

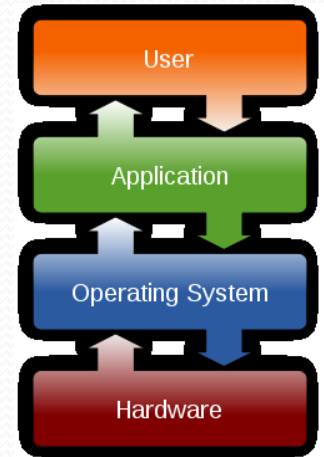
Operační systémy

Úvod

Počítačový systém

Běžný číslicový počítačový systém lze popsat následující hierarchickou strukturou

- Hardware
 - CPU, paměť, vstupní a výstupní zařízení, ...
- Operační systém
 - Řídí a koordinuje efektivní využití hardware
 - Vytváří rozhraní pro aplikační programy.
- Aplikační programy
 - programy, které využívají uživatelé pro řešení problémů.
 - Vývojové nástroje, kancelářské programy, popřípadě hry atd.
- Uživatelé
 - Lidé, stroje, jiné počítače



„Co je operační systém?“

- základní otázka, ale neexistuje jednoduchá odpověď
 - V šedesátých letech byla definice OS **„software, který řídí hardware“**
- V dnešní době je to mnohem složitější
 - Velké množství paralelně běžícího SW
 - Odděluje HW od SW
 - Spravuje HW
 - Poskytuje služby SW

Proč studujeme operační systém?

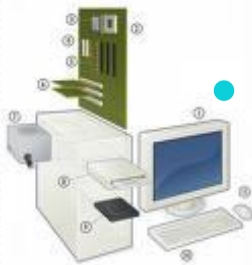
- Jedná se o nejsložitější SW systémy
- Uplatňují se zde různé oblasti vědění
 - Lze uplatnit i v jiných oblastech
 - Krizové rozhodování, souběžnost, správa zdrojů, algoritmizace, programování
- Cíle:
 - Pochopit roli OS
 - Porozumět používání služeb a rozhraní OS

Co všechno je OS

- Operační systém je vše, co dodavatel poskytuje v jednom balíku.
 - Problém je v tom, že součástí dnešních operačních systémů jsou i aplikační programy.
- Operační systém je pouze jádro (kernel), které běží po celou dobu běhu počítačového systému.
 - Problém je opět v tom, že některé nízkoúrovňové části operačního systému jsou zaváděny na žádost

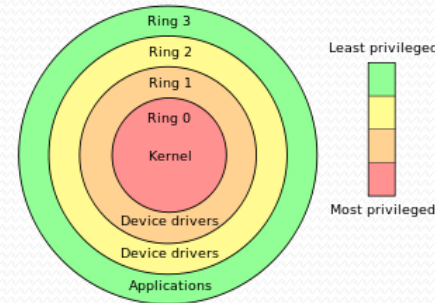
Pohled na OS

- **Pohled na operační systém jako na rozšíření stroje (pohled shora dolů)**
 - Operační systém skrývá před programy „pravdu“ o hardwaru a poskytuje jednoduché rozhraní pomocí vysokoúrovňových služeb, které nazýváme systémové volání.
 - Operační systém vytváří virtuální HW, potom mluvíme o virtuálním počítači – virtual machine.
- **Pohled na operační systém jako na správce zdrojů (pohled zdola nahoru)**
 - Operační systém působí jako řídicí program, který spravuje zdroje a poskytuje je programům (resource allocator/manager).
 - Operační systém zabraňuje nesprávnému použití počítače a chybám (přístupová práva, uváznutí, ...). Jestliže nastanou konfliktní situace, operační systém musí rozhodnout



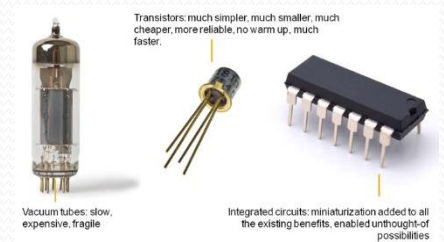
Vyvolání služby systému

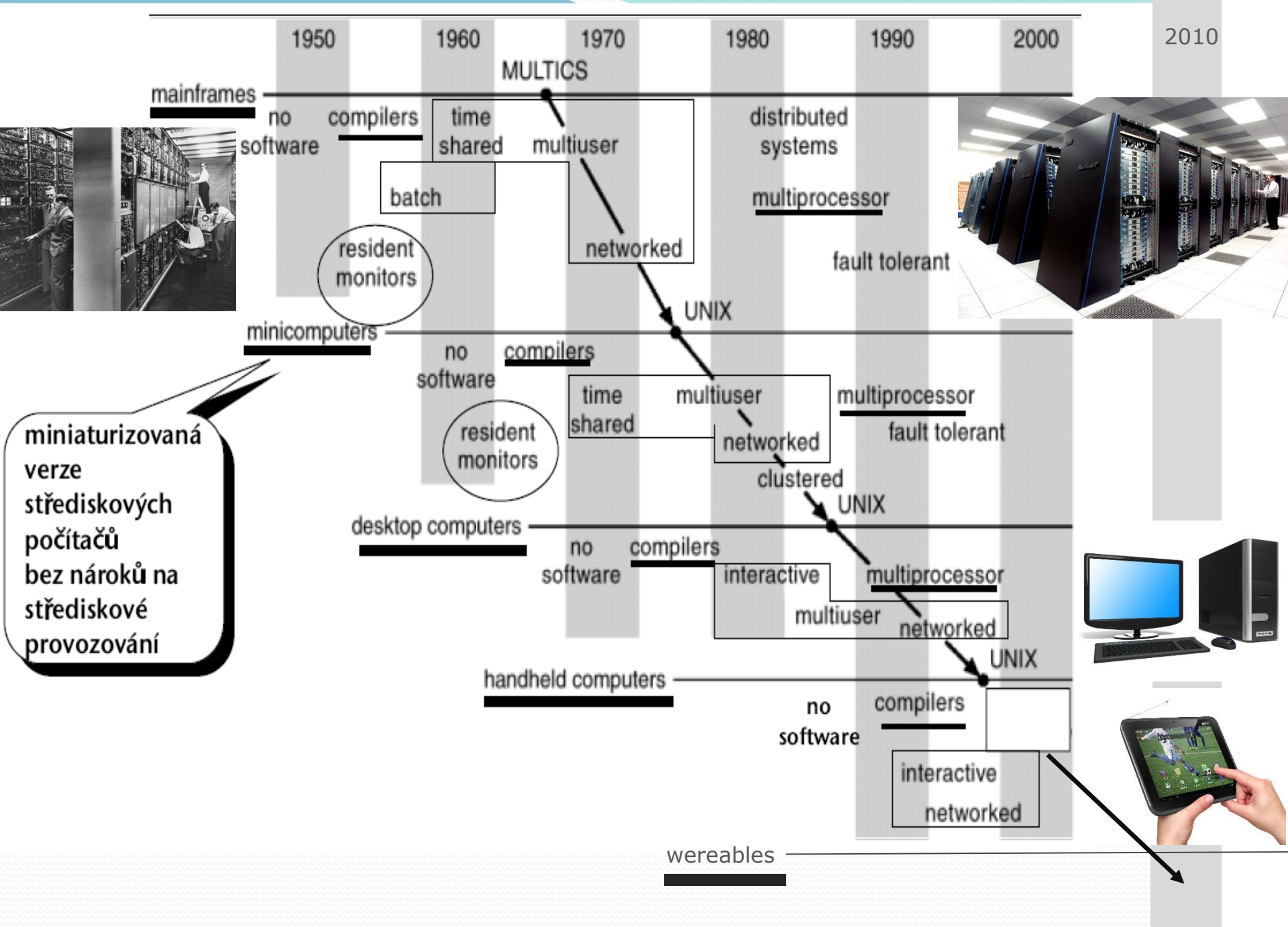
- Většina moderních procesorů pracuje ve dvou režimech:
 - **Privilegovaný** (režim jádra) –povoleny všechny instrukce
 - pouze vlastní operační systém (jádro operačního systému).
 - **Uživatelský** – I/O a některé další instrukce jsou zakázány
 - aplikace a systémové programy
- Různé operační systémy poskytují různé služby
 - liší se i mechanismy volání jádra.
- Programovací jazyky zakrývají služby systému
 - Jeví se jako běžné knihovní funkce.
- Postup vyvolání služby aplikací:
 - Parametry se uloží na určené místo (registry, zásobník)
 - Vyvolá se speciální instrukce, která vyvolá obslužnou proceduru v jádře a zároveň se přepne do privilegovaného režimu
 - Operační systém zjistí, která služba je vyvolána, převezme parametry a vyvolá službu
 - Návrat do aplikace a přepnutí zpět do uživatelského režimu



Historie a vývoj

- Operační systémy se vyvíjely v několika odlišných fázích (generacích)
 - korespondují zhruba s desetiletími.
- První generace (40. a 50. léta)
- Druhá generace (60. léta)
- Třetí generace (70. léta)
- Čtvrtá generace (od roku 1980)
- Čekání na pátou generaci





Základní komponenty OS

- **Procesy** –potřebuje přidělený výpočetní čas CPU, místo v paměti kde může běžet (pro data i program) a dále vstupy a výstupy (soubory).
- **Správa hlavní paměti**
 - Alokace a dealokace paměti podle potřeby
 - Udržuje informaci, která část paměti je používána a kým
- **I/O podsystém, vstupy a výstupy**
 - Správa paměti pro buffering, caching, spooling (Simultaneous Peripheral Operation On Line)
 - Vytváření a rušení souborů a adresářů
 - Rozvrh diskových operací
 - Společné rozhraní ovladačů zařízení, ovladače pro specifická zařízení
- **Ochrana a bezpečnost** – přístup ke zdrojům pouze autorizované procesy
 - Specifikace přístupu
 - Mechanismus ochrany (paměti, souborů, ...)
- **Uživatelské rozhraní**
 - Součást jádra systému nebo jako samostatný program
 - CLI (Command Line Interface) – řádkově orientované zadávání příkazů
 - GUI (Graphical User Interface) – ikony reprezentující programy, soubory a funkce systému. Výběr z nabízených možností.
- **Síť (Network)**

Architektura OS

- **Monolitický operační systém**
 - Všechny části OS jsou obsaženy v kernelu a přímo komunikují s ostatními.
 - Přímá komunikace velmi efektivní.
 - Kernel spuštěn s neomezeným přístupem k celému počítačovému systému.
 - Většina starších operačních systémů (OS/360, VMS, Linux).

Monolitický OS

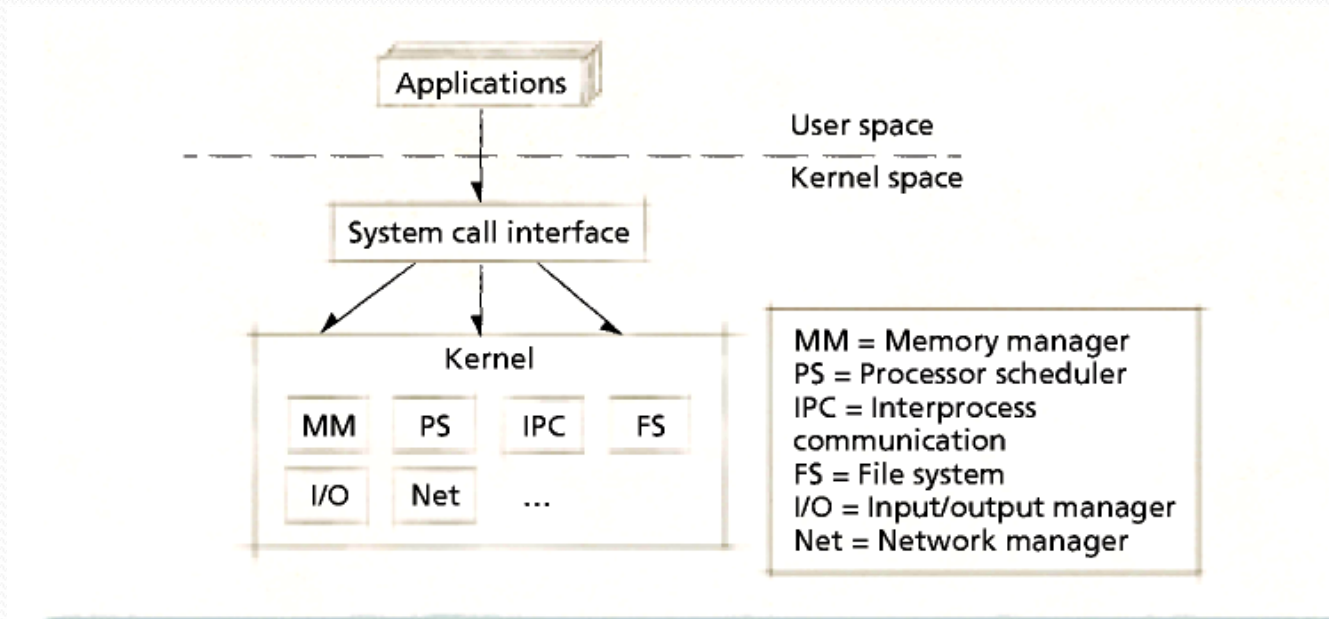


Figure 1.3 | Monolithic operating system kernel architecture.

Architektura OS

- **Microkernel architektura**

- Poskytuje základní služby v pokusu udržet jádro malé a škálovatelné.
 - nízkoúrovňový paměťový management
 - komunikaci mezi procesy
 - základní synchronizaci procesů potřebnou pro spolupráci mezi procesy
 - obsluhu přerušení.
- Další části OS spuštěny mimo jádro OS s menší úrovní práv
- OS Mach, GNU Hurd, Win NT, ...

Microkernel architektura

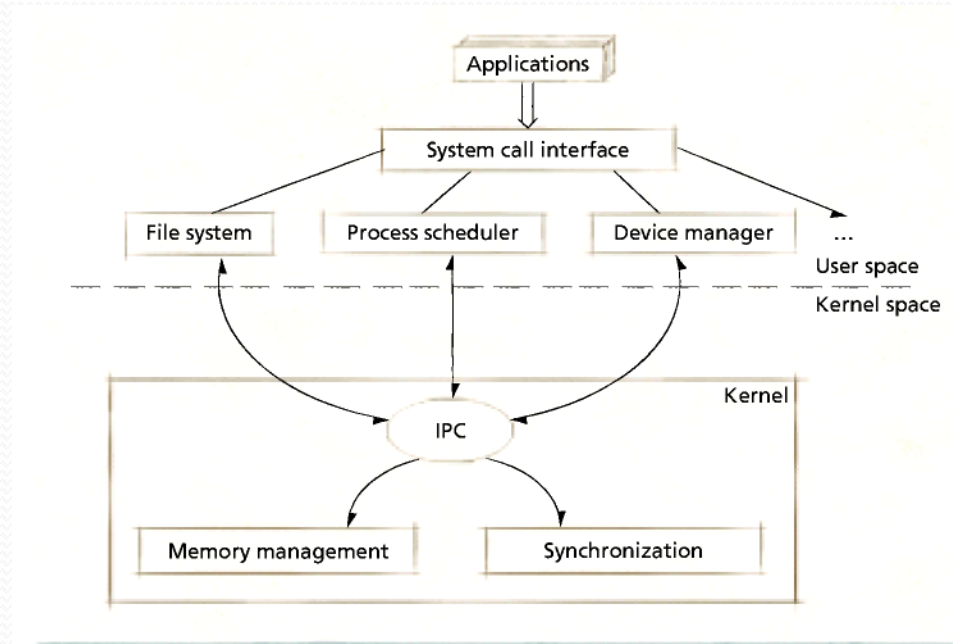


Figure 1.5 | Microkernel operating system architecture.

Architektura OS

- **Vrstvený operační systém**
 - Vytvořena hierarchie procesů.
 - Nejnižše položené vrstvy komunikují s hardwarem, každá další vyšší vrstva poskytuje abstraktnější virtuální stroj.
 - Jednotlivé vrstvy nelze obcházet, každá vrstva komunikuje výhradně se sousední vrstvou.
 - Výhodou je možnost výstavby systému od nejnižších vrstev, modularita.
 - Typickým představitelem je OS „THE“
 - dnešní OS obsahují prvky vrstvené arch.

Vrstvený OS

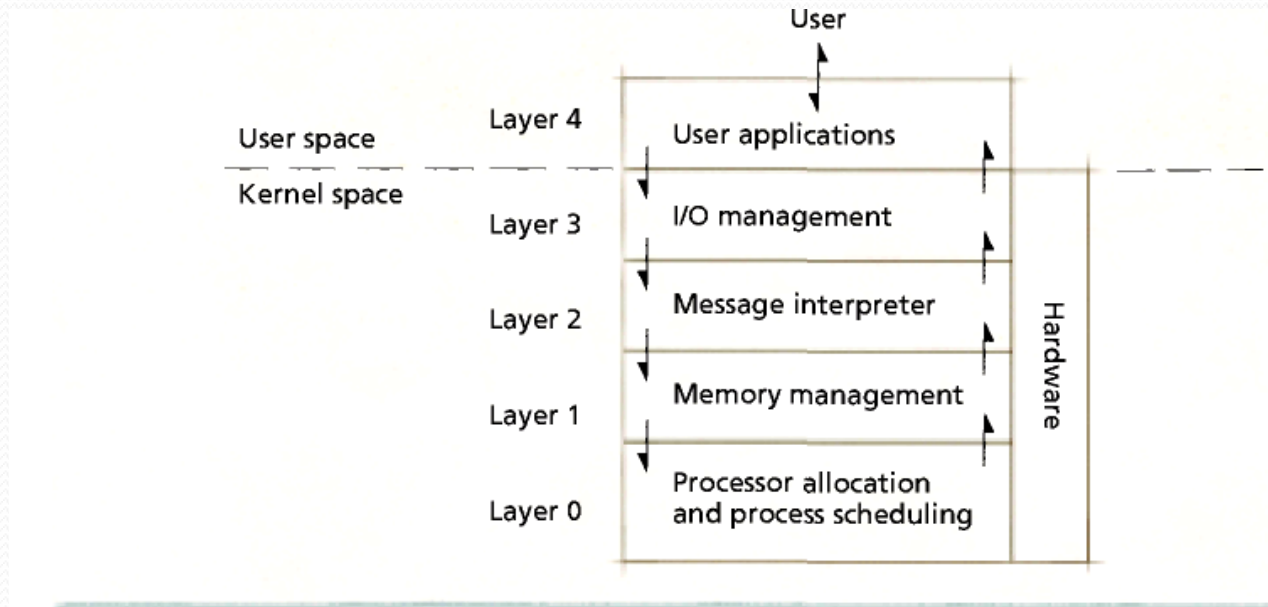


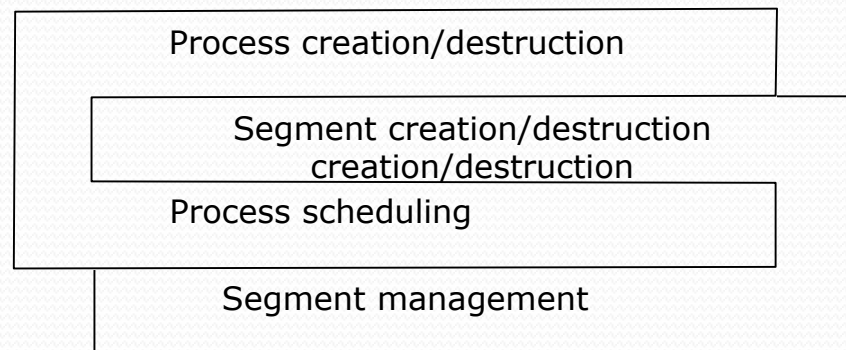
Figure 1.4 | Layers of the TFE operating system.

Vrstvený OS

- **funkční hierarchie**

- U vrstvených systémů je obtížné rozčlenit OS do striktní hierarchie
 - vznikají závislosti

Process management



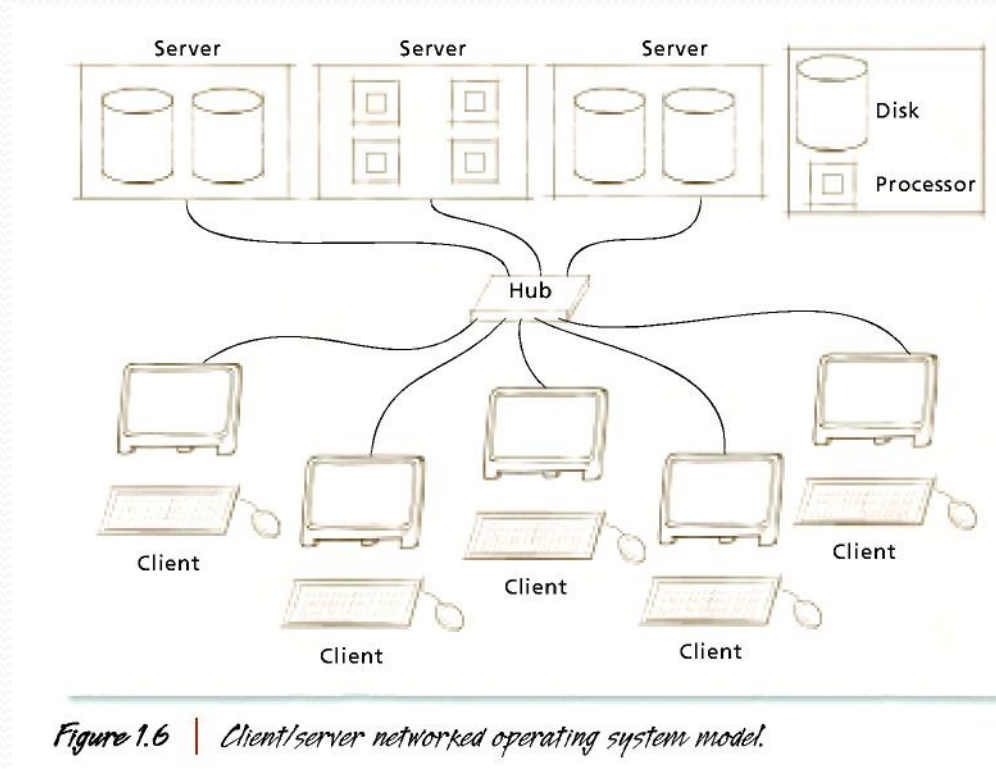
Memory management

Architektura OS

- **Sít'ové OS**

- Umožňují procesům přistupovat ke zdrojům jiného počítače v síti.
 - Důležitý je sít'ový souborový systém
- Model klient-server.
- Komunikace přes sít'ový protokol.

Model klient-server



Architektura OS

- **Distribuované OS**

- Spravuje zdroje na více jak jednom počítači.
 - Vytváří iluzi, že mnoho počítačů jsou vlastně jeden superpočítač.
- Proces přistupuje ke všem systémovým zdrojům bez ohledu na umístění procesu nebo požadovaných zdrojů.
- Distribuovaný OS je těžké implementovat
 - Chord (MIT)
 - Amoeba (Vrije Universiteit in Amsterdam)
 - Sprite (University of California at Berkley)
 - Plan 9 (Bell Labs)

Dělení operačních systémů

- Podle úrovně sdílení CPU
- Podle počtu uživatelů
- Podle způsobu nasazení/požadavků na odezvu
- Podle velikosti hardwaru
- Podle počtu procesorů
- Podle míry distribuovanosti
- Podle funkcí
- ...

Cíle operačního systému

- Efektivita
- Robustnost
- Škálovatelnost
- Rozšiřitelnost
- Přenositelnost
- Bezpečnost
- Interaktivita
- Použitelnost



**Existují stovky
operačních systémů,
pouze jste o nich
nikdy neslyšeli.**