Gerenciamento de processos

Davidson Rodrigo Boccardo

flitzdavidson@gmail.com

Revisão

- Critérios de alocação:
  - Utilização da CPU
  - Produtividade (*Throughput*)
    - Número de processos finalizados por unidade de tempo
  - Tempo de entrega (*Turnaround time*)
    - Tempo que o processo leva para ser executado
  - Tempo de espera
    - Tempo que o processo fica aguardando na fila de prontos
  - Tempo de resposta
  - Intervalo de tempo entre o envio de uma requisição e a primeira resposta a essa requisição

Revisão

- Objetivo
  - Maximizar a utilização da CPU
  - Maximizar a produtividade (*throughput*)
  - Minimizar o tempo de entrega
  - Minimizar o tempo de espera
  - Minimizar o tempo de resposta

Algoritmos de escalonamento

- **First-Come, First-Served (FCFS)**
- **Shortest-Job-First (SJR)**
- **Prioridade**
- **Round-Robin (RR)**
- **Filas Multinível**

First-Come, First-Served (FCFS)

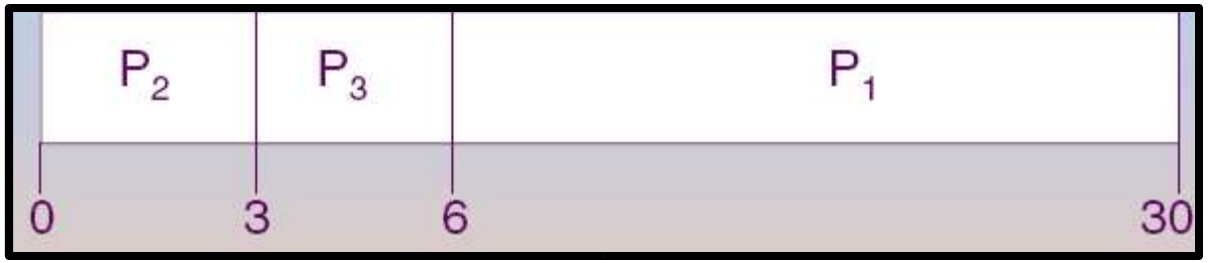
Process	Burst Time
P_1	24
P_2	3
P_3	3

- **Ordem de chegada: P_1, P_2, P_3**

- **Tempo de espera médio?**
  - **$P_1 = 0$; $P_2 = 24$; $P_3 = 27$**
- **Tempo de espera médio: (0+24+27)/3 = 17**

First-Come, First-Served (FCFS)

- Suponha agora a ordem de chegada: P_2, P_3, P_1



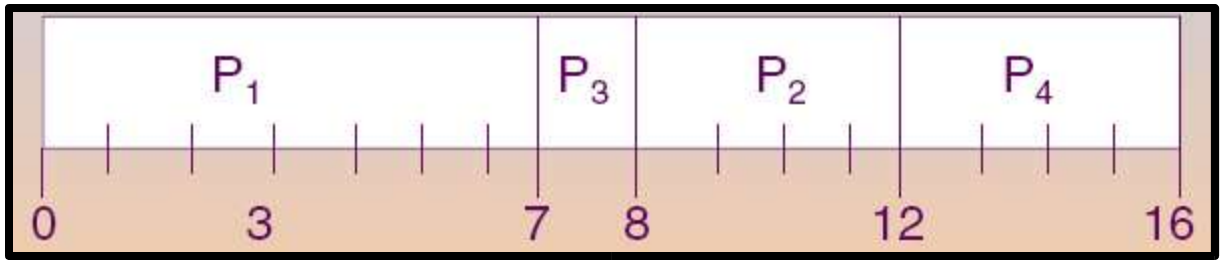
- Tempo de espera médio?
  - $P_1 = 6$; $P_2 = 0$; $P_3 = 3$
- Tempo de espera médio: (6+0+3)/3 = 3
- É um algoritmo não-preemptivo.
- Problemas do FCFS?

Shortest-Job-First (SJF)

- **Associa a cada processo a duração da próxima fase de uso da CPU (CPU burst)**
- **Escalona sempre o processo de menor duração da próxima fase de uso da CPU. Empate?**
- **Duas maneiras:**
  - **Não preemptiva**
  - **Preemptiva: se novos processos chegam com CPU burst menores que o tempo restante do processo em execução, este é tirado a força. Este algoritmo é conhecido como Shortest-Remaining-Time-First (SRTF).**
- **SJF é ótimo – dá um tempo de espera médio mínimo para um dado conjunto de processos**

Exemplo SJF não preemptivo

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

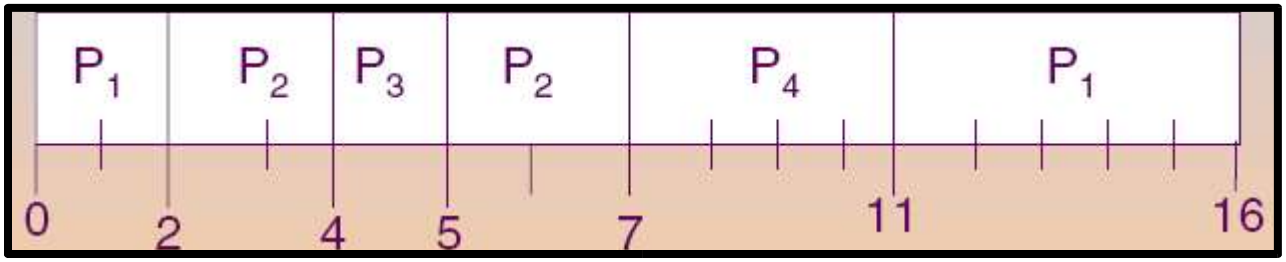


- **Tempo de espera médio?** Problema?
  - **(0+6+3+7)/4 = 4**

Exemplo SJF preemptivo (SRTF)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- **SJF (preemptivo) ou SRTF (Shortest-Remaining-Time-First)**



- **Tempo de espera médio?**
  - **(9+1+0+2)/4 = 3**

Já que este algoritmo ótimo
qual é o problema deste algoritmo?

Shortest-Job-First (SJF)

- **Como determinar o tamanho do próximo CPU burst?**
- **Pode ser feita uma estimativa, através dos CPU bursts anteriores, usando média exponencial**

1. t_n = valor do n^{th} CPU burst

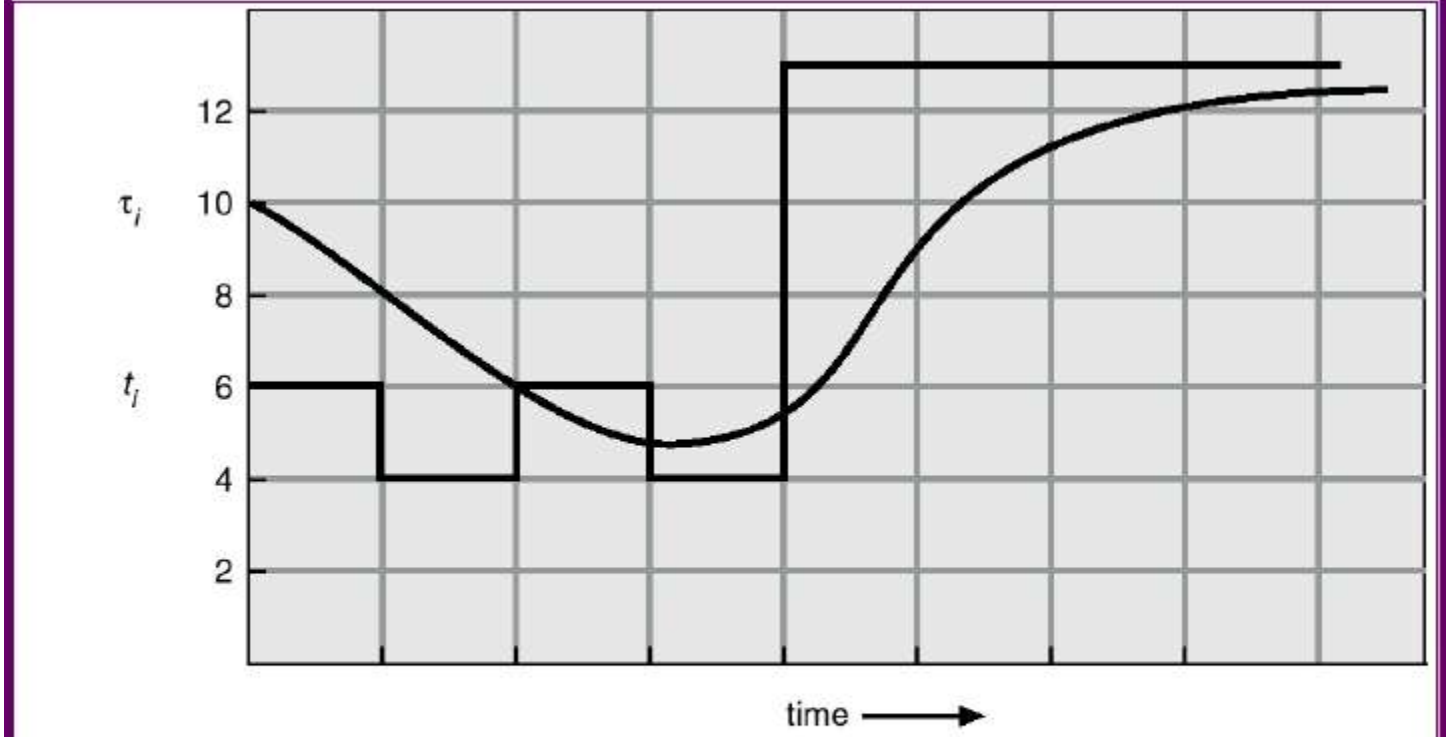
. $_{n+1}$ = valor previsto para o próximo CPU burst

3. , 0 $\leq$ $\leq$1

4. Define:

$$_{n+1} = t_n + (1 -) \ _n$$

Previsão do próximo CPU burst



CPU burst (t_i)	6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

Prioridade

- Uma prioridade (inteiro) é associada a cada processo
- A CPU é alocada para o processo com maior prioridade (menor inteiro, maior prioridade)
- SJF é um escalonamento por prioridade cuja prioridade é o tempo previsto do próximo CPU burst
- Duas maneiras:
  - Não preemptiva
  - Preemptiva
- Problemas?
  - Starvation - Processos com menor prioridade nunca executam
- Solução? Aging (Envelhecimento)
  - Aumento da prioridade dos processos com o passar do tempo

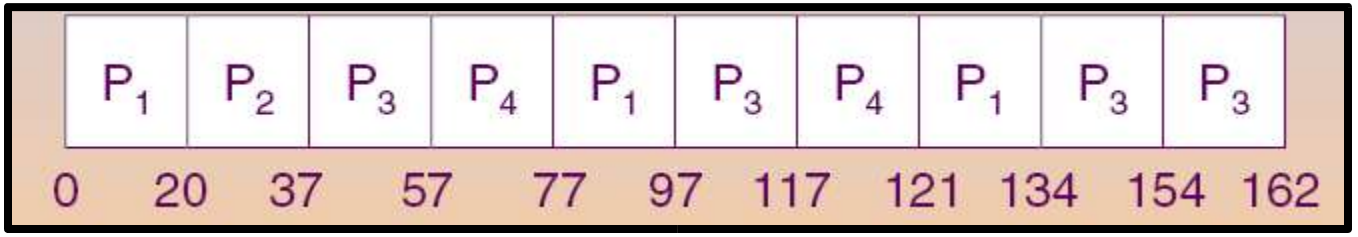
Round Robin (RR)

- Cada processo tem uma fatia de tempo (quantum time)
- Similar a um FCFS, porém com preempção
- Pergunta: Se existir *n* processos na fila de prontos e o quantum time for *q*, qual é o máximo tempo de espera?
- Desempenho
  - *q* grande FIFO
  - *q* pequeno alto overhead devido as mudanças de contexo

Exemplo Round Robin

- **Quantum time = 20**

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

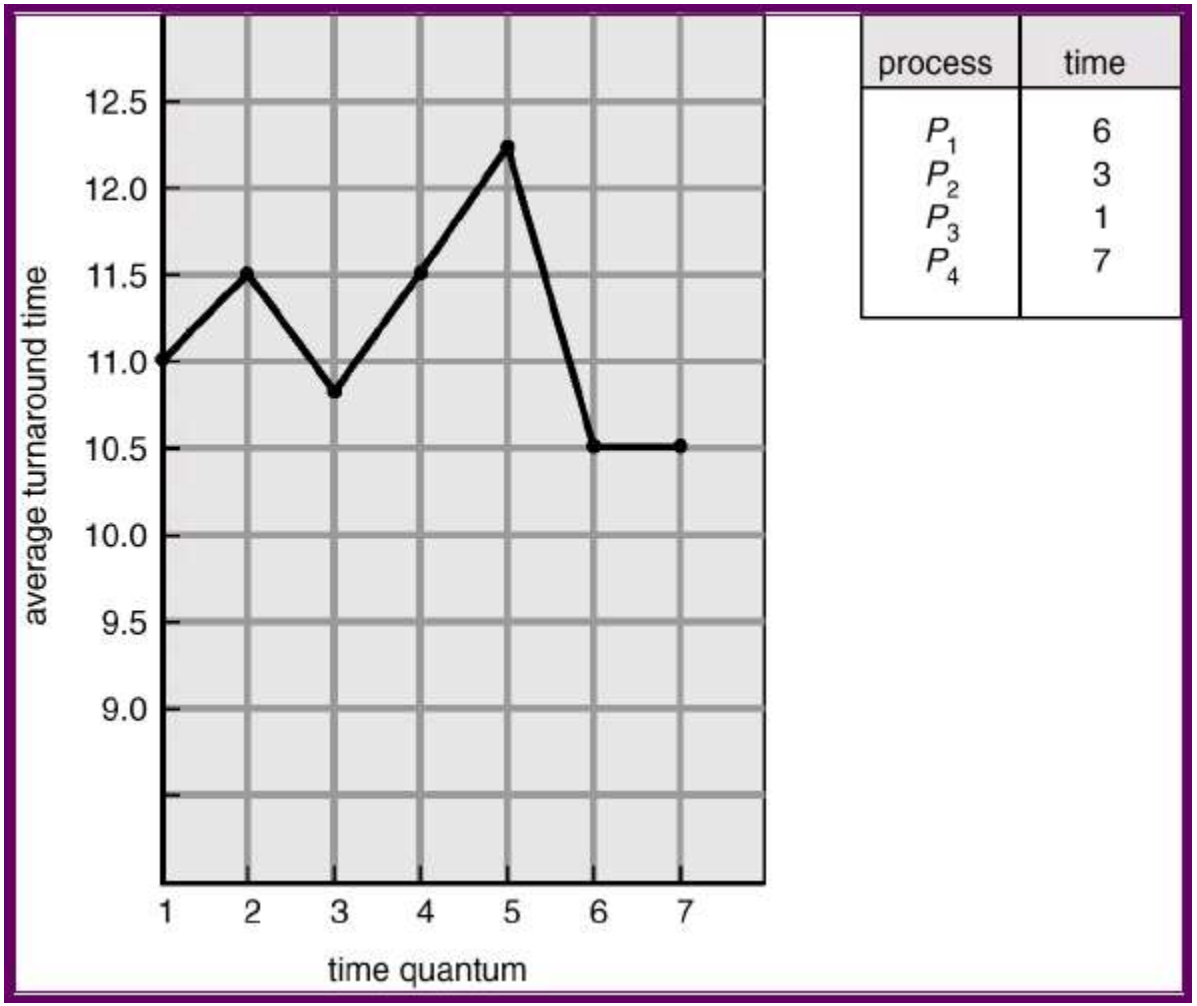


- **Geralmente, maior tempo de entrega médio do que o SJF, mas melhor resposta**

Quantum x nº trocas de contexto

quantum	context switches
12	0
6	1
1	9

Variação do tempo de entrega médio x quantum

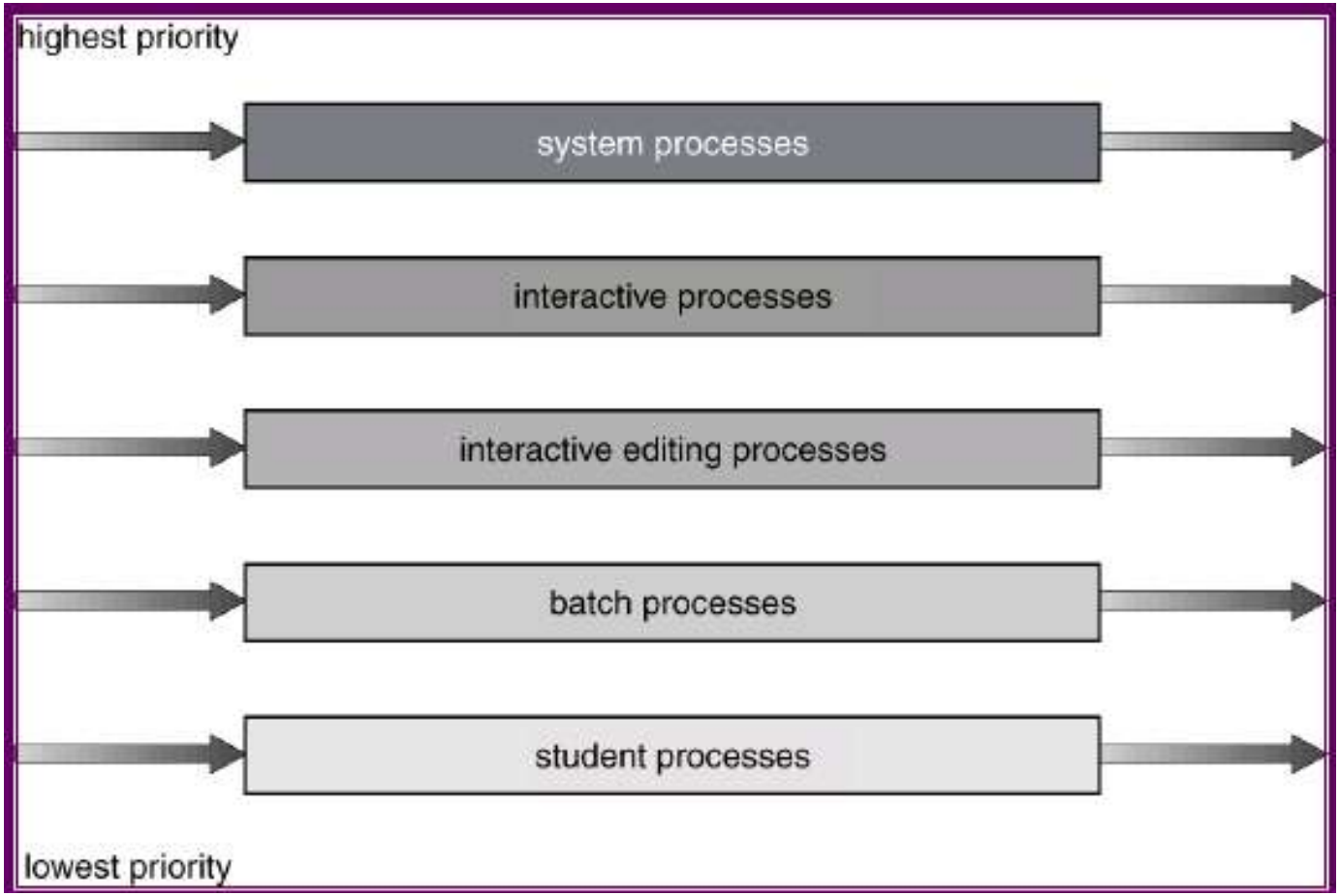


process	time
P_1	6
P_2	3
P_3	1
P_4	7

Filas Multinível

- Fila de prontos é dividida de acordo com a classificação dos processos:
  - Processos interativos (execução em primeiro plano)
  - Processos não interativos (execução em segundo plano)
- Cada fila pode ter difentes algoritmos:
  - Primeiro Plano – RR
  - Segundo Plano – FCFS
- Deve existir escalonamento entre as filas
  - Geralmente, alocação preemptiva com prioridades fixas. Ex. Todos processos foreground depois os background. Problema?
  - Fatias de tempo – cada fila tem uma certo montante de tempo de CPU para escalonamento de seus processos.

Escalonamento em Filas Multinível



Filas Multinível de retorno

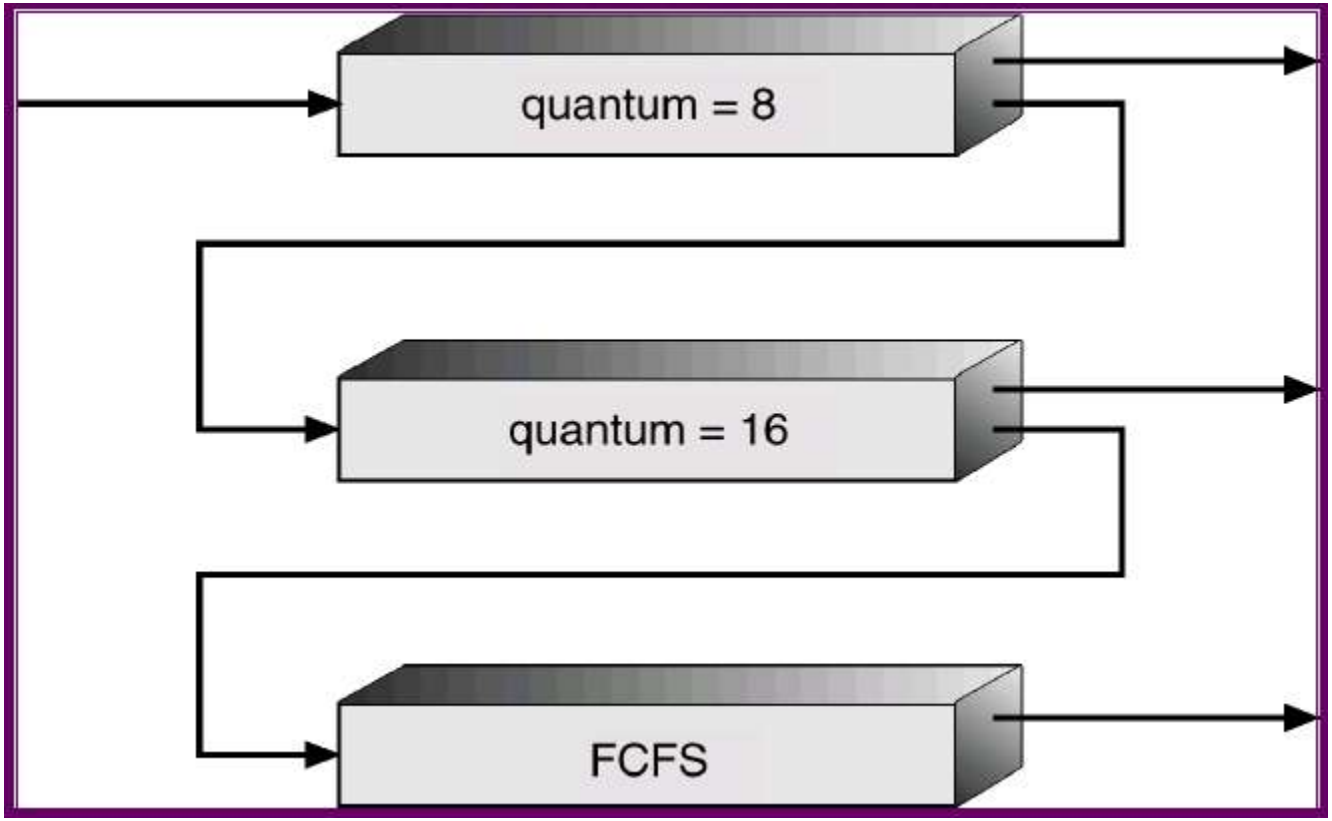
- Um processo pode mover entre as filas
  - Técnica aging pode ser implementado desta maneira
- Parâmetros?
  - Número de filas
  - Algoritmo de alocação da CPU usado para cada fila
  - Método usado para determinar quando transferir um processo para uma fila de prioridade mais alta
  - Método usado para determinar quando transferir um processo para uma fila de prioridade mais baixa
  - Método usado para determinar em qual fila um processo deve ser colocado, quando precisar usar a CPU

Exemplo Filas Multinível

- **Três filas:**
  - **Q_0 – time quantum 8 milisegundos**
  - **Q_1 – time quantum 16 milisegundos**
  - **Q_2 – FCFS**

- **Escalonamento**
  - **Um novo processo entra na fila Q_0 e espera ser escalonado pela CPU, se demorar mais que 8 milisegundos o processo é movido para a fila Q_1;**
  - **Quando não houver processos na fila Q_0, o processo terá mais 16 milisegundos, caso não termine neste tempo é movido para a fila Q_2 que é FCFS.**

Exemplo Filas Multinível



Dúvidas?