

OS – Deadlock a prevence

Tomáš Hudec

`Tomas.Hudec@upce.cz`

`http://asuei01.upceucebny.cz/usr/hudec/vyuka/os/`

Prostředky poskytované OS

- OS poskytuje procesům systémové **prostředky** (zdroje, resources)
 - HW zařízení, soubory, semaforey, (sdílená) paměť
- zajímají nás prostředky s exkluzivním přístupem
- prostředky můžeme rozdělit na
 - **odejmutelné** (preemptable)
 - např. CPU, operační paměť (lze odložit na swap)
 - **neodejmutelné** (non-preemptable)
 - např. DVD-RW (při vypalování), tiskárna

Řízení přístupu k prostředkům

- k některým prostředkům si mohou přístup řídit samy procesy
 - např. pomocí semaforů
- příklad – prostředky A a B, procesy P_1 a P_2
 - oba prostředky jsou vyžadované současně
 - prostředky alokujeme postupně
 - alokujeme semafor pro každý prostředek
- **záleží na pořadí alokace!**

Řízení přístupu – příklad

- uvažujme postupnou alokaci pro každý proces
 - fungující příklad
- pokud ovšem alokace proběhne u některého procesu v opačném pořadí
 - může nastat **DEADLOCK**

proces P_1

```
mutex_lock (&mA) ; // získá A
mutex_lock (&mB) ; // čeká na B
použij_prostředky (A, B) ;
mutex_unlock (&mB) ;
mutex_unlock (&mA) ;
```

proces P_2

```
mutex_lock (&mB) ; // získá B
mutex_lock (&mA) ; // čeká na A
použij_prostředky (A, B) ;
mutex_unlock (&mA) ;
mutex_unlock (&mB) ;
```

Deadlock (stav uváznutí)

- vzájemné zablokování, smrtící obětí, uváznutí
- definice **stavu uváznutí** (zablokování)
 - skupina procesů je ve **stavu uváznutí**, když **každý proces** ve skupině **čeká na událost**, kterou může vyvolat **pouze jiný proces ze skupiny**
 - protože událost nelze vyvolat jinak než procesem ve skupině, budou tyto procesy nekonečně čekat
 - předpokládáme, že procesy nelze probudit asynchronně
 - např. signálem, při jehož obsluze by se mohla událost vyvolat

Livelock

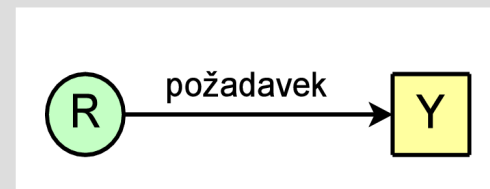
- procesy běží, ale nečiní žádný pokrok
- procesy mohou reagovat na stav jiného procesu, ale tyto změny nevedou k pokroku
- na rozdíl od stavu deadlock **procesy nejsou ve stavu čekání**
- př.: dva lidé potkavše se v úzké uličce
 - oba neustále současně uhýbajíce na stejnou stranu
- definice stavu livelock se v pramenech liší

Starvation (vyhladovění)

- nekončící čekání procesu na přidělení prostředku, proces nečiní žádný pokrok
 - proces v soupeření o prostředek neustále prohrává
 - např. nízká priorita při soutěži o procesorový čas
- nejedná se o deadlock
 - proces by učinil pokrok, kdyby jej ostatní procesy neustále „nepředbíhaly“
 - v jistém smyslu lze však uvažovat, že důsledkem stavu uváznutí (deadlock / livelock) je vyhladovění

Grafické modelování požadavků

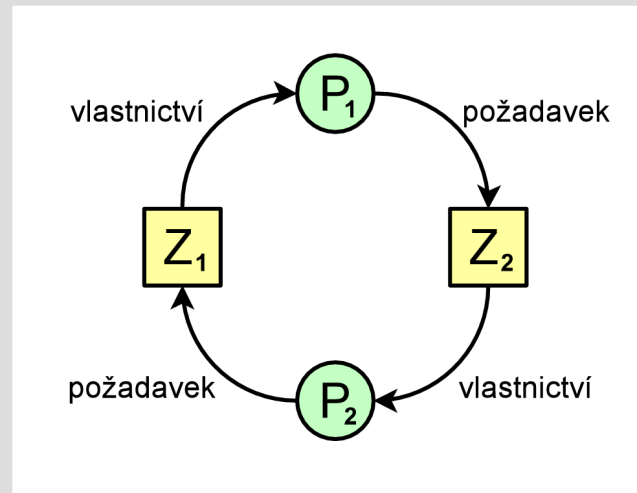
- prostředky reprezentujeme pravoúhelníky
- procesy reprezentujeme kruhy
- alokaci (držení) prostředku Z procesem P znázorňujeme šipkou od Z k P
- blokující požadavek procesu R na přidělení prostředku Y znázorňujeme šipkou od R k Y



Coffmanovy podmínky vzniku stavu uvážnutí (publ. 1971)

- **vzájemné vylučování** (mutual exclusion)
 - prostředky lze vlastnit pouze jediným procesem
- **alokace a čekání** (hold and wait)
 - proces vlastníci prostředek může požadovat další
- **neodnímatelné prostředky** (non-preemptable)
 - OS je nemůže odejmout, musí být explicitně uvolněny vlastníčím procesem
- **cyklické čekání** (circular wait)
 - řetěz vzájemně čekajících procesů uzavírá cyklus

Cyklické čekání (obrázek)



- proces P_1 vlastní prostředek Z_1
a je blokován požadavkem na prostředek Z_2
- proces P_2 vlastní prostředek Z_2
a je blokován požadavkem na prostředek Z_1

Řešení a prevence stavu uváznutí

- **ignorování** problému (přetrosí přístup)
- **detekce** a obnovení
 - obnova (rollback) stavu bez uváznutí (checkpoint)
 - násilné odebrání prostředku
 - zabití některého procesu (př.: Linux OOM-killer)
- vyloučení možnosti vzniku (**prevence**)
 - negování alespoň jedné z předešlých podmínek
 - splnění **všech** předešlých podmínek je **nutné**, aby vznikl deadlock – **stačí vyloučit jedinou**

Prevence vzniku uváznutí (1)

- negace vzájemného vylučování
 - spooling (např. tiskárna)
- negace alokace a čekání (hold and wait)
 - zajistit alokaci všech potřebných prostředků naráz
 - proces je blokován, dokud vše není dostupné
 - proces pak může čekat velmi dlouho
 - některé prostředky zůstávají dlouho nevyužity
- negace neodnímatelných prostředků
 - zavést možnost násilného odebrání prostředku
 - nelze vždy – např. vypalování médií (DVD)

Prevence vzniku uváznutí (2)

- negace cyklického čekání (circular wait)
 - nedovolit alokaci prostředku, pokud by vznikl cyklus
 - definování lineárního uspořádání na prostředcích
 - přidělují se vždy pouze prostředky s vyšším pořadovým číslem než má jakýkoli procesu již přidělený prostředek
 - může nastat zbytečné odmítnutí požadavku na přidělení

Prevence vzniku uváznutí (3)

- dovolení pouze bezpečných stavů
 - bezpečný stav
 - není stav uváznutí a **existuje plánovací pořadí**, při kterém všechny procesy mohou být dokončeny, i když všechny procesy naráz budou požadovat **maximum svých** potřebných zdrojů (prostředků)
 - nebezpečný stav
 - stav, kdy může (ale nemusí vždy) nastat uváznutí
 - pokud by každý proces požadoval maximum svých deklarovaných zdrojů a ani jednomu nebude moci systém vyhovět, jedná se o nebezpečný stav

Bankéřův algoritmus

- algoritmus publikoval Dijkstra 1965
- řešení situace bankéře při jednání s klienty, kterým poskytuje půjčku
 - pokud požadavek klienta vede k nebezpečnému stavu, je tento požadavek odmítnut
- výchozí předpoklady
 - pevný počet prostředků
 - každý proces deklaruje své maximální požadavky
 - postupná alokace prostředků

Bankéřův algoritmus

- prostředek je na požadavek přidělen jen tehdy, vede-li situace do bezpečného stavu
- slabiny algoritmu
 - nelze vždy garantovat pevný počet prostředků
 - proces nemusí znát maximální požadavky předem
- důsledek: algoritmus je mnohdy prakticky nepoužitelný

Prevence uváznutí – shrnutí

podmínka

vzájemné vylučování

alokace a čekání

neodnímatelné prostředky

cyklické čekání

možné řešení

spooling prostředků

alokovat vše naráz

dovolit násilné odebrání

uspořádání prostředků,
bankéřův algoritmus