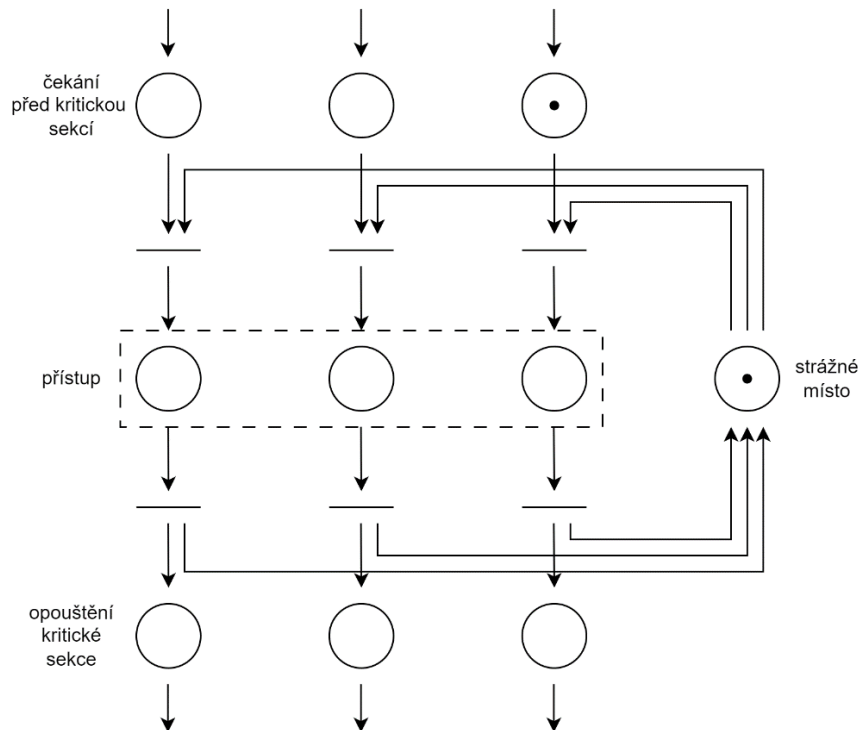


17.

Synchronizace procesů – kritická sekce, producent - konzument, čtenáři a písaři, 5 hladových filosofů

Kritická sekce: řeší problém výlučného přístupu ke sdíleným prostředkům

- Aby proces mohl provést svou část kódu, musí získat přístup ze strážného místa
- Zajišťuje konzistenci dat ve všech časových intervalech



Implementace čekání před kritickou sekcí:

Pasivní čekání: čekající proces nedostává procesor

- **Zákaz přerušení:** proces zakáže přerušení a tím znemožní přepínání kontextů
 - procesor nemůže být přidělen jinému procesu
- **nevýhody:** pouze pro jednoprocessorové systémy
 - ne všechna přerušení lze zakázat
 - proces nelze ukončit ani přerušit
- **Zákaz přepnutí kontextu:** musí podporovat OS
 - nevýhoda: může dojít k uváznutí a stárnutí
- **Navýšení priority:** proces má na procesoru vždy přednost -> zaručení výhradního přístupu k procesoru
- **MUTEX:** uzamknutí prostředku na úrovni jádra

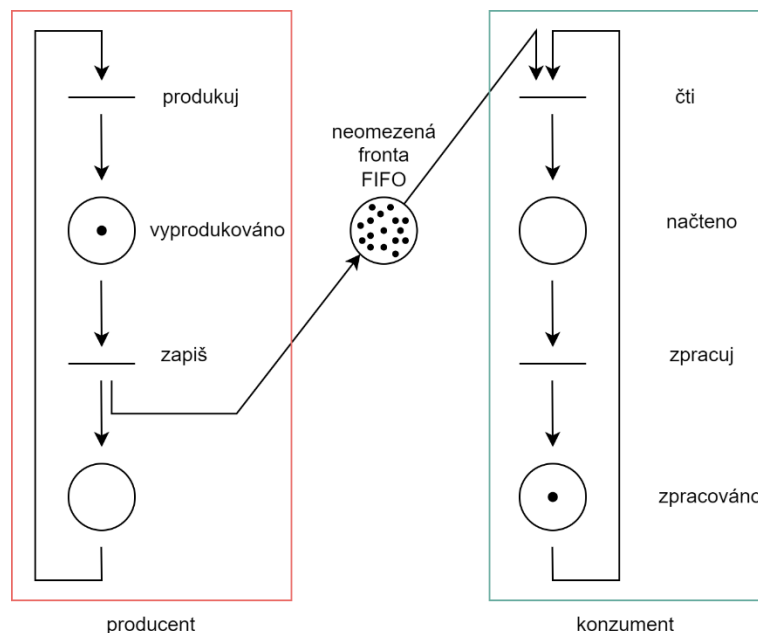
Aktivní čekání: proces dostává přidělen procesor (provádí jiné činnosti / ukládá čekací smyčky)

- **Sdílená zamykací proměnná:** proměnná obsazeno – má možné stavy (0/1)
 - v kritické sekci se může nacházet pouze jeden proces
 - ostatní procesy provádějí test na proměnnou obsazeno
 - dovnitř se dostane ten proces jehož test se provádí jako první po nastavené proměnné na 0

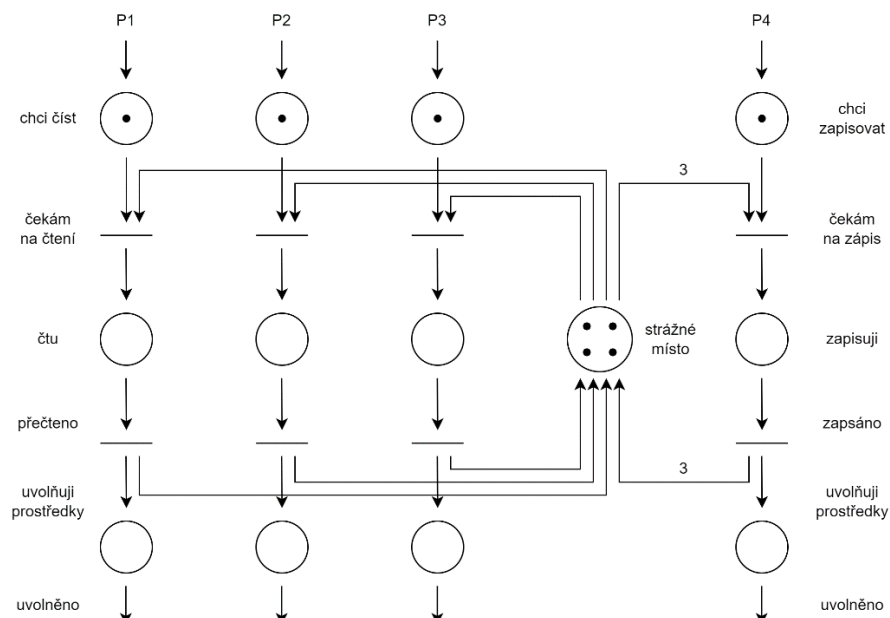
- **Střídání procesů:** prostředek je střídavě přidělován všem procesům
 - pokud proces prostředek nepotřebuje zbytečně zdržuje ostatní
- **Pekařův algoritmus:** proces při žádosti o přístup dostane přidělené pořadové číslo -> proces s nejmenším pořadovým číslem se dostává na řadu
- **Hardwarové řešení:** nutnost podpory ze strany procesoru
 - všechny instrukce provádějí zamykání a odemykání kritické sekce (TLS, swap, XCHG)

Producent-konzument (Buffer): účelem je možnost, aby každý mohl pracovat jinou rychlostí

- Producent je proces generující data
- Konzument je proces, který data zpracuje



Čtenáři a písaři: dvě skupiny: **čtenáři** (pouze čtou sdílená data), **písaři** (mohou měnit tato data)
Cílem je umožnit více čtenářům číst zároveň, ale zamezit situaci, kdy by jeden proces mohl číst zrovna, když jiný proces zapisuje.



Problém 5 hladových filozofů: Tento problém představuje situaci, kdy je několik filozofů usazeno kolem stolu s vidličkami. Každý filozof potřebuje k jídlu dvě vidličky (umístěné vedle něj). Pokud si všichni filozofové vezmou najednou vidličky vedle sebe, může dojít k situaci vzájemného blokování, kde žádný z nich nemůže pokračovat (hodování) kvůli nedostatku vidliček. Řešením je správná synchronizace tak, aby se minimalizovalo vzájemné blokování a zároveň se předešlo hladovění některých filozofů.

