

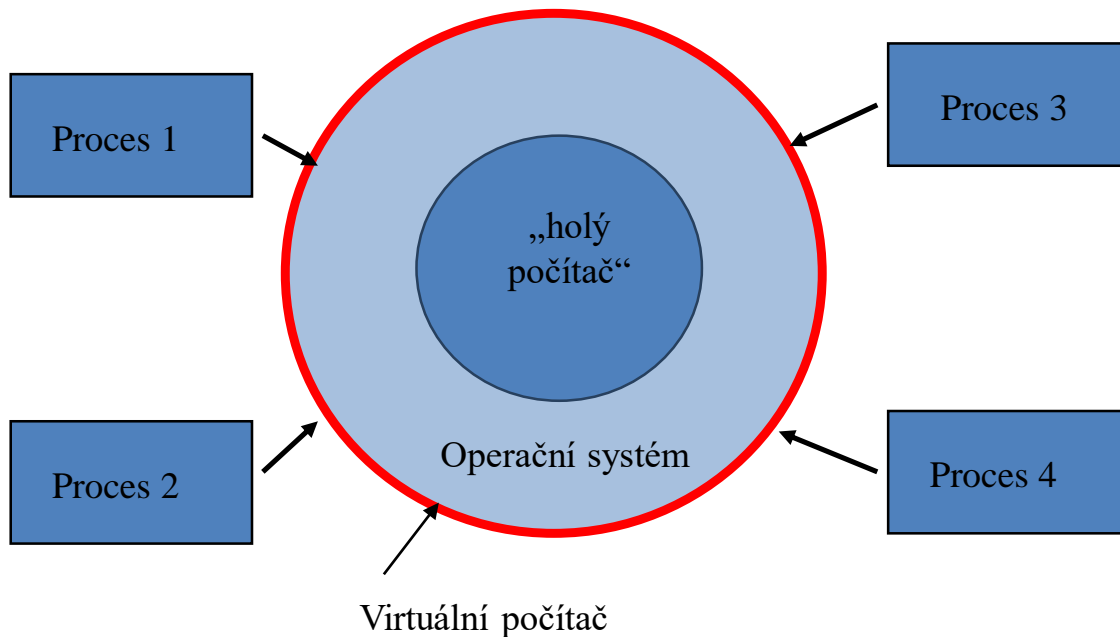


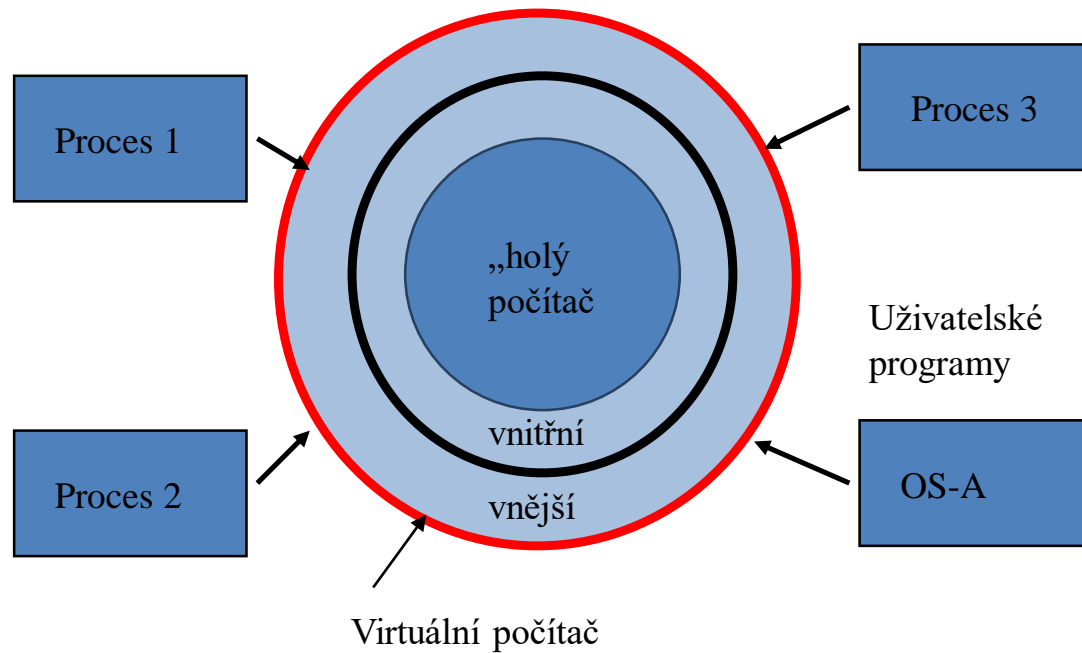
OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA

Architektura operačních systémů, moderní přístupy k výstavbě operačních systémů

Ing. Pavel Smolka, Ph.D.

Koncepce virtuálního počítače





Základní pojmy v oblasti OS

- **Proces** - spuštěná instance počítačového programu.
 - je umístěn v operační paměti počítače v podobě sledu strojových instrukcí vykonávaných procesorem,
 - obsahuje nejen kód vykonávaného programu, ale i dynamicky měnící se data, která proces zpracovává,
 - program může běžet v podobě jednoho či více procesů s různými.
- **Soubor** – abstrakce s kterou pracuje OS prostřednictvím systémových volání.
 - jedná se o uloženou pojmenovanou sadu dat,
 - fyzické uspořádání soubor je závislé na souborovém systému.



Základní pojmy v oblasti OS

- **Systémové programy**
 - poskytují pohodlné prostředí pro vývoj a spouštění programů,
 - typická kategorizace systémových programů:
 - práce se soubory, kopírování, editace ...,
 - získávání, definování a údržba systémových informací,
 - podpora jazykových prostředí,
 - zavádění a provádění programů,
 - Komunikace,
 - aplikační programy.
- Významným systémovým programem je shell
 - shell implementuje příkazy sám nebo může volat externí programy.
- Uživatelský pohled na OS je dán především systémovými programy a ne voláními systému (system calls).



Typy operačních systémů

- Dle doby odezvy a způsobu vstupu úloh:
 - dávkové,
 - Interaktivní,
 - systémy pracující v reálném čase (RT),
 - hybridní systémy.
- Dle použití:
 - univerzální,
 - Specializované.



Dělení operačních systémů dle vnitřní struktury

- velké monolitické jádro,
- modulární, hierarchický přístup,
- malé jádro a samostatné procesy,

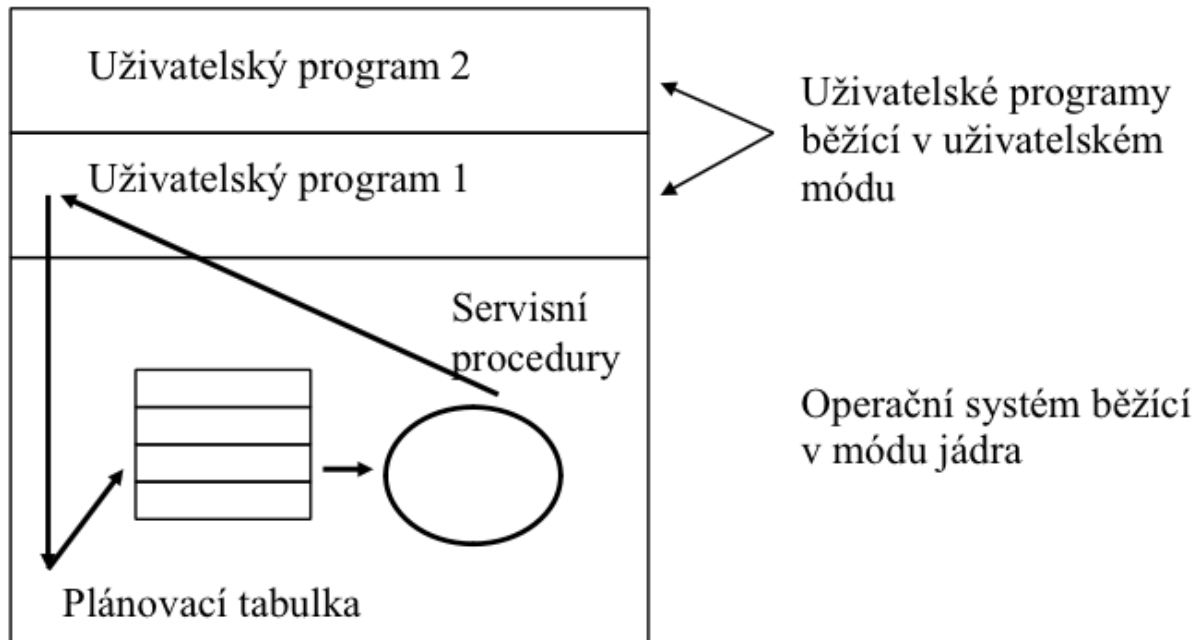


Monolitický operační systém

- velmi rozšířený,
- model jádra OS + systémových služeb,
- bez jasné vnitřní struktury,
- Unix, CP/M, MS-DOS
- různé režimy činnosti procesoru → různé fáze činnosti OS

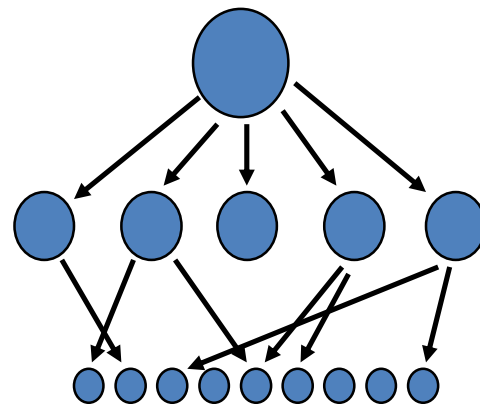


Schéma monolitického



Monolitický systém – popis vnitřní struktury

- hlavní program – volání obslužné procedury
- obslužné procedury pro jednotlivá systémová volání
- pomocné procedury pro obslužné procedury.



Výhody a nevýhody monolitických OS

Nevýhody:

- velké jádro (chyby)
- obtížné ladění
- složitá rozhraní procedur
- **vliv HW**

Výhody:

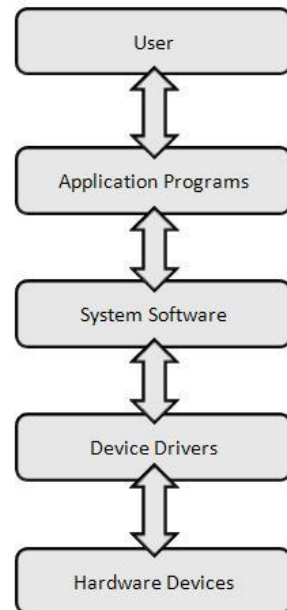
- relativně snadná implementace
- **velký výkon**



Vrstevnatý operační systém

- hierarchické uspořádání vrstev
- vyšší vrstva využívá pouze služeb nejbližší nižší vrstvy
- příklad návrh OS „THE System“ (Technische Hogeschool Eindhoven – Dijkstra 1968): 6 vrstev

5	operátor
4	uživatelské programy
3	správa I/O zařízení
2	komunikace s procesy
1	správa paměti
0	přidělování procesoru



Základní rysy vrstevnaté struktury OS

- OS se dělí do jistého počtu vrstev (úrovní).
- Každá vrstva je budována na funkcionalitě nižších vrstev.
- Nejnižší vrstva (0) je hardware.
- Nejvyšší vrstva je uživatelské rozhraní.
- Pomocí principu modulů jsou vrstvy vybírány tak, aby každá používala funkcí (operací) a služeb pouze vrstvy $n-1$.



Základní rysy vrstevnaté struktury OS

- Řeší problém přílišné složitosti velkého systému.
- Provádí se dekompozice velkého problému na několik menších zvládnutelných problémů.
- Každá úroveň řeší konzistentní podmnožinu funkcí.
- Nižší vrstva nabízí vyšší vrstvě „primitivní“ funkce (služby).
- Nižší vrstva nemůže požadovat provedení služeb vyšší vrstvy.
- Používají se přesně definovaná rozhraní.
- Jednu vrstvu lze uvnitř modifikovat, aniž to ovlivní ostatní vrstvy – princip modularity.



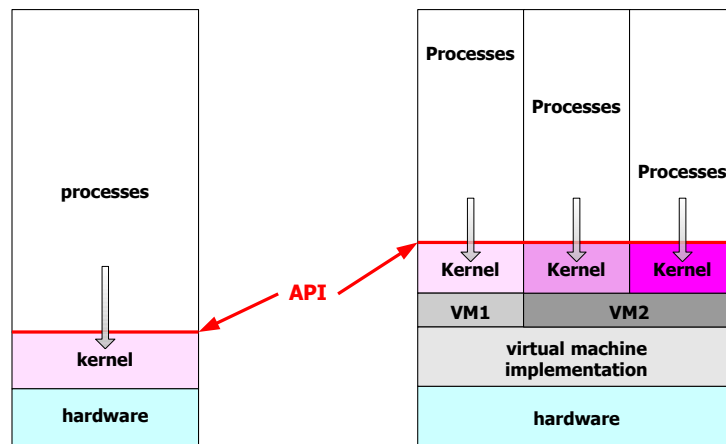
Výhody a nevýhody hierarchické struktury

- Výhodou je modularita OS
- Nevýhodou je především vyšší režie a tím pomalejší vykonávání systémových volání
- Protože efektivita hraje v jádře OS významnou roli je třeba volit kompromis
 - pouze omezený počet úrovní pokrývající vyšší funkcionalitu
 - příklad: první verze Windows NT měli hierarchickou strukturu s řadou vrstev, avšak pro zvýšení výkonu OS bylo ve verzi NT 4.0 rozhodnuto přesunout více funkcionality do jádra a sloučit některé vrstvy



Virtuální počítač – implementace hierarchického přístupu

- Operační systém virtualizuje hardwarové prostředky
- Emulovaný běh různých hostovaných OS

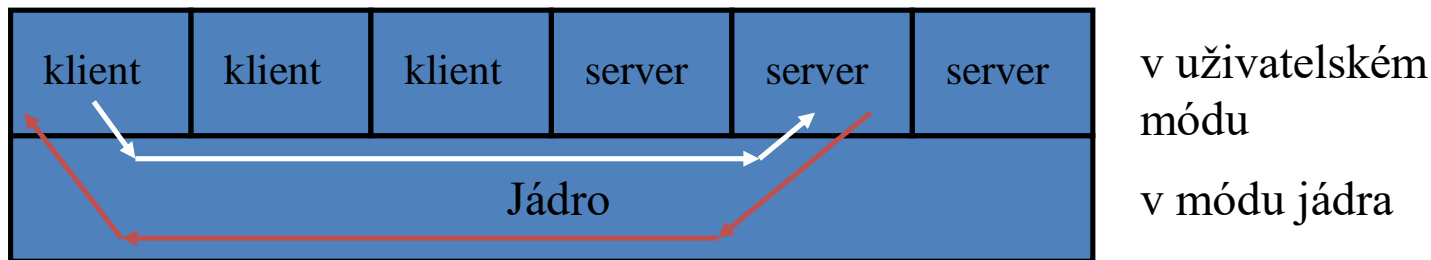


Standardní nevirtualizovaný stroj

Jeden hardware
Dvě virtuální hardwarové platformy
Tři virtuální operační systémy

Operační systém klient - server

- Trend minimalizace rozsahu jádra OS (mikrokernel),
- část funkcí OS implementována do úrovně uživatelských procesů (zasílání požadavků mezi procesy),
- mikrokernel zajišťuje zejména komunikaci.



Výhody a nevýhody klient – server OS

Výhody:

- Oddělení adresových prostorů procesů → dobrá strukturovanost OS,
- odolnost systému v případě selhání serverového procesu,
- použitelnost v síťovém a distribuovaném prostředí.

Nevýhody:

- Aplikace musí být napsány s ohledem na tuto architekturu.



Charakteristické rysy při vývoji moderních OS

Reakce na rozvoj hardware

- Multiprocesorové a multijádrové stroje.
- Vysokorychlostní sítě.
- Rychlejší procesory, větší kapacity pamětí.

Požadavky nových softwarových technologií

- Multimédia, Internet a web.
- Klient/Server technologie.

Distribuované OS

- Iluze jedné společné paměti.
- Iluze jednoho paralelního stroje.
- Distribuované systém souborů s jednotnou tvář.

Objektově orientovaný návrh

- Modulární rozšiřování funkcionality z malého jádra.
- Přizpůsobování konkrétním potřebám bez porušení integrity celého systému.



Cíle při návrhu operačního systému

Cíle pro uživatele

- OS musí být snadno použitelný, snadno naučitelný, spolehlivý, bezpečný, rychlý, ...

Systémový cíl

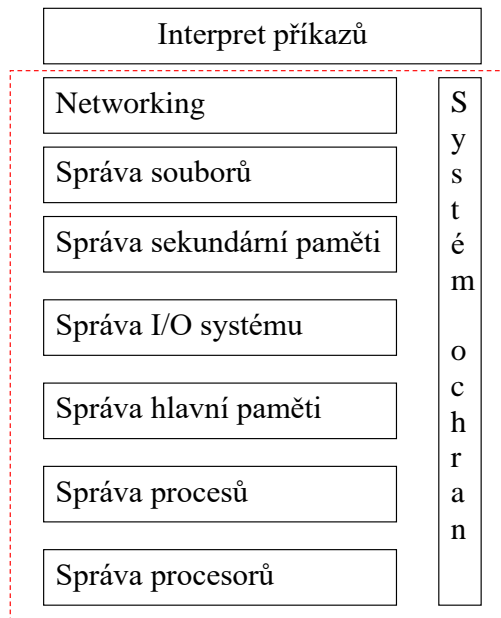
- OS se musí dát snadno navrhnout, implementovat, udržovat a musí být přizpůsobivý, spolehlivý, bezchybný, spolehlivý.

Realita - OS jsou:

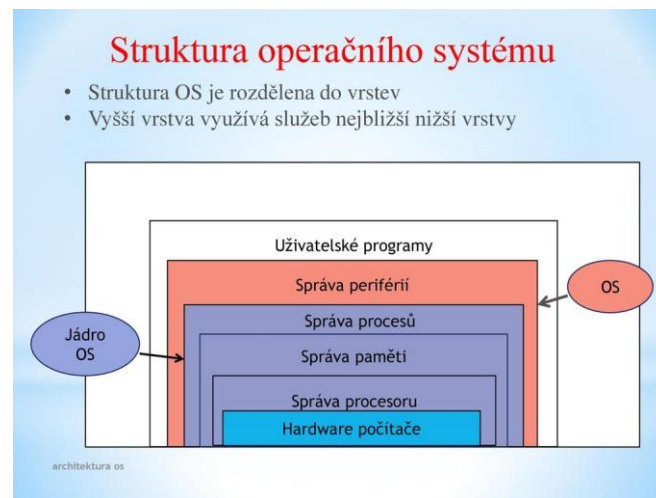
- Obrovské – až desítky milionů řádků zdrojového kódu,
- Složité,
- asynchronní (interaktivní),
- (vždy) plné chyb a (často) nespolehlivé,
- silně závislé na konkrétním hardware – obtížně přenositelné.



Generické komponenty OS



Jádro OS



Správa procesů

- Proces (aktivní entita) je spuštěný program (pasivní entita)
 - jeden program může být zároveň spuštěn řadou uživatelů
 - proces je jednotkou práce systému
 - procesy jádra OS
 - uživatelské procesy
- Proces pro svou činnost potřebuje zdroje
 - procesor, paměť, soubory
 - zdroje se alokují při spuštění nebo při běhu
- Typické aktivity OS v oblasti správy procesů
 - vytváření a ukončování uživatelských a systémových procesů
 - potlačování (suspending) a obnovování (resuming) procesů
 - mechanismy pro synchronizaci procesů
 - mechanismy pro komunikaci mezi procesy
 - mechanismy pro detekci a řešení uváznutí (deadlock)

Správa operační paměti

- Při spuštění procesu musíme program nahrát do paměti
- Později může proces vyžadovat dodatečnou paměť pro data a tuto paměť OS vracet
- Jakmile proces končí, musí OS veškerou paměť užívanou procesem opět uvolnit
- Typické aktivity OS v oblasti správy paměti
 - znalost, která část paměti je využívána a kým (kterým procesem)
 - přidělování (alokace) a uvolňování (dealokace) paměti podle požadavků procesů
 - rozhodování o tom, který proces kdy zavést do paměti



Správa souborů

- Soubor
 - kolekce souvisejících informací definovaná svým tvůrcem
 - může mít strukturu, ale nemusí
 - z hlediska OS typicky jen posloupnost bajtů
- I/O zařízení na kterých jsou soubory uloženy mohou být nejrůznějšího typu
 - magnetické disky, optické disky, magnetické pásky
 - OS zavádí abstraktní koncept souboru
- Typické aktivity OS v oblasti správy souborů
 - vytváření a rušení souborů
 - vytváření a rušení adresářů (složek)
 - základní operace pro manipulaci se soubory a adresáři (např. čtení ze souboru, zápis do souboru, seznam souborů v adresáři)
 - zálohování na nevolatilní (energeticky nezávislá) média



Správa I/O zařízení

- Snaha o skrytí specifik jednotlivých I/O zařízení
 - co nejjednodušší přístup k jednotlivým I/O zařízením
 - mnoho OS zpřístupňuje I/O zařízení přes speciální soubory
 - /dev/sda, /dev/lp0 v UNIXu
 - \\.\PHYSICALDRIVE2, CONIN\$ ve Win32
- Ovladače jednotlivých HW komponent
- Řízení bufferů, kešování, spooling



Správa sekundární paměti

- Typické sekundární paměti jsou disky
- Správa sekundárních pamětí obvykle formou souborů – souborového systému
- Typické aktivity OS
 - správa volného místa
 - přidělování místa
 - plánování činnosti disku (které požadavky kdy vyřídit)



Správa síťových služeb

- Komunikace je řízena protokolem
 - protokol je několikastranný algoritmus pro dosažení určitého cíle
- Snaha o transparentnost
 - síťové souborové systémy (SMB, NFS, ...)
 - API podobné jako přístup k souborům



Ochranný systém

- V multitaskingovém a multiuživatelském OS musíme jednotlivé procesy navzájem chránit
- **Ochrana** je mechanismus, který řídí přístup programů, procesů a uživatelů ke zdrojům počítačového systému
 - HW za pomoci OS zajišťuje, že proces může používat pouze adresy svého adresového prostoru
 - časovač brání jednomu procesu v získání plného přístupu k CPU
 - režim CPU brání uživatelským procesům spouštět privilegované instrukce
- Ochranný mechanismus rozlišuje autorizované a neautorizované použití prostředků



Interpret příkazů

- Interpret příkazů, neboli command-line interpreter, neboli shell
 - úkolem je získávat příkazy od uživatele a provádět je
- Za tento interpret příkazů můžeme považovat i moderní GUI
- Někdy je interpret příkazů přímo součástí jádra operačního systému, někdy je to jen speciální program



Služby OS

- Spouštění programů
 - zavedení programu do paměti, běh procesu
- I/O operace
 - uživatelské procesy nemají přístup k I/O instrukcím, vše musí poskytovat OS
- Souborový systém
 - manipulace se soubory a adresáři
- Komunikace
 - mezi procesy nebo počítači
- Detekce chyb
 - musí počítat s chybami HW (např. parita paměti)



Interní služby OS

- Zabezpečují efektivní provoz samotného OS
- Alokace zdrojů
 - plánovací algoritmy pro přidělování CPU
 - přidělování přístupu k tiskárnám apod.
- Účtování
 - máme přehled o tom, kteří uživatelé kdy využily které zdroje
- Ochrana
 - Autentizace - identifikace, řízení přístupu - autorizace



Systémová volání

- Systémová volání tvoří rozhraní mezi uživatelským procesem a OS
 - typicky jsou popsána jako instrukce assembleru a jsou uvedena v programátorském manuálu k OS
 - vyšší programovací jazyky obsahují některé funkce, které odpovídají systémovým voláním (např. open, write) a dále knihovní funkce, které poskytují vyšší funkčnost a v rámci této spouští (třeba hned několik) systémových volání (např. fopen, fwrite).



Systemová volání (2)

- Různé OS a různé HW platformy mívají různé způsoby jak volat služby OS a různou strukturu těchto služeb
- Nicméně existují určité standardy, které usnadňují přenositelnost programového kódu
 - v oblasti UNIXu: POSIX
 - V oblasti Windows: Win32
- Teoreticky kód který bude psán podle standardu bude přeložitelný na kompatibilních platformách, v praxi však existuje celá řada verzí standardu a mnoho výjimek co je a není implementováno



Kategorie systémových volání

- Správa procesů
 - spust', ukonči, čekej, nastav atributy
- Správa souborů
 - vytvoř, smaž, otevři, čti, zapiš, zavři
- Správa I/O
 - získej přístup, připoj, čti, zapiš, odpoj
- Správa informací
 - datum a čas, atributy procesů a souborů
- Komunikace
 - vytvoř spojení, odešli, přijmi



Další funkce operačního systému

- Monitorování a „účtování“ systému
 - Sběr statistiky o využití systému
 - Pro účtování „placených služeb“
 - Pro dlouhodobé plánování v systému
 - Monitorování výkonnosti a chování systému
 - Slouží k podpoře dalšího vývoje systému
- Je třeba rozlišovat
 - služby OS jako celku
 - služby jádra OS (**JOS**) (*system calls*)



Základní pojmy v oblasti OS II

- Bootování
 - Spuštění činnosti počítače zavedením jádra a předáním řízení na vstupní bod jádra pro spuštění činnosti
- Bootstrap program
 - Program uchovávaný v ROM, který je schopný najít jádro, zavést ho do paměti a spustit jeho provedení
- SYSGEN
 - Program pro získání informací týkajících se konkrétní konfigurace konkrétního hardwarového systému



Proces Bootování

Řídí BIOS nebo UEFI (Unified Extensible Firmware Interface)

- provede se inicializace HW komponent,
- na základě uložené konfigurace zjistíme z kterého zařízení se má OS zavést,
- v případě pevného disku se spustí kód uložený v Master Boot Record (MBR),
- tento kód například zjistí, která partition je aktivní a spustí boot sektor této partition. Kód uložený v boot sektoru načte soubory s jádrem OS do paměti,
- zavaděče, např. LILO, GRUB umožní interaktivně vybrat který OS bude zaveden (bootsektor které partition se má spustit, kde je soubor s jádrem OS),
- jelikož tento kód může být delší než je délka MBR, musí pak být uložen v jiné oblasti disku.



Děkuji za pozornost

