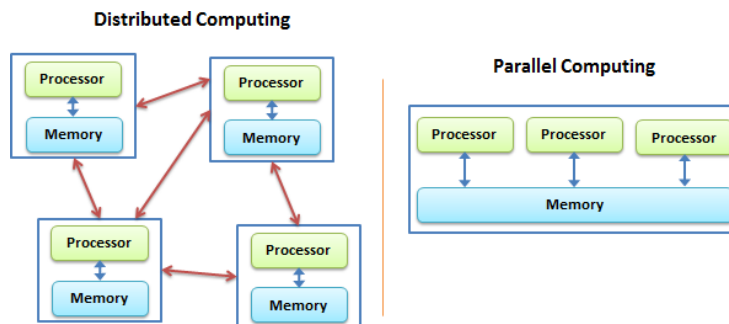


Víceprocesorové a víceúlohové systémy - MO. 16

Paralelizace

- **Definice**
 - Paralelizace je technika rozdělení úlohy na menší části, které se pak zpracovávají souběžně
 - Může to vést k výraznému zkrácení doby potřebné k dokončení úlohy, zvláště u úloh, které jsou náročné na výpočty
- **Důvody**
 - *Zvýšení výkonu:* Paralelizace umožňuje využít více výpočetních zdrojů (např. procesorů, jader) najednou, čímž se zkrátí doba potřebná k dokončení úlohy
 - *Zlepšení škálovatelnosti:* Paralelizované úlohy se obvykle lépe škálují na více procesorů, tzn. že jejich výkon se s rostoucím počtem procesorů zvyšuje
 - *Zjednodušení algoritmu:* V některých případech může paralelizace vést k jednoduššímu algoritmu pro řešení dané úlohy
- **Výhody**
 - Zvýšení výkonu
 - Zlepšení škálovatelnosti
 - Zjednodušení algoritmu
 - Lepší využití dostupných hardwarových zdrojů
 - Možnost řešit složitější úlohy v reálném čase
- **Nevýhody**
 - Nutnost koordinace mezi paralelně běžícími částmi úlohy
 - Případné problémy s pamětí a synchronizací
 - Nárůst spotřeby energie
 - Ne všechny úlohy jsou vhodné pro paralelizaci



Symetrický a nesymetrický multiprocessing

- **Symetrický multiprocessing (SMP)**
 - Systém, ve kterém všechny procesory sdílí společnou paměť a periferní zařízení
 - Všechny procesory jsou si rovny a mohou provádět libovolnou úlohu
- **Nesymetrický multiprocessing (NUMA)**
 - Systém, ve kterém každý procesor má vlastní paměť a periferní zařízení
 - Procesory mohou přistupovat k paměti jiných procesorů, ale s určitou penalizací za přístup

Srovnání výkonu s klasickým řešením

- **SMP** systémy obvykle poskytují lepší výkon než klasické systémy s jedním procesorem, zvláště u úloh, které jsou náročné na výpočty
- **NUMA** systémy se hodí pro úlohy, které vyžadují velkou paměť a / nebo hodně komunikace mezi procesory

HW podpora pro systémy se souběžným zpracováním více úloh

- **Více jader**
 - Moderní procesory obvykle obsahují více jader, která mohou zpracovávat úlohy souběžně
- **Více procesorů**
 - V systémech s více procesory může každý procesor zpracovávat úlohy souběžně
- **Hyper-threading**
 - Tato technologie umožňuje jednomu jádru procesoru zpracovávat více vláken souběžně
- **Virtualizace**
 - Virtualizační software umožňuje na jednom fyzickém počítači spouštět více operačních systémů, které pak mohou zpracovávat úlohy souběžně

Preemptivní a nepreemptivní multitasking

- **Preemptivní multitasking**
 - Umožňuje operačnímu systému pozastavit běh jednoho procesu a spustit jiný proces, což dovolí operačnímu systému spravovat dostupné výpočetní zdroje a zajistit, aby všechny procesy měly možnost běžet
- **Nepreemptivní multitasking**
 - Umožňuje procesu běžet tak dlouho, dokud se sám nevzdá procesoru
 - Může to vést k tomu, že jeden proces zabere všechny dostupné výpočetní zdroje a zabrání tak běhu jiných procesů
- **Příklady**
 - *Preemptivní multitasking*: Většina moderních operačních systémů, jako je Windows a Linux, používá preemptivní multitasking
 - *Nepreemptivní multitasking*: Některé starší operační systémy, jako je MS-DOS, používaly nepreemptivní multitasking

Paralelní stroje

- **Flynnův taxonomie**
 - Klasifikuje paralelní stroje podle úrovně paralelismu instrukcí a dat
- **Multiprocesorové systémy (MPS)**
 - Skládají se z více procesorů sdílejících paměť a další periferní zařízení
- **Distribuované systémy**
 - Skládají se z více počítačů propojených sítí, které spolupracují na řešení úloh
- **Masivně paralelní procesory (MPP)**
 - Skládají se z velkého počtu procesorů, které pracují na stejném úkolu současně

Zrychlení není podle počtu jader -> režie, potřeba synchronizovat

- Zatímco více jader v MCU může vést ke zvýšení výkonu, nárůst výkonu nemusí být přímo úměrný počtu jader
- **Režie**
 - Více jader znamená více hardwaru a softwaru, které je nutné spravovat, což může vést ke ztrátě výkonu
- **Potřeba synchronizace**
 - Pokud jádra pracují na stejných datech nebo sdílených zdrojích, je nutné je synchronizovat, aby se zabránilo konfliktům
 - Synchronizace může vést k prodlevám a snižovat výkon
- **Synchronizační problémy**
 - Některé synchronizační problémy, jako je problém s obědvajícími filozofy, se obtížně řeší a mohou vést k významným ztrátám výkonu

Synchronizační problémy => obědvající filozofové, hladový proces

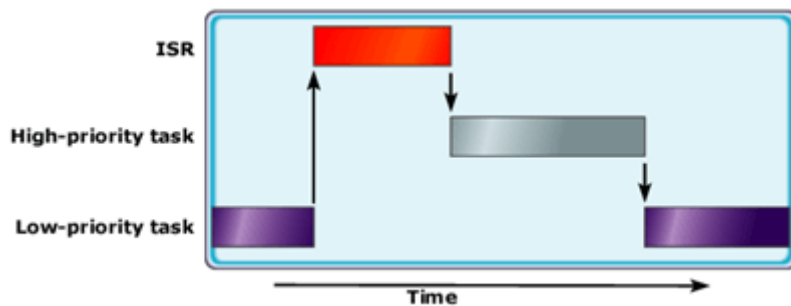
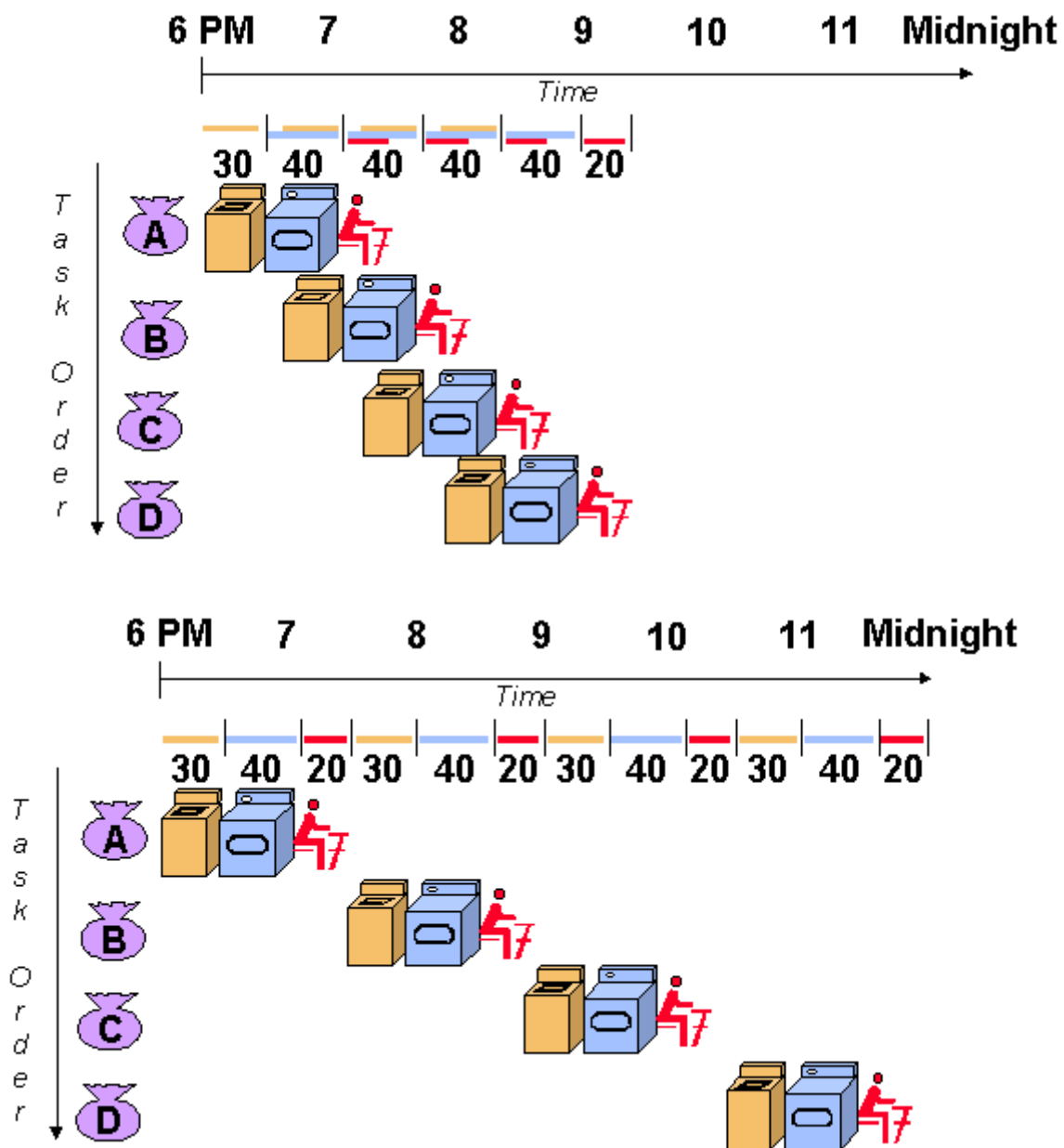
- Synchronizační problémy jsou důležitou součástí paralelního programování a jejich pochopení je nezbytné pro navrhování a implementaci efektivních a spravedlivých paralelních systémů
- Příklady, jako je problém s obědvajícími filozofy a hladový proces, nám pomáhají porozumět komplexnosti těchto problémů a zvážit filozofické aspekty, které s nimi souvisejí

Obědvající filozofové

- Synchronizační problémy otevírají i filozofické otázky týkající se spravedlnosti, priority a etiky v paralelních systémech
- Například v problému s obědvajícími filozofy můžeme diskutovat o tom, zda je spravedlivé, aby jeden filozof čekal déle než ostatní, a zda je vhodné, abychom dali prioritu hladovému procesu v problému s hladovým procesem
- Tyto otázky nemají snadné odpovědi a záleží na konkrétním kontextu a požadavcích daného systému

Více jader v jednom MCU

- Mikrořadiče (MCU) s více jádry se stávají stále běžnějšími, jelikož nabízí řadu výhod oproti jednojádrovým MCU
- **Zvýšený výkon**
 - Více jader umožňuje paralelní zpracování úloh, čímž se zkracuje doba potřebná pro dokončení úkolů a zvyšuje se celkový výkon
- **Zvýšená propustnost**
 - Více jader umožňuje zpracovávat více dat současně, čímž se zvyšuje propustnost systému
- **Lepší energetická účinnost**
 - Více jader umožňuje efektivnější využití energie, jelikož jádra se mohou zapínat a vypínat v závislosti na aktuální zátěži
- **Možnost implementace složitějších algoritmů**
 - Více jader umožňuje implementovat složitější algoritmy, které by na jednojádrovém MCU nebyly proveditelné v reálném čase



ISR = Kód, který se spustí v reakci na přerušení, s ohledem na prioritu

High-priority task = Úloha s vyšší prioritou

Low-priority task = Úloha s nižší prioritou