Plánování procesů

Multi-Level Feedback Queue

Milan Radojčić

SSPŠaG – Operační systémy

10. 11. 2025

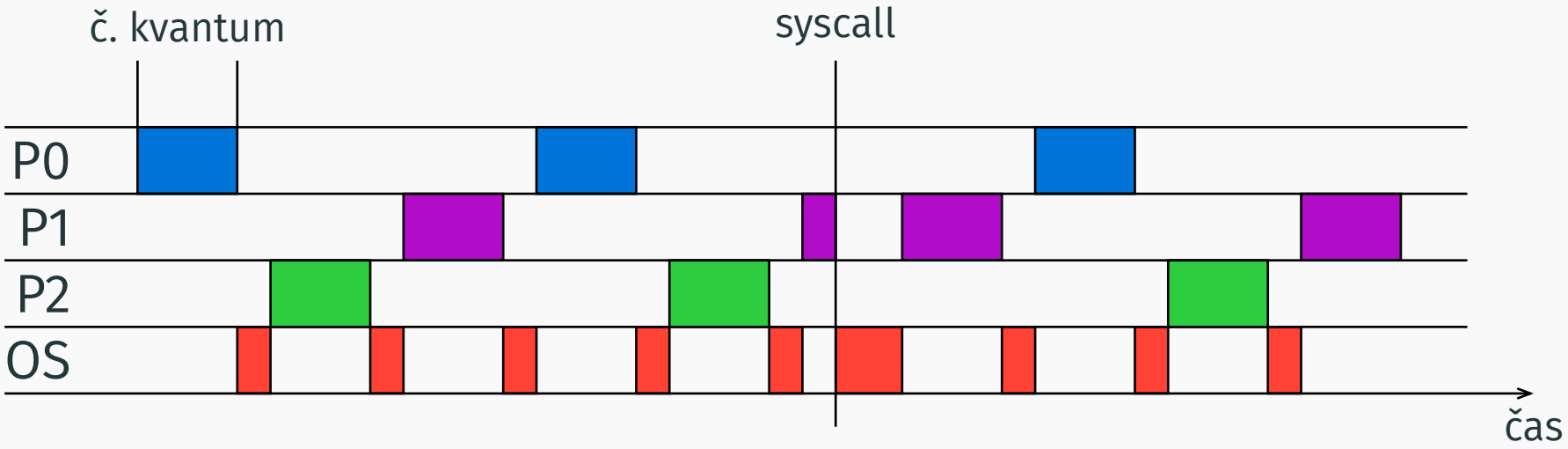

IBM

Historické okénko

- Na předchozím slidu je F. Corbato, který s jeho kolegy vymyslel **multi-level feedback queue** (MLFQ).
  - Za tuto práci později obdržel Turingovu cenu od ACM.[1]
- Jejich systém s MLFQ se jmenoval **compatible time-sharing system** (CTSS).
- CTSS začal být používán v roce 1963.
- Počítač vlevo dole je IBM 7090.

[1] Stejnou cenu dostali např. E. Dijkstra, vynálezci RSA, W. Diffie, M. Hellman, D. Knuth, R. Hamming nebo D. Richie a K. Thompson. ACM – Association for Computing Machinery

Opakování: Preemptivní plánování



Při **preemptivním přístupu** je vláknům přiřazeno nějaké **časové kvantum**. Po něm je jim procesor odebrán. Vlákna se mohou vzdát procesoru před vypršením kvanta pomocí systémového volání.

Opakování: Metriky

Doba k dokončení (turnaround time)

$$T_{\text{turnaround}} = T_{\text{completion}} - T_{\text{arrival}}$$

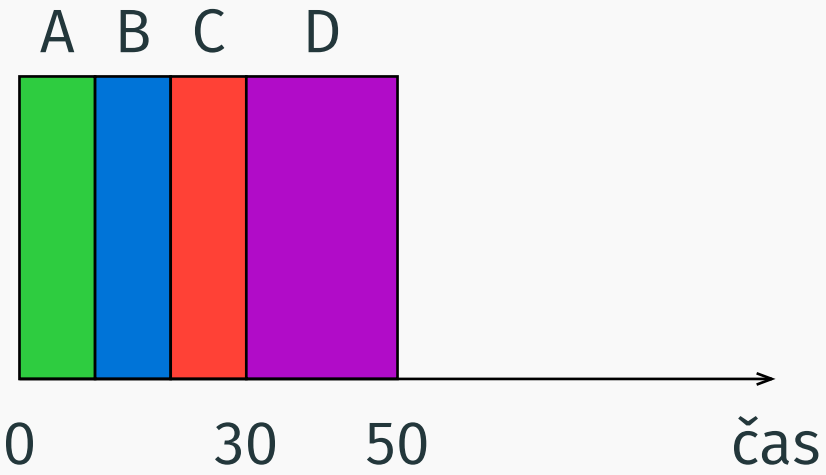
Doba odezvy (response time)

$$T_{\text{response}} = T_{\text{firstrun}} - T_{\text{arrival}}$$

Opakování: Doba odezvy vs. doba k dokončení

A, B, C trvají 10s. D trvá 20s. Round robin střídá procesy po 5s.

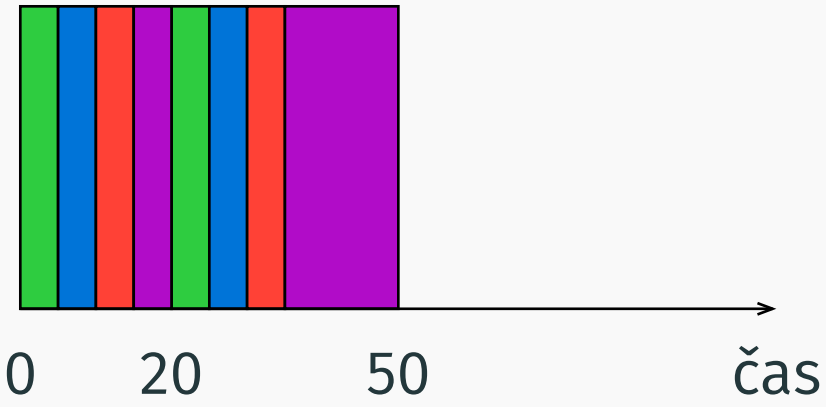
SJF



$$T^{\text{average}}_{\text{response}} = \frac{10 + 20 + 30}{4} = 15\text{s}$$

$$T^{\text{average}}_{\text{turnaround}} = \frac{10 + 20 + 30 + 50}{4} = 27,5\text{s}$$

Round Robin



$$T^{\text{average}}_{\text{response}} = \frac{5 + 10 + 15}{4} = 7,5\text{s}$$

$$T^{\text{average}}_{\text{turnaround}} = \frac{25 + 30 + 35 + 50}{4} = 35\text{s}$$

Opakování: Doba k dokončení vs doba odezvy

- Vidíme, že tyto algoritmy umí optimalizovat dobu k dokončení nebo dobu odezvy, **ale ne oboje**.
  - Pro programy, které provádí hodně výpočtů (**CPU-bound**) je lepší **doba k dokončení**.
  - Pro programy, které jsou interaktivní (**IO-bound**) je lepší **doba odezvy**.

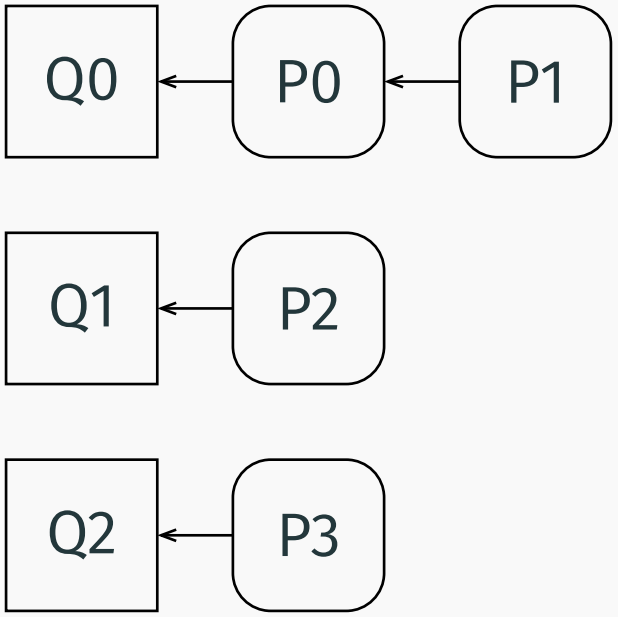
Opakování: Co dál?

- Pořád nám zbývá zbavit se posledního předpokladu.
  - Doopravdy neznáme dobu běhu každého procesu.
- Jak tohle překonat si povíme teď! To povede na **multi-level feedback queue** (MLFQ).
  - To bude náš první opravdový plánovač, který je stále používán (např. v Solaris).

Zkuste krátce ve skupinkách po dvou až třech vymyslet vlastní nápady, jak navrhnout plánovač.

Základní principy MLFQ

MLFQ používá několik **front**, každá odpovídá nějaké **prioritě**.
Procesy jsou spouštěny na nějaké časové **kvantum**.



Pravidla pro výběr:

1. Priorita(A) > Priorita(B): A je spuštěno.
2. Priorita(A) = Priorita(B): A a B se střídají pomocí RR.

Základní principy MLFQ

- Důležitá otázka teď je: **jak bude MLFQ určovat priority procesů?**
- Procesům bude přiřazeno nějaké množství času, které mohou strávit v dané frontě.
  - Ze začátku budeme předpokládat, že tohle množství času je stejné jako kvantum na které jsou plánovány.

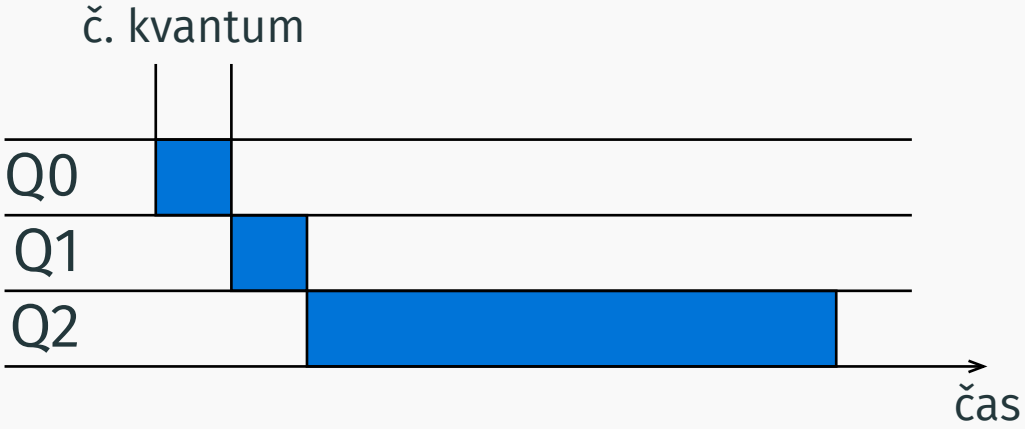
Pravidla pro přiřazení priority:

3. Nové procesy dostanou nejvyšší prioritu.
4. Jakmile proces využije přiřazené množství času, tak je snížena jeho priorita.
5. Pokud nevyužije přidělené množství času[<2>], tak mu ponecháme jeho prioritu.

[<2>] Vzdá se CPU před koncem č. kvanta.

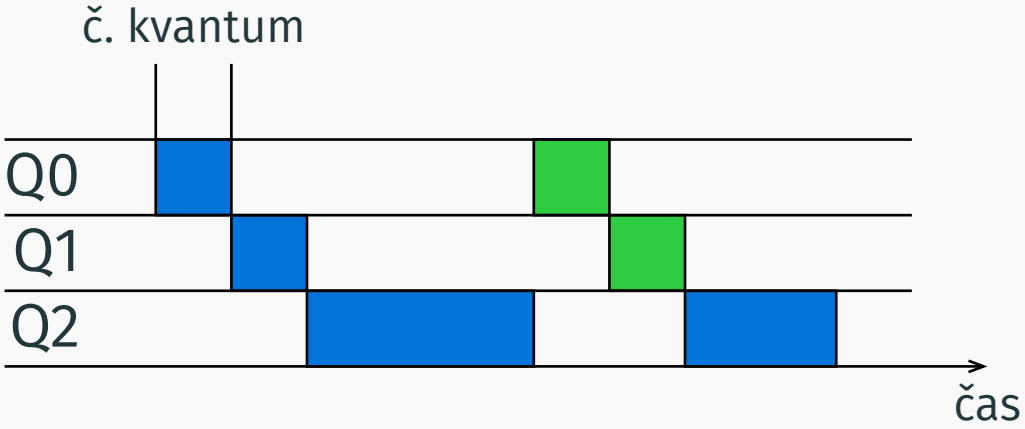
Příklad 1: jeden dlouhy proces

Ukážeme si fungování MLFQ na jednoduchém příkladu s jedním procesem, který používá jen CPU (neprovádí IO).



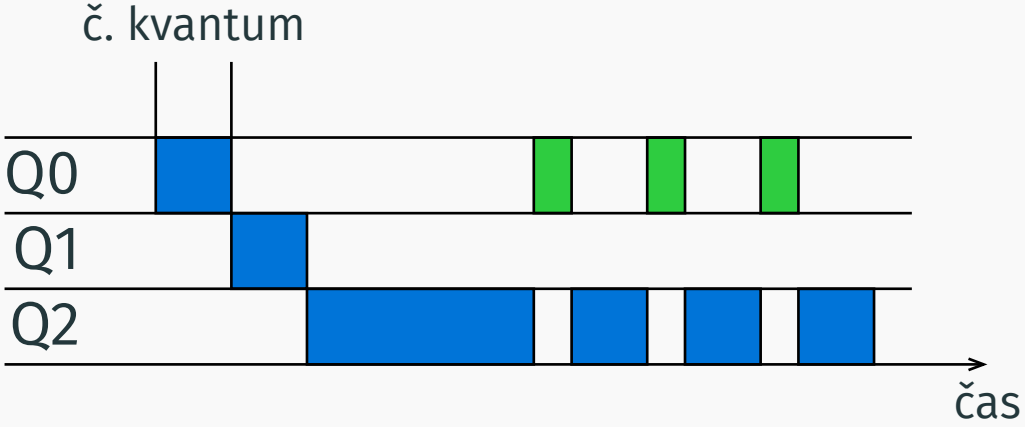
Příklad 2: dlouhý a krátký proces

Stejný příklad jako minule, ale později se spustí krátký proces.



Příklad 3: CPU-bound a interaktivní proces

Místo krátkého procesu se nyní spustí interaktivní proces.



Problémy s MLFQ

Zatím popsaný princip má nějaké problémy, jaké?

1. Procesy s nejnižší prioritou se nemusí dostat k CPU, když je hodně krátkých nebo interaktivních procesů.
   - Tomu se říká **hladovění** (starvation).
2. Je možnost „přechytračit“ plánovač. Když se proces vzdá CPU vždy těsně před vypršením času, tak může mít stále nejvyšší prioritu.
3. Procesy nemusí být celou dobu jen interaktivní nebo CPU-bound.
4. Není jasné, jak nastavit počet front a množství času, které procesy mohou strávit v dané frontě.

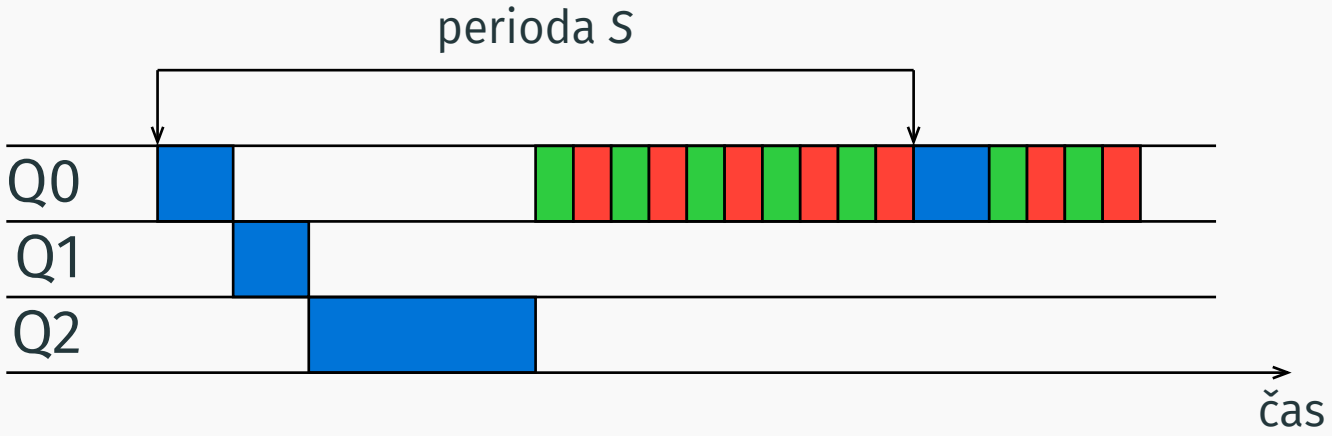
Oprava hladovění

Máte nápady, jak předejít hladovění?

Přidáme nové pravidlo:

6. Periodicky po nějakém času *S* dáme všem procesům nejvyšší prioritu.

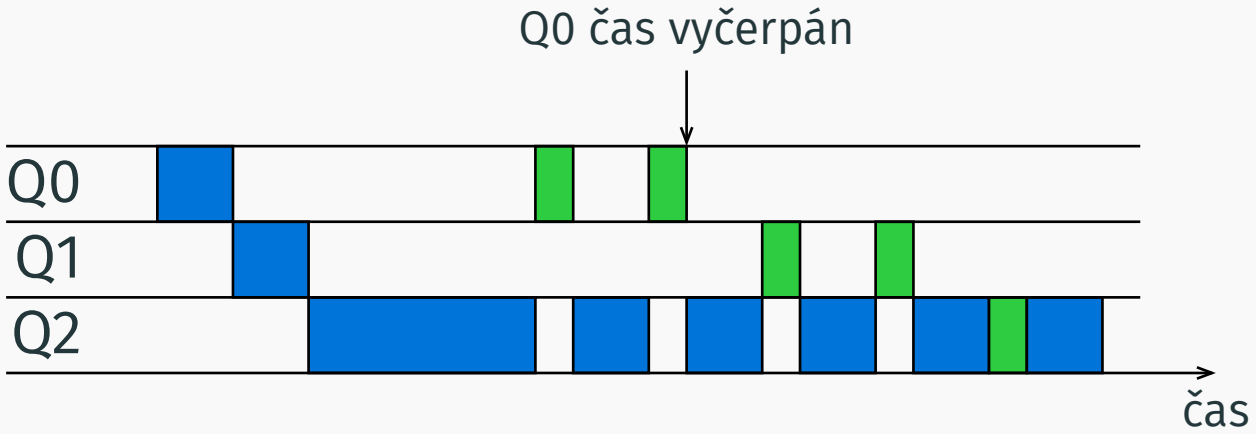
Tohle nám zároveň řeší problém se změnou chování procesů.



Ukázka s interaktivními procesy

Oprava sledování času procesů

Přidělený čas nebude resetován mezi spuštěním procesu.



Ukázka s interaktivními procesy

Poznámky k MLFQ

- Pořád jsme si neřekli jak určit počet front, časové limity pro fronty atd.
  - Většina implementací má nějak empiricky určené výchozí nastavení, které funguje ve většině případů.[3]
- Naše verze také nedovoluje dávat manuálně procesům různé priority (např. pomocí `nice`).

[3] Pro víc info viz třeba https://ostep.org kapitola Scheduling: The Multi-Level Feedback Queue, předposlední podkapitola.

Shrnutí MLFQ

Pravidla pro výběr:

1. Priorita(A) > Priorita(B): A je spuštěno.
2. Priorita(A) = Priorita(B): A a B se střídají pomocí RR.

Pravidla pro přiřazení priority:

3. Nové procesy dostanou nejvyšší prioritu.
4. Jakmile proces využije přiřazené množství času, tak je snížena jeho priorita.
5. Pokud nevyužil přidělené množství času, tak mu ponecháme jeho prioritu.

Prevence hladovění:

6. Periodicky po nějakém čase S dáme všem procesům nejvyšší prioritu.

Poznámky k plánování

- Obecně je efektivita plánování důležitá pro efektivitu celého systému.
- Plánování procesů je komplikovaná věc a MLFQ není jediný přístup.[4]
  - Aktuální výchozí plánovač v Linuxu je založený na **principu earliest eligible virtual deadline first** (EEVDF).
- Dál jsme se vůbec nebavili o tom **jak plánovat na více jádrech**.[5]
- Jiný způsob, jak postavit plánovač je např. proporcionální rozdělení (proportional share).[6]

[4] Viz např. nedávná práce o optimalizaci plánování na Steam Decku.

[5] Viz kapitola Multiprocessor Scheduling z https://ostep.org. Pozor, vyžaduje nějaké znalosti o synchronizaci vláken.

[6] Viz kapitola Scheduling: Proportional Share z https://ostep.org.

Cvičení

- Na https://radojcic.cz/osy/handouts/ máte handout s kódem: 3_planovani_mlfq.zip.
- Deadline je prespříští hodina.