Principy počítačů a operačních systémů

Operační systémy
Procesy a vlákna, plánování

Zimní semestr 2011/2012

Procesy a vlákna

Jak mohou aplikace (a OS) sdílet procesor(y)?

Aplikace si myslí, že systém má neomezený počet procesorů...

Procesy a vlákna

Proces = abstrakce počítače

- instance běžící aplikace (posloupnost instrukcí)
  - jedna aplikace může běžet ve více instancích
- představuje výpočetní prostředí z pohledu aplikace
  - v podstatě poskytuje aplikaci virtuální stroj
- existuje pouze po dobu běhu aplikace

Vlákno je součástí procesu

- představuje konkrétní běžící výpočet – aktivitu
- každý proces musí mít alespoň jedno vlákno
  - proces může mít samozřejmě více vláken

Proces poskytuje kontext

Proces jako abstrakce počítače definován

- stavem procesoru (hodnoty registrů)
- adresovým prostorem (obsah paměti)
- prostředím (struktury operačního systému)

Registry CPU obsahují současný stav (výpočtu)

Stav CPU určen obsahem registrů

- Processor Status Word (PSW)
  - režim procesoru, příznaky výsledku poslední aritmetické operace (zero, negative, overflow, carry), zakázaná/povolená přerušení
- Program Counter (PC) / Instruction Pointer (IP)
  - adresa následující instrukce
- Stack Pointer (SP)
  - adresa aktuálního stack frame, obsahuje návratovou adresu a lokální proměnné funkce
- General Purpose Registers
  - obsah OS nezajímá, to má (měl) na starosti překladač aplikace

Paměť obsahuje (dosavadní) výsledky výpočtu

Oblasti v adresovém prostoru procesu

- text
  - kód aplikace, typicky pouze ke čtení, může být sdílen více procesy (tj. více instancemi stejné aplikace)
- data
  - statická (definovaná při překladu) data aplikace, typicky konstanty a globální proměnné
- heap
  - "natahovací" oblast, ze které si může proces "ukusovat" paměť za běhu podle potřeby (pomocí operátoru **`new`**, resp. funkce **`malloc`**)
- stack
  - lokální úložiště pro registry CPU a lokální proměnné funkcí

Prostředí obsahuje vztahy k jiným entitám

Proces neexistuje ve vakuu, je vázán na...

- uživatelský terminál
  - textové/grafické rozhraní pro uživatele
- otevřené soubory používané pro vstup a výstup
- komunikační kanály a sokety
  - spojení mezi procesy, potenciálně na různých počítačích

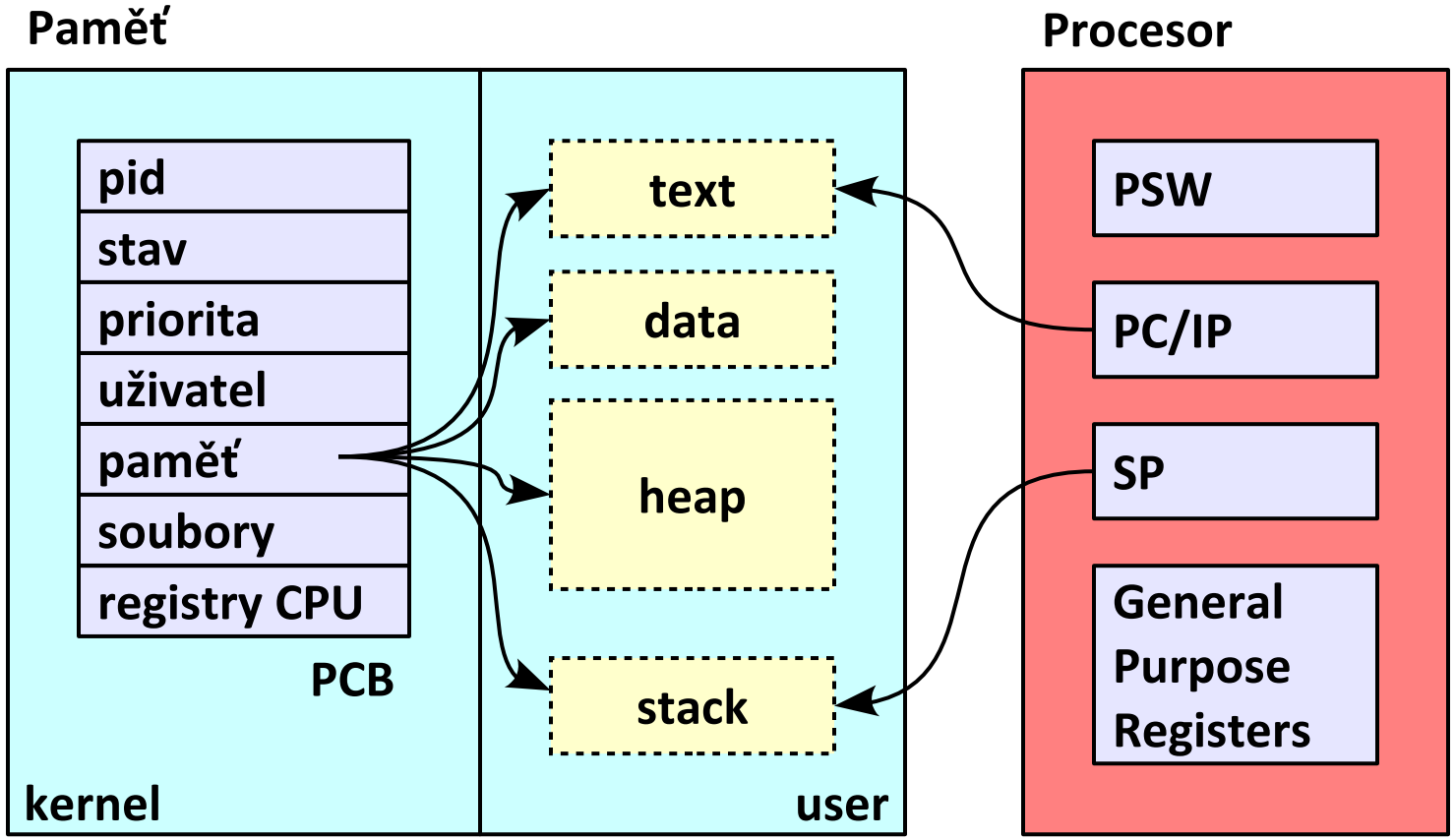
Vazby zachycuje OS ve svých strukturách...

- tabulka otevřených souborů/soketů

Veškerá data o procesu udržuje OS

Process Control Block

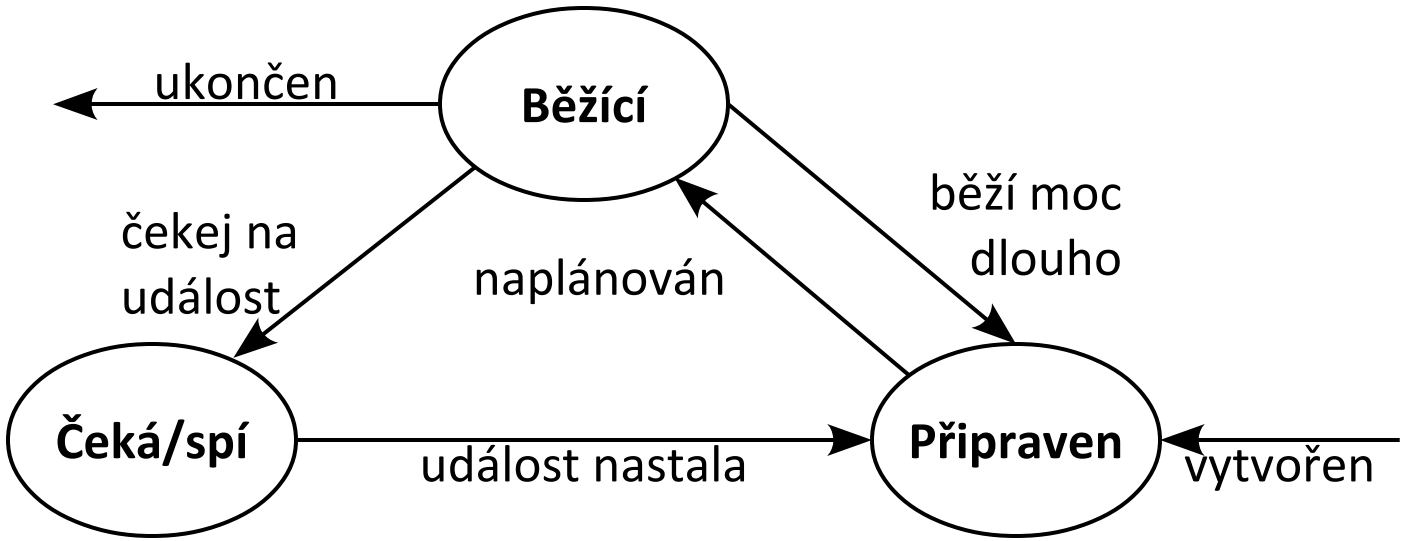
- datová struktura OS popisující proces
- z pohledu OS reprezentuje proces



Základní stavy procesu

Přechody mezi stavy

- v důsledku událostí v systému



Změna stavu jako reakce na události

Události v systému

- synchronní – vznikají v důsledku běhu procesu
  - traps, speciální instrukce (např. pro volání OS)
  - exceptions, nesprávné chování procesu
- asynchronní – vznikají vně systému
  - žádost zařízení o přerušení

Obsluha přerušení

- procesor předá řízení OS, uloží se kontext CPU
- analyzuje se příčina přerušení, vyvolá se obsluha
- obsluha přerušení, obnovení kontextu CPU
- návrat do přerušené aplikace

Vlákna

Vícevláknový proces

- více současně vykonávaných výpočtů (aktivit)
  - každý běžící výpočet definován stavem CPU, tj. registry PSW, PC/IP, SP, a GPRs
- abstrakce stroje s více procesory
  - procesory mohou (nemusí) běžet současně
    - concurrent vs. parallel
  - všechy "procesory" (vlákna) sdílí paměť a prostředky

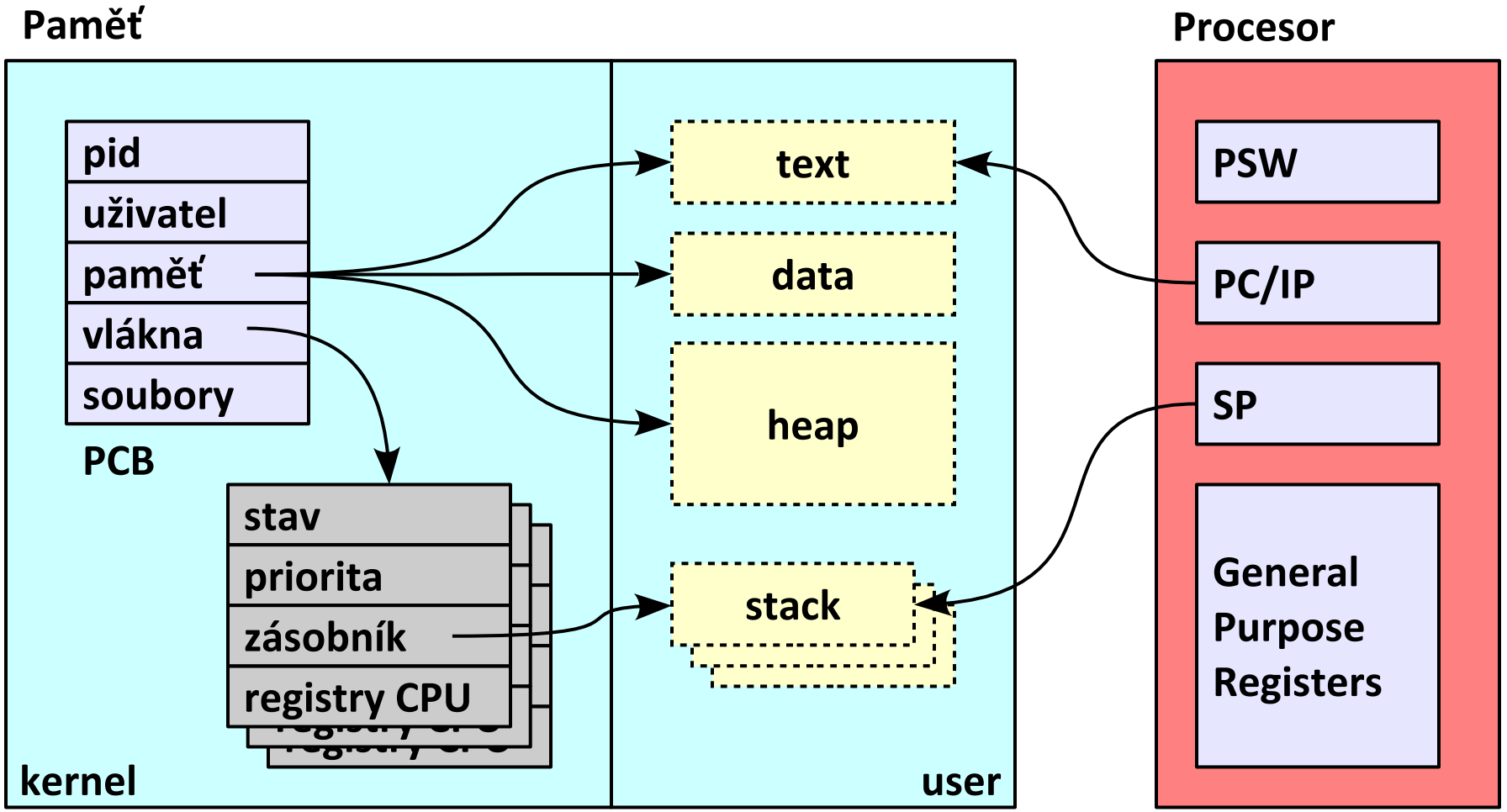
Vlákna procesu sdílí

- kontext paměti, kontext prostředí

Vlákna nesdílí

- kontext CPU, zásobník (historie běhu výpočtu)

PCB pro vícevláknové procesy



K čemu se používají vlákna?

Strukturování programu

- samostatná vlákna pro zpracování požadavků na serveru – vlákno se nemusí starat o další požadavky, které přicházejí zatímco pracuje

Implementace asynchronních I/O operací

- překrytí výpočetních operací s I/O operacemi – pomalé I/O operace vykonává jedno vlákno (při čtení/zápisu se zablokuje), zatímco jiné vlákno "počítá"

K využití více CPU je nutná implementace v OS

- copak jdou vlákna implementovat i jinak?

Operace s procesy a vlákny

Vytvoření nového (fork/spawn/create)

- nový proces má typicky pouze 1 (hlavní) vlákno
- nové vlákno lze vytvořit jen v rámci procesu

Ukončení existujícího (exit)

- návrat z hlavní funkce (main), nebo volání exit

Pozastavení

- dočasné (sleep), čekání na událost (wait)

Zaslání signálu/zprávy

- obdoba přerušení u procesoru, obsluhuje běhové prostředí, typicky ukončí proces (možno předefinovat)

Multiprogramming

... aneb když je v systému více procesů...

K čemu se hodí více procesů?

Zlepšení odezvy systému

- dlouhé odezvy při dávkovém zpracování úloh
  - při sdílení procesoru by kratší úlohy byly hotovy dříve

Zlepšení využití prostředků

- aplikace typicky něco počítá, nebo čeká na data
  - doba čekání na data z disku v řádu ***ms***
- během čekání mohou jiné aplikace "něco počítat"
  - zlepšuje odezvu – jiný proces může postupovat vpřed

Současný běh více procesů

- přepínání mezi více aplikacemi, spojování procesů pro práci na stejném problému, ...

Plánování procesů a vláken

Plánování procesů a vláken

OS musí rozhodnout, který proces (vlákno) poběží

- rozhodnutí typicky optimalizuje nějakou metriku

Typické metriky

- doba odezvy (response time, turnaround)
  - do ukončení procesu, do první odezvy, ...
- propustnost (throughput)
  - počet dokončených úloh za jednotku času
- využití procesoru (utilization)
- spravedlnost (fairness)

Off-line plánování

Předpoklady

- všechny procesy jsou k dispozici od začátku a žádné již nepřibydou
- o všech procesech je známo jak dlouho poběží

Výsledky

- dávkové zpracování s ohledem na cílové metriky
- běh procesů není nutné přerušovat, plán je optimální

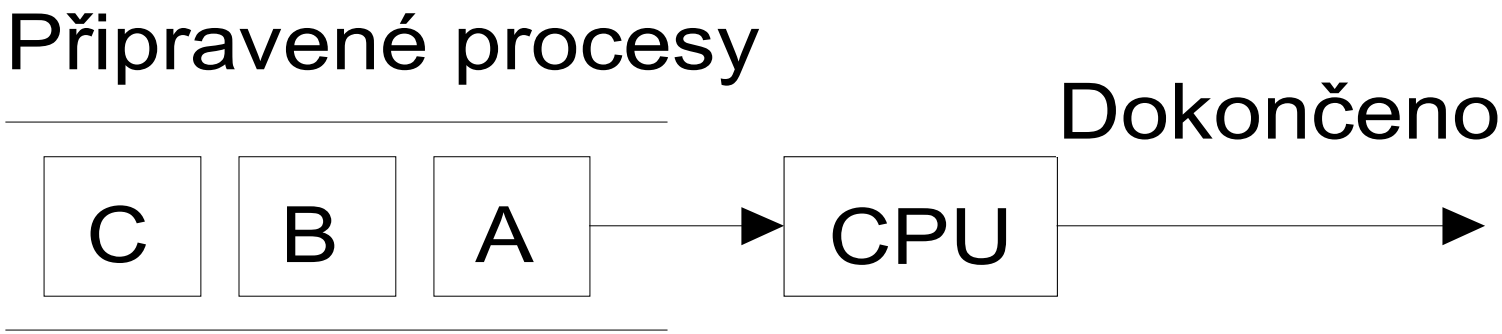
Problém

- předpoklady jsou málo realistické
- poskytují teoretické meze, pokud bychom měli požadované informace

Základní off-line algoritmy

FCFS – First Come First Served

- základní algoritmus dávkového zpracování
  - procesy plánovány v pořadí, v jakém přicházejí
  - procesy běží dokud neskončí



SJF – Shortest Job First

- kratší úlohy plánovány přednostně
  - minimalizuje průměrnou dobu odezvy

On-line plánování

Předpoklady

- procesy se objevují libovolně a neočekávaně
- doba běhu procesů je neznámá

Kritéria plánování

- vázanost na CPU nebo I/O, interaktivní/dávkový proces
- chování procesu v minulosti, výpadky stránek, priorita

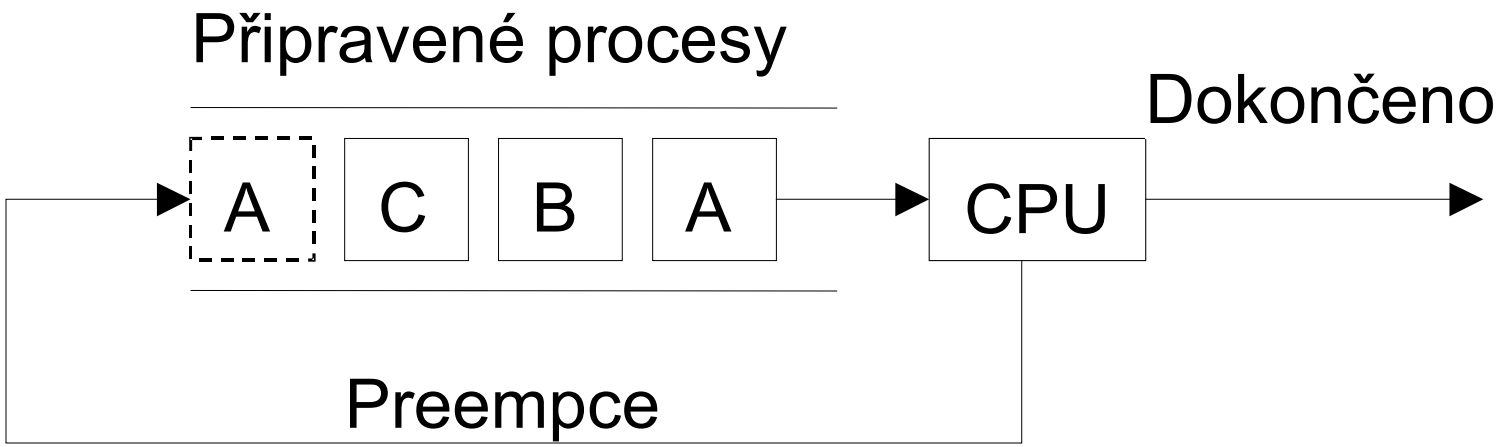
Preemptivní plánování

- potřebuje podporu HW (časovač)
- možnost měnit plán na základě nových informací
- ***context switch*** – přepnutí na jiný proces / vlákno

Round robin plánování

Sdílení procesoru

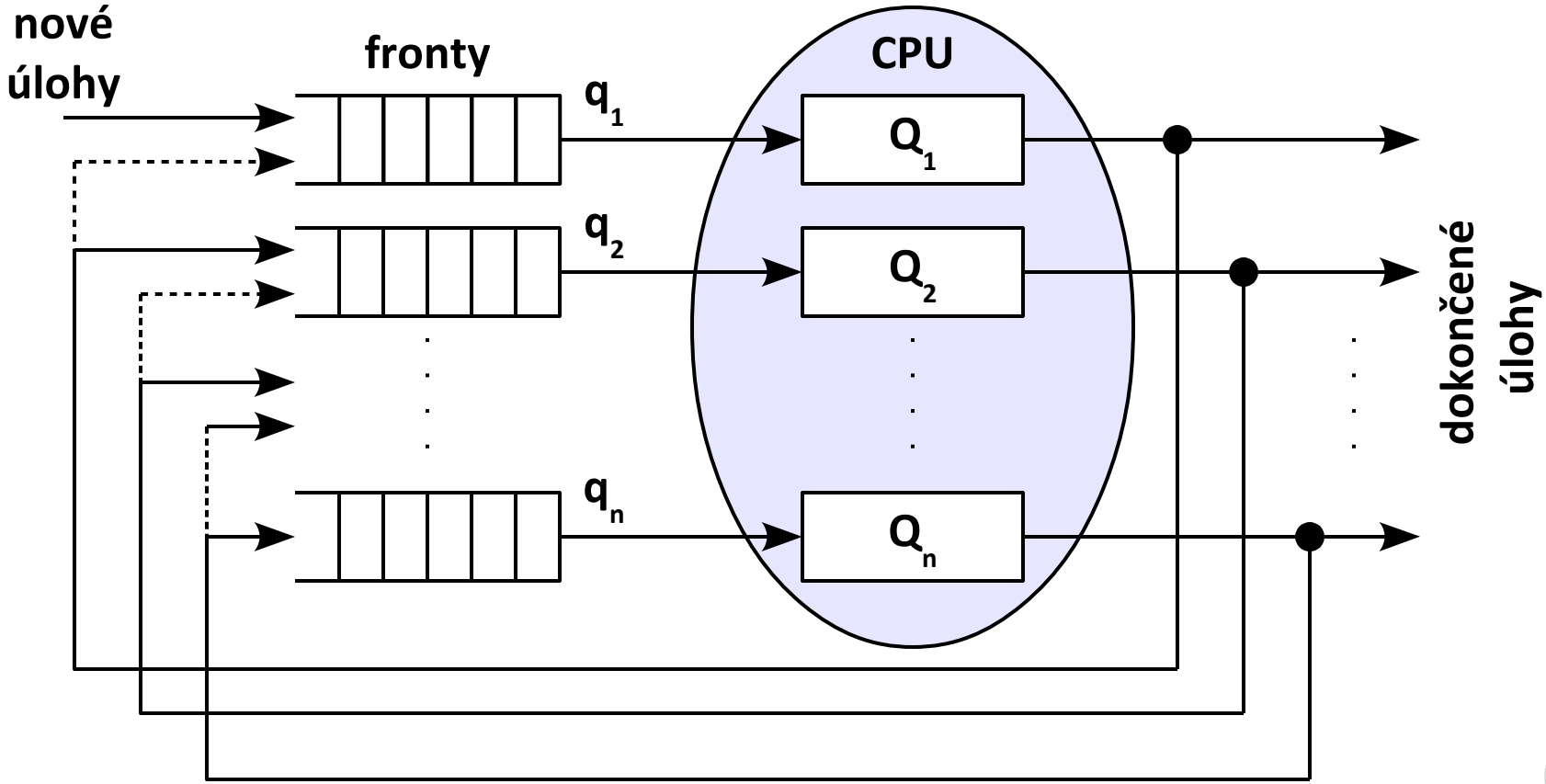
- časové kvantum (time slice)
- preemptivní plánování, fairness



Prioritní plánování s více frontami

Reaguje na chování úloh

- rozlišuje interaktivní a dávkové úlohy
- priorita úlohy (určuje frontu a časové kvantum Q_i)



Speciální plánovače

... pro více procesorů

- každý procesor má vlastní "ready queue"
- vyvažování zátěže, zohlednění afinity

... pro real-time systémy

- aplikace řízené událostmi
  - procesy ~ obsluha událostí, příjem a zpracování dat, generování výstupu
- běh omezen reálným časem dokončení – deadline
- hard real-time, soft real-time
- často off-line plánování