

# Планиране на процеси

Ангел Чолчев<sup>1</sup>  
atcholtchev@elsys-bg.org

<sup>1</sup>Технологическо училище "Електронни системи"  
Технически университет, София

6 ноември 2009 г.



# Процеси

- Поведение на процесите
  - работещи предимно с процесора - cpu bound processes
  - работещи предимно с IO операции - IO bound processes

# Планиране

- Кога да планираме процесите
  - при multiprocessing имаме нужда от стратегия за следващите процеси
  - прекъсвания
- Политики относно процесите - policies
  - непрекъване - non preemptive
  - прекъсване - preemptive

# Планиране - системи

- Три основни вида системи
  - Системи с последователно изпълнение - batch systems
  - Интерактивни системи - interactive systems
  - Системи работещи в реално време - real time systems

# Планиране - цели

- Глобални цели
  - Честно разпределение на процесора - Fairness
  - Баланс - системата трябва да бъде постоянно заета

# Планиране - цели

- Batch Systems
  - Производителност(Throughput) - възможно най-много задачи за единица време
  - Turnaround time - времето от планирането до завършването на задачата
  - CPU Utilization

# Планиране - цели

- Interactive Systems
  - Response time - трябва да отговаря на заявките бързо
  - Пропорционалност - трябва да отговаря на очакванията на потребителя

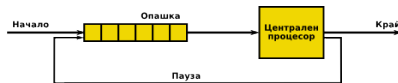
# Планиране - цели

- Realtime Systems
  - meeting deadlines - процеса трябва да се завърши за точно определено време
  - Predictability - не трябва да се влошава качеството при мултимедия



# Планиране в последователни системи

- Scheduling in Batch Systems
  - nonpreemptive nature



# Планиране в последователни системи

- First Come First Served
  - лесен за разбиране и имплементиране
  - nonpreemptive - много бавен с io bound процесите

# Планиране в последователни системи

- Shortest Job First
  - постигаме доста по-добро средно turnaround време
  - всички процеси трябва да известни предварително
  - всички процеси трябва да са готови за изпълнение едновременно

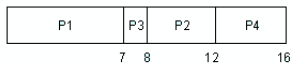
# Планиране в последователни системи

## Non-preemptive Shortest Job First

- Example:

<b>Process:</b>	<b>p1</b>	<b>p2</b>	<b>p3</b>	<b>p4</b>
<b>Arrival time:</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>5</b>
<b>Burst time:</b>	<b>7</b>	<b>4</b>	<b>1</b>	<b>4</b>

- Schedule:



**Waiting time:** 0      6      3      7

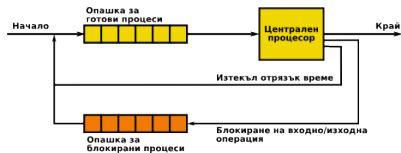
**Average waiting time:** 4.

# Планиране в последователни системи

- Shortest Remaining time
  - preemptive версия на Shortest Job First.
  - използвана в Interactive Systems

# Планиране в интерактивни системи

- Scheduling in Interactive System
  - nonpreemptive nature

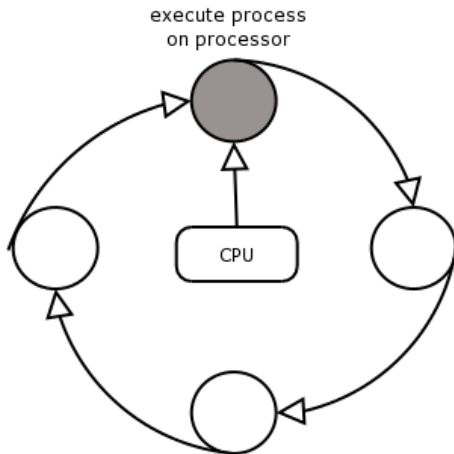


# Планиране в интерактивни системи

- Round Robin Algorithm

- quantum - единица време, за която процеса се изпълнява
- колко голям трябва да бъде отчитайки contex switch time
  - малка стойност - големи загуби от чест context switch
  - голема стойност - процес в края на опашката чака твърде дълго

# Планиране в интерактивни системи- RoundRobin





# Планиране в интерактивни системи

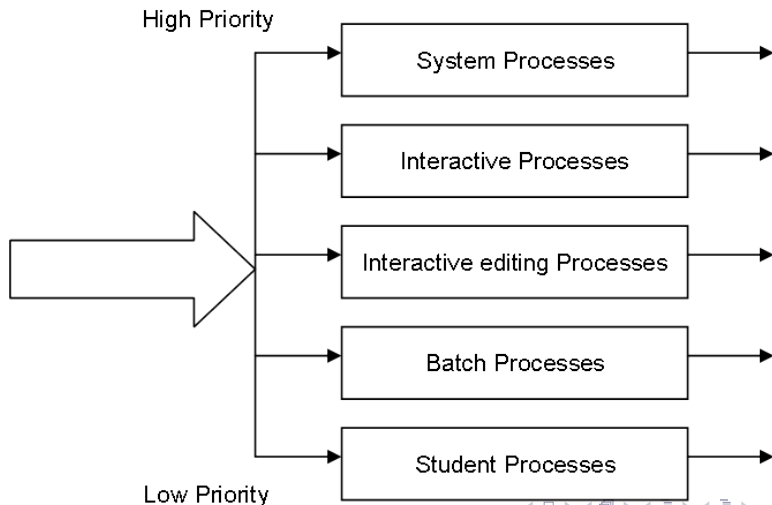
- Priority Scheduling Algorithm
  - roundRobin базиран - всеки процес работи една quanta
  - всеки процес има приоритет
  - бива избран процесът с най-голям приоритет в опашката

# Планиране в интерактивни системи

- Priority Scheduling Algorithm с много опашки(multiple queues)
  - IO bound - трябва да бъдат планирани бързо, за да бъде системата предсказуема
  - CPU bound - трябва да работят по-дълго, за да не губим време в context switch
  - наличие на много опашки с различен приоритет.
  - опашките биват редувани по round robin algorithm
  - процес с най-висок приоритет бива пускан 1 quantum време
  - процес с по-нисък приоритет бива пускан 2 quantum време
  - процес с най-нисък приоритет бива пускан 4 quantum време
  - приоритетите се задават динамично

# Планиране в интерактивни системи

- Priority Scheduling Algorithm с много опашки(multiple queues)



# Планиране в интерактивни системи

- Shortest process next
  - версия на shortest process fist за интерактивни системи
  - предложения за дължината на процеса се правят на базата на предишните изпълнения

# Планиране в интерактивни системи

## ● Lottary Scheduling

- всеки процес получава лотарийни билети
- победителят от лотарията печели процесорно време
- важните процеси имат много билет - ще бъдат пускани често
- лесен за имплементиране - ако даден процес има  $n$  билета, ще бъде пуснат  $n$  време
- новите процеси могат да бъдат пуснати веднага - highly responsive

# Планиране в интерактивни системи

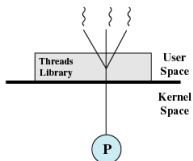
- Guaranteed Scheduling
  - $n$  потребителя - всеки потребител получава гарантирано  $1/n$  процесорно време
  - single user system,  $m$  процеса - всеки процес получава  $1/m$

# Планиране в интерактивни системи

- Fair-share scheduling
  - Problem - 2 потребителя с Round Robin
    - User 1 - 9 Processes - 90% computing time
    - User 2 - 1 processes - 10% computing time
  - Fair-share algorithm подsigурява равнопоставеност между потребителите
    - User 1 - 4 Processes A B C D
    - User 2 - 1 processes E
    - Нека обещаем на 2 потребителя по равно CPU
    - Scheduling A E B E C E D E A E B E C E D E...
    - Нека обещаем на потребител 1 2x повече CPU
    - Scheduling A B E C D E A B E C D E...

# Планиране на нишки

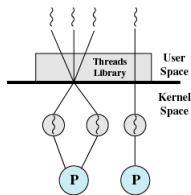
- User level threads
  - Позволява използването на custom process scheduler





# Планиране на нишки

- Kernel level threads
  - Планирането може да бъде доста скъпо - може изисква context switch(ако нишките са в различни процеси)



(c) Combined