, kam má byť záznam umiestnený. Na určenie miesta slúži ukazovateľ zápisu, ktorý musí byť správne nastavený vždy, keď sa má do súboru zapisovať Alternatívou či skôr doplňujúcou funkciou je *append* – pridanie záznamu na koniec súboru, *merge* – spojenie súborov a pod.
- **čítanie zo súboru** taktiež predpokladá zadanie mena súboru, informáciu o tom, kde v pamäti sa nachádza ďalší blok súboru a ukazovateľ čítania. Mnohé systémy namiesto ukazovateľa zápisu a ukazovateľa čítania používajú iba jediný, a to ukazovateľ aktuálnej pozície. Niekedy sa za špeciálnu operáciu považuje aj premiestnenie ukazovateľa v súbore a pripravenie novej aktuálnej pozície pre ďalší prístup ku súboru.
- **premenovanie súboru** predpokladá vyhľadanie záznamu o atribútoch súboru a patričnú úpravu v záznamoch. Obdobou premenovania súboru je jeho premiestnenie, kopírovanie, za určitých okolností aj tlač súboru.
- **vyprázdnenie súboru** znamená, že súbor ako taký ostáva zachovaný vrátane mena, umiestnenia, práv aj ďalších atribútov, ale obsah súboru sa zmaže. Veľkosť súboru sa nastaví na nulu.
- **zmazanie súboru** si vyžaduje vyhľadanie informácie o názve a umiestnení súboru , zmazanie tejto informácie a označenie miesta obsadeného súborom ako voľného

Uvedené operácie a ich podskupiny patria medzi hlavné operácie so subormi. ďalšou skupinou operácií sú operácie pre prácu s atribútmi súborov – ich čítanie a príp. nastavenie. Ďalej je nevyhnutné zabezpečiť operácie vyhľadania súboru, otvorenie súboru, udržiavanie prehľadu o otvorených súboroch a operácie uzatvorenia súboru po ukončení práce. Komplikovanejšia situácia nastáva v multiužívateľkých systémoch, kedy je nevyhnutné zabezpečiť obsluhu otvoreného súboru viacerými užívateľmi pre základné režimy *read* a *write*.

1.1.8.5.1 typy súborov

Jednou z hlavných úloh operačného systému je rozoznávanie typov súborov. V takom prípade je možné mnohé úlohy automatizovať a zabezpečiť aj prijateľný stupeň ich ochrany .
Typ súboru okrem iného udáva aj jeho vnútornú štruktúru, a teda spôsob jeho interpretácie programovým vybavením. Operačný systém môže pre každý typ súboru priradiť príslušnú aplikáciu, ktorá vie s jeho štruktúrou pracovať.
Jednou z možností označenia typu súboru je vloženie typu súboru priamo ako súčasti jeho názvu. Takto to zabezpečuje napríklad MS DOS a jeho varianty, Windows a niektoré ďalšie. Operačný systém OS Mac Apple Macintosh používa pre označenie typu súboru atribút „typ“. Unixové systémy rozoznávanie typu súboru podľa častí v názve nepoznajú, ale na identifikáciu typu používajú tzv. magic number, ktorý býva umiestnený na začiatku súboru a ktorý môže, ale nemusí byť súčasťou súboru.

1.1.8.6 Metódy prístupu k súborom

Ďalším problémom je metóda, ktoru sa pristupuje ku súboru . Niektoré systémy používajú iba jedinú metódu prístupu ku súboru, iné systémy, napr. strediskové počítače IBM podporujú viacero prístupových metód v rámci jediného systému.

1.1.8.6.1 Sekvenčný prístup

Je to najjednoduchšia metóda prístupu ku súboru. Je založená na modelu páskového pamäťového média. Dáta zo súboru sú spracovávané postupne, po záznamoch, po jednotlivých bytoch. Tento prístup používajú napr. textové editory, kompilátory a pod.

1.1.8.6.2 Priamy prístup k súboru

Súbor pozostáva s logických záznamov pevnej dĺžky, čo umožňuje operačnému systému načítavať záznamy po celých blokoch, čím sa výrazne urýchľuje načítavanie dát. Táto metóda je založená na diskovom princípe záznamu. V moderných OS býva blok dát stotožnený so sektorom alebo clusterom.

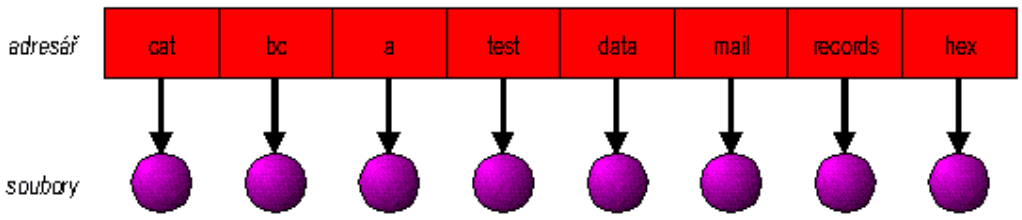
1.1.8.7 Adresárová štruktúra

1.1.8.7.1 Adresárový systém

Informácie o všetkých súboroch sú uložené v adresárovej štruktúre, ktorá býva uložená na príslušnom pamäťovom médiu.
Súborový systém môže byť značne rozsiahly, môže obsahovať milióny súborov na diskoch s kapacitou rádovo stovky gigabyte. Také množstvo súborov je nevyhnutné vhodným spôsobom prehľadne organizovať.
Organizácia súborov je spravidla rozdelená do dvoch úrovní
partitions (volumes) – predstavuje spravidla úroveň logického disku. S každou partition pracuje operačný systém ako so samostatnou nezávislou oblasťou. Niekedy obsahuje fyzický disk viacero partitions, menej častým prípadom je rozprestretie jedinej partition cez viacero fyzických diskov.
adresare (directories) – uchovávajú informácie o mene a atribútoch súboru. Adresár musí umožniť vložiť do štruktúry nový súbor, vyhľadať požadovanú položku, zmazať súbor atď.

1.1.8.7.2 Adresár jednej úrovne

Najjednoduchšia adresárová štruktúra je jednoúrovňová. Všetky súbory sú uložené v jedinom adresári. Táto štruktúra je jednoduchá, prístup ku súborom sa ľahko programuje, je však nevýhodná až nepoužiteľná pre systémy s veľkým množstvom súborov alebo u viacužívateľských systémov.
Znázornenie jednoúrovňovej štruktúry adresárov:

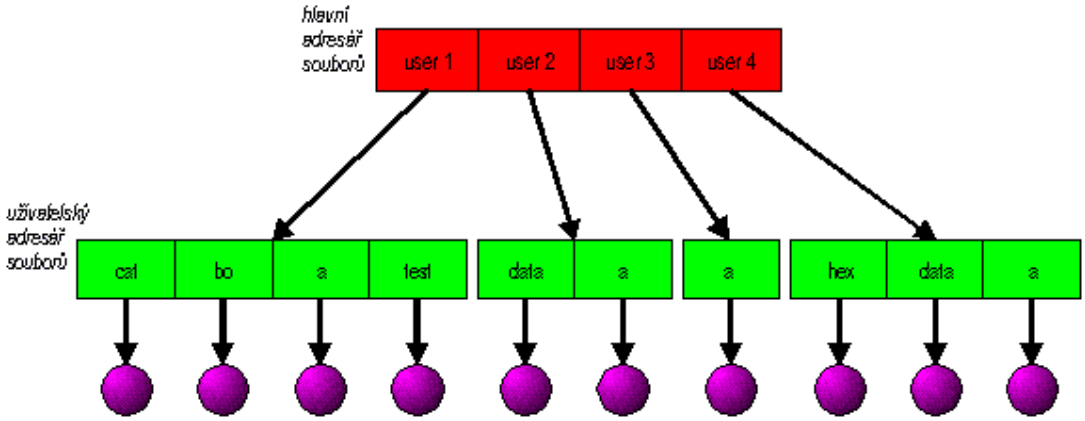


Hlavnými problémami sú neprehľadnosť takého systému a problémy s pomenovávaním súborov u viacužívateľských systémov.

1.1.8.7.3 Dvojúrovňový adresár

Princípom dvojúronňového adresára je vytvorenie samostatného adresára pre každého používateľa (*user file directory – UDF*). Každý UDF má rovnakú štruktúru, ale obsahuje súbory patriace iba určenému užívateľovi. Nad UDF je nadradený hlavný adresár *(master file directory - MFD).*

Znázornenie dvojúrovňovej štruktúry adresárov:

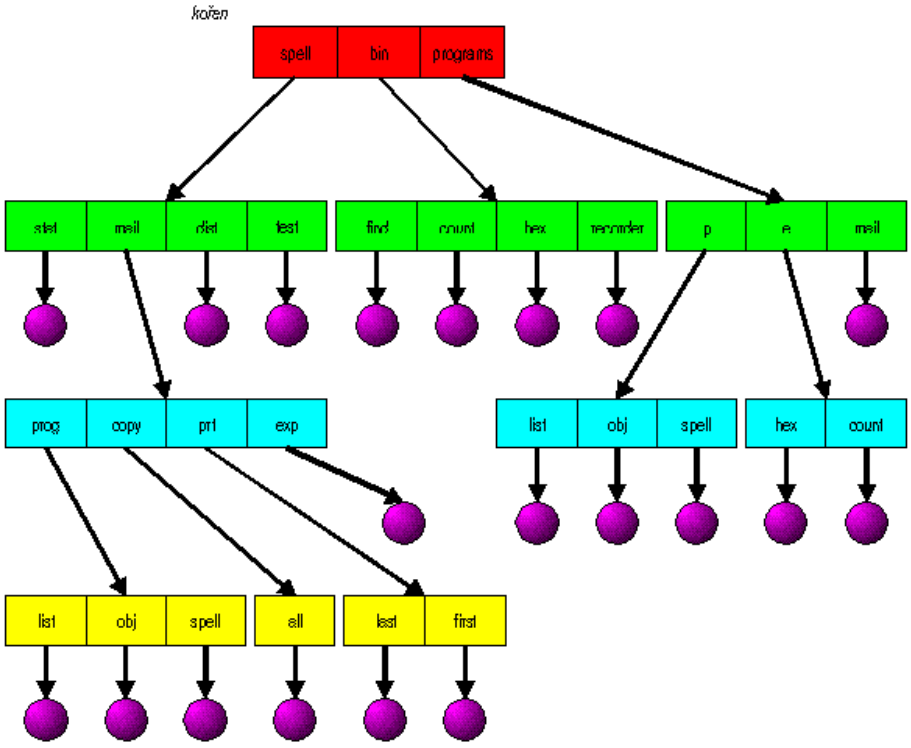


1.1.8.7.4 Stromová štruktúra adresárov

Stromová štruktúra je logickým pokračovaním dvojúrovňovej štruktúry. Táto štruktúra umožňuje každému používateľovi vytvárať si vlastné štruktúry podadresárov a organizovať si súbory do prehľadných štruktúr podľa svojich potrieb. Stromová štruktúra môže mať prncipiálne ľubovoľnú výšku, prakticky je počet úrovní daný adresovacími schopnosťami konkrétneho operačného systému.

Strom je najrozšírenejšou adresárovou štruktúrou súčasnosti. Každý strom má jediný hlavný – koreňový – adresár, v niektorých systémoch nazývaný *root*. Ku každému súboru existuje od koreňa jednoznačná cesta.

Znázornenie stromovej (hierarchickej) štruktúry adresárov:



Adresár obsahuje množinu súborov, adresárov a podadresárov. Adresár je v podstate súbor, s ktorým OS pracuje špeciálnym spôsobom.
Adresár, s ktorým užívateľ práve pracuje, sa nazýva ***aktuálny adresár***. Operačný systém hľadá všetky volané súbory automaticky v aktuálnom adresári. Zmena aktuálneho adresára sa uskutočňuje systémovým príkazom *change directory*.
Cesta ku súboru môže mať dva tvary:

- absolútna cesta – je definovaná od koreňového adresára ku hľadanému súboru
- relatívna cesta - definuje cestu z aktuálneho adresára

Problémom v systéme stromovej štruktúry môže byť zmazanie adresára. Systém musí zabezpečiť ochranu v prípade, ak adresár určený na zmazanie nie je prázdny. Niektoré systémy (napr. MS DOS) nedovoľuje mazať adresár ktorý nie je prázdny, užívateľ musí najskôr postupne zmazať všetky súbory v adresári a prípadne aj vo všetkých vnorených podadresároch. Tento systém je pomerne bezpečný, ale namáhavý pre užívateľa a v určitých situáciách nepraktický. Iné systémy, napr Unix, umožňujú zmazanie všetkých vnorených súborov aj podadresárov jediným príkazom, čo však predstavuje značné riziko nechteného zmazania dôležitých súborov.

1.1.8.8 Ochrana súborov pred stratou, zničením a neoprávneným použitím

Prvoradou metódou na ochranu súboru pred stratou a zničením je pravideľné zálohovanie. Mnohé operačné systémy disponujú nástrojmi na automatické zálohovanie dát.
Poškodenie súboru môže nastať hardwarovou chybou či nepredvídateľnou udalosťou, chybou používateľa, úmyselným poškodením atď.
Ochrana súboru pred neoprávneným prístupom sa zabezpečuje systémom prístupových práv , ochranou heslom a u dôverných súborov šifrovaním.

1.1.8.8.1 Ochrana súborov v Unixe

S každým súborom aj s adresárom sú asociované tri položky (vlastník, skupina, ostatní), pričom každá položka obsahuje tri bity úrovní zabezpečenia (read, write, execute).

- príznak read r umožňuje, aby užívateľovi sa vypísal názov adresára a jeho atribúty
- príznak write w dovoľuje užívateľovi zapisovať do adresára
- príznak execute x dovoľuje užívateľovi nastaviť adresár ako aktuálny

Príklad: Zobrazenie vlastností a prístupových práv v OS Linux

```
-rw-r--r--   1 mrazek   operator     546 Mar 22  1995 .signature
-rw-r--r--   1 mrazek   operator     630 Nov  7 14:15 .xsession-errors
-rw-------   1 mrazek   operator   70144 Nov  9 18:23 3c59xx.exe
-rw-------   1 mrazek   operator  202727 Aug 17 22:03 Connect_Map.mono.ps
-rw-------   1 mrazek   operator   32432 Jan 29  1995 ELNK3.EXE
-rw-rw----   1 mrazek   operator   24574 Nov  7 09:33 HTML.TXT
-rw-rw----   1 mrazek   operator    7308 Nov  7 09:32 INET.TXT
-rw-------   1 mrazek   operator     222 Feb  7  1994 Imakefile
-rw-------   1 mrazek   operator   20649 Aug  3 12:58 Nologin
-rw-r-----   1 mrazek   users      29378 Oct  2 17:55 PKUNZIP.EXE
-rw-------   1 mrazek   operator   10604 Mar 13  1995 USENET
-rw-rw----   1 mrazek   operator    7604 Nov  7 09:32 WWW.TXT
-rw-r--r--   1 root     operator    3775 Jun  5 15:45 XF86Config.old
-rw-r--r--   1 root     operator    4300 Nov 23 11:08 bbb.gif
drwx------   2 mrazek   operator    8192 Nov 20 08:01 bin
drwxr--r--   2 mrazek   operator    8192 Jan 16  1995 buttons-3d-border
drwxr--r--   3 mrazek   operator    8192 Aug  2 11:59 hrivnac
drwxr--r--   3 mrazek   operator    8192 Nov  7 10:45 ikony
-rw-------   1 mrazek   operator 7023616 May 26  1995 infosys.doc
```

Výpis ukazuje v prvom stĺpci nastavenie práv a ochrany každého súboru a adresára, ďalej sa vypisuje počet linkov na príslušný súbor či adresár, meno vlastníka či skupiny, veľkosť v bytoch a dátum a čas vytvorenia súboru a na záver meno súboru, resp. adresára.

1.1.8.8.2

1.1.8.8.3 Najpouzivajsie typy file systemov podporovane UNIXom

- **ufs** - ***U***ser ***F***ile ***S***ystem
- **s5** - ***S***ystem ***V*** File System
- **bfs** - ***B***oot ***F***ile ***S***ystem
- **ext2** - second extended file system (druhy rozsireny souborovy system) je zakladnym file systemom Linuxu
- **ext3** - third extended file system (treti rozsireny souborovy system); na rozdiel od ext2 si vytvara zurnalny subor, ktory umoznuje obnovit data pri havarii disku
- **reiserfs** - zurnalny suborovy system, ktory uklada subory do vyvazeneho stromu, co umoznuje velmi rychly pristup k suborom a obnovu havarovanych diskov
- **swap** - file system pouzivany v odkladacom priestore disku

1.1.8.8.4 File systémy ďalších operačných systémov

DOS podporuje systémy FAT (pre diskety, 12-bitový) a FAT 16, natívny je FAT 16
Windows radu 9x podporuje VFAT, FAT32 aj staršie verzie FAT (natívny je FAT 32)
Windows radu NT /2K/XP/Vista/Seven podporuje FAT32, NTFS kde NTFS je natívny
Mac OS podporuje HFS a novší HFS+ kde HFS+ je natívny, vie pracovať aj s FAT verziami, NTFS vie iba čítať
Novell Netware podporuje NWFS (staršie verzie) a NSS (od ver. 5.0)
Linux používa natívne súborový systém ext, ext2, ext3, ext4 (podľa typu distribúcie), vie pracovať aj s FAT, FAT32 a NTFS (závisí od konkrétnej distribúcie), ale pri zápisu na NTFS môžu nastať problémy s vlastnosťami a s právami.
Prehľad súborových systémov a ich porovnanie nájdete napríklad na:
http://sk.wikipedia.org/wiki/Zoznam_s%C3%BAborov%C3%BDch_syst%C3%A9mov
http://sk.wikipedia.org/wiki/Porovnanie_s%C3%BAborov%C3%BDch_syst%C3%A9mov