

# Sistemas Operativos

LEI - 2021/2022

**:: Escalonamento de processos ::**

**Escola Superior de Tecnologia de Setúbal - IPS**

# Conteúdos

- Escalonamento de processos no CPU
- Algoritmos de escalonamento

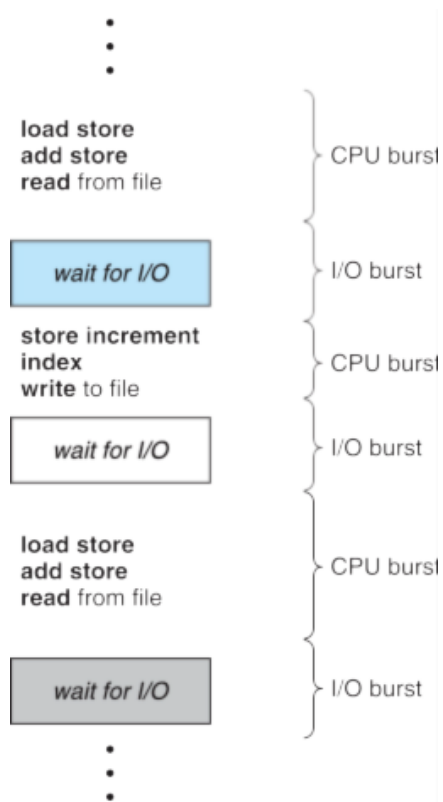
# Porquê escalonamento de processos?

- Utilização máxima do CPU
- Ciclos de CPU/IO
  - Conjunto de instruções CPU seguido de um conjunto de instruções de I/O

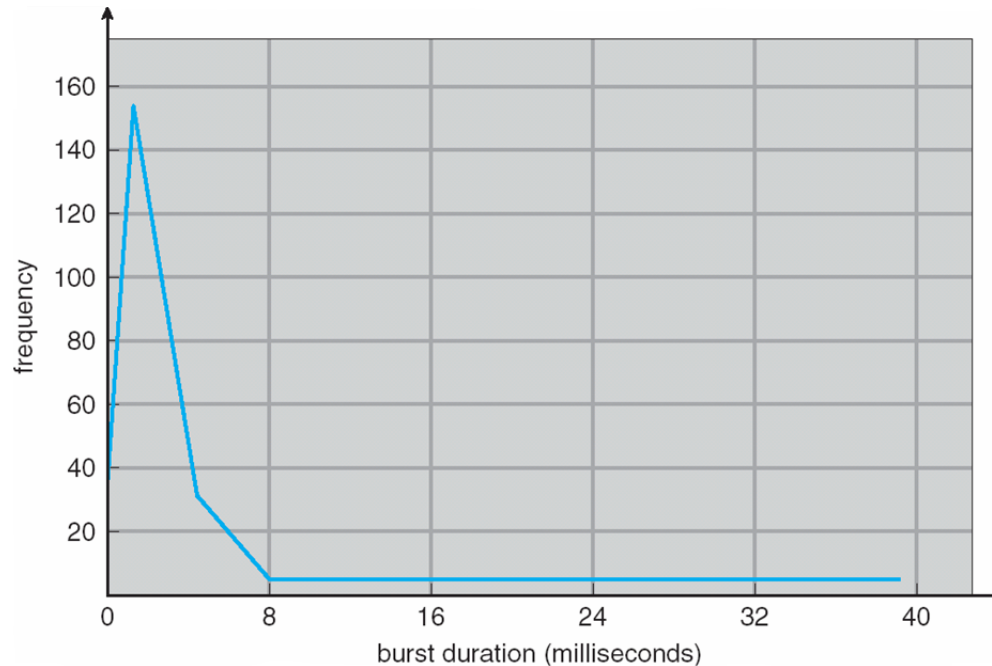
⚠ Para maximizar o uso do CPU, um processo pode ser trocado enquanto espera por I/O

💻 Escalonamento de processos é central ao sistema operativo

# Ciclo CPU/IO



# Histograma de picos de CPU



Tipicamente muitos ciclos de CPU de curta duração..

# Escalonador de CPU

Decisões sobre escalonamento do CPU podem ocorrer quando um processo:

1. Muda de "running" para "waiting" (ex: wait())
2. Muda de "running" para "ready" (ex: interrupt)
3. Muda de "waiting" para "ready" (ex: fim de I/O)
4. Termina (ex: exit(0))

Os pontos 1 e 4 são **não-preemptivos** ou cooperativos. Os restantes são **preemptivos**.

# Preemptivo vs não-preemptivo

## Não-preemptivo ou cooperativo

- Os processos libertam o processador ao terminarem ou passarem para espera (wait)

## Preemptivo

- Os processos libertam o processador por meios externos (interrupts, time-slots, etc.)

# Critérios de escalonamento

- **Utilização de CPU** - manter o CPU ocupado
- **Taxa de transferência** - número de processos que completam a execução por unidade de tempo
- **Tempo de execução** - tempo que um processo demora a executar
- **Tempo de espera** - tempo que um processo aguarda na fila "ready"
- **Tempo de resposta** - tempo desde que um processo foi iniciado até começar a executar



# Algoritmo: First-come, First-Served

Execução por ordem de chegada..

Ex: P1=24ms, P2=3ms, P3=3ms



Tempos de espera: P1=0, P2=24, P3=27

Média =  $(0+24+27)/3 = 17$  ms....

## Algoritmo: First-come, First-Served (2)

Se ordem for P2, P3, P1:



Tempos de espera: P<sub>1</sub>=6, P<sub>2</sub>=0, P<sub>3</sub>=3

Média =  $(6+0+3)/3 = 3$  ms....

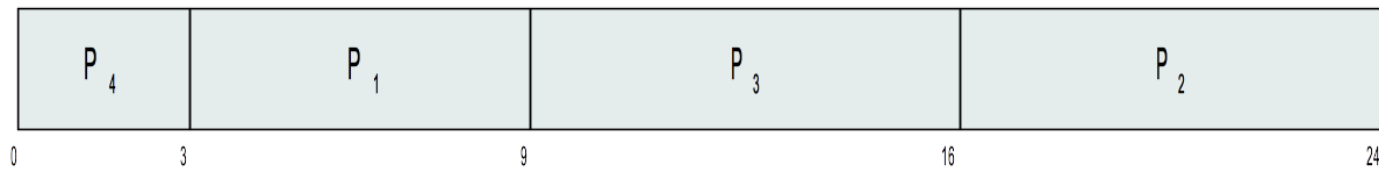


Bastante melhor que os 17ms anteriores!!!

# Algoritmo: Shortest-Job First

Execução por ordem de CPU burst (existem algoritmos de estimativa)..

Ex: P1=6ms, P2=8ms, P3=7ms, P4=3ms



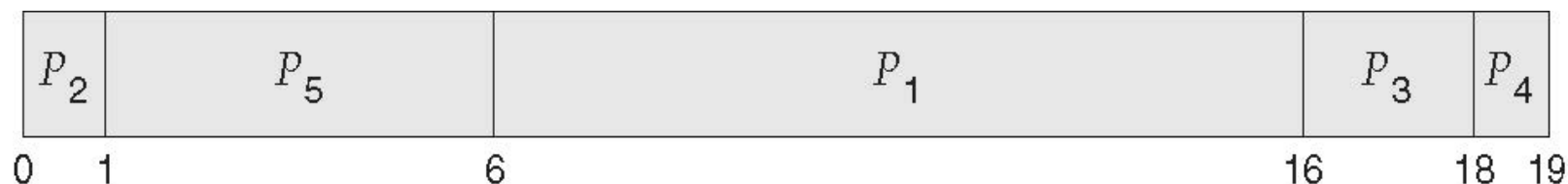
$$\text{Média} = (3+16+9+0)/4 = 7\text{ms}$$

# Algoritmo: Priority scheduling

Execução por ordem de prioridade..

Processo	Burst Time	Prioridade
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

## Algoritmo: Priority scheduling (2)



Média = 8.2ms

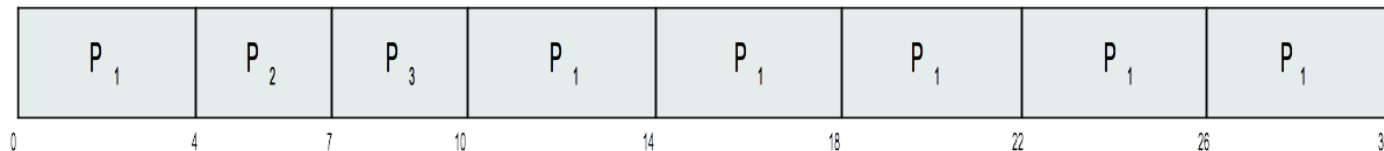
**Problemas:** Fome - processos com prioridade baixa podem nunca ser executados

**Solução:** Idades - aumentar prioridade à medida que o tempo passa..

# Algoritmo: Round-Robin

Algoritmo preemptivo para sistemas com partilha de tempo.

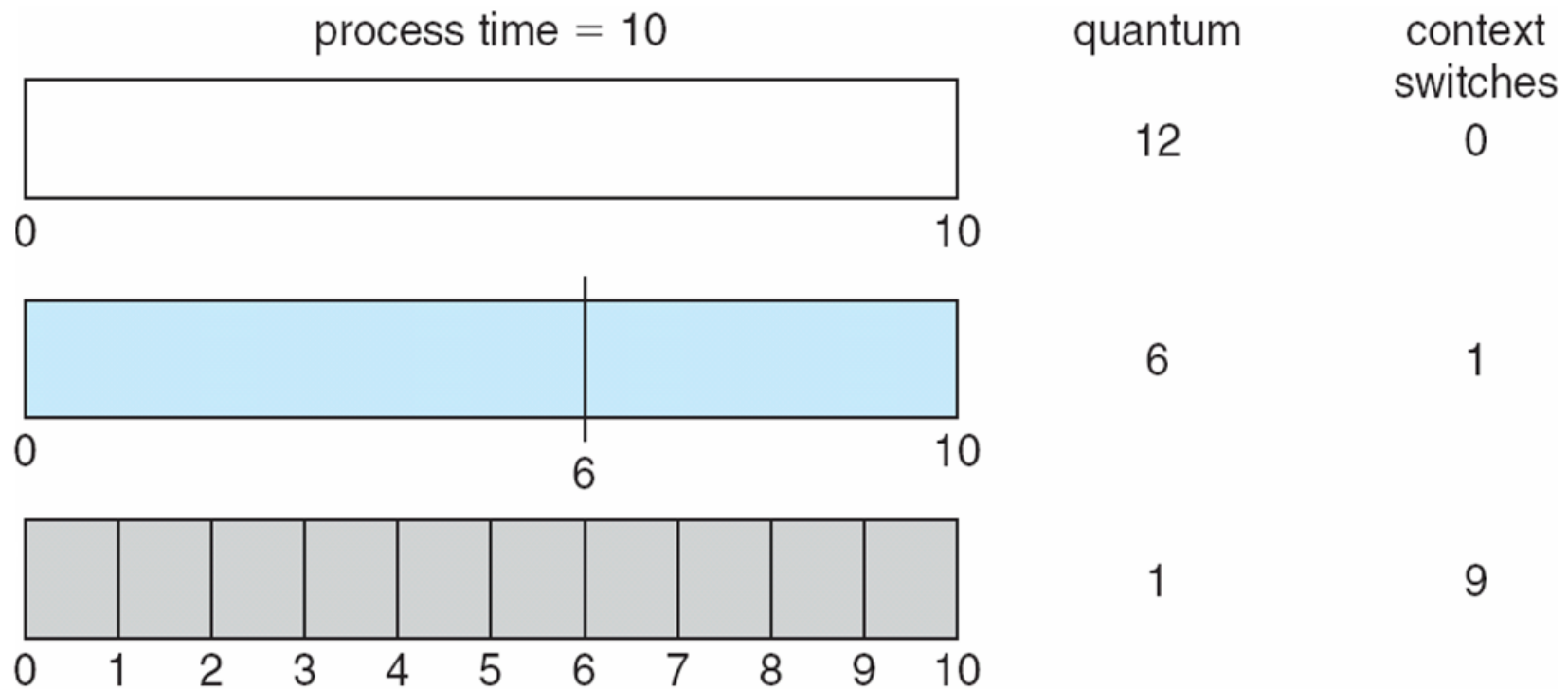
Ex:  $P_1=24\text{ms}$ ,  $P_2=3\text{ms}$ ,  $P_3=3\text{ms}$ , "quantum"= $4\text{ms}$



Tempos de espera:  $P_1=(10-4)=6$ ,  $P_2=4$ ,  $P_3=7$

Média =  $5.66\text{ms}$

# Time quantum & context-switches



# Outros algoritmos

## Multilevel Queue Scheduling

- Para processos em diferentes grupos
- Ex: processos interactivos vs background

## Multilevel Feedback Queue Scheduling

- Semelhante ao anterior
- Permite que processos transitem entre filas de espera

## Sistemas em Tempo Real

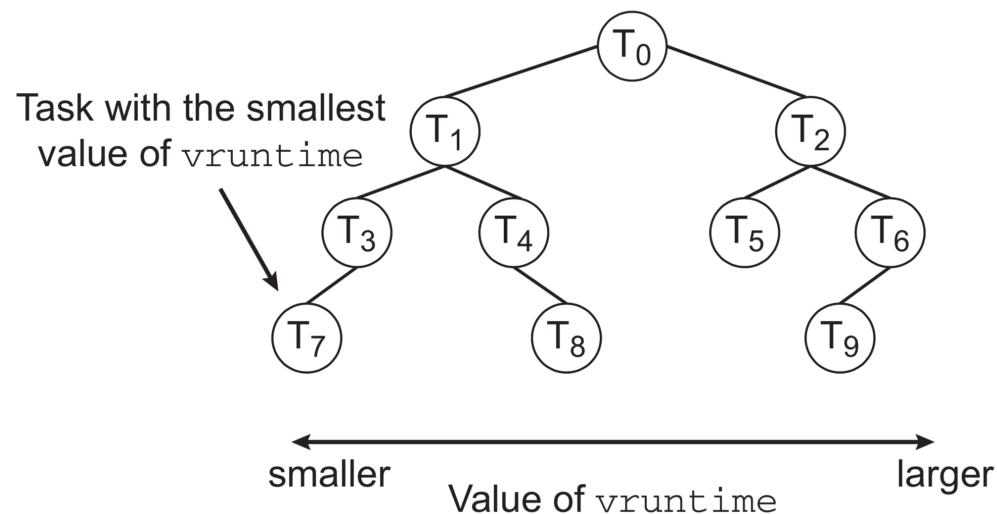
- Minimizar latência
- Processos devem cumprir requisitos em termos de tempos



# Linux: escalonamento de processos

CFS (Completely Fair Scheduler - kernel 2.6.23):

- Preemptivo, baseado em prioridades



 vruntime = tempo de execução até ao momento

# Windows: escalonamento de processos

- Preemptivo, baseado em prioridades
- 6 classes de prioridades
  - IDLE\_PRIORITY\_CLASS
  - BELOW\_NORMAL\_PRIORITY\_CLASS
  - NORMAL\_PRIORITY\_CLASS
  - ABOVE\_NORMAL\_PRIORITY\_CLASS
  - HIGH\_PRIORITY\_CLASS
  - REALTIME\_PRIORITY\_CLASS
- 7 níveis de prioridade relativa
  - IDLE; LOWEST; BELOW\_NORMAL; NORMAL; ABOVE\_NORMAL; HIGHEST; TIME\_CRITICAL

# Windows: escalonamento de processos (2)

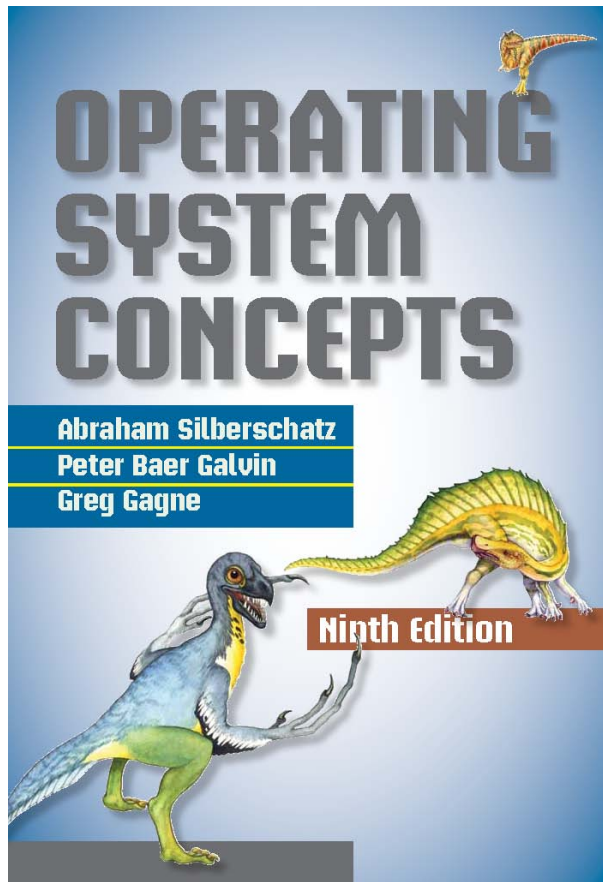
Prioridades calculadas a partir das classes e níveis relativos.

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

**Quiz...**

# Sumário

- Escalonamento é a escolha do próximo processo a ser executado
- First-come first-served é o mais simples.
- Restantes tentam melhorar.
- Round-Robin é mais apropriado para sistemas de partilha de tempo.
- Time-quantum pode dar origem a muitos context-switches.



Ler capítulo 6...