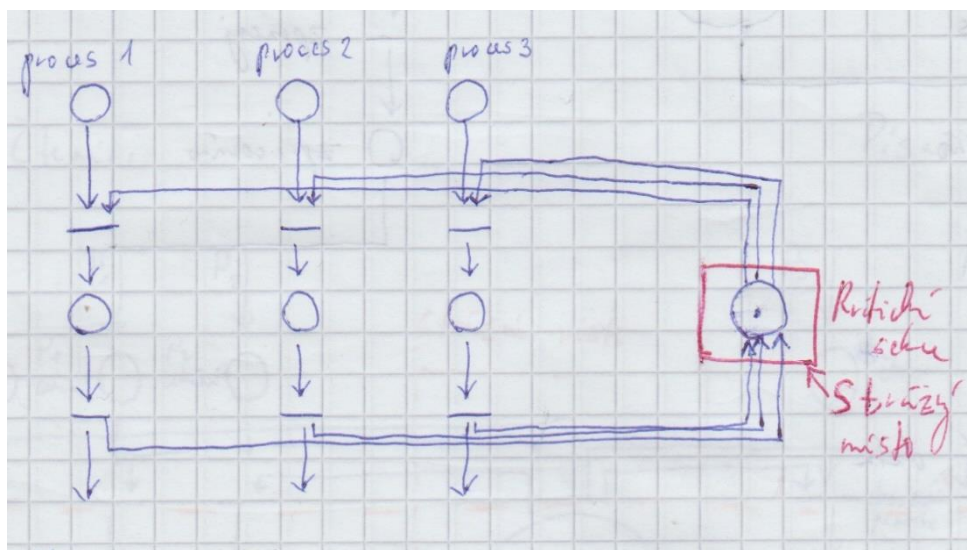


## 19. Synchronizace procesů – kritická sekce, producent – konzument, čtenáři a písáři, 5 hladových filozofů

### HARDWARE A APLIKAČNÍ SOFTWARE

#### Kritická sekce

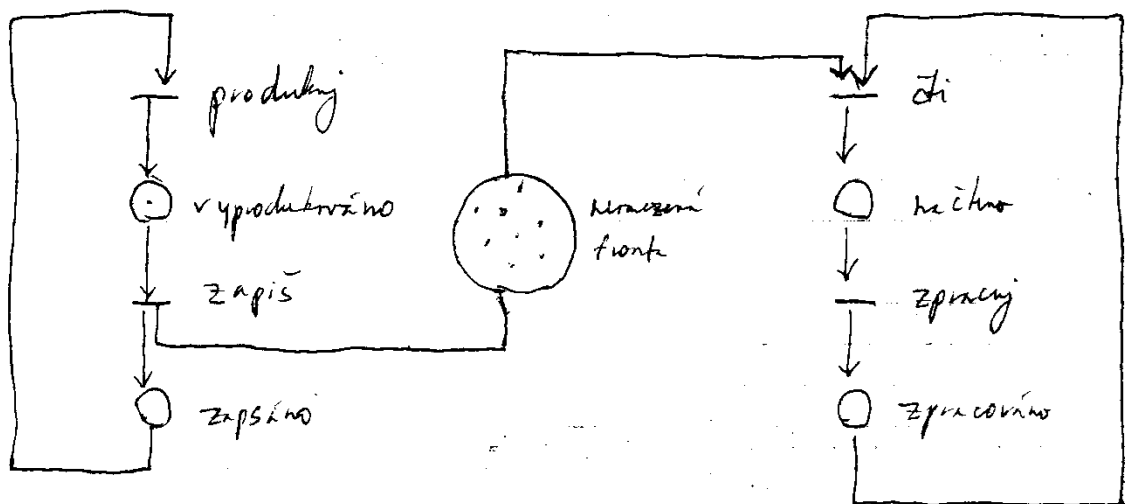
- Řeší problém výlučného přístupu ke sdíleným prostředkům.
- Aby proces mohl provést svou část kódu, přistupující ke kritické musí být ve strážném místě.
- Kdyby další proces chtěl vstoupit do kritické sekce, musí počkat, dokud pracující proces neopustí strážné místo.
- Zajišťuje konzistenci dat ve všech časových intervalech.



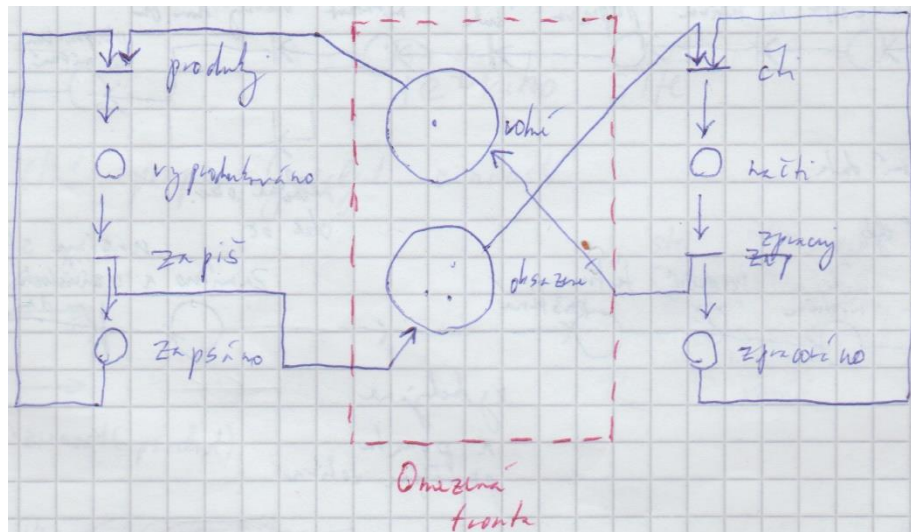
#### Producent – Konzument

- Producent = proces produkující data
- Konzument = proces, který data přijímá a dále zpracovává
- Účel je, aby každý mohl pracovat jinou rychlostí (nezávisle na sobě).

#### Neomezená fronta

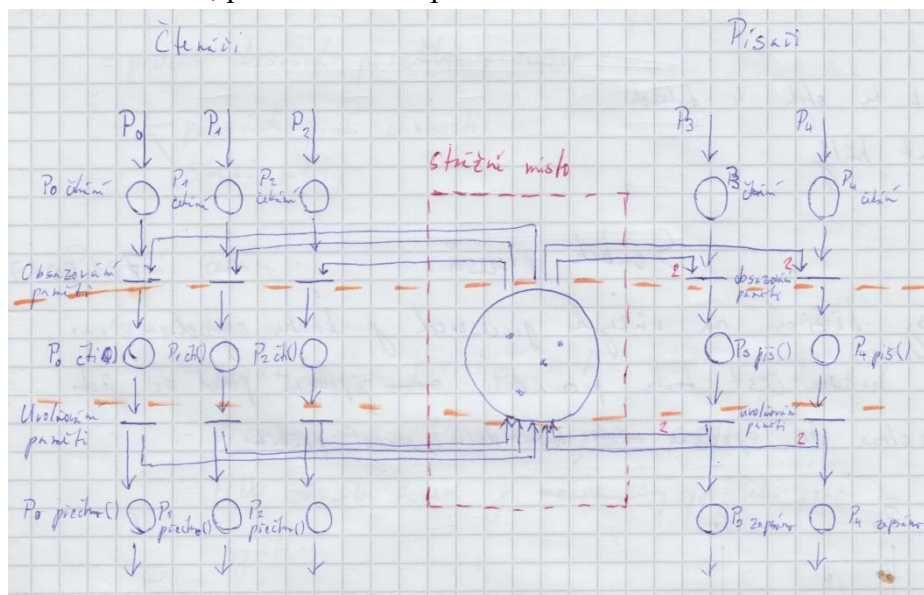


## Omezená fronta



## Čtenáři a písaři

- Procesy jsou rozděleny podle přístupu ke sdílenému prostředku.
- Čtenáři mohou číst, písaři mohou zapisovat.

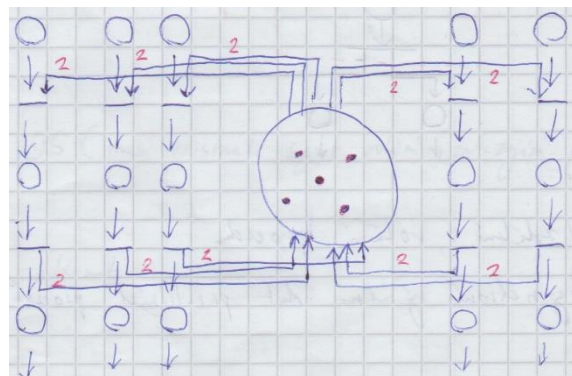


## 5 hladových filozofů

- U kulatého stolu sedí 5 filozofů, každý k jídlu potřebuje 2 hůlky.
- Pokud nemá hůlku po pravé a po levé ruce, tak se nenají.
- Filozof, jehož hůlku mu stále berou zůstává hladový a nedostane se k jídlu, tím dochází ke stárnutí procesu.

## Řešení

- Pustit ke stolu 4 filozofy.
- Přidat hůlky.



## **Deadlock**

- Uváznutí
- Vzájemné čekání na vstup do kritické sekce a požadavku na další prostředky.
- Dochází pouze při splnění všech Coffmanových podmínek.

### **Coffmanovy podmínky:**

#### **Vzájemné vyloučení**

- Prostředek v 1 okamžiku používá jen 1 proces.

#### **Drž a čekej**

- Proces může žádat o další prostředky i když už nějaké má.

#### **Neodnímatelnost**

- Jakmile proces prostředek vlastní, nelze mu ho bezpečně odejmout, musí ho sám vrátit.

#### **Cyklické čekání**

- Je možné uzavřít cyklus z procesu čekajících každý na svého předchůdce.

### **Řešení zablokování**

#### **Prevence**

- Napadení jedné z podmínek.

#### **Vyhýbání se**

- Prostředky se půjčují jen napůl.

#### **Detekce a zotavení**

- Odebírání prostředků
- Zabíjení procesů
- Roll-back (vracení se k před koliznímu stavu)

#### **Ignorování problému**

- Windows, Unix
- Pštroší algoritmus

### **Čekání před kritickou sekcí**

#### **Zamykací proměnná**

- Proměnná osazeno může nabývat hodnoty 0 nebo 1.
- V kritické sekci se může nacházet jen 1 proces.
- Ostatní procesy provádí test proměnné obsazeno.
- Do kritické sekce se dostane ten proces, jehož test se provádí jako první po nastavení proměnné na hodnotu 0.

#### **Střídání procesů**

- Prostředek je střídavě přidělován všem procesům, i těm co ji nepotřebují.
- Pokud proces prostředek nepotřebuje, zbytečně zdržuje ostatní procesy.

#### **Pekařův algoritmus**

- Procesu je při žádosti o přístup přiděleno pořadové číslo.

- Přednost má proces s nejnižším pořadovým číslem.

### **Hardwarové řešení**

- Některé procesy nabízí HW řešení pro aktivní čekání.
- **Instrukce:**
  - TLS
  - SWAP
  - XCHG
- Instrukce provádějí zamykání a odemykání kritické sekce.

### **Bankéřův algoritmus**

- Způsob předcházení
- Bankéř (OS) půjčí prostředky pouze tehdy, zůstane-li banka (systém) v bezpečném stavu.
- To znamená, že zůstane cesta, jak v každém okamžiku je možné přidělit alespoň jednomu klientu (procesu) max. množství prostředků, a poté co všechny své prostředky vrátí, uspokojit ostatní (klienty) procesy.