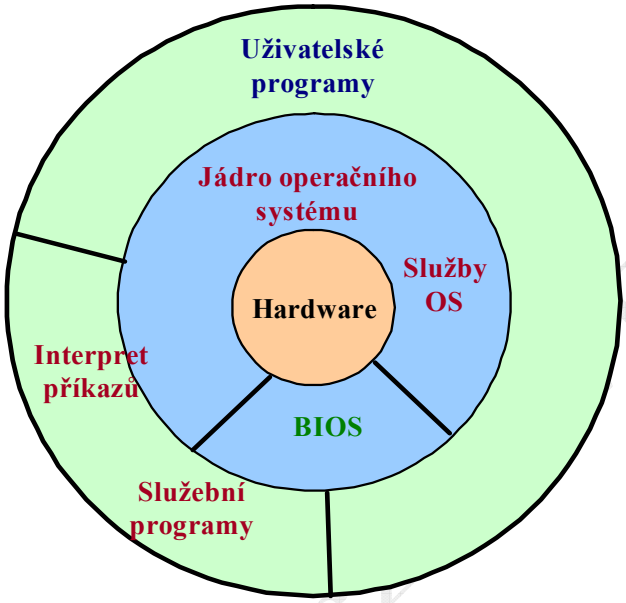
Principy operačních systémů

Struktura programového vybavení



Operační systém

- **Operační systém** je základní program, který se spouští automaticky po zapnutí počítače a který řídí činnost jeho technických prostředků podle požadavků uživatele.

- Dnešní počítače se mohou ovládat pouze pokud na nich běží operační systém. Bez spuštěného operačního systému jsou pro uživatele nepřístupné.

- **Úkoly operačního systému**
  - **Řídit** a synchronizovat procesy a **přidělovat** jim zdroje (prostředky),
  - Poskytovat služby programátorům a uživatelům. Vytváří tak zvané **virtuální prostředky** (např. adresářová struktura souborů),
  - Poskytuje procesům **systémové údaje** (např. datum, systémový čas, …).

Druhy operačních systémů

- Podle **počtu uživatelů**:
  - **jednouživatelské** (MS DOS)
  - **víceuživatelské** – je třeba kontrolovat práva uživatelů (Windows® 3.X, 95, 98, 2000, XP, Linux, …)

- Podle **počtu procesů**, které mohou probíhat „současně“: *(V každém okamžiku může být aktivní nejvýše jeden.)*
  - **jednoprocesový** (jednotaskový, jednoprogramový) – spuštěn pouze jeden uživatelský proces. (MS DOS)
  - **víceprocesový** (multitaskingový) - současně spuštěno více uživatelských procesů + operační systém (Windows® 3.X, 95, 98, 2000, XP, Linux, …)

- Podle **interaktivnosti**:
  - **dávkové zpracování** – programy byly načteny do vstupní fronty úloh a pak postupně zpracovány, výsledky procesů byly uloženy do výstupní fronty (případně vytištěny)
  - **interaktivní** operační systémy – zpracování programů řídí uživatel interaktivně z terminálu. (všechny současné operační systémy)

- Podle **způsoby práce**:
  - **lokální** – pracující místně, na lokálním počítači
  - **síťové** – pracující na dálku, v rámci počítačové sítě

Multitasking

- **Výhody multitaskingu**
  - lze přejít na jiný úkol bez ukončení rozpracovaného úkolu,
  - umožňuje činnosti, které mají probíhat paralelně (např. správa sítě),
  - programy mohou spolupracovat i jinak než předáním celých souborů dat,
  - lepší využití možností stroje (zatím co jeden proces čeká, může být čas procesor vyžíván jiným procesem,
  - je nutností pro víceuživatelský provoz.

- **Nevýhody (nebezpečí)**
  - vyšší spotřeba času na režii,
  - složitější, tedy dražší, s vyššími nároky na zdroje,
  - malá bezpečnost a vyšší poruchovost, není-li dobře navržen (nebezpečí chybné obsluhy determinismu, uváznutí a pod.).

- **Způsoby zajištění multitaskingu**
  - **kooperativní multitasking**. Procesy musely samy nabízet přerušení své práce. Dnes se již neužívá (Windows® 3.X)
  - **preemptivní multitasking**. Proces je přerušen vždy, když nastane událost (událostí je i uplynutí časového kvanta), která může mít vliv na přidělení prostředků procesům. Přerušení aktivuje operační systém a ten rozhodne, který proces bude dále pokračovat. (Windows® 95, 98, 2000, XP, Linux)

Procesy

- Základním pojmem v teorii operačních systémů je **proces**. Stručně řečeno proces je operačním systémem spuštěný a zpracovávaný program. Obsahuje paměť pro instrukce, pro data, vlastní registry, zásobník.

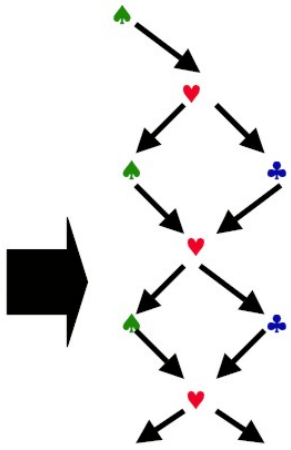
- **Procesy mohou být:**
  - **nezávislé** - nesmí se vzájemně ovlivňovat. Je nutné je oddělit.
  - **spolupracující** – procesy spolu spolupracují a sdílejí společná data. Přístup ke společným datům nutno řídit – **synchronizace procesů**

- **Determinismus** = požadavek stejného výsledku při spuštění se stejnými daty bez ohledu na konstelaci současně probíhajících procesů.
  - U nezávislých procesů zajistí operační systém tím, že oddělí data.
  - Pokud procesy spolupracují, musí mít společná data. Přístup ke společným datům nutno řídit.

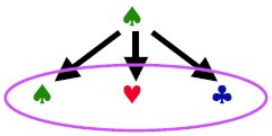
  - **Příklad**: Kopírování dat

♠ čti blok dat do S
♣ zapiš blok dat z T
♥ přesuň v paměti S → T

DETERMINISTICKÉ ZPRACOVÁNÍ



Současné spuštění všech tří procesů vede k nedeterminismu!



NEDERMNISMUS

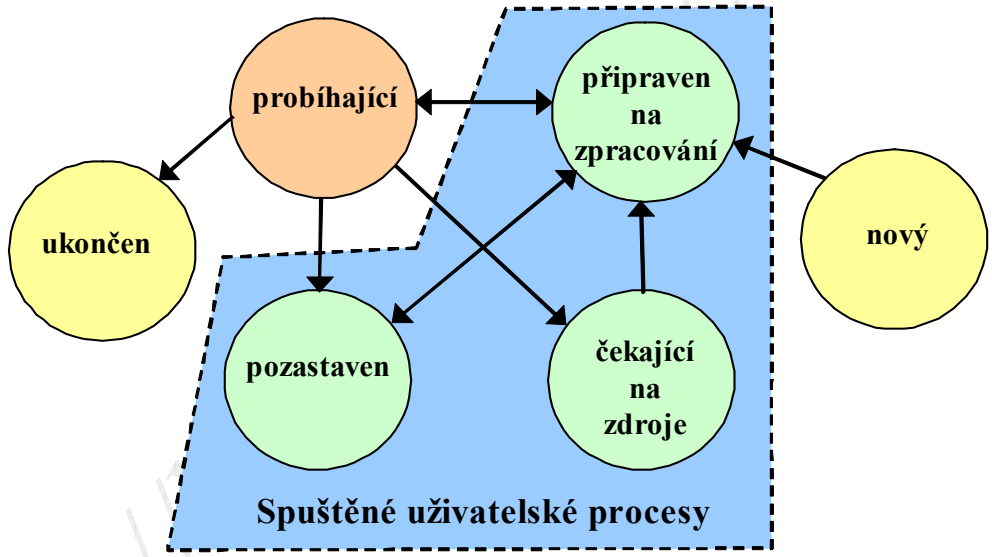
Pokud ♣ začne až po ukončení ♥, ale před druhým ♥ a ♠ až po ♥ bude prvý záznam překopírován správně. Začne-li ♣ dříve, překopíruje se nesmysl (původní obsah paměti). Při jiném průběhu se záznam může ztratit (být přepsán dříve než zapsán) nebo být zapsán několikrát.

Nedeterminismus byl způsoben tím, že procesy užívají společné proměnné S a T a přístup k nim není řízen.

Stavy procesů

- Proces může být během své existence v systému v různých stavech. Během své činnosti přechází z jednoho stavu do druhého. Všechny možné přechody mezi jednotlivými stavy se nejčastěji vyjadřují stavovým grafem.

- **Možné stavy procesu jsou:**
  - **probíhající** (*running*) - proces má přidělen procesor a je vykonáván.
  - **připraven na zpracování** (*ready*) - proces je připraven ke zpracování, ale nemá přidělen procesor
  - **pozastaven** (*stoped*) - proces je z nějakého důvodu pozastaven (zásahem uživatele nebo po uplynutí přiděleného časového kvanta) a nedostává přidělen procesor. Do stavu připraven se dostane až po svém probuzení.
  - **čekající na zdroje** (*waiting, blocked*) - proces čeká na splnění nějaké podmínky, aby mu byl přidělen procesor, čeká například na uvolnění nějakého sdíleného prostředku
  - **nový** (*new*) - proces byl nově vložen a po vytvoření všech potřebných systémových struktur bude zařazen ke zpracování
  - **ukončen** (*terminated*) – proces ukončil svoji činnost a bude uvolněn z paměti



- **Výběr procesu pro stav probíhá** (je-li více připravených procesů)

  - **Automaticky operačním systémem**
    - **FIFO** (nejdéle čekající)
    - Podle **priorit** (zvláštní případ je proces na popředí a na pozadí)
    - **Rozdělení času** –„time sharing“ (přidělí se časové kvantum, na základě přerušení od hodin, cyklická fronta)
    - **Kombinované** – (prevence tak zvaného „vyhladovění“)

  - **Obsluhou** (u PC)

Služby operačního systému

- Hranice mezi operačním systémem a programy uživatele není ostrá. Služby operačního systému lze pak rozdělit na:
  - **POVINNÉ** - Některé činnosti nemůže vykonávat proces přímo Program o ně musí požádat operační systém
  - **NEPOVINNÉ** - Některé problémy v principu vlastním programem řešit lze, ale není to výhodné.

Služby vstupu a výstupu

- Služby **vstupu a výstupu** – **I/O** (*Input*/*Output*) patří mezi **povinné** a to z důvodů:
  - Jen operační systém má přehled o tom, která zařízení jsou volná – musí být **arbitr** (na dvou úrovních)
    - řízení procesů
    - pro nesdílitelná zařízení ještě řízení úloh).
  - Nelze ohrozit HW ani další procesy.
  - Přímé programování by bylo příliš náročné.
  - Uživatelský program nemůže záviset na jemných rozdílech funkce jednotlivých zařízení.
- Proto se rozlišuje:
  - **Logická úroveň** I/O (v programovacích jazycích)
  - **Fyzická úroveň** I/O (realizuje operační systém)

Logická úroveň I/O

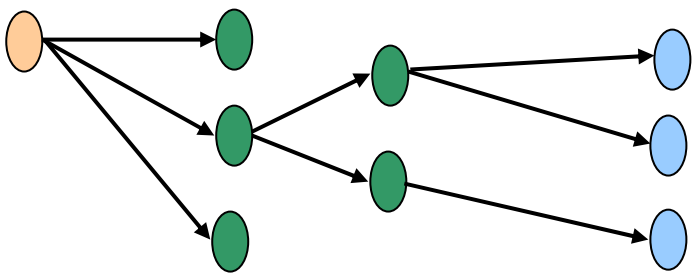
- **Data členěna** do logických vět (=**záznamů**)
  - Vždy je k dispozici celý záznam.
  - Záznam může mít svoji **vnitřní strukturu** (i v několika úrovních).
- Množina záznamů stejného typu tvoří **SOUBOR** nebo část **BÁZE DAT** (např. tabulku).
  - **Souborová organizace** je jednodušší, snáze se s ní pracuje, ale může být nadbytečná a náročná na údržbu
    - Podle přístupu lze rozdělit na **sekvenční** nebo **přímé**
  - **Databázová organizace** je složitější, náročnější, ale úsporná (tatáž data pro různý účel) a jednodušší na údržbu.
    - Při databázovém přístupu neprobíhá komunikace s vnějším zařízením a programem přímo, ale pouze prostřednictvím **systému pro řízení báze dat** **(SŘBD)**. Ten data plně odstiňuje od programu.
    - Databáze lze rozdělit na **síťové**, **relační** (tabulkové) nebo **objektové**

Fyzická úroveň I/O

- Realizuje operační systém **privilegovanými instrukcemi**
- Na základě znalostí o skutečných parametrech připojeného zařízení se vytvoří tak zvaný „**kanálový program**“ = program v kódu řadiče příslušného vnějšího zařízení (vnější paměti).
  - Pomocí privilegované instrukce START IO vyšle po sběrnici (kanálu) tento program, včetně informace o umístění vysílaných nebo přijímaných dat v hlavní paměti.
  - Kanálový program provádí **řadič** (= řídicí jednotka) vnějšího zařízení již autonomně.
  - Procesor se může věnovat další práci na stejném nebo jiném procese.
  - Po ukončení své činnosti vyvolá řadič příslušného zařízení přerušení. Tím sdělí operačnímu systému, že přenos dat je ukončen a že zařízení může přijímat další úkoly.
- **Od fyzického I/O je uživatel plně ostíněn**.

Další služby operačního systému

- **PODPORA TELEKOMUNIKACÍ**
  - **Internet**
  - **Elektronická pošta** – různí „poštovní agenti“
  - **Prohlížeče www**
  - **...**
- **ZABEZPEČENÍ DAT**
  - Proti záměrnému i nezáměrnému použití a zničení
  - Přístupu k počítači, přístupu k souborům, komunikace ...
- **KOMPILÁTORY A INTERPRETY**
  - Dnes obvykle s celým prostředím obsahujícím knihovny, prostředky pro zkoušení programů.
- **OBSLUHA KNIHOVEN**
  - Obvykle organizovány jako členěný soubor ve více úrovních.
  - Hierarchická **struktura adresářů**:



kořen **adresáře** **soubory**

- **Programy** mohou být
  - **v proveditelném kódu** (kód stroje)
  - **v relativních adresách** (příznak, které adresy posunout)
  - **v** různých „**mezikódech**“ (např. pro zajištění přenosu na jinou platformu nebo interpretaci na této platformě)
  - **v zdrojovém jazyku**

STANDARDNÍ SLUŽEBNÍ PROGRAMY

- **utility** (konverzní programy mezi různými formáty, komprimace dat, řazení, vyhledávání ...)
- **standardní algoritmy** (matematické funkce, generátory pseudonáhodných čísel, ...)
- **základní funkční programy** (zpracování různých formátů informace, např. digitálních obrázků, vytváření grafických rozhraní pro interakční styk s počítačem,...)
- **podpora různých typických aplikací** (textové editory, tabulkové procesory, presentační grafika, plánovaní práce, návrh staveb, strojových součástek, kreslení map, ...)
- **...**

Operační systém Microsoft® Windows

Uživatelé systému

- Operační systém Microsoft® Windows je obecně **jednouživatelský**, tj. v daném okamžiku může obsluhovat pouze jednoho uživatele. Přesto umožňuje ochranu systému i uživatelů před jinými uživateli a jejich programy, a to přihlašováním uživatelů pomocí svého uživatelského jména a hesla.

Systém souborů

- **Soubor** je sekvence bitů, bytů, řádek nebo záznamů, jejichž význam je definován zakladatelem a uživatelem souboru. Soubory jsou ukládány na vnější paměťová zařízení počítače (např. pevné disky) a tvoří dohromady **systém souborů**.

- Systémy souborů jsou v operačním systému Windows oddělené na jednotlivé disky a označují se velkými písmeny (A:, B:, C:, …). Z uživatelského hlediska jsou soubory logicky organizovány do **adresářů** a adresářových struktur, tj. určitých množin souborů.

Architektura operačního systému

- V **architektuře operačního systému** Windows® (Windows NT®, 2000, XP) je použito několik modelů:

  - **Vrstvená architektura** - Programy vždy pracují v neprivilegovaném (uživatelském) režimu, mají omezený přístup k systémovým zdrojům, při volání služby operačního systému OS toto volání zachytí a přepne do privilegovaného režimu, kód patřící do určité vrstvy může volat pouze kód nižší vrstvy. Tento model se uplatňuje například v jádře systému, vrstvě abstrakce HW (vrstva přímo pracující s HW), a dále v I/O systému.

  - **Modulární architektura** - Systém se skládá z uzavřených modulů, z nichž každý poskytuje určité služby přes stanovené rozhraní. Modulární architekturu používá především exekutiva (řícící program operačního systému), a to:
    - správce procesů,
    - správce paměti,
    - I/O systém, atd.

  - **Architektura klient-server** - Systém je rozdělen na **mikrojádro** (běží v privilegovaném režimu) a další (systémové) procesy (tzv. servery, běží v uživatelském režimu a proto nemohou zasahovat do jádra), každý tento proces zajišťuje určité služby. Když server selže nebo je poškozen, může být znovu spuštěn, jádro zůstane stabilní. Ostatní procesy jsou klienti, kteří využívají služeb serverů.

Procesy a multitasking

- **MS-DOS** (*Microsoft Disc Operating System*) - Byl navržen tak, aby byl malý a oddělený od aplikací, takže nenabízel o moc víc, než nahrání aplikace do paměti a přístup k souborovému systému. Některé rezidentní programy (například obsluha tisku) se pověsily na přerušení od hardwarových hodin a prováděly zpracování na pozadí. Jiné utility jako např. SideKick simulovaly jakési přepínáni úloh - aplikace byla pozastavena a běžela utilita.

- **Microsoft® Windows** - Prostředí oken umožňuje programům současně běžet na jediné obrazovce. Je velmi jednoduché přepínat se mezi aplikacemi a dokonce přenášet data mezi programy. Mechanismus přepínání úloh (multitasking) prošel následujícím vývojem:

  - **nepreemptivní multitasking** - Tato forma multitaskingu byla umožněna díky architektuře Windows založené na zprávách. V obecném případě aplikace nečinně sedí v paměti a čeká, až dostane zprávu. Tyto zprávy bývají obvykle přímým či nepřímým důsledkem uživatelského vstupu - například stisku klávesy nebo pohybu myši. Jakmile program zprávu zpracuje, předává řízení zpět do Windows. Přepínáni mezi úlohami se odehrávalo jen tehdy, když aplikace dokončila zpracování zprávy a předala řízení Windows. Této formě nepreemptivního multitaskingu se někdy říká také **kooperativní multitasking**, protože vyžaduje jistou spolupráci aplikací. Pokud by program zpracovával nějakou zprávu příliš dlouho, mohl tím zablokovat celý systém.

  - **multithreading** - V multithreadingovém prostředí se program může rozdělit na několik samostatných částí, takzvaných prováděných vláken (*threads*), která běží současně. Z pohledu programu je vlákno reprezentováno funkcí, která může volat další funkce. Program začíná pracovat hlavním vláknem, které v prostředí Windows vykonává funkci WinMain. Jakmile program běží, může speciálním systémovým voláním (*CreateThread*) vytvářet další vlákna. Operační systém pak preemptivně přepíná řízení mezi vlákny. Hlavní vlákno vytvoří všechna okna, která program potřebuje, včetně všech procedur těchto oken, a zpracovává všechny zprávy těchto oken. Všechna ostatní vlákna pak představují výkonné mechanismy na pozadí. Nijak nekomunikují s uživatelem vyjma komunikace s hlavním vláknem. Vlákna v jednom programu jsou součástí stejného procesu, takže sdílejí všechny prostředky procesu, jako je například paměť nebo otevřené soubory. K usnadnění koordinace činnosti vláken nabízí operační systém různé metody pro synchronizaci vláken. Jednou z metod jsou semafory, které umožňují programátorovi pozastavit činnost jednoho vlákna v nějakém místě až do doby, dokud jiné vlákno neoznámí, že je možno pokračovat. Semaforům se podobají také kritické sekce, což jsou sekce kódu, které může vykonávat vždy jen jediné vlákno.

Operační systém LINUX

- Linux je založen na operačním systému UNIX s víceuživatelskou a **víceprocesovou** architekturou.

Uživatelé systému

- Každý soubor, služba a aplikace jsou exkluzivně přiděleny konkrétnímu uživateli nebo skupině uživatelů.
- **Každý uživatel** má v systému přiděleno jednoznačné **uživatelské číslo** (*UID, user identifier*), **uživatelské jméno**, přístupové **heslo** a svůj **osobní** (domovský) **adresář**, který je nepřístupný (nebo i skrytý) pro ostatní uživatele. V případě týmové práce je však možné změnit nastavení adresáře nebo souboru (změnou přístupových práv) a sdílet je s ostatními uživateli.
- Uživatelé jsou rozděleni do skupin. Každá **skupina** má své jméno a identifikační číslo (*GID, group identifier*).
- Existuje zde i **superuživatel**, který slouží jako administrátor všech ostatních uživatelů a má přístup ke všem souborům všech uživatelů.

Systém souborů

- **Soubor** je v LINUXu posloupnost určitého počtu bytů.
- Diskové soubory jsou organizovány do větších celků, které se nazývají **systémy souborů**. Na jednom fyzickém disku může být vytvořen jeden či několik systémů souborů.
- Systémy souborů jsou dostupné prostřednictvím tzv. **bodů připojení**. Z pohledu uživatele se jedná o adresáře, které jsou součástí adresářové struktury. Uživatel se pouhým přepnutím do požadovaného adresáře ocitne v jiném systému souborů.
- Soubory systému souborů jsou pomocí adresářů organizovány do **stromové adresářové struktury**, začínající **hlavním adresářem** (kořenovým adresářem).
- Se souborem nemůže proces pracovat přímo, ale musí použít některou ze služeb systému určenou pro práci se soubory. Tyto služby realizuje ta část operačního systému, která se nazývá **správce souborů**.

Procesy a multitasking

- **Proces je spuštěný program**, zapsaný v některém z programovacích jazyků (obvykle v jazyku C) a přeložený do strojového kódu. Linux jako velmi výkonný operační systém umožňuje běh více procesů současně (multitasking).

- Program může být **spuštěn na popředí** nebo **na pozadí**.
  - Pokud je program spuštěn na popředí, operační systém čeká na jeho skončení. Teprve poté může uživatel spustit další program.
  - Pokud je program spuštěn na pozadí, operační systém nečeká na jeho ukončení a je schopen okamžitě přijmout další program. Spuštění na pozadí se provede tak, že spouštěný program se ukončí znakem **&**.

- Během své činnosti proces obvykle potřebuje od jádra operačního systému celou řadu **služeb**. V tom případě musí požádat o službu systému. Například potřebuje:
  - pracovat se souborem
  - zaslat jinému procesu zprávu či signál
  - použít některé I/O zařízení
  - zjistit stav některého systémového prostředku (např systémového času)

- **Systémové služby** jsou v Unixu realizovány pomocí vnitřního přerušení.
  - To znamená, že proces použije pro volání služby systému instrukci, která generuje vnitřní **přerušení**. Následuje přerušení činnosti procesu a skok do jádra systému a to do místa, ve kterém systém analyzuje, o kterou službu se jedná.
  - Po zjištění, o kterou službu se jedná, je proveden skok na tu část kódu jádra, která požadovanou službu realizuje.

- V určitém okamžiku operační systém zpracovává řádově desítky procesů. Některé z nich plní systémové úkoly, některé řeší úlohy zadané uživateli. Aby procesy mohly být jednoznačně identifikovány, systém jim při jejich vzniku přiřadí uvnitř systému jednoznačné kladné číslo, tzv. **identifikační číslo procesu** (*PID, process identifier*).

- Operační systém udržuje v **tabulce procesů** informace, které potřebuje o procesech znát. Jsou to například informace nutné pro rozhodnutí, zda bude proces vybrán ke spuštění, tj. stav procesu, jeho priorita, dosud spotřebovaný procesorový čas atd.

Stavy procesů

- Po spuštění se procesy mohou nacházet v různých **stavech**. Všechny možné stavy a přechody mezi jednotlivými stavy znázorňuje obrázek.

- Základním prostředkem, pomocí kterého lze ovlivnit stav procesu, jsou **signály**. Signál může procesu zaslat jádro systému nebo jiný proces. Zasílání signálů procesu je možné systémovým programem **kill**.