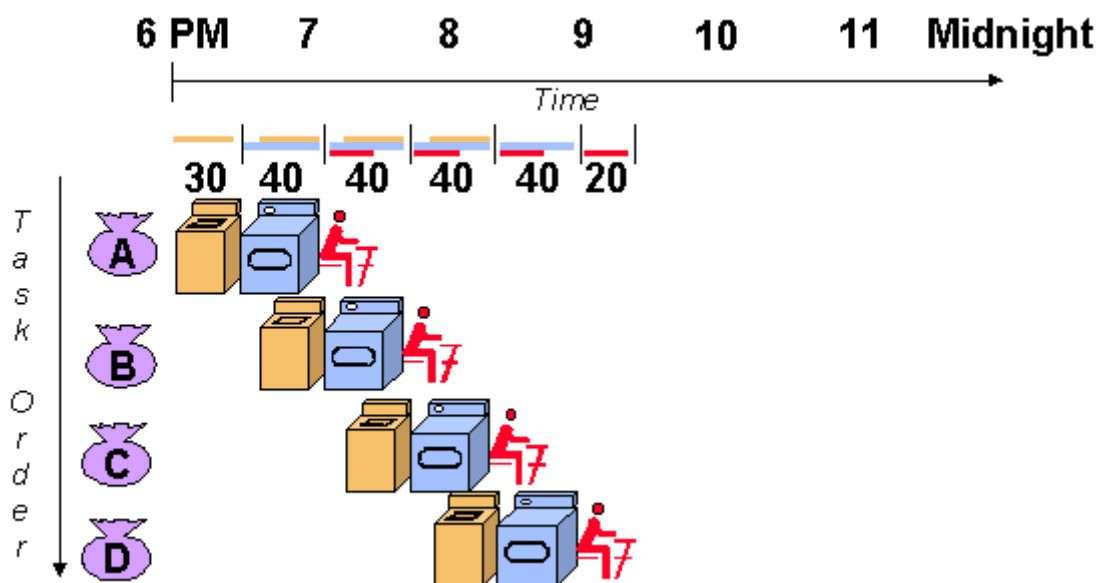


# M16 Víceprocesorové a víceúlohové

#technicke\_vybaveni\_pocitacu

## Paralelizace

- rozdělení úlohy na menší části, které jsou následně prováděny současně
- zvyšuje efektivitu a rychlost zpracování
- implementace na úrovni *hardwaru* - vícejádrové procesory softwaru - algoritmy



## Důvody

- zkrácení doby zpracování - podprogramy jsou spouštěny současně, dosahující rychlejšího výsledku než sériové zprac. (viz [M12 Pipeline CPU](#))
- zjednodušení algoritmu - použito ke zjednodušení problému na menší části řešené samostatně
- lepší využití zdrojů - rozdělením práce se zabráňuje situace, kdy jeden procesor pracuje naplno a ostatní čekají

## Výhody a nedostatky

- výhody
  - rychlejší zpracování
  - zvládnutí více práce současně
  - snadnější rozšíření - přidáním stroji další procesor program poběží ještě rychleji
  - čistší kód (*v některých případech*) - snadnější řešení složitých úloh
- nevýhody
  - složitější kód a odhalování chyb v něm (+ větší riziko výskytu)

- komunikace - podprogramy potřebují mezi sebou komunikovat; komunikace je zdrojem režie, předávání mezivýsledků aj.
- nemožnost paralelizace - pokud vlákno musí čekat na výsledek druhé, pak nemusí být paralelizace nejoptimálnější řešení

## Multiprocessing

- počítačový koncept se schopností nezávisle provádět procesy současně za pomoci více procesorů

### symetrický

- všechny procesory jsou si rovnocenné
- mají přístup ke stejným zdrojům (paměť, IO zařízení)
- jsou stejného typu
- operační systém plánuje úlohy tak, aby byly úlohy rovnoměrně rozděleny mezi všechny dostupné procesory
- dají se jednoduše implementovat a rozšiřovat; rozšiřitelnost je omezená
- určena pro širokou škálu využití

### nesymetrický

- procesory mohou být rozdílné (různý typ, odlišný výkon, jiný přístup k různým zdrojům)
- procesory jsou specializované pro různé úlohy
- paměti či periférie nemusí být sdíleny stejně mezi procesory
- operační systém nemusí mít plnou kontrolu pro všechny procesory
- master CPU - řídí celý systém; slave CPU - vykonává specifické úlohy
- grafické karty - mají CPU navíc pro ostatní úlohy
- menší využitelnost tohoto konceptu - úpadek

## HW podpora pro systémy se souběžným zpracováním více úloh

- multijádrové procesory
  - každé jádro zpracovává instrukce a provádí úkoly nezávisle na ostatních jádrech
  - počet jader od 2 do 64
- hyperthreading
  - schopnost jednoho fyzického jádra simulovat více (logických) jader (asi 2 jádra?)
  - schopnost dosažena sdílením instrukční fronty a registrů
  - menší nárůst výkonu oproti multicore CPU
- víceprocesorové systémy
  - pro náročné úkoly vyžadující maximální výpoč. výkon

- každý procesor má svou paměť nebo část společné
- třeba optimalizovat pro každý procesor takt
- paměť
  - kapacita RAM by měla odpovídat požadavkům náročnosti na výkon
  - vícekanálová paměť umožňuje paralelní přístup → větší propustnost
  - využití cache paměti
- IO systémy - rychlejší čtení a zápis výhodou (načítání z/do paměti)
- akcelerátory - specifické procesory pro zrychlení určitých úloh, jako je zpracování signálu nebo strojového učení

## Multitasking

- schopnost vykonávat několik úloh zdánlivě současně
- užitečný pro vyplnění krátkých prodlev (například čekání na stažení souboru)
- při nenáročných kombinacích může vést ke zvýšení produktivity

## preemptivní

- operační systém má plnou kontrolu na přidělování procesoru jednotlivým úlohám - neposlouchá požadavky aplikace
- operační systém uděluje procesorový čas různým programům v krátkých úsecích (v řádu milisekund); po uplynutí časového úseku systém přepne pozornost na jinou úlohu ve frontě, i přes nedokončení předešlé úlohy
- před přepnutím úlohy je stav předešlé úlohy uložen do paměti
- spravedlivé rozdělení procesoru - jeden program "nezablokuje" celý systém
- oddělení úloh zajišťuje, že chyby v jedné úloze nemají vliv na ostatní úlohy

## nepreemptivní

- nechává operační systém samotné programy rozhodovat, kdy uvolní procesor pro další úlohy
- postup multitaskingu
  1. OS přidělí procesor spuštěnému programu
  2. program využívá procesor, dokud ho sám dobrovolně neuvolní pomocí systémového volání
  3. OS následně přepne procesor na jiný program z fronty úloh
- z hlediska návrhu a implementace je jednodušší než preemptivní
- vhodný pro aplikace s pevně daným chováním v reálném čase
- využito ve starších verzích OS (např. Win3.x)