INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA STROJNÍ

OPERAČNÍ SYSTÉMY

ÚVOD DO TEORIE OPERAČNÍCH SYSTÉMŮ

doc. Dr. Ing. Oldřich Kodym

Ostrava 2013

ISBN 978-80-248-3053-7

Tento studijní materiál vznikl za finanční podpory Evropského sociálního fondu (ESF) a rozpočtu České republiky v rámci řešení projektu: CZ.1.07/2.2.00/15.0463, MODERNIZACE VÝUKOVÝCH MATERIÁLŮ A DIDAKTICKÝCH METOD

OBSAH

1. ÚVOD DO TEORIE OPERAČNÍCH SYSTÉMŮ

OBSAH KAPITOLY:

Základní pojmy.

Připomenutí funkce procesoru.

Procesy.

Struktura operačního systému.

MOTIVACE:

Výběr operačního systému a stanovení jeho možností je u výpočetního systému při daném technickém vybavení nejdůležitějším rozhodnutím. Každý uživatel se setkává s operačním systémem při zadávání úloh, neboť operační systém poskytuje uživateli „základní spojení s počítačem“. Mnohé pojmy a techniky uplatněné v operačních systémech mají obecnější použití i v některých jiných aplikacích. Možnost vytvořit pro speciální účely vlastní operační systém nebo stávající systém modifikovat.

CÍL:

Základní pojmy teorie operačních systémů, funkce procesoru, procesy v OS, hierarchická struktura OS

1. ZÁKLADNÍ POJMY TEORIE OPERAČNÍCH SYSTÉMŮ

Holý počítač – počítač pouze s nejzákladnějším softwarovým vybavením; pro běžného uživatele zcela neovladatelný. V následujícím textu bude pro zjednodušení, a nebude-li uvedeno jinak, předpokládán holý počítač s jedním jednojádrovým procesorem.

Operační systém – ovládá základní technické prostředky počítače a vytváří vhodnější podmínky pro jejich využívaní v uživatelských programech. Funkce operačního systému tvoří podstatnou složku činností počítače a mnozí uživatelé je ani nerozlišují od funkcí technického vybavení.

Proč studovat operační systémy?

- Výběr operačního systému a stanovení jeho možností je u výpočetního systému při daném technickém vybavení nejdůležitějším rozhodnutím.
- Každý uživatel se setkává s operačním systémem při zadávání úloh, neboť operační systém poskytuje uživateli "základní spojení s počítačem".
- Mnohé pojmy a techniky uplatněné v operačních systémech mají obecnější použití i v některých jiných aplikacích.
- Možnost vytvořit pro speciální účely vlastní operační systém nebo stávající systém modifikovat.

Operační systém jsou ty programové moduly ve výpočetním systému, jež ovládají řízení prostředků, jimiž je tento výpočetní systém vybaven, jako jsou procesory, operační paměť, vnější paměť, I/O zařízení a soubory dat. Tyto moduly "rozhodují spory" (např. o užití téhož prostředku více úlohami), snaží se optimalizovat výkon a zjednodušují efektivní využívání výpočetního systému.

– definice nezahrnuje problémově orientované moduly OS – překladače, knihovny podprogramů a ladicí prostředky, neboť tyto již jsou uživateli OS

Uživatel – každý, kdo dává svou zakázku ke zpracování výpočetnímu systému.

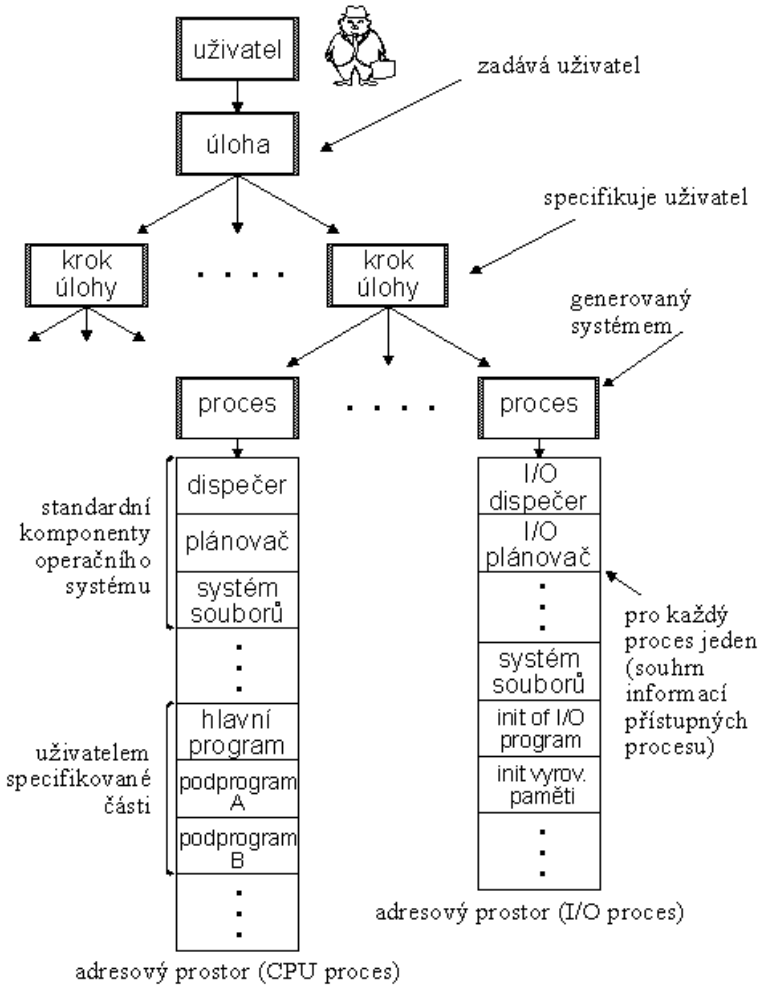
Úloha (Job) – souhrn činností potřebných k provedení teto zakázky; může být rozdělena na kroky.

Kroky úlohy – jednotky činností, které musí byt provedeny postupně v určitém pořadí (např. překlad programu, zavedeni programu, spuštěni programu apod.)

Proces – instance úlohy, kterou vytváří procesor a která může být prováděna paralelně s jinými výpočty.

Adresový prostor – souhrn programu (instrukcí) a dat v procesu.

- Nutno zobrazit adresové prostory jednotlivých procesů do operační paměti – stránkování (paged system) nebo technika výměn (swapping). Blíže viz kap. Modul přidělování paměti.



Obrázek 1 - Uživatel, úloha, krok úlohy, proces a adresový prostor

Multiprogramový systém – systém, v němž může být více procesů najednou ve **stavu provádění**. Proces je ve stavu provádění, jestliže byl zahájen a nebyl ještě dokončen nebo zastaven (popř. ukončen s chybou).

- Proces může být ve stavu provádění, ale ve skutečnosti nemusí být právě prováděn, tj. některé mezivýsledky jsou spočítány, ale procesor provádí v daném okamžiku některý jiný proces.
- Současný běh více procesů je zdánlivý, protože v daném okamžiku může procesor provádět vždy jen jeden z nich.

Privilegovaný stav CPU (supervisor state) – procesor může provádět i privilegované instrukce (změna stavu CPU, zahájeni I/O operace, změna způsobu ošetření přerušení apod.) a nemůže být přerušen.

Uživatelský stav CPU (user state) – běžný stav procesoru.

Ochrana paměti (protection hardware) – OS může zakázat zápis do určité části paměti. Může tak např. znemožnit uživatelským programům měnit OS.

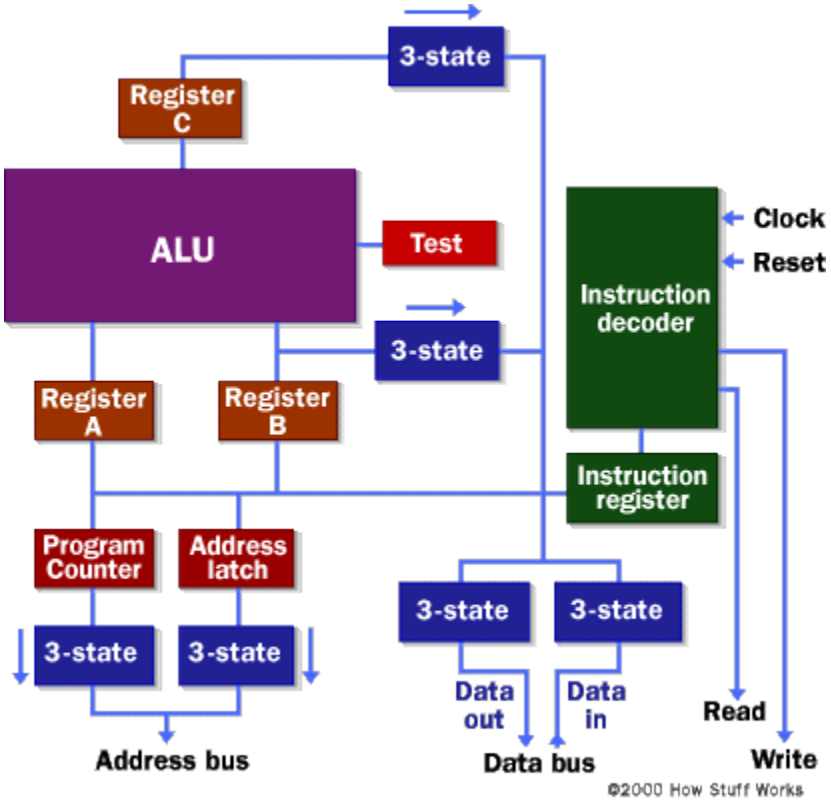
Prostředky přerušení (interrupt hardware) – dovolují OS koordinovat paralelně probíhající operace – tím je umožněn paralelní běh uživatelských programů. **Přerušení** je proces, během kterého je procesor nucen zaznamenat nějakou událost. Stejně tak existují prostředky k **maskování přerušení** (tj. jeho potlačení).

V oblasti výpočetní techniky (a tedy i v následujícím textu) se jako velikosti dat historicky používají jednotky B, KB, MB atd., které mají poněkud jiné velikosti než je obvyklé

v ostatních oblastech. Je to dáno použitím nikoliv soustavy desítkové, ale dvojkové. Platí tedy 1 MB = 1024 KB a 1 KB = 1024 B. Dnes se pro předejití nejasnostem oba způsoby rozlišují. V naší zájmové oblasti se tedy nově používá označení KiB, MiB, GiB atd. – počítané jako mocniny dvou, tedy 2^{10}, 2^{20}, 2^{30} atd., na rozdíl od běžně používaných 10^3, 10^6, 10^9 atd.

2. PŘIPOMENUTÍ: JAK PRACUJE PROCESOR

Operační systémy jsou vždy vázány na konkrétní hardware výpočetního systému. Musí tomu tak být, mají-li být využity veškeré možnosti technických prostředků, má-li celý systém pracovat optimálně, maximálně efektivně. Určujícím prvkem výpočetního systému je především procesor. Připomeňme si proto základy funkce procesoru. Mějme na mysli, že níže uvedené připomenutí je velmi významně zjednodušeno a jednotlivé procesory se mohou v některých aspektech lišit. I přes toto riziko považuji níže uvedené připomenutí za užitečné.



Obrázek 2 - Blokové schéma jednojádrového procesoru

- Základními funkčními bloky procesoru je ALU – *aritmeticko-logická jednotka*, Instruction decoder – *řídící jednotka* a *registry* (A, B, C, Test – příznakový registr a Instruction). *3-state* jsou spínače, které řídí propojování jednotlivých bloků a tím tok informací mezi nimi a vnějšími sběrnicemi (adresovou a datovou.
- Instruction decoder vygeneruje adresu uložení následující instrukce a uloží ji do registru Program Counter.
- Obsah registru Program Counter je aktivací příslušného třístavového spínače (3-state) přenesen na adresovou sběrnici systému.
- Aktivací signálu Read je z definované adresy operační paměti načten přes datovou sběrnici obsah do registru instrukce (Instruction register) – zavedli jsme instrukci programu, tato bude dále zpracována.

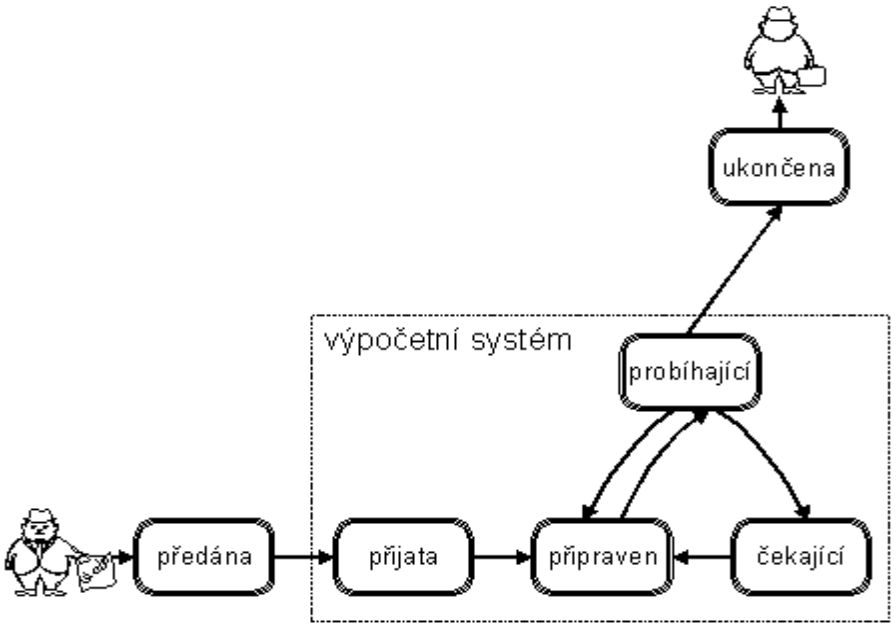
- Instrukce je dekódována a obsahuje-li adresu operandů, pak je tato adresa uložena do registru Address latch. Odtud ji lze přivést na adresovou sběrnici.
- Aktivací signálu Read je z definované adresy operační paměti načten přes datovou sběrnici obsah do registru A nebo B – zavedli jsme data.
- Je-li obsahem instrukce příkaz pro ALU (aritmeticko-logickou jednotku), je tento vykonán a výsledek je uložen do registru C (akumulátor). Současně podle výsledku je modifikován obsah registru Test.
- Jde-li o instrukci zápisu do paměti, pak aktivací signálu Write bude obsah registru C přes datovou sběrnici zapsán do paměti na adresu uloženou v registru Address latch.
- Po vykonání instrukce je načtena a zpracována instrukce další (ta, která je umístěna v operační paměti na adrese uložené v registru Program Counter). Současně je obsah programového čítače nastaven na adresu další instrukce.

3. OPERAČNÍ SYSTÉM Z HLEDISKA PROCESŮ

3.1 Stavy procesů

Životní cyklus procesu v OS se skládá z přechodů mezi třemi hlavními stavy:

- **Stav probíhající (running)** – procesu je přidělen procesor a právě se provádí příslušné programy.
- **Stav čekající (waiting)** – proces čeká na určitou událost, např. dokončeni I/O operace.
- **Stav připraven (ready)** – proces je přípraven k vykonání a čeká pouze na přidělení procesoru.



Obrázek 3 - Model stavů procesu

Tyto 3 hlavní stavy procesu nestačí pro úplný popis životního cyklu úlohy v OS. Pro úplnost doplňme ještě další významné stavy:

- **Stav předána (submit)** – uživatel předal svou úlohu systému a ten na ni musí reagovat. Stav mírně archaický, spočívající např. ve vložení sady děrných štítků do čtečky. Dnes odpovídá spíše kliknutí na ikonu.
- **Stav přijata (hold)** – úloha je na disku počítače ve vnitřní reprezentaci. Očekává přidělení prostředků.
- **Stav ukončena (complete)** – výpočet úlohy skončil a všechny přidělené prostředky jsou uvolněny k dalšímu použiti.

3.2 Moduly operačního systému

Přechody mezi stavy procesu zajištují moduly OS, které musí jako správa prostředků počítače:

- Mít přehled o jednotlivých prostředcích.
- Realizovat pravidla, která určují, komu bude prostředek přidělen, kdy a v jakém rozsahu.
- Prostředky přidělovat a vyžadovat jejich navrácení.

Modul přidělování procesoru

- *Plánovač úloh* – sleduje a eviduje stav všech úloh v systému, které si uchovává ve frontě. Různá priorita jednotlivých úloh! – systémové úlohy mají vždy vyšší prioritu než uživatelské.
- *Plánovač procesů* – sleduje frontu procesů a rozhoduje, který proces a na jak dlouho dostane přidělen procesor.
- *Dispečer (traffic controler)* – sleduje procesor a stav procesů.

Modul přidělování periferií

- *I/O dispečer* – sleduje stav periferních zařízení, kanálů, řídicí jednotky.
- *I/O plánovač* – rozhoduje o efektivním přidělení periferních zařízení. Pokud má být sdíleno, rozhoduje o tom, kdo ho dostane, a v jakém rozsahu. Přirázuje periferii a zahajuje I/O operaci. Požaduje navrácení prostředků, většinou se u I/O ukončuje automaticky.

Systém správy souborů

- Sleduje každý soubor – jeho umístěni, užití, stav apod.
- Rozhoduje, komu bude soubor poskytnut – realizuje požadavky na ochranu dat a operace přístupu k nim.
- Přiděluje prostředek – otevírá (zpřístupňuje) soubor.
- Odebírá prostředek – uzavírá soubor.

3.3 Průběh vykonávání procesu

Jak proces prochází jednotlivými stavy, ukazuje obr. 3b. Rámečky označují jednotlivé stavy procesu a obláčky moduly OS, které tyto změny stavu zajištují.

Uživatel uvede úlohu do stavu **předána**, pokud ji *I/O dispečer* za pomoci modulu *spooling* je schopen vyhradit dostatečné místo na pevném disku.

Tím úloha přechází do stavu **přijata** a ujímá se jí *plánovač úloh*, který provádí:

- Dotaz na požadovanou kapacitu operační paměti u *modulu přidělováni paměti.*
- Dotaz na požadované periferie u *I/O dispečera.*

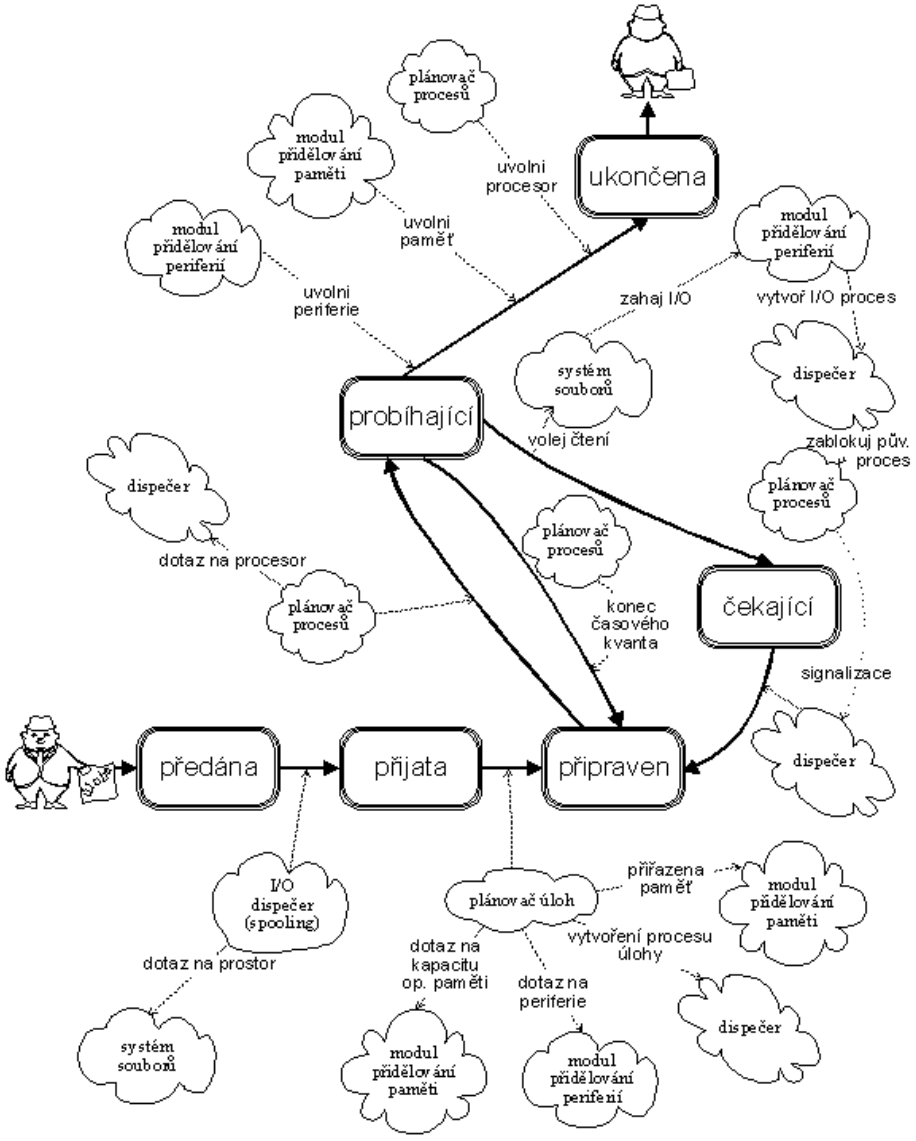
- Jestliže se obého dostává, je volán *dispečer*, aby vytvořil příslušné záznamy a *modul přidělování paměti*, aby procesu přidělil dostatečnou operační paměť.

Potom je úloha zavedena do paměti a jí odpovídající proces je ve stavu **připraven**. Ujímá se ho *plánovač procesu*, a jakmile je možné úloze přidělit procesor, plánovač to udělá.

Procesor je úloze přidělen pouze na určitou dobu. Pokud během ní úloha není dokončena, *dispečer* uloží stav registrů a procesu, *plánovač procesu* jí procesor odebere a vrátí do stavu **připraven**.

Je-li úloha ve stavu **probíhající** a žádá o čtení ze souboru (nebo jinou I/O operaci), *modul správy souborů* volá *modul přidělování periferií*, aby zahájil čtení (nebo jinou I/O operaci). *Modul přidělování periferií* ji zahájí a zároveň požádá *plánovač procesů*, aby proces převedl do stavu **čekající**. Je-li I/O operace dokončena, je patřičný signál přerušení vyhodnocen jako žádost o navrácení úlohy do stavu **připraven**.

Pokud je úloha ve stavu **probíhající** dokončena, *modul přidělování periferií* jí odebere přidělené periferie, *modul přidělování paměti* uvolní paměť, která byla úloze alokována a *plánovač procesu* ji odebere procesor. Tím je výpočet úlohy **ukončen**.

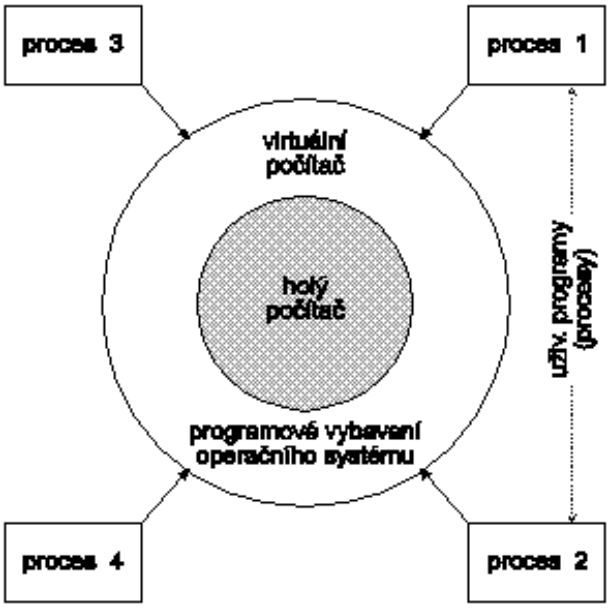


Obrázek 4 - Přechod mezi stavy procesu v OS

3.4 Hierarchická struktura OS, virtuální počítač

virtuální počítač = počítač + operační systém

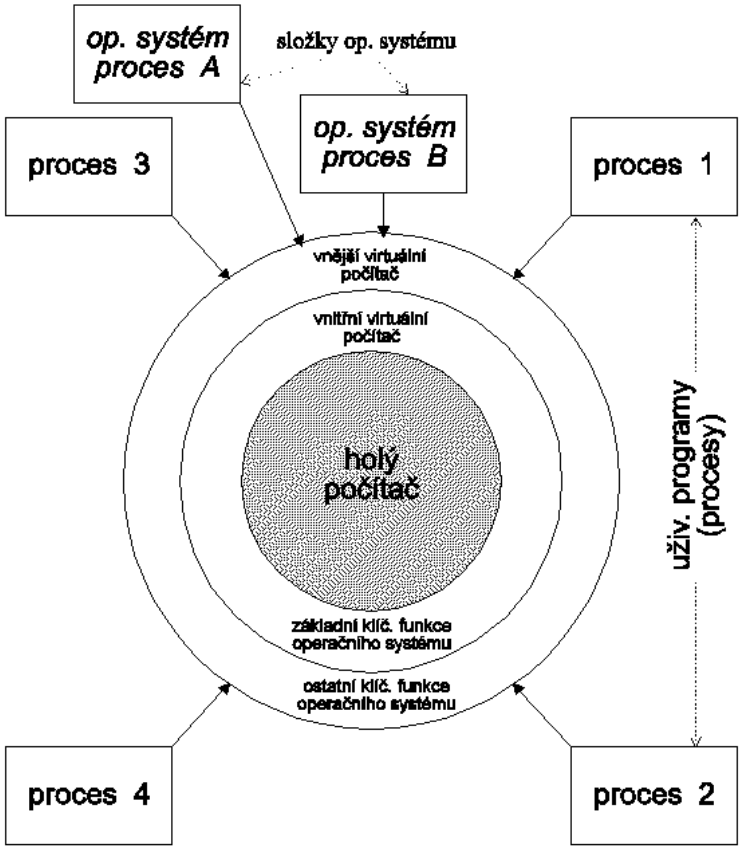
OS umožňuje ovládání počítače na přijatelné úrovni. Ke strojovým instrukcím počítače přidá množinu svých příkazů, které strojové instrukce používají. Vzniká **množina instrukcí virtuálního počítače.** Uživatelské programy zpracovává virtuální počítač.



Obrázek 5 - Virtuální počítač

U velkých systémů není možné řešit OS jako jediný program. Struktura dnešních operačních systémů je hierarchická, modulární:

- Klíčové funkce používané mnoha systémovými moduly jsou včleněny do "vnitřního virtuálního počítače".
- Jiné funkce jsou ve vlastním virtuálním počítači a jsou prováděny v podstatě stejně jako uživatelské procesy.



Obrázek 6 - Jednoduchá struktura virtuálního počítače

Kam tedy s jednotlivými moduly OS?

- Procesy OS mohou být spolu různě provázané (např. mohou komunikovat); lze je uspořádat do tzv. **vrstev procesu**. (tady vzniká modulární programování).
- V hierarchickém systému je dovoleno volat pouze moduly na stejné nebo nižší úrovni.
- Které funkce na kterou úroveň?

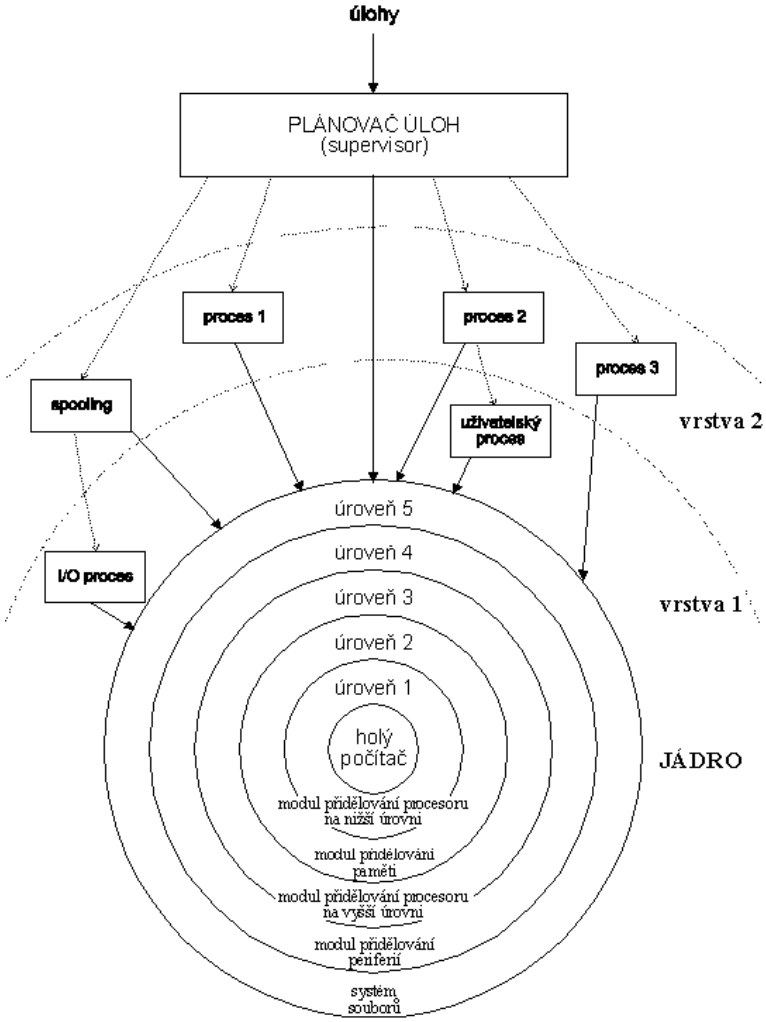
Na nejnižší úrovni musí byt funkce, které jsou volány všemi moduly, funkce, jejichž úlohou je přidělování prostředků:

- **P operace** – zaznamenávají přidělení nebo žádost o přidělení prostředku, synchronizuji žádosti (operace čekej na semaforu);
- **V operace** – zaznamenávají, že byl prostředek uvolněn (operace signál na semaforu);

Tyto operace se provádějí nad datovou strukturou zvanou semafor.

Žádost o prostředek: P operace nad semaforem onoho prostředku. Je-li *semafor otevřen*, P operace ho převede do stavu uzavřen a její provádění končí. Je-li původně *semafor uzavřen*, převede P operace žádající proces do stavu **čekající** a její provádění se ukončí.

Uvolnění prostředku: V operace nad semaforem. Převede jeden z čekajících procesů do stavu **připraven**; pokud žádný takový proces není, převede semafor do stavu *otevřen*.



Obrázek 7 - Hierarchická struktura operačního systému

Příklady primitivních funkcí na různých úrovních jádra:

úroveň 1: modul přidělování procesoru na nižší úrovni

- provádění P operací,
- provádění V operací,
- plánování procesů – multiprogramování,

úroveň 2: modul přidělování paměti

- přiděl paměť,
- uvolni paměť.

úroveň 3: modul přidělování procesoru na vyšší úrovni

- vytvoř / zruš proces,
- zašli / přijmi zprávu mezi procesy,
- zastav proces,
- zahaj proces.

úroveň 4: modul přidělování periferních zařízení:

- sleduj stav všech I/O zařízení,

- naplánuj I/O operace,
- zahaj I/O proces.

úroveň 5: modul ovládání systému souborů

- vytvoř / zruš soubor,
- otevři / uzavři soubor,
- operace čtení a zápisu do souboru.

Příklad:

Použijme analogii mezi operačním systémem (resp. aplikačním programátorem) a tesařem, který staví dům. Jeho základní používané součástky jsou hřebíky, sklo, cement, tmel a dřevo.

1. S jakými problémy se potýká, pokud nemá žádné prefabrikáty (polotovary)?
2. Nechť má tesař k dispozici tyto prvky vyšší úrovně, např. okna, dveře apod.

V prvním případě bude mít tesař každopádně více práce:

- Musí polotovary sám vyrábět - protože se bez nich stejně neobejde a jeho výtvory budou jistě horší než výrobky specialistů.
- Jeho pracoviště nebude příliš přehledné, protože bude potřeba spousty základních surovin.
- Tesař sám se více nadře, protože musí všechno dělat sám.

Ve druhém případě bude všechno snazší:

- Tesař využije kvalifikované práce specialistů – zkvalitnění výstupu a urychlení práce.
- Staveniště bude přehledné, protože nebude potřeba tolik (dílčích) surovin.