OS = program (kolekce programů),
který spojuje hardware (může být virtualizovaný) s aplikacemi a jejich uživateli

Cíle OS = - maximální využití zdrojů počítače (dříve)
- jednoduchost použití počítače (dnes)

Role OS
1. správce prostředků - více programů sdílí paměť, více uživatelů a souborů sdílí diskový prostor
2. Tvůrce prostředí pro uživatele a jejich aplikační programy (tzv. virtuálního počítače).

OS obsahuje
jádro, systémové knihovny a utility, textrové a/nebo grafické uživatelské rozhraní

Jádro
Jádro OS je nejnižší a nejzákladnější část OS
Zavádí se první a běží po celou dobu běhu počítačového systému
Navazuje přímo na hardware
Běží v privilegovaném režimu
zajišťuje základní správu prostředků a tvorbu prostředí
2 typy rozhraní: Kernel interface, Library interface

Typy jader OS
Monolitická jádra – mají vysokou efektivitu, vytváří rozhraní se službami nabízenými vyšším vrstvám (služby – správa paměti, plánování, komunikace mezi procesy, souborové systémy, síťová komunikace, apod.) PŘÍKLADY – FreeBSD, Linux

Mikrojádra - Minimalizují rozsah jádra a nabízí jednoduché rozhraní, malý počet služeb – základní spáva procesů, in/out zařízení, pamětí a komunikace mezi procesy PŘÍKLADY – Mach, QNX, L4

výhody – flexibilita (více současně běžících služeb), zabezpečení
nevýhody – výšší režie

Hybridní jádra - Mikrojádra rozšířená o kód, je za účelem menší režie PŘÍKLADY – Mac OS X, Windows NT (a vyšší)

Exojádra - Experimentální jádra poskytující velmi nízké rozhraní zaměřené hlavně na bezpečné sdílení prostředků PŘÍKLADY – Aegis, Nemesis, …

Klasifikace OS
podle účelu - univerzální (UNIX, Windows, Linux, ...),
specializované:
– real-time (QNX, RXS, RT-Linux, ...),
– databáze, web, ... (napˇr. z/VSE),
– mobilní zaˇrízení (Android, iOS, Windows Phone, ...), …
podle počtu uživatelů:
jednouživatelské (CP/M, MS-DOS, ...)
víceuživatelské (UNIX, Windows, ...).
podle počtu současně běžících úloh:
jednoúlohové
víceúlohové

Základní koncepty v UNIXu
Procesy – abstrakce probíhajícího výpočtu (komunikují spolu pomocí IPC – roury, signály, semafory, sdílená paměť, sockets, zprávy, ...)
Soubory – abstrakce zdroje dat (komunikují s procesy pomocí I/O operací)

Komunikace s jádrem: Služby jádra – operace, realizované jádrem – volané systémovým voláním
PŘÍKLADY – open, close, read, write, kill, fork, exec, exit

HW přerušení
(synchronní – proces-jádro)
(asynchronní - hardware-jádro)
mechanismus, kterým HW zařízení asychronně oznamují jádru vznik událostí, které je zapotřebí obsloužit

žádosti o přerušení příchází do řadiče přerušení
přerušení mají priority, podle kterých se oznamují procesoru
Přijme-li procesor přerušení vyvolá obslužnou rutinu, přičemž přejde do privilegovaného režimu
Řadič může být naprogramován tak, že některá přerušení jsou maskována – používá se pro nastavení priority

Přerušení v procesoru
trap - po obsluze se pokračuje další instrukcí

fault - po obsluze se (případně) opakuje instrukce, která výjimku vyvolala (např. výpadek stránky, dělení nulou, chybný operační kód apod.)

abort - nelze určit, jak pokračovat, provádění se ukončí (např. některé zanořené výjimky typu fault, chyby HW detekované procesorem – tzv. „machine check mechanism“).

Skriptování
Interpret: program, který provádí činnost programu, který je jeho vstupem.
Skript: textový soubor s programem pro interpret.
Nevýhody: pomalejší, je třeba interpret.
Výhody: nemusí se překládat (okamžitě spustitelné), čitelný obsah programu.

“Magic number” = číslo uvedené na začátku souboru a charakterizující jeho obsah
U spustitelných souborů jádro zjistí na základě magic number, jak soubor spustit
Výhoda: možnost psát programy v libovolném skriptovacím jazyku

Pevný disk
Diskový sektor: nejmenší jednotka, kterou disk umožňuje načíst/zapsat
přípojení – ATA, SATA, USB, FireWire atd.
Vzniká hierarchie pamětí, ve které stoupá kapacita a klesá rychlost a cena:
• primární paměť’: RAM (nad ní ještě registry, cache L1-L3)
• sekundární paměť’: pevné disky, SSD (mají také své cache)
• terciární paměť’: pásky, CD, DVD, …
Parametry – přístupová doba, kapacita, otáčky, přenosová rychlost

SSD – Solid State Drive
výhody – rychlý náběh, náhodný přístup, větší přenosová rychlost, tichý, nízká spotřeba
nevýhody – výšší cena, omezený počet přepisů, možné komplikace se zabezpečením

Zabezpečení disků:
ECC – Error Correction Code - k užitečným datům sektoru si ukládá redundantní data, která umožňují opravu nebo alespoň detekci chyb
S.M.A.R.T. - moderní disky si automaticky shromažďují řadu statistik, které lze použít k předpovídání/diagnostice chyb
Rozpoznávání a označování vadných bloků

Disková pole:
RAID (0,1,2,3,4,5,6)

RAID 0 – následné bloky dat rozmístěny na různých discích, vyšší výkonnost, žádná redundance.
RAID 1 – disk mirroring, všechna data ukládána na dva disky, velká redundance
RAID 2 – data rozdělena mezi disky po bitech
(pro 4 datové: chybu na 1 disku lze automaticky opravit, na 2 discích detekovat).
RAID 3 – data uložena po bajtech (nebo i bitech) na různých discích, navíc je užit disk s paritami.
RAID 4 – bloky (sektory či jejich násobky) dat na různých discích a paritní bloky na zvláštním disku.
RAID 5 – jako RAID 4, ale paritní a datové bloky jsou rozloženy na všech discích, redukce kolizí u paritního disku při zápisu.
RAID 6 – jako RAID 5, ale parita uložena 2x, vyrovná se i se ztrátou 2 disk°u.

Uložení souboru na disku:
Alokační blok:
skupina pevného počtu sektorů, typicky 2n pro nějaké n, následujících logicky (tj. v souboru) i fyzicky (tj. na disku) za sebou, která je nejmenší jednotkou diskového prostoru, kterou OS čte či zapisuje při běžných operacích.

Fragmentace:
externí fragmentace: dochází k ní při přidělování a uvolňování prostoru pro soubory na disku vzniká posloupnost volných oblastí a oblastí použitých různými soubory

důsledky:
Vzniknou některé nevyužité oblasti příliš malé na to, aby se daly využít
Data souboru jsou na disku uložena nespojitě – složitější a pomalejší přístup

Moderní souborové systémy užívají různé techniky k minimalizaci externí fragmentace (rozložení souborů po disku, předalokace, odložená alokace)
Přesto bývají k dispozici nástroje pro defragmentaci

interní fragmentace:
nevyužité místo v posledním přiděleném alokačním bloku – plítvání místem.
Některé souborové systémy umožňují sdílení posledních alokačních bloků více soubory

Přístup na disk:
Prostřednictvím I/O portů a/nebo paměťově mapovaných I/O operací

Přenos z/na disk je typicky řízen řadičem disku s využitím technologie přímého přístupu do paměti (DMA). O ukončení operací či chybách informuje řadič procesor pomocí přerušení.

Plánování přístupu na disk:
Pořadí bloků čtených/zapisovaných na disk ovlivňuje plánovač diskových operací

Přicházející požadavky na čtení/zápis jsou ukládány do vyrovnávací paměti a jejich pořadí je případně měněno tak, aby se minimalizovala režie diskových operací

typy souborových systémů:
fs, ufs, ufs2, ext2, ext3, ext4, FAT, FAT32, NTFS

- Virtuální souborový systém (VFS) – vrstva, která zastřešuje všechny použité souborové systémy a umožňuje pracovat s nimi jednotným, abstraktním způsobem.

- Síťové souborové systémy

- Speciální souborové systémy

Žurnálování:
Žurnál slouží pro záznam modifikovaných metadat (příp. i dat) před jejich zápisem na disk
Obvykle implementován jako cyklicky přepisovaný buffer ve speciální oblasti disku
Systémy souborů se žurnálem: ext3, ext4, ufs, NTFS,
Umožňuje spolehlivější a rychlejší návrat do konzistentního stavu po chybách
Data obvykle nejsou žurnálována (byt' mohou být): velká režie

Implementace:
Implementace na základě dokončení transakcí (REDO)
sekvence dílčích operací se uloží nejprve do žurnálu mezi značky označující
začátek a konec transakce (příp. spolu s kontrolním součtem),
• poté se dílčí operace provádí na disku,
• uspějí-li všechny dílčí operace, transakce se ze žurnálu uvolní,
• při selhání se dokončí všechny transakce, které jsou v žurnálu zapsány celé
(a s korektním kontrolním součtem).

Implementace na základě anulace transakcí (UNDO)
záznam dílčích operací do žurnálu a na disk se prokládá,
• proběhne-li celá transakce, ze žurnálu se uvolní,
• při chybě se eliminují nedokončené transakce.

UNDO a REDO je možno kombinovat

Implementace žurnálování musí zajišťovat správné pořadí zápisu operací.

Alternativy k žurnálování
Copy-on-write (např. ZFS, btrfs) – nejprve zapisuje nová data či metadata na disk, pak
je zpřístupní:

I-uzel:
Základní datová struktura popisující soubor v UNIXu.
i-uzel obsahuje metadata: jeho stav, typ souboru, dílka souboru v bajtech, přístupová práva, počet pevných odkazů, přímé a nepřímé odkazy
v-uzel = rozšířený i-uzel

Deduplikace:
Snaha odhalit opakované ukládání těchže dat, uložit je jednou a odkázat vícenásobně.
podporovaná na různých úrovních: sekvence bytů, bloky, extenty, soubory
Může uspořit diskový prostor při virtualizaci, paměťový prostor i čas
Při menším objemu duplikace může naopak zvýšit spotřebu procesorového času, paměťového i diskového prostoru.
Podporuje např.: NTFS

Přístupová práva:
V UNIXu jsou typicky rozlišena práva pro vlastníka, skupinu a ostatní.

Uživatelé
Uživatele definuje administrátor systému
UID: číslo identifikující uživatele

Skupiny:
Skupiny definuje administrátor systému
GID: číslo identifikující skupinu uživatelů
Uživatel může být členem více skupin, jedna z nich je aktuální (používá se při vytváření souborů).

Sticky bit:
Sticky bit je příznak, který nedovoluje rušit cizí soubory v adresáři, i když mají všichni právo zápisu.

SUID, SGID:

Vlastník programu může propůjčit svoje práva komukoli, kdo spustí program s nastaveným SUID.

Buffering – čtení a zápis:
buffer – vyrovnávací paměťové
Cílem je minimalizace počtu pomalých operací s periferiemi
čtení:
1. přidělení VP a načtení bloku,
2. kopie požadovaných dat do adresového prostoru procesu (RAM→RAM).
Zápis:
1. přidělení VP a čtení bloku do VP (pokud se nezapisuje na konec souboru),
2. zápis dat do VP (RAM→RAM), nastaví se příznak modifikace (dirty bit),
3. zpožděný zápis na disk, nuluje příznak.

Otevření souboru pro čtení:
fd = open("/dir/file", O_RDONLY);
1. Vyhodnotí cestu a naleznečíslo i-uzlu
2. V systémové tabulce aktivních i-uzlů vyhradí novou položku a načte do ní i-uzel, vzniká v-uzel
3. V systémové tabulce otevřených souborů vyhradí novou položku a naplní ji
4. V poli deskriptorů souborů v uživatelské oblasti procesu vyhradí novou položku a naplní ji odkazem na položku v tabulce otevřených souborů.
5. Vrátí index položky v poli deskriptorů (nebo -1 při chybě).

Zavření souboru:
x = close(fd);
1. Kontrola platnosti souborového deskriptoru.
2. Uvolní se odpovídající položka v tabulce deskriptorů, sníží se počitadlo odkazů v odpovídající položce tabulky otevřených souborů.
3. Pokud je počitadlo odkazů nulové, uvolní se odpovídající položka v tabulce otevřených souborů a sníží se počitadlo odkazů ve v-uzlu.
4. Pokud je počitadlo odkazů nulové, i-uzel se z v-uzlu okopíruje do VP a uvolní.
5. Funkce vrací nulu nebo -1 při chybě.

Duplikace deskriptoru souboru:
fd2 = dup(fd);
fd2 = dup2(fd,newfd);
Postup při duplikaci deskriptoru:
1. Kontrola platnosti fd.
2. Kopíruje danou položku v tabulce deskriptorů do první volné položky (dup) nebo do zadané položky (dup2). Je-li deskriptor newfd otevřen, dup2 ho automaticky uzavře.
3. Zvýší počitadlo odkazů v odpovídající položce tabulky otevřených souborů.
4. Funkce vrací index nové položky nebo -1 při chybě.

Terminály:
Terminály – fyzická nebo logická zařízení umožňující (primárně) textový vstup/výstup systému: vstup/výstup po řádcích, editace na vstupním řádku, speciální znaky (Ctrl-C, Ctrl-D, ...), …

Roury:
Roury (pipes) – jeden z typů speciálních souborů. Rozlišujeme:
• roury nepojmenované
• Roury pojmenované – vytvoření mknod či mkfifo.
Roury representují jeden z mechanismů meziprocesové komunikace.
Implementace: kruhový buffer s omezenou kapacitou.
Procesy komunikující přes rouru (producent a konzument) jsou synchronizovány.

Sockets:
Umožňují síťovou i lokální komunikaci.
Lokální komunikace může probíhat přes sockety pojmenované a zpřístupněné v souborovém systému

VFS:
Virtuální souborový systém vytváří jednotné rozhraní pro práci se souborovými systémy, odděluje vyšší vrstvy OS od implementace operací na souborových systémech
Pro popis souborů používá rozšířené i-uzly (tzv. V-uzly)

NFS:
NFS (Network File System) – zpřístupňuje soubory uložené na vzdálených systémech.
Umožňuje kaskádování

Spooling:
spool = vyrovnávací paměť (typicky soubor) pro zařízení (nejčastěji tiskárny), které neumožňují prokládané zpracování dat různých procesů.
Výstup je proveden do souboru, požadavek na jeho fyzické zpracování se zařadí do fronty, uživatelský proces může pokračovat a zpracování dat se provede, až ně přijde řada.