Operační systémy

Přednáška 6: Uváznutí procesů/vláken

Výpočetní prostředky

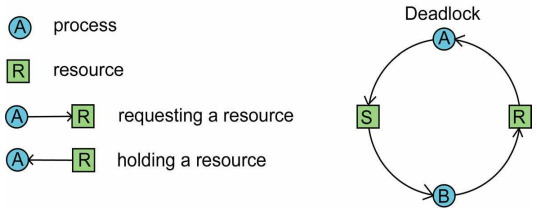
- **Výpočetní prostředek**
  - **Hardwarové zařízení** (např. pásková mechanika, tiskárna, CD vypalovačka,…).
  - **Informace** (např. položka v databázi, řádek v systémové interní tabulce, soubor,…).
- **Výlučný přístup** k prostředkům
  - Většina výpočetních prostředků může být **v jednom okamžiku používána pouze jedním procesem** (např. tiskárna, páska,…).
- Většina procesů potřebuje **výlučný přístup k více prostředkům.**
- **Dva typy výpočetních prostředků**
  - **Odnímatelné (Preemptable)**: mohou být odebrány procesu bez rizika dalších problémů (např.odložení procesu z fyzické paměti na disk).
  - **Neodnímatelné (Nonpreemptable)**: nemohou být odebrány bez rizika (např. CD-ROM vypalovačka).

Výpočetní prostředky (2)

- **Sekvence kroků** při použití prostředku:
  1. **Žádost** o prostředek.
  2. **Použití** prostředku.
  3. **Uvolnění** prostředku.
- **Žádný prostředek** není dostupný ⇒ žádající proces bude **zablokován**.
  - Automaticky zablokován prostřednictvím OS a probuzen až daný prostředek bude k dispozici.
  - Žádost skončí s chybou a žádající proces po nějaké době žádost opakuje dokud prostředek nezíská.
- **Definice uváznutí (deadlock):** Množina procesů uvázne pokud každý proces v množině čeká na nějakou událost, kterou může vyvolat pouze některý proces z množiny.

Modelování uváznutí

- Uváznutí můžeme modelovat pomocí **alokačního grafu**.



- **Každá smyčka v grafu představuje uváznutí** (procesy ve smyčce čekají a nemohou pokračovat).

Příklad: uváznutí

proces A	proces B	proces C
Žádost o R	Žádost o S	Žádost o T
Žádost o S	Žádost o T	Žádost o R
uvolnění R	uvolnění S	uvolnění T
uvolnění S	uvolnění T	uvolnění R

Alokace s uváznutím:
A: Žádost o R
B: Žádost o S
C: Žádost o T
A: Žádost o S ⇒ proces je uspán
B: Žádost o T ⇒ proces je uspán
C: Žádost o R ⇒ **uváznutí !!!**

Alokace bez uváznutí:
A: Žádost o R
C: Žádost o T
A: Žádost o S
C: Žádost o R ⇒ proces je uspán
A: uvolnění S . . . **bez uváznutí!!!**

Coffmanovy podmínky

- Nutné podmínky uváznutí
  1. **Vzájemné vyloučení.** Každý prostředek je buď přidělen právě jednomu procesu a nebo je volný (prostředek nemůže být sdílen více procesy).
  2. **Podmínka „drž a čekej".** Proces, který má již přiděleny nějaké prostředky, může žádat o další prostředky (proces může žádat o prostředky postupně).
  3. **Podmínka neodnímatelnosti.** Prostředek, který byl již přidělen nějakému procesu, nemůže mu být násilím odebrán. Musí být dobrovolně uvolněn daným procesem.
  4. **Podmínka cyklického čekání.** Musí existovat smyčka dvou nebo více procesů, ve které každý proces čeká na prostředek přidělený následujícímu procesu ve smyčce.
- **Pokud aspoň jedna z podmínek není splněna, nemůže dojít k uváznutí.**

Strategie řešení uváznutí

1. **Pštrosí algoritmus.** Úplné ignorování celého problému.
2. **Detekce a zotavení.** K uváznutí může dojít, ale pak je detekováno a odstraněno.
3. **Dynamické zamezení vzniku uváznutí** pomocí pečlivé alokace prostředků.
4. **Prevence** pomocí nesplnění aspoň jedné z Coffmanových podmínek.

Pštrosí algoritmus

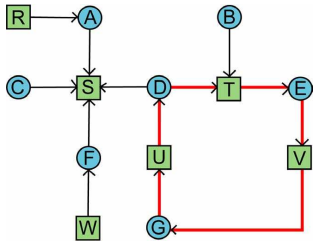
- Praktické řešení ve většině univerzálních OS (UNIX, MS windows,…).
- **Uváznutí** se vyskytuje relativně **zřídka**, např. jednou za rok.
- **Hardwarové chyby** a **chyby v OS** se vyskytují **mnohem častěji**, např. každý měsíc (záplaty,…).
- **Zabránění uváznutí je drahé.**
- Např. v Unixu, nejčastěji používané prostředky v jádru jsou položky v tabulce procesů a položky v tabulce i-nodů (mající samozřejmě omezenou kapacitu).
- **Řešení:**
  - Proces se pokusí alokovat prostředek. Pokud není k dispozici, tak skončí s chybou.
  - Administrátor zjistí uvázlé procesy a ukončí je.
- Toto řešení **není přijatelné ve„fault tolerant" systémech**.

Detekce a zotavení

- Systém se **nesnaží předcházet výskytu uváznutí**, tzn. že procesy mohou uváznout.
- V případě výskytu uváznutí se tuto situaci snaží **detekovat** a na základě toho se pokusí **uváznutí odstranit**.

Detekce uváznutí

- Předpokládejme, že existuje pouze **jeden prostředek od každého typu**.
- Sestrojíme **alokační graf**.
  - Jakýkoliv proces, který je součástí smyčky, je uváznutý.
  - Pokud v grafu neexistuje žádná smyčka, tak v systému nedošlo k uváznutí.



Detekce uváznutí (2)

- Předpokládejme **více prostředků od každého typu** (*m*).
- Algoritmus pro detekci mezi *n* procesy:

Exitující prostředky

$$E = (E_1, E_2, E_3, \ldots, E_m)$$

Volné prostředky

$$A = (A_1, A_2, A_3, \ldots, A_m)$$

Již alokované prostředky

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \cdots & C_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1} & C_{n2} & C_{n3} \cdots & C_{nm} \end{bmatrix}$$

Ještě požadované prostředky

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \cdots & R_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ R_{n1} & R_{n2} & R_{n3} \cdots & R_{nm} \end{bmatrix}$$

Každý prostředek je již alokován nebo je volný:

$$\sum_{i=1}^{n} C_{ij} + A_j = E_j$$

Detekce uváznutí (3)

- Každý proces je **na začátku neoznačen**.
- **Algoritmus pro detekci uváznutí**:
  1. Najdi neoznačený proces P_i, jehož *i*-tý řádek v R má menší nebo stejné hodnoty jako A.
  2. Pokud takový proces existuje, přičti hodnoty *i*-tého řádku z C k hodnotám v A, označ tento proces a pokračuj bodem 1.
  3. Pokud žádný takový proces neexistuje, algoritmus končí.
- Po skončení algoritmu, **všechny neoznačené procesy jsou uvázlé.**

Příklad: detekce uváznutí

- Máme **3 procesy** a **4 typy prostředků** (např. pásky, plotry, skenery, CR-ROM).
- Aktuální rozdělení je E = (4, 2, 3, 1), A = (2, 1, 0, 0)

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

- Třetí proces může být upokojen a potom uvolní své alokované prostředky ⇒ A=(2, 2, 2, 0).
- Dále může být uspokojen druhý proces, který také vrátí své prostředky ⇒ A=(4, 2, 2, 1).
- Nyní poslední proces může pokračovat ⇒ **Bez uváznutí!**

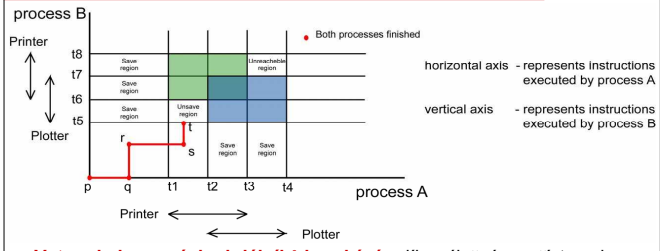
Zotavení z uváznutí

- **Zotavení pomocí odebrání**
  - Dočesné násilné odebrání prostředku a půjčení jinému procesu. Ve většině případů to vyžaduje manuální intervenci, (např. odebrání pásky nebo tiskárny).
- **Zotavení promocí návratu**
  - **Periodicky si ukládáme důležité informace o procesech**, tak abychom později byli schopni je vrátit do předchozích stavů.
  - Při detekci uváznutí, **proces, který vlastní kritický prostředek, je vrácen zpět** v čase do stavu, kdy daný prostředek ještě nevlastnil. Uvolněný kritický prostředek je přidělen jednomu z uváznutých procesů.
- **Zotavení pomocí ukončení procesů**
  - Ukončíme proces, který je součástí **smyčky** v alokačním grafu.
  - Problém s některými typy procesů (překlad x modifikace databáze).

Zamezení vzniku uváznutí

- Většinou procesy alokují prostředky postupně jeden po druhém.
- V okamžiku žádosti o další prostředek, systém musí rozhodnout, zda přidělení prostředku je **bezpečné** (tzn. že nedojde k uváznutí) či **nebezpečné**.
- Prostředek bude **přidělen pouze pokud to bude bezpečné**.
- Systém se snaží předejít uváznutí **opatrnou alokací prostředků**.
- **Postupový prostor** (postupová cesta) lze použít pro modelování vzniku uváznutí.

Postupový prostor



- **Vstup do barevných obdélníků je zakázán** díky výlučnému přístupu k prostředku (uvnitř barevné oblasti by prostředek byl alokován současně dvěma procesy).
- **Procesy nesmí vstoupit do nebezpečné oblasti (Unsave region).** Vstup do této oblasti vždy skončí uváznutím.
- V bodě *t*, systém musí rozhodnout, zda přidělí ploter procesu B.
- Abychom se vyhnuli uváznutí, B musí být pozastaven.
- Proces B bude pokračovat, až proces A použije a opět uvolní ploter.

Bankéřův algoritmus

- Algoritmus, který předchází vzniku uváznutí (Dijkstra, 1965).
- **Bezpečný stav**
  - Pokud nedošlo k uváznutí a existuje takové alokační pořadí, které zaručuje, že každý proces bude postupně uspokojen a skončí.
- Bankéřův algoritmus kontroluje, zda **přidělení prostředku vede na bezpečný stav**.
  - Pokud ne, pak je žádost odmítnuta.
  - Pokud ano, prostředek je přidělen.
- **Nevýhoda:** proces musí dopředu vědět, které prostředky bude během svého života potřebovat.

Příklad: bankéřův algoritmus

- V systému je 10 prostředků stejného typu.
- Procesy A, B, C a D budou dohromady potřebovat 22 prostředků.

Počáteční stav

proces	má	max
A	0	6
B	0	5
C	0	4
D	0	7

Bezpečný stav

proces	má	max
A	1	6
B	1	5
C	2	4
D	4	7

Nebezpečný stav

proces	má	max
A	1	6
B	1	5
C	2	4
D	5	7

- **Bezpečný stav:** (2 volné prostředky), pokud dáme 2 prostředky procesu C, ten po skončení uvolní 4 prostředky a ostatní procesy mohou pokračovat.
- **Nebezpečný stav:** (pouze 1 volný prostředek), žádný proces nemůže pokračovat.

Prevence uváznutí

- Nesplnění aspoň jedné z Coffmanových podmínek

1. Porušení výlučného přístupu

- **Sdílení prostředku** s výlučným přístupem pomocí **virtualizace**.

Příklad: sdílení tiskárny

- Každý proces pošle svůj požadavek do tiskové fronty.
- Tiskový démon postupně požadavky z tiskové fronty posílá na fyzickou tiskárnu.
- Pouze tiskový démon přistupuje přímo k fyzické tiskárně.

Bohužel toto řešení nelze použít ve všech případech, např. pásky.

Prevence uváznutí (2)

2. Porušení podmínky „drž a čekej“

- Každý proces musí **alokovat všechny požadované prostředky v okamžiku spuštěním** (např. OS/360).
- **Prostředky nebudou využívány optimálně.**
- Například:
  Proces si alokuje pásku, 5 minut z ní načítá vstupní data, potom 50 minut provádí výpočet a nakonec výsledek tiskne 5 minut na tiskárnu.
- Páska i tiskárna jsou alokovány po celých 60 minut.

Pokud budeme znát požadavky na prostředky v okamžiku spuštění procesu můžeme použít **bankéřův algoritmus.**

Prevence uváznutí (3)

3. Porušení podmínky neodnímatelnosti

- V praxi těžko realizovatelné.

4. Porušení podmínky kruhového čekání

a) Proces může mít přidělen v daném okamžiku pouze jeden prostředek.

b) Proces může alokovat více prostředků, ale pouze v přesně definovaném pořadí.
   - Prostředky mají přiřazena čísla (např. ptiskárna 1, ploter 2, páska 3).
   - Proces může požádat pouze o prostředek s vyšším číslem než je maximum z již alokovaných prostředků.

- Problém: pro daný počet procesů a prostředků **nemusí existovat vhodné očíslování prostředků**.