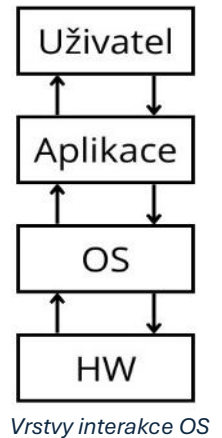
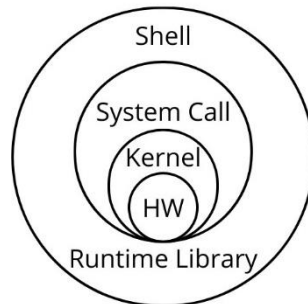


6. Operační systém a plánování procesů

- Charakteristika OS
- Typy jader OS
- Proces vs. vlákno
 - Charakteristika
 - PCB vs. TCB
- Přepínání kontextu
- Plánovače OS
 - Preemptivní vs. nepreemptivní plánování
- Plánovací algoritmy
 - FCFS
 - SJF
 - SRTF
 - RR
 - OPS
 - MFQS

1. Charakteristika OS

- Základní programové vybavení PC
 - Kolekce programů tvořící spojující vrstvu mezi HW a uživatelskými programy
- Při startu je zaveden do OP, kde zůstává až do konce činnosti
- Zajišťuje abstrakci pro různorodý hardware (není potřeba znát daný HW a psát program přímo pro něj)
- Zajišťuje správu:
 - Procesů
 - Paměti (vnitřní, vnější, virtuální)
 - Souborového systému
 - Periferií
 - UI
- Shell = terminál
- Runtime library = knihovna s příkazy určené pro uživatele pro komunikaci s Kernelem

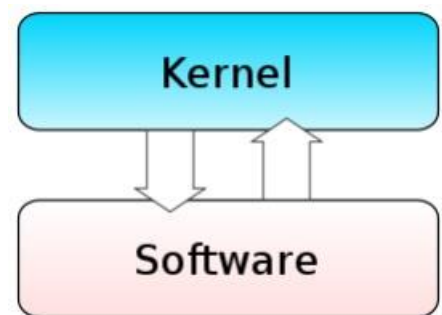


Jádro OS

- Říká se mu CORE nebo KERNEL
- Je to nejnižší, nejdůležitější a nejzákladnější část OS
- Je zaveden jako první při startu OS do operační paměti, zůstává tam až do vypnutí PC
- Jádro běží v privilegovaném režimu -> nikdy neztratí kontrolu nad PC
- Jádro komunikuje s HW, na který přímo navazuje a od uživatele je zcela zapouzdřen
- Systém a uživatelské programy žádají jádro o služby prostřednictvím systémových volání (system call)

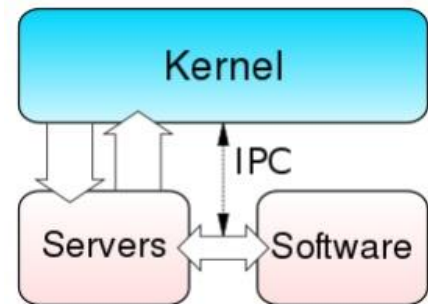
2. Typy jader OS

- Monolitické
 - Tradiční struktura OS
 - Veškerý kód běží v privilegovaném režimu za účelem vysoké efektivity
 - Jeden program obsahuje celý kód jádra a poskytuje všechny služby OS
 - Systémové volání
 - Méně přenositelné a těžké na údržbu
 - Chyba v jednom subsystému může ovlivnit další
 - MS-DOS, WIN 95/98, Mac OS do 8.6



- Mikrojádro

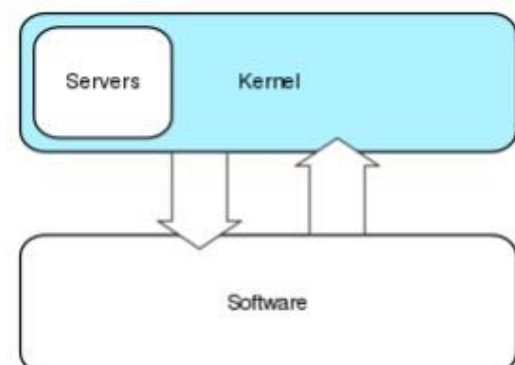
- Minimalizuje rozsah jádra a nabízí jednoduché rozhraní s jednoduchými abstrakcemi a malým počtem služeb pro nejzákladnější správu CPU, I/O zařízení, paměti a IPC
- Ostatní služby jako jsou plánování, správa SW systému, ovladače zařízení jsou implementovány mimo jádro v podobě serveru (každá komponenta má svůj server)
- Aby to vypadalo, že běží více úloh současně, přepíná úlohy 240mil./s. => může se stát, že bude trávit více času přepínáním kontextu, než zpracováváním
- Výhody
 - Jednodušší programování díky rozdělení na logické celky
 - Flexibilita – možnost více současně běžících implementací služeb
 - Bezpečnost – v případě napadení, nehrozí pád PC
- Nevýhoda
 - Vysoká režie IPC = mezi-procesová komunikace
- Minix, Symbian OS



Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux,BSDs(FreeBSD,OpenBSD,NetBSD)etc.

- Hybridní

- Kombinace monolitického a micro jádra
- Objevuje se v dnešních OS
- Micro jádro je rozšířeno o kód, který by mohlo běžet v podobě serveru v uživatelském režimu, ale za účelem zmenšení režie IPC je těsněji provázán a běží v privilegovaném režimu v podobě serveru

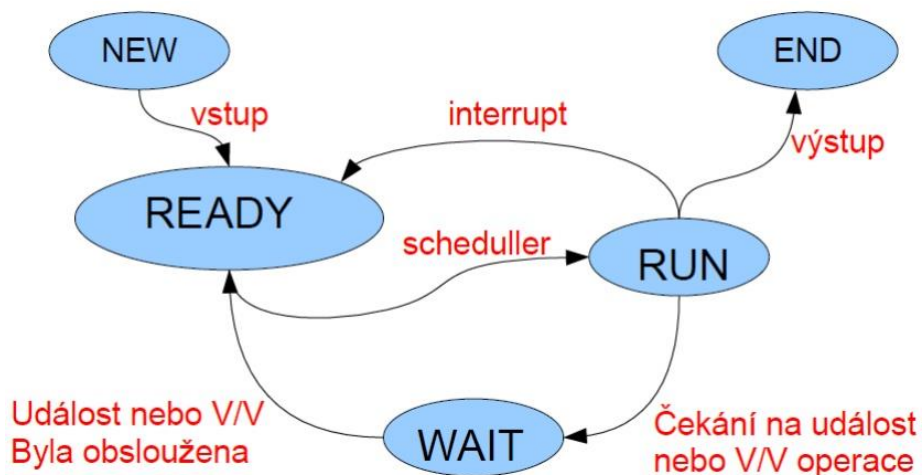


- Nedokáže za běhu samo zavádět moduly
- Windows, Mac OS X

3. Proces vs. vlákno

- Charakteristika

- Proces = program je název pro spuštěný (běžící) PC program
- Proces je umístěn v operační paměti v podobě sledu strojových instrukcí vykonávané procesorem
- Obsahuje strojový kód a dynamicky měnící se data, které procesor zpracovává
- Jeden program může v PC běžet jako více procesů s různými daty (více krát spuštěný web zobrazující různé stránky)
- Správu procesů vykonává OS, který zajišťuje jejich běh, přiděluje jim systémové prostředky PC a umožňuje uživateli procesy spravovat – spouštět, ukončovat,...
- Proces je v OS definován
 - Identifikátorem – PID
 - Programem, který je řízen
 - Obsahem registru - čítač instrukcí, adresa zásobníku
 - Daty
- Procesů běží v OS spousty a je nutné je spravovat
 - Proces management – správa procesů
 - Přepínání kontextů - velmi náročné (vznik vláken)
 - Plánovač = dispatcher – plánuje na základě plánovacího algoritmu
 - Správa paměti
 - Podpora meziprocessorové komunikace
- Proces se může nacházet v 5 stavech
 - NEW – vytvořený, nový
 - Proces je vytvořen buď příkazem uživatelem, nebo na žádost OS
 - Strojový kód je schedulerem zaveden do operační paměti
 - READY – připravený
 - Proces je připravený pro vstup do stavu run, čeká pouze na přidělení procesoru
 - RUN – běžící
 - Procesu je přidělen procesor, a právě provádí příslušné operace
 - WAIT – čekající, blokový
 - Proces je převeden do tohoto stavu v případě, kdy čeká na dokončení nějaké vstupně-výstupní operace, případně na skončení jiného procesu
 - fronta procesů čekajících na vstupní, nebo výstupní události anebo, na nějaký signál, po obslužení se přesunou do fronty ready
 - END – ukončený
 - Proces se zpracoval a je ukončený



PCB vs. TCB

- PCB (Process Control Block) je datová struktura v OS, která obsahuje informace o běžícím procesu
- Obsah PCB
 - Ukazatel
 - Odkaz na další PCB
 - Stav procesu, v jakém stavu momentálně proces je
 - Číslo procesu – PID
 - Jedinečné číslo procesu, pod kterým vystupuje
 - PC – Program counter
 - Číslo instrukce, kde skončil
 - Registry
 - Záloha registrů, protože by je jiné programy mohly využívat
 - Registry mají své vlákna
 - Vlákna vytvářejí kritickou sekci
 - Limit paměti
 - Vyhrazený adresní prostor procesu
 - Seznam otevřených souborů
 - Slouží pro případ, že programátor nezavře soubor, aby jej systém mohl zavřít za něj
 - Priorita, ukazatel na paměť, kdy naposled běžel, jak dlouho, ...
- TCB (Thread Control Block) je odlehčený proces a je jeho součástí
 - Sám o sobě nemůže existovat
 - Pomocí vlákna se snižuje režie OS, protože jsou schopny rychleji přepínat kontext na uživatelské úrovni než proces v privilegovaném režimu
 - Vlákna mají sdílenou paměť
 - Vlákna mají stejná práva jako proces
 - Hyperthreading – jsme schopni zpracovávat více instrukcí, než máme vláken
- Obsah TCB
 - Tabulka vlákna nacházející se v jádře OS
 - Datová struktura obsahující informace o vlákně
 - Jednoznačný identifikátor vlákna – TID
 - Ukazatel na proces – PCB, ke kterému se vztahuje (žije v něm)
 - Ukazatel na aktuální programovou instrukci vlákna

- Stav vlákna
- Registry
- Vlákno na uživatelské úrovni
 - Tím, že jsou vlákna spravována na uživatelské úrovni, tak o nich OS neví – jsou nezávislé
 - Není nutno volat jádro OS pro práci s vlákny
 - Jsou v plné režii programátora
 - Výhody:
 - Rychlé přepínání mezi vlákny
 - Uživatelský proces má nad vlákny plnou kontrolu
 - Nevýhody:
 - Vlákno neví o jádrech, tudíž přiděluje procesorový čas procesům a není možné, aby 2 vlákna stejného procesu běžela současně, a to i v případě víceprocesorového systému
- Vlákno na úrovni jádra OS
 - O všechno se stará jádro OS
 - Jeden proces může využívat více procesorů
 - Volání služby neblokuje ostatní vlákna z CPU
 - Náročnější správa
 - Nespravedlivé plánování – CPU čas je přidělován vláknům a ne procesu
 - Použití vláken
 - Obsluha periférií – 1 vlákno komunikace s HW, 2 vlákno komunikace s uživatelem
 - U serverů pro každého uživatele, který se připojí na server, bude vyhrazené 1 vlákno
 - Výhody vláken:
 - Menší režie
 - Urychlení výpočtu, odezvy programu, celkového běhu
 - Efektivní využití systému
 - Jednodušší sdílení a komunikace než mezi procesy
 - Paralelní běh
 - Lepší a přehlednější strukturalizace programu
 - Nevýhody vláken
 - Omezení počtu vytvořených vláken
 - Náročnější kód pro řešení souběhu vláken a pro sdílení prostředků

4. Přepínání kontextu

- Přepínání kontextu je operace, při kterém operační systém ukládá stav běžícího procesu nebo vlákna a načítá stav jiného procesu nebo vlákna, aby mohl pokračovat ve svém provádění.

5. Plánovače OS

- **Preemptivní vs. nepreemptivní plánování**
 - V preemptivním plánování může OS přerušit běžící úlohu a přiřadit CPU jiné úloze, zatímco v nepreemptivním plánování úloha běží, dokud se nedokončí nebo nevyvolá I/O operace.

6. Plánovací algoritmy

- FCFS
 - **(First-Come, First-Served):** Procesy jsou plánovány podle jejich příchodu.
- SJF
 - **(Shortest Job First):** Proces s nejkratší dobou běhu je vybrán pro provádění jako první.
- SRTF
 - **(Shortest Remaining Time First):** Proces s nejkratší zbývajícím dobou běhu je vybrán pro provádění jako první, pokud se objeví proces s ještě kratší dobou běhu, je dosavadní proces přerušen.
- RR
 - **(Round Robin):** Každý proces dostává časový kvantum na běh, po kterém je přepnut na další proces.
- PS
 - **(Priority Scheduling):** Procesy jsou plánovány podle jejich priority.
- MFQS
 - **(Multilevel Feedback Queue Scheduling):** Procesy jsou rozděleny do front podle priority a každá fronta má svůj vlastní plánovač s různým kvantem času. Procesy se mohou přesouvat mezi frontami na základě jejich chování.