

# Sistemas Operacionais

## *Escalonamento de Processos*

**Edeyson Andrade Gomes**

[www.edeyson.com.br](http://www.edeyson.com.br)

# Roteiro

---

## ▶ Escalonamento de Processos

### ▶ Metas

### ▶ Algoritmos

#### ▶ FIFO

#### ▶ SJF

- Menor Job Primeiro

#### ▶ SJFP

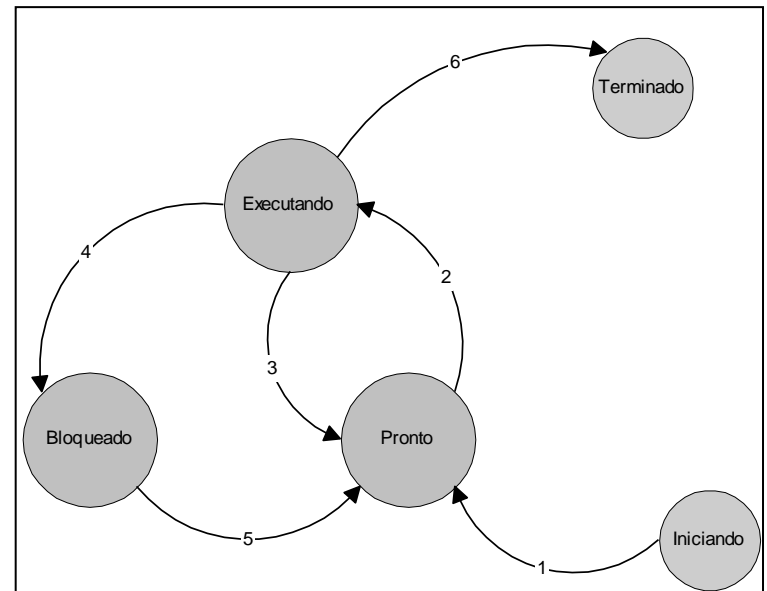
- Menor Job Primeiro Preemptivo

#### ▶ Round Robin

# Definição

## ▶ Algoritmo de Escalonamento de CPU

- ▶ Algoritmo do S.O. que determina qual o próximo processo a ocupar a CPU
  - ▶ Executado quando ocorre estouro de Quantum ou interrupção do processo (I/O, Evento, Sinal, etc.) ou o processo acaba
    - Transições 3, 4 e 6



# Escalonador de Processos

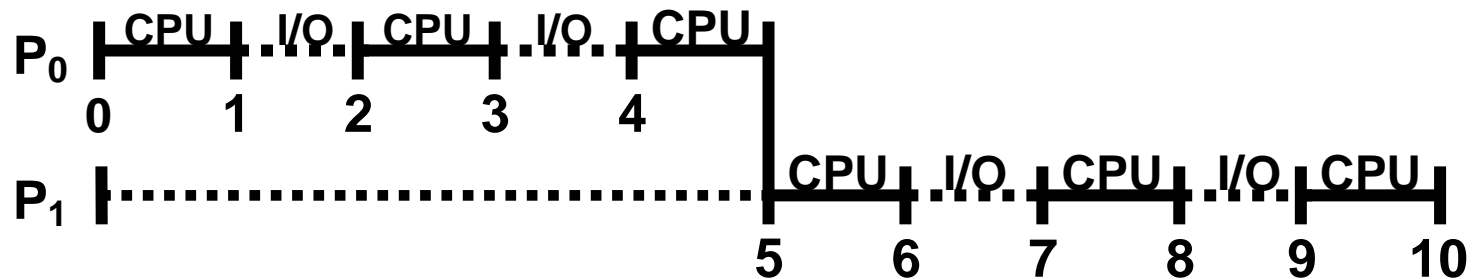
---

- ▶ **Sistema Multiprogramado ou Multiprocessado**
  - ▶ Processos no estado de Pronto concorrem pela CPU
    - ▶ SO necessita de critério de escolha dos processos para execução
      - Política de Escalonamento
- ▶ **Critérios mudam com características dos Processos**
  - ▶ Batch
    - ▶ Processos em lote, normalmente não interativos
  - ▶ Intensivamente consumidores de CPU
  - ▶ Intensivamente consumidores de I/O
  - ▶ Híbridos (revezam consumindo CPU e I/O)

# Escalonador de Processos

## ▶ Sem multiprogramação

- ▶ Suponha a execução de 2 processos ( $P_0$  e  $P_1$ )
  - ▶ Uso de CPU – linha contínua
  - ▶ Bloqueio aguardando I/O – linha pontilhada
- ▶ Tempo Total de Execução = 10 unidades de tempo (unidade de tempo - ut)
- ▶ **Throughput** = 0,2 p/ut (Nº Processos Executados por ut)
  - ▶  $\Delta t P_0 = 5ut$        $\Delta t P_1 = 10ut$       ( $\Delta t =$  Tempo Total)
- ▶ **Tempo médio de execução** = 7,5 ut =  $(\Delta t P_0 + \Delta t P_1) / 2$
- ▶ **Utilização da CPU** = 60 % (Desprezando-se tempo de Kernel)
  - ▶ 40% de I/O

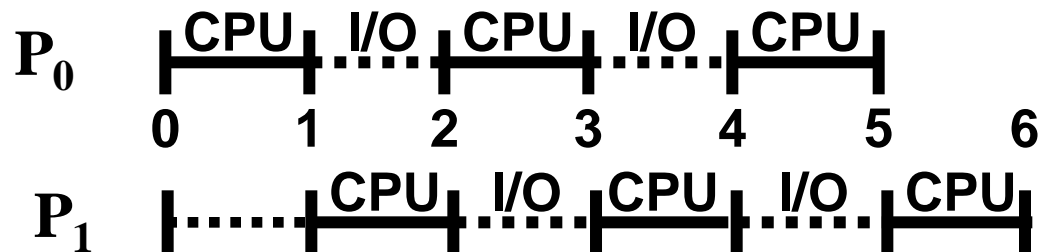


# Escalonador de Processos

---

## ▶ Com multiprogramação

- ▶ Suponha a execução dos mesmos 2 processos ( $P_0$  e  $P_1$ )
- ▶ Tempo Total de Execução = 6 ut
- ▶ **Throughput** = 0,33
  - ▶  $\Delta t P_1 = 5ut$        $\Delta t P_2 = 6ut$       ( $\Delta t =$  Tempo Total)
- ▶ **Tempo médio de execução** = 5,5 ut
- ▶ **Utilização da CPU** = 100 %
  - ▶ Desprezando-se tempo de Kernel
- ▶ Note que enquanto  $P_0$  está bloqueado (fora da CPU)  $P_1$  executa, e vice-versa.



# Metas do Escalonamento

---

## ▶ Eficiência

- ▶ Manter a CPU ocupada 100% do tempo

## ▶ Throughput

- ▶ Maximizar o número de processos (tarefas, jobs) executados em um dado intervalo de tempo

## ▶ Turnaround

- ▶ Minimizar o tempo de um processo no sistema, desde seu início até o término
  - ▶ Tempo médio de execução
  - ▶ Fundamental a processos Batch

# Metas do Escalonamento

---

- ▶ **Igualdade**

- ▶ Todo Processo tem direito de ocupar a CPU

- ▶ **Tempo de resposta**

- ▶ Minimizar o tempo decorrido entre a submissão de um pedido e a resposta produzida num processo interativo



# Conflito entra Metas

---

- ▶ **Atender a uma meta pode prejudicar outra**
  - ▶ Qualquer algoritmo de escalonamento favorecerá um tipo de processo (CPU Bound, I/O Bound, Tempo Real, etc) em detrimento de outros
    - ▶ **Propósito Geral**

# Tipos de Escalonamento

---

- ▶ **Dois tipos:**
  - ▶ Escalonamento não-preemptivo;
  - ▶ Escalonamento preemptivo.

# Escalonamento

---

## ▶ Conceitos Básicos

- ▶ Multiprogramação visa maximizar a utilização da CPU
- ▶ Processos têm surtos de CPU e I/O

# Escalonamento

---

- ▶ Escalonador de CPU ou de Curto Prazo
  - ▶ Seleciona processo pronto para CPU não ficar ociosa
  - ▶ Algoritmo para seleção do processo pronto
    - ▶ FIFO, com prioridade, árvore, etc

# Escalonamento

---

- ▶ Escalonador de CPU (Curto Prazo ou Baixo Nível)
  - ▶ Transições de Estado para processos na Memória
  - ▶ Acionamento do escalonador
    1. Processo em execução para bloqueado/espera
    2. Processo em execução para pronto
    3. Processo em execução termina
      - ➔ CPU Livre
    1. Processo em espera para pronto
    2. Processo de Pronto para Execução

# Escalonamento

---

- ▶ Escalonador de CPU ou de Curto Prazo
  - ▶ Escalonamento Não-Preemptivo
    - ▶ Escalonamento Cooperativo
    - ▶ Processo mantém a CPU até terminar, executar um I/O ou ocorrer uma interrupção no sistema
    - ▶ Não requer recursos especiais de hardware
    - ▶ Usado até o Windows 95
    - ▶ Não existe Quantum
      - Devolução voluntária do controle ao S.O.

# Escalonamento

---

- ▶ Escalonador de CPU ou de Curto Prazo
  - ▶ Escalonamento Preemptivo
    - ▶ Requer temporizador na CPU
      - Fatia de Quantum
      - Uso do Clock
    - ▶ Requer suporte do SO para coordenar acesso a dados compartilhados de forma consistente
      - Proteção

# Escalonamento

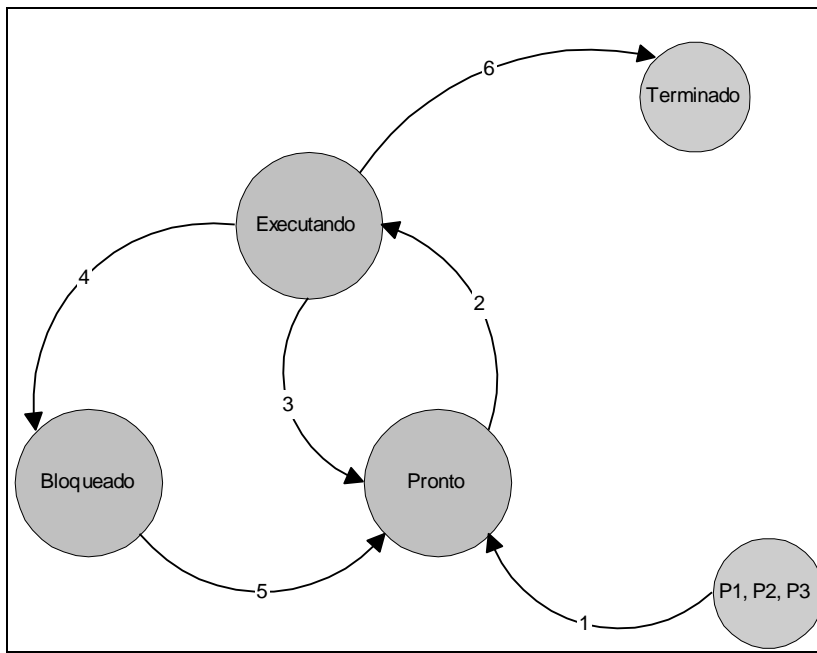
---

- ▶ **Dispatcher e Latência**
  - ▶ P<sub>x</sub> Estoura tempo de Quantum
  - ▶ Troca de contexto
  - ▶ Interrupção de Clock
    - ▶ Firmware
      - Modo Kernel (instruções privilegiadas)
      - SO
    - ▶ Mudança do modo de operação para Usuário
  - ▶ Reinício do programa na posição correta



# Escalonamento FIFO

- ▶ First Come First Served (FCFS, FIFO, PEPS)
  - ▶ Não preemptivo



Processo	Início	Duração (ut)
P <sub>1</sub>	0	24
P <sub>2</sub>	0	3
P <sub>3</sub>	0	3

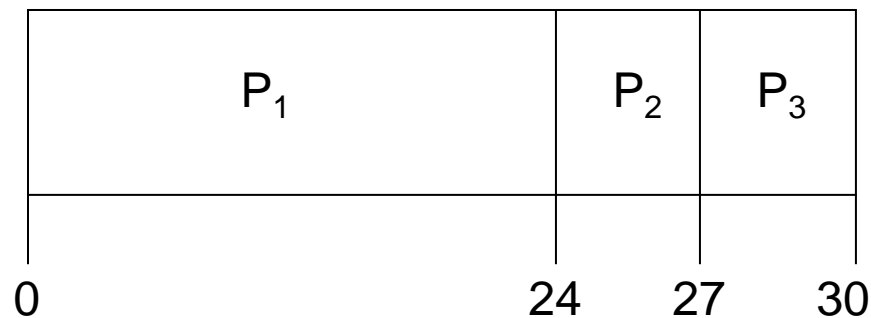
# Escalonamento FIFO

---

- ◆ **Ordem de chegada dos processos:**

$P_1, P_2, P_3$

- ◆ **Diagrama de Gantt**



# Escalonamento FIFO

---

- ▶ Tempos de espera

$$P_1 = 0$$

$$P_2 = 24$$

$$P_3 = 27$$

Dica: Tempo de Espera é o tempo que o processo passa no estado de Pronto.

- ▶ Tempo médio de espera

$$(0 + 24 + 27) / 3 = 17$$

$$\text{Throughput} = 0,1 \text{ (3/30)}$$

# Escalonamento FIFO

---

▶ Tempos de saída

$$P_1 = 24$$

$$P_2 = 27$$

$$P_3 = 30$$

▶ Tempo médio de saída

$$(24 + 27 + 30) / 3 = 27$$

$$\text{Throughput} = 0,1 \text{ (3/30)}$$

# Escalonamento FIFO

---

- ▶ Outra ordem de chegada

$P_2, P_3, P_1$

- ▶ Diagrama de Gantt



# Escalonamento FIFO

---

- ▶ **FIFO ordenado (SJF / MPP)**
  - ▶ Menor Processo Primeiro
    - ▶ Menor = menor tempo de execução
- ▶ **Tempos de espera**  
 $TEP_1 = 6; \quad TEP_2 = 0; \quad TEP_3 = 3$
- ▶ **Tempo médio de espera melhora**  
 $(6 + 0 + 3) / 3 = 3$
- ▶ **Tempo médio de espera não é mínimo**
  - ▶ Pode variar muito (com os surtos de CPU)
- ▶ **Efeito Comboio**
  - ▶ Processos I/O bound esperam por CPU bound

# Escalonamento FIFO

---

- ▶ Tempos de saída

$$P_1 = 30; \quad P_2 = 3; \quad P_3 = 6$$

- ▶ Tempo médio de saída melhora

$$(30 + 3 + 6) / 3 = 13$$

- ▶ Tempo médio de saída não é mínimo

- ▶ Pode variar muito (com os surtos de CPU)

# Escalonamento SJF

---

- ▶ Shortest-Job-First (Menor Job Primeiro)
  - ▶ Deveria ser “próximo surto de CPU menor primeiro”

Usado para Processos Batch.

Sua execução diária permite determinar seu tempo total.

PID	Início	Duração de surto
P <sub>1</sub>	0	6
P <sub>2</sub>	0	8
P <sub>3</sub>	0	7
P <sub>4</sub>	0	3

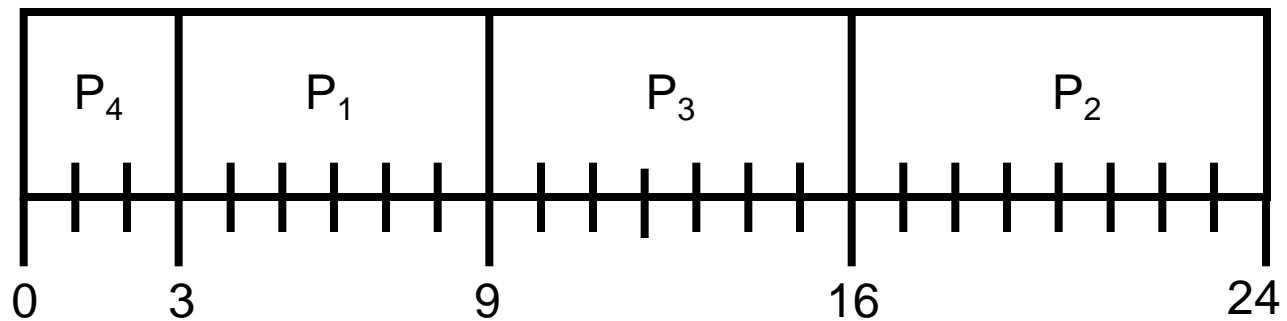


# Escalonamento SJF

---

## ◆ Tempos de espera

$$P_1 = 3; \quad P_2 = 16; \quad P_3 = 9; \quad P_4 = 0$$



# Escalonamento SJF

---

- ◆ **Tempo médio de espera melhora**

$$(3 + 16 + 9 + 0) / 4 = 7$$

Para FIFO, nesta situação, seria  $10,25 = (0 + 6 + 14 + 21) / 4$

- ◆ **Tempo médio de espera é mínimo**

- Algoritmo considerado *ótimo*

# Exercício com SJF

---

Processo	Instante de chegada (hora de agendamento)	Duração de surto (tempo total de execução)
$P_1$	0	7
$P_2$	2	4
$P_3$	4	1
$P_4$	5	3

# Escalonamento SJF

---

- ▶ **Problema: determinação da duração do próximo surto de CPU é impossível**
  - ▶ SJF é usado para escalonamento de jobs em sistemas batch
    - ▶ Usuário especifica o tempo de CPU do job
  - ▶ Em escalonamento de CPU é usada estimativa
    - ▶ Baseada na duração dos surtos anteriores
      - Média exponencial

# Preempção em SJF

---

## ▶ Não preemptivo

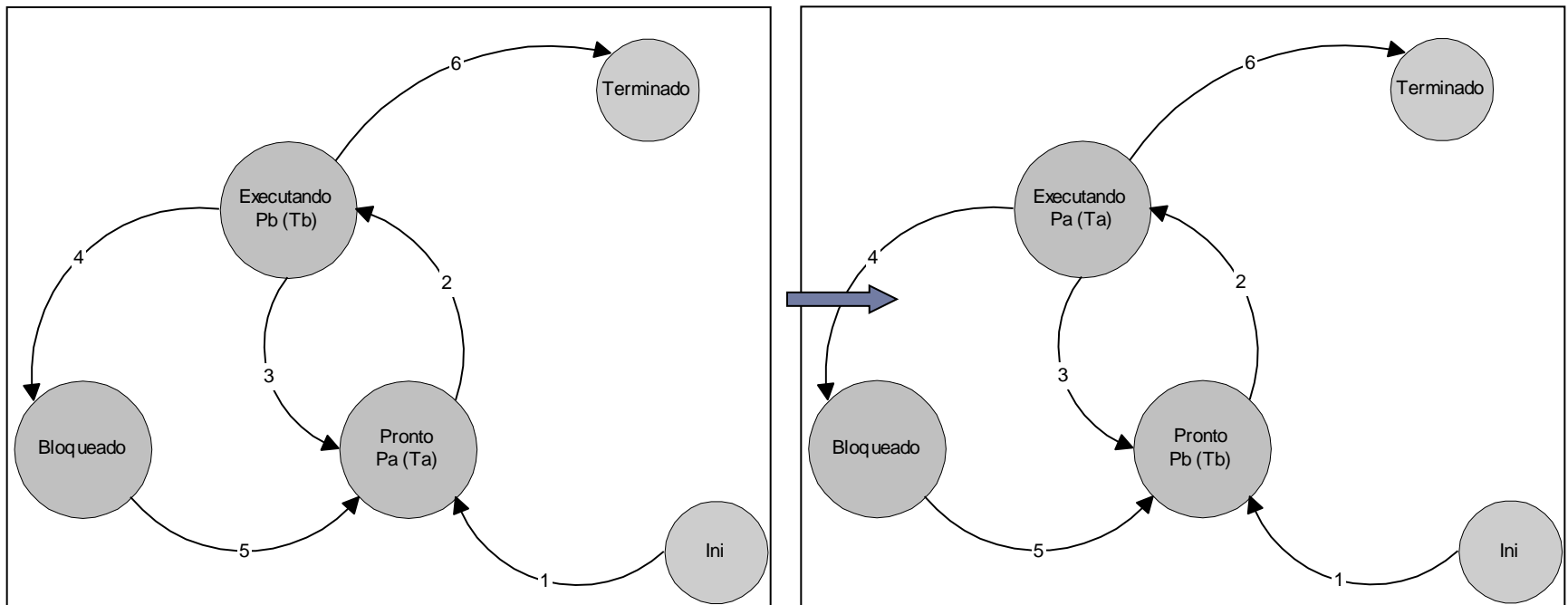
- ▶ Processo usa CPU até completar surto

## ▶ Preemptivo

- ▶ Novo processo pronto com surto previsto ( $T_A$ )
- ▶ Tempo restante previsto para o processo em execução ( $T_B$ )
- ▶ Se  $T_A < T_B \Rightarrow$  preempção por prioridade
- ▶ Shortest-Remaining-Time-First (SRTF)

# Preempção em SJF

$T_a < T_b$



# Preempção em SJF

---

Processo	Instante de chegada	Duração de surto
$P_1$	0	7
$P_2$	2	4
$P_3$	4	1
$P_4$	5	4

# Preempção em SJF

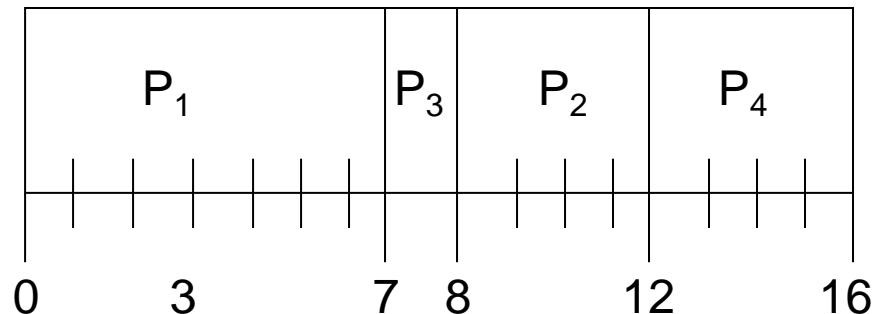
---

## ▶ SJF não preemptivo

▶ Tempo de espera médio =  $(0 + 6 + 3 + 7) / 4 = 4$

▶  $TEP_1 = 0$                        $TEP_2 = 6$  (8 - 2)

▶  $TEP_3 = 3$                        $TEP_4 = 7$





# Preempção em SJF

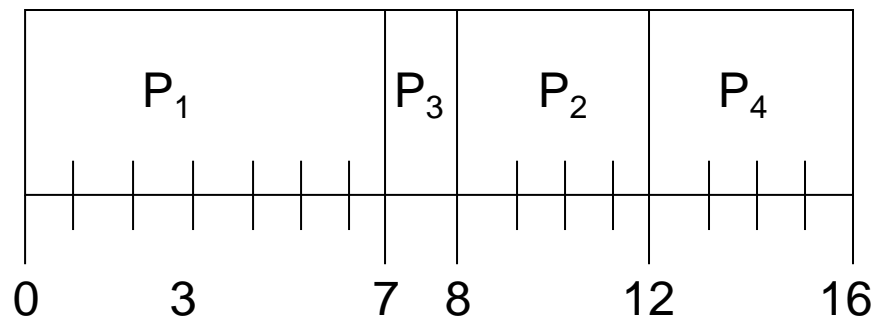
---

## ▶ SJF não preemptivo

▶ Tempo de saída médio =  $(7 + 10 + 4 + 11) / 4 = 8$

▶  $TSP_1 = 7$                        $TSP_2 = 10$

▶  $TSP_3 = 4$                          $TSP_4 = 11$



# Preempção em SJF

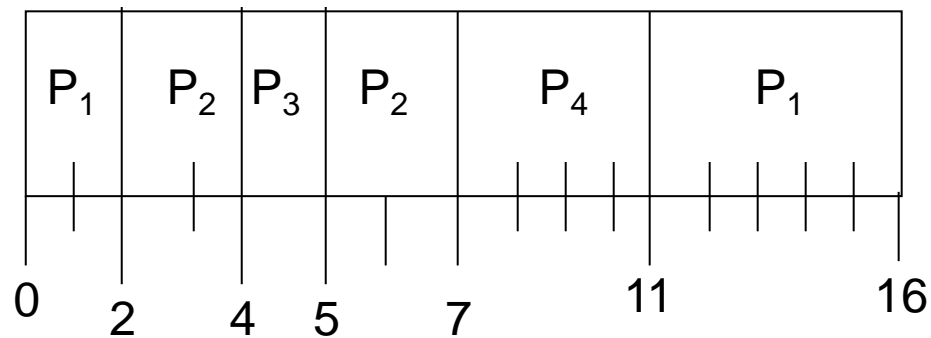
---

## ▶ SJF preemptivo

▶ Tempo de espera médio =  $(9 + 1 + 0 + 2) / 4 = 3$

▶  $TEP_1 = 9$                        $TEP_2 = 1$

▶  $TEP_3 = 0$                        $TEP_4 = 2$



# Preempção em SJF

---

## ▶ SJF preemptivo

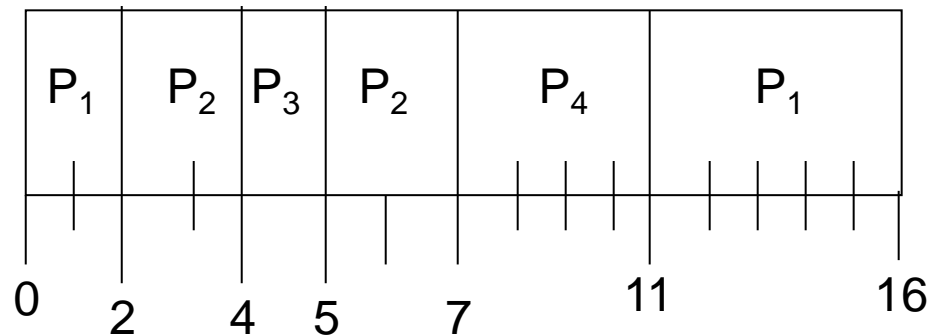
▶ Tempo de saída médio =  $(16 + 5 + 1 + 6) / 4 = 7$

▶  $TSP_1 = 16$

$TSP_2 = 5$

▶  $TSP_3 = 1$

$TSP_4 = 6$



# Escalonamento Round Robin

---

- ▶ Round-Robin (revezamento circular)
  - ▶ Sistema Preemptivo
    - ▶ Interrupção do Clock (existe Quantum)
  - ▶ Tempo de espera médio é longo
  - ▶ Tempo de saída maior que SJF
  - ▶ Tempo de resposta melhor que SJF

# Escalonamento Round Robin

---

## ▶ Preemptivo

- ▶ Quantum de tempo (10 ~ 100 ms)
  - ▶ Necessita temporizador
- ▶ Fila circular de processos prontos
- ▶ Com quantum  $q$  e  $n+1$  processos prontos:
  - ▶ Tempo máximo de espera:  $n*q$

# Escalonamento Round Robin

---

- ▶ Com quantum  $q$  e  $n+1$  processos prontos:
  - ▶ Tempo máximo de espera:  $n*q$
- ▶ Suponha uma fila de pronto com 101 processos, Quantum de 100 ms
- ▶ Um processo interativo:
  - I. executa;
  - II. faz uma requisição;
  - III vai para bloqueado;
  - IV. de lá, vai para o fim da fila de pronto.
  - ▶ Quando a resposta será entregue ao usuário do processo interativo?

# Escalonamento Round Robin

---

Processo	Início	Duração de surto
P <sub>1</sub>	0	24
P <sub>2</sub>	0	3
P <sub>3</sub>	0	3

Exemplo com quantum de 4 UT

TEP1 = 6    TEP2 = 4    TEP3 = 7

Tempo médio de espera:  $17 / 3 = 5,66$  UT

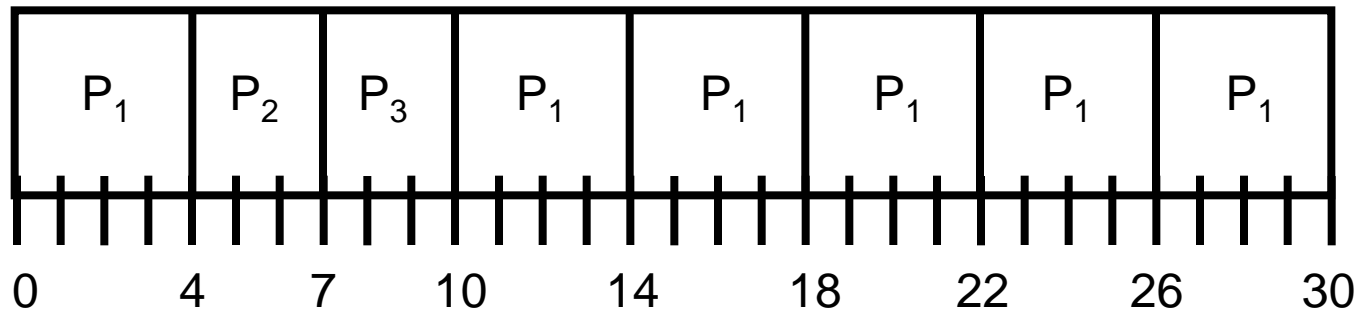
# Escalonamento Round Robin

Processo	Início	Duração de surto
P <sub>1</sub>	0	24
P <sub>2</sub>	0	3
P <sub>3</sub>	0	3

Exemplo com quantum de 4 ms

TEP1 = 6    TEP2 = 4    TEP3 = 7

Tempo médio de espera:  $17 / 3 = 5,66$  ms





# Escalonamento Round Robin

---

## ▶ Desempenho

### ▶ Depende do quantum (q)

▶ q grande  $\Rightarrow$  FCFS/FIFO (Fila)

▶ q pequeno  $\Rightarrow$  Compartilhamento de processador

### ▶ Efeito da troca de contexto

▶ q deve ser grande em relação ao tempo da troca de contexto para evitar aumento de overhead

# Escalonamento Round Robin

---

- ▶ Tempo de retorno x Tamanho do quantum
  - ▶ Tempo de retorno não melhora sempre com aumento do quantum
  - ▶ Há melhora quando processos acabam com surto de Iq
    - » Exemplo: 3 processos que necessitam de 10 ms para executar:
      - ◆ Quantum 1 ms  $\Rightarrow$  tempo de saída médio 29 ms
      - ◆ Quantum 10 ms  $\Rightarrow$  tempo de saída médio 20 ms
      - ◆ Sem considerar tempo para troca de contexto
  - ◆ Regra geral: 80% dos surtos devem ser menores que Iq

# Escalonamento Round Robin

» Exemplo: 3 processos com 10 ms:

- ◆ Quantum 1 ms  $\Rightarrow$  tempo de saída médio 29 ms

P1	P2	P3	P1	P2	P3		P1	P2	P3
0 1	1 2	2 3	3 4	4 5	5 6	...	27 28	28 29	29 30

TS1 = 28   TS2=29   TS3 = 30  
::Com 201 processos na fila o  
TR = 200ms = 0,2 seg

- ◆ Quantum 10 ms  $\Rightarrow$  tempo de saída médio = 20 ms

P1	P2	P3
0   10	10   20	20   30

TS1 = 10   TS2=20   TS3 = 30  
::Com 201 processos na fila o  
TR = 2000 ms = 2 seg

- ◆ Sem considerar tempo para troca de contexto
- ◆ Se o Quantum = 100ms, o TR seria de 20 seg para 201 processos na fila

# Exercício

---

Determine a Tabela de Troca de Estados para os seguintes dados, usando Round Robin:

PID	Chegada	Tempo
1	0	32
2	0	18
3	0	12

Quantum =  
6ms