

Sistemas Operacionais

Escalonamento

Edeyson Andrade Gomes

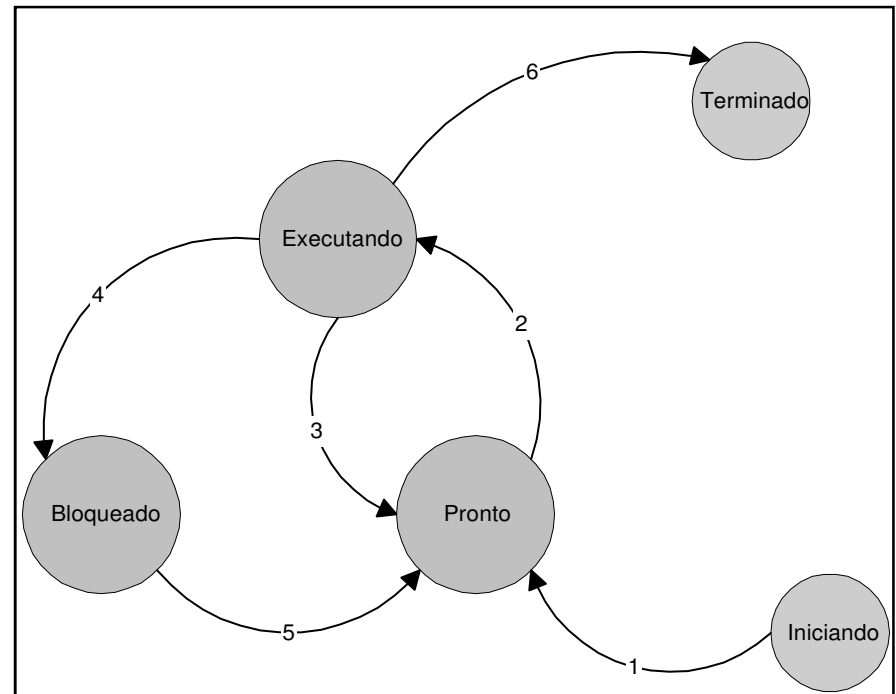
www.edeyson.com.br

Roteiro da Aula

- ▶ **Escalonamento de Processos**
 - ▶ Metas
 - ▶ Algoritmos
 - ▶ FIFO
 - ▶ SJF
 - ▶ Prioridades

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



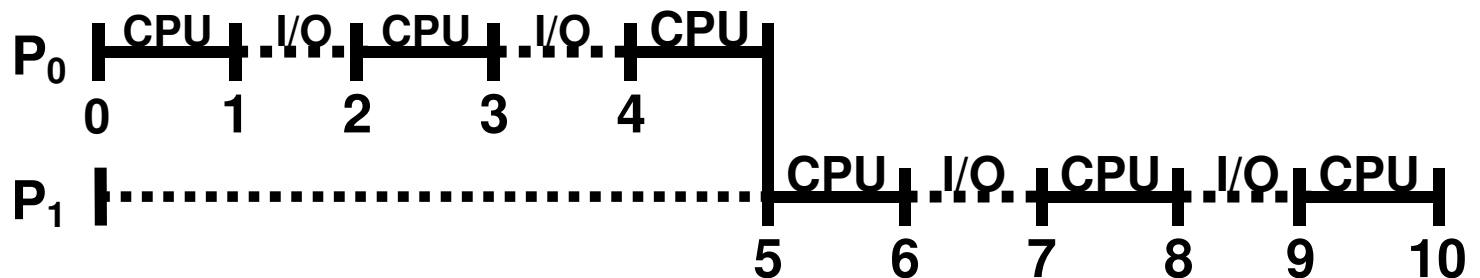
Escalonador de Processos

- ▶ **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, CPU Bound, I/O Bound, Interativos

Escalonador de Processos

▶ Sem multiprogramação

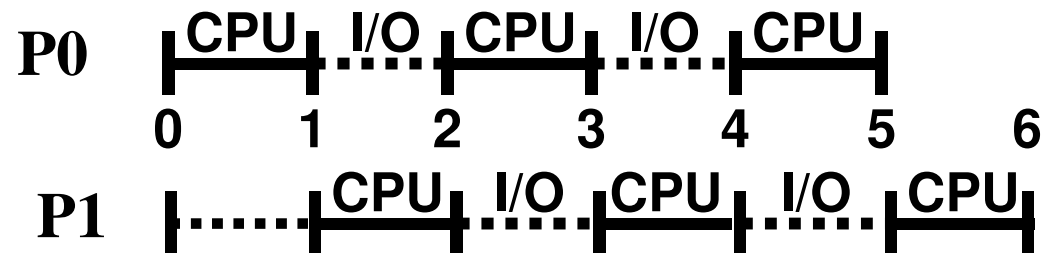
- ▶ Tempo Total de Execução = 10 unidades de tempo (ut)
- ▶ **Throughput** = 0,2 p/ut (No. Processos Executados por ut)
- ▶ $\Delta t P_0 = 5ut$ $\Delta t P_1 = 10ut$ ($\Delta t =$ Tempo Total)
- ▶ **Tempo médio de execução** = 7,5 ut = $(\Delta t P_0 + \Delta t P_1) / 2$
- ▶ **Utilização da CPU** = 60 % (Desprezando-se tempo de Kernel)
 - ▶ 40% de I/O



Escalonador de Processos

▶ Com multiprogramação

- ▶ Tempo Total de Execução = 6 ut
- ▶ **Throughput** = 0,33 (No. Processos / ut)
- ▶ $\Delta t P1 = 5ut$ $\Delta t P2 = 6ut$ ($\Delta t =$ Tempo Total)
- ▶ **Tempo médio de execução** = 5,5 ut
- ▶ **Utilização da CPU** = 100 %
 - ▶ Desprezando-se tempo de Kernel



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**

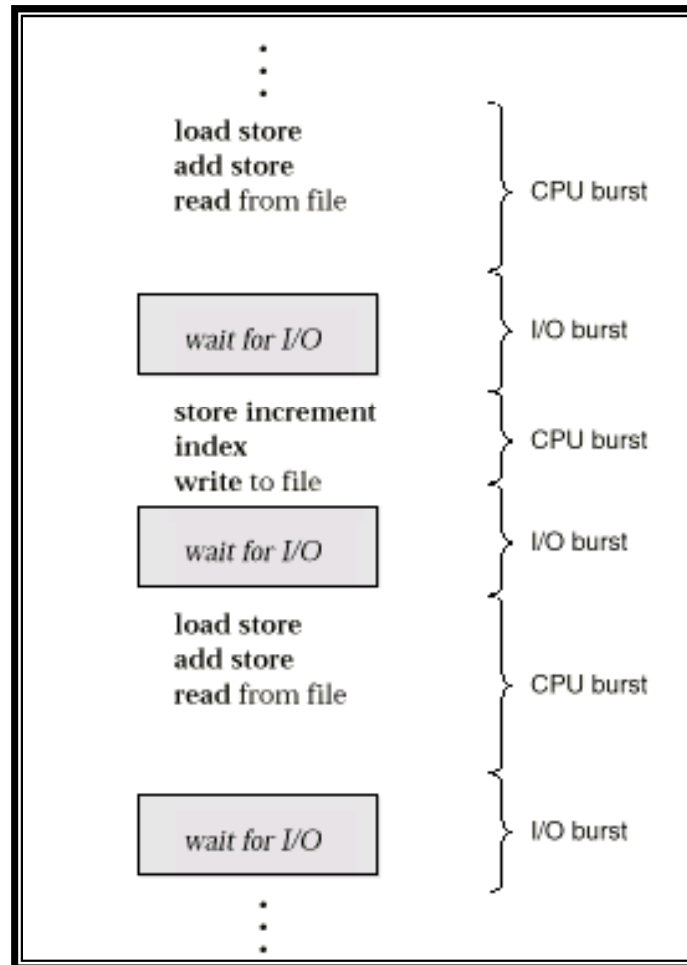
Tipos de Escalonamento

- ▶ **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

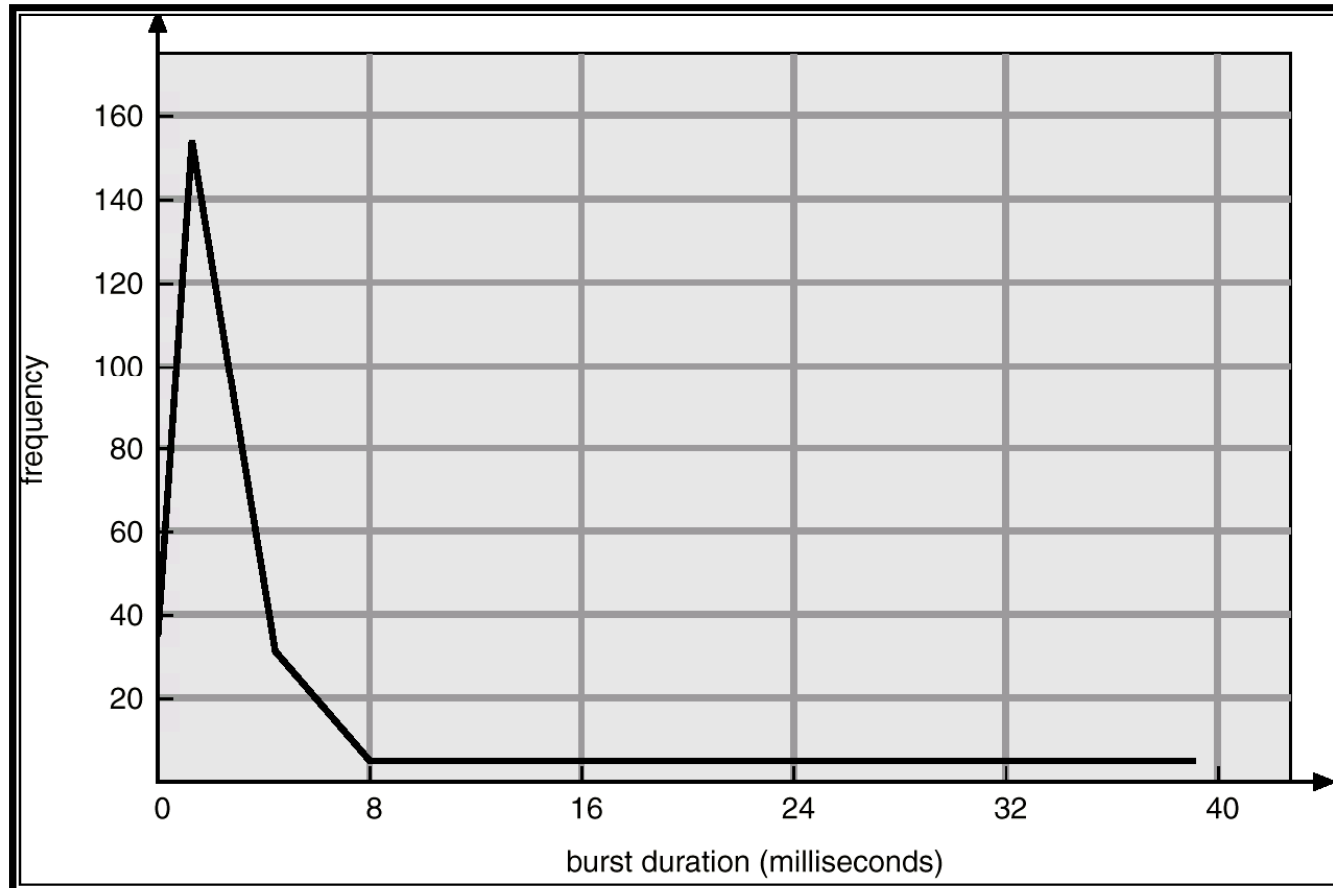
Escalonamento



Escalonamento

- ▶ Curva de frequência da duração dos surtos de CPU
 - ▶ Muitos surtos curtos
 - ▶ Poucos surtos longos

Escalonamento



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

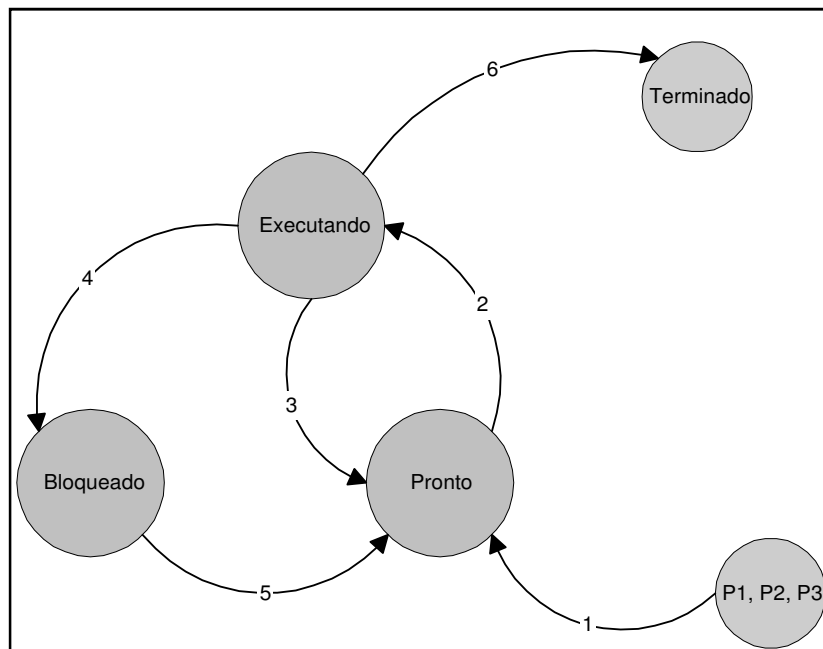
- ▶ Escalonador de CPU ou de Curto Prazo
 - ▶ Escalonamento Preemptivo
 - ▶ Requer temporizador na CPU
 - Fatia de Quantum
 - Uso do Clock
 - ▶ Requer suporte do SO para coordenar acesso a dados compartilhados de forma consistente
 - Proteção

Escalonamento

- ▶ **Dispatcher e Latência**
 - ▶ P_x 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



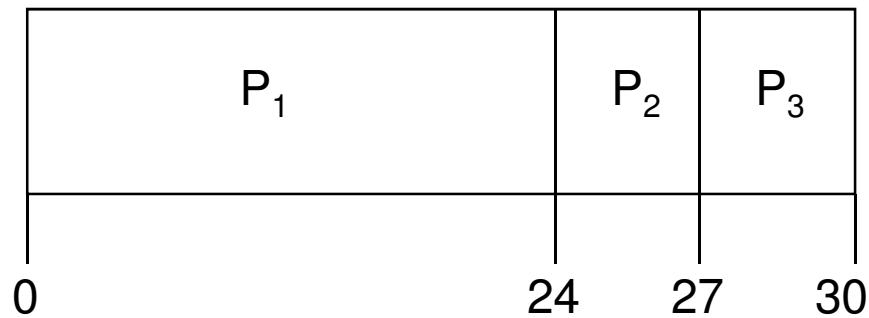
| Processo | Início | Duração (ut) |
|----------------|--------|--------------|
| P ₁ | 0 | 24 |
| P ₂ | 0 | 3 |
| P ₃ | 0 | 3 |

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

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

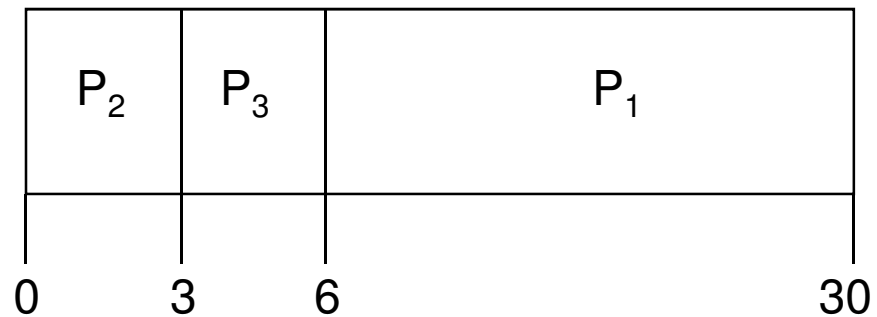
$$\text{Throughput} = 0,1 \text{ (3/30)}$$

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.

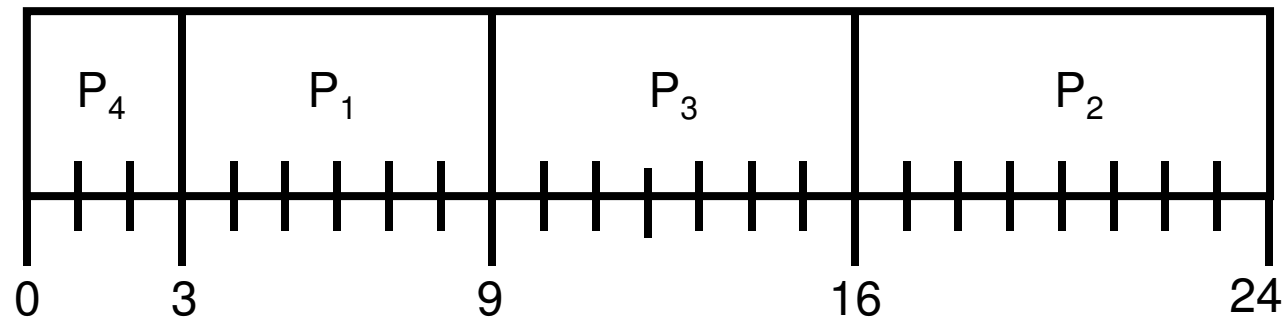
Sua execução diária permite determinar seu tempo total.

| PID | Início | Duração de surto |
|----------------|--------|------------------|
| P ₁ | 0 | 6 |
| P ₂ | 0 | 8 |
| P ₃ | 0 | 7 |
| P ₄ | 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*

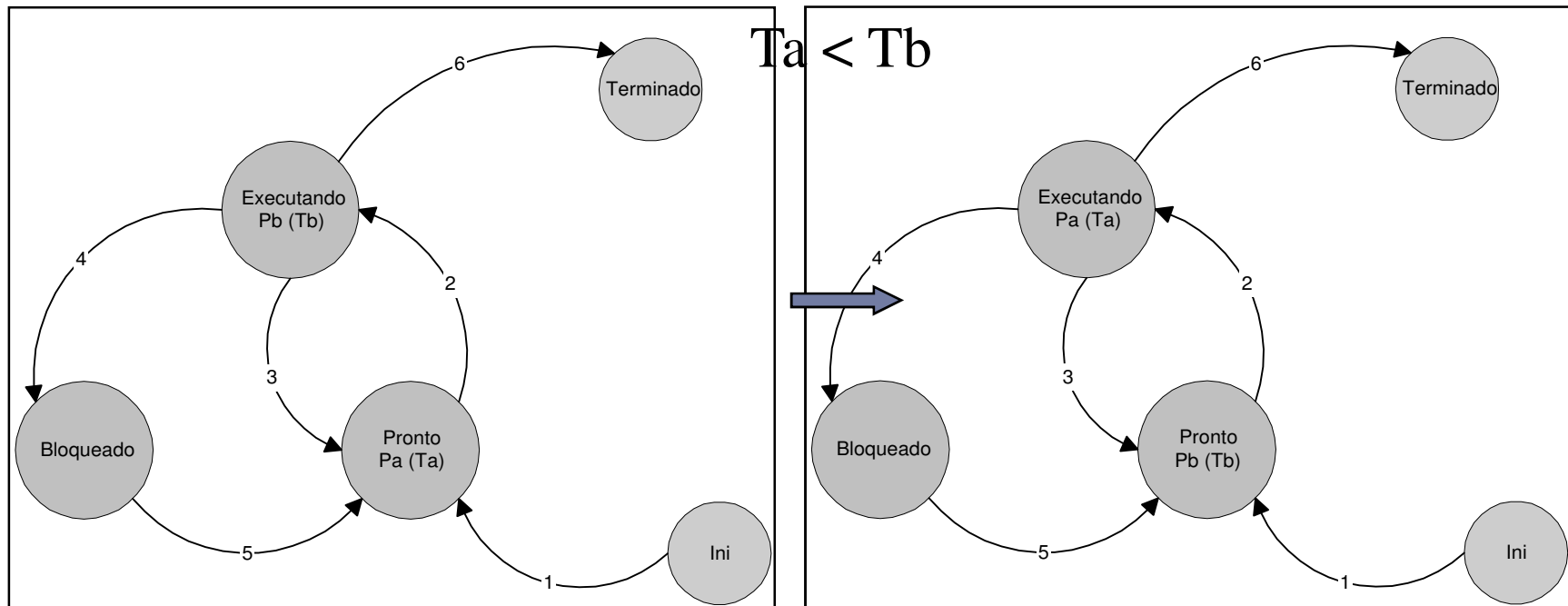
Escalonamento SJF

- ▶ **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



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 |

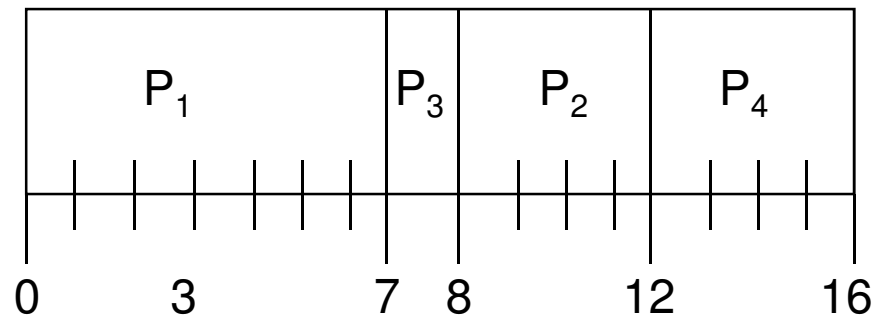
Preempção em SJF

- ▶ 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 = $(7 + 10 + 4 + 11) / 4 = 8$

- ▶ $TSP_1 = 7$ $TSP_2 = 10$

- ▶ $TSP_3 = 4$ $TSP_4 = 11$

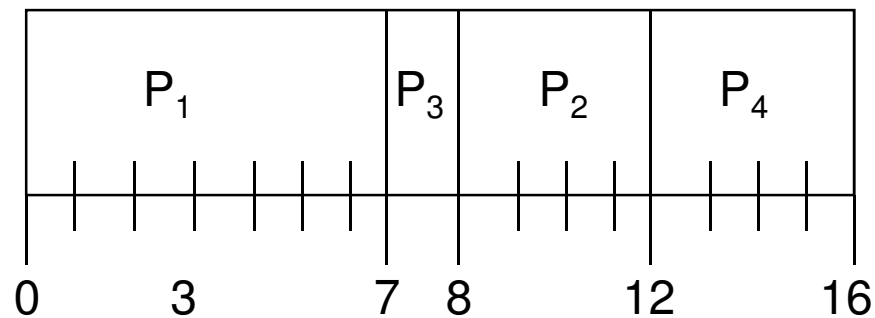


Tabela de Estados

| Tempo | PR | EX | TER |
|-------|---------------------|--------|-----|
| 0 | P1 (7) | | |
| 0 | | P1 (7) | |
| 2 | P2 (4) | P1 (5) | |
| 2 | P1(5) | P2(4) | |
| 4 | P3(1), P1(5) | P2(2) | |
| 4 | P2(2), P1(5) | P3(1) | |
| 5 | P2(2), P4(4), P1(5) | | P3 |
| 5 | P4(4), P1(5) | P2(2) | |
| 7 | P4(4), P1(5) | | P2 |
| 7 | P1(5) | P4(4) | |
| 11 | P1(5) | | P4 |
| 11 | | P1 | |
| 16 | | | P1 |

TP2 < TP1 => Preempção

TP3 < TP2 => Preempção

Escalonador por Término de P3

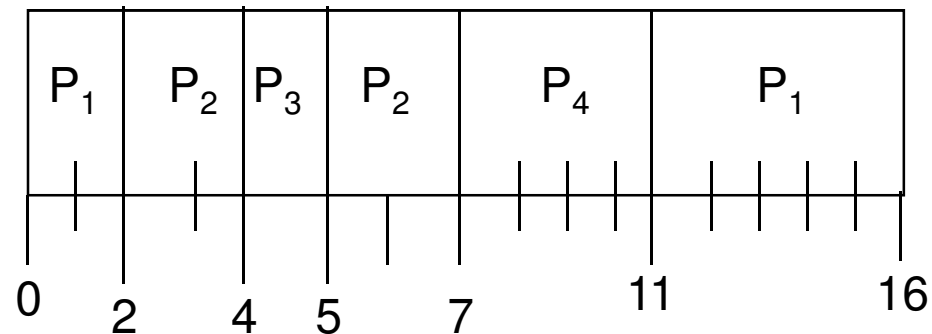
Preempção em SJF

- ▶ SJF preemptivo

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

- ▶ $TEP_1 = 9$ $TEP_2 = 1$

- ▶ $TEP_3 = 0$ $TEP_4 = 2$



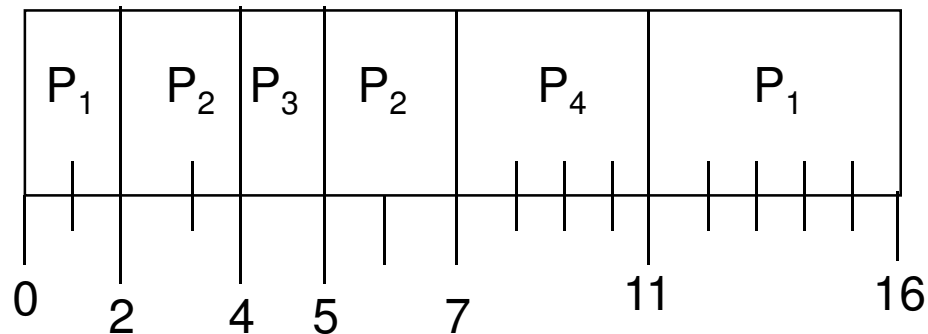
Preempção em SJF

▶ SJF preemptivo

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

▶ $TSP_1 = 16$ $TSP_2 = 5$

▶ $TSP_3 = 1$ $TSP_4 = 6$



Preempção em SJF

| PID | Chegada | Tempo CPU |
|-----|---------|-----------|
| 1 | 0 | 15 |
| 2 | 5 | 5 |
| 3 | 10 | 10 |
| 4 | 20 | 4 |

| Tempo | PR | EX | TER |
|-------|----------------|--------|-----|
| 0 | P1(15) | | |
| 0 | | P1(15) | |
| 5 | P2(5) | P1(10) | |
| 5 | P1(10) | P2(5) | |
| 10 | P1(10), P3(10) | | P2 |
| 10 | P3(10) | P1(10) | |
| 20 | P4(4), P3(10) | | P1 |
| 20 | P3(10) | P4(4) | |
| 24 | P3(10) | | P4 |
| 24 | | P3(10) | |
| 34 | | | P3 |

Preempção em SJF

| PID | Chegada | Tempo de Surto de CPU |
|-----|---------|-----------------------|
| 1 | 0 | 7 |
| 2 | 2 | 4 |
| 3 | 4 | 1 |
| 4 | 5 | 4 |

| Tempo | PR | EX | TER |
|-------|---------------------|-------|-----|
| 0 | P1 (7) | | |
| 0 | | P1(7) | |
| 2 | P2(4) | P1(5) | |
| 2 | P1(5) | P2(4) | |
| 4 | P3(1), P1(5) | P2(2) | |
| 4 | P2(2), P1(5) | P3(1) | |
| 5 | P2(2), P4(4), P1(5) | | P3 |
| 5 | P4(4), P1(5) | P2(2) | |
| 7 | P4(4), P1(5) | | P2 |
| 7 | P1(5) | P4(4) | |
| 11 | P1(5) | | P4 |
| 11 | | P1(5) | |
| 16 | | | P1 |

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 executa, faz uma requisição, vai para bloqueado e de lá para o fim da fila
 - ▶ Quando a resposta será entregue ao usuário do processo interativo?

Escalonamento Round Robin

| Processo | Início | Duração de surto |
|----------------|--------|------------------|
| P ₁ | 0 | 24 |
| P ₂ | 0 | 3 |
| P ₃ | 0 | 3 |

Exemplo com quantum de 4 UT

TEP1 = 6 TEP2 = 4 TEP3 = 7

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

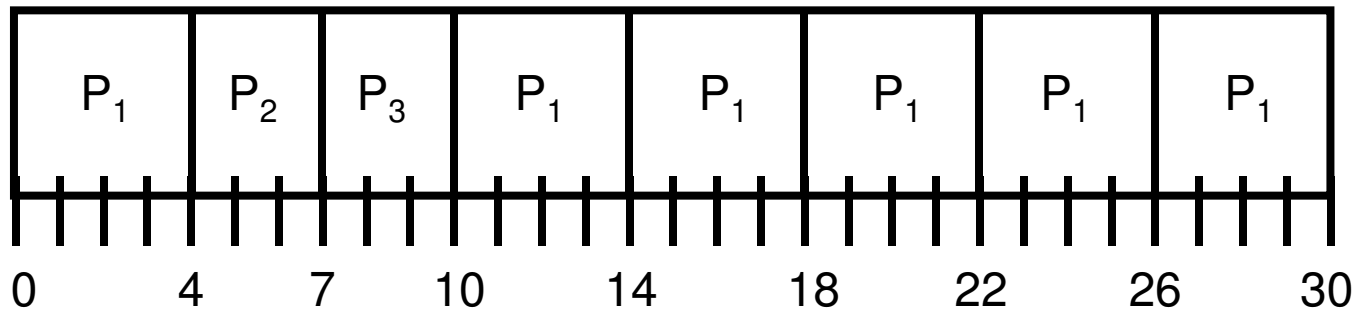
Escalonamento Round Robin

| Processo | Início | Duração de surto |
|----------------|--------|------------------|
| P ₁ | 0 | 24 |
| P ₂ | 0 | 3 |
| P ₃ | 0 | 3 |

Exemplo com quantum de 4 ms

TEP1 = 6 TEP2 = 4 TEP3 = 7

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

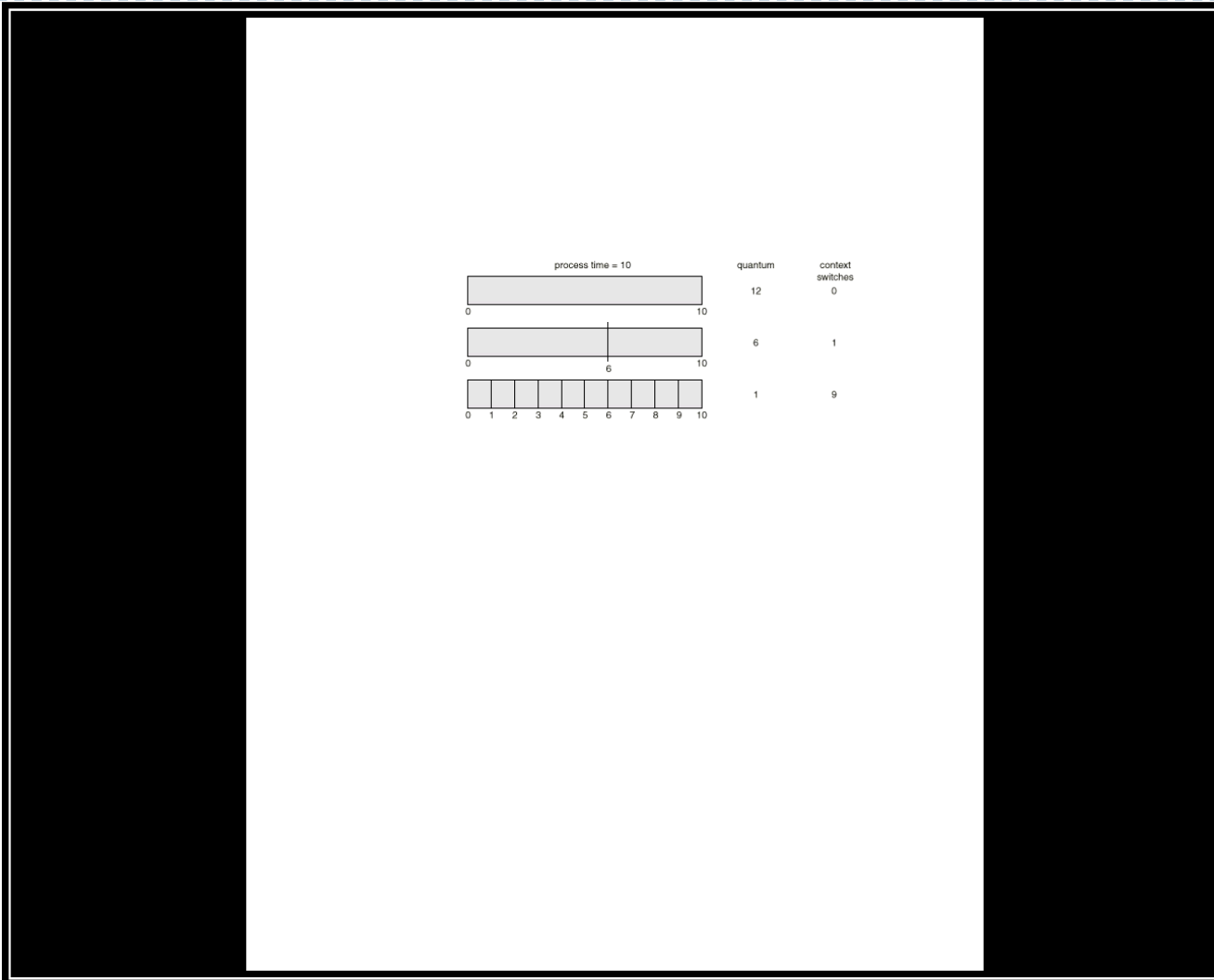


Escalonamento Round Robin

▶ 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



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 com 10 ms:
 - ◆ 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

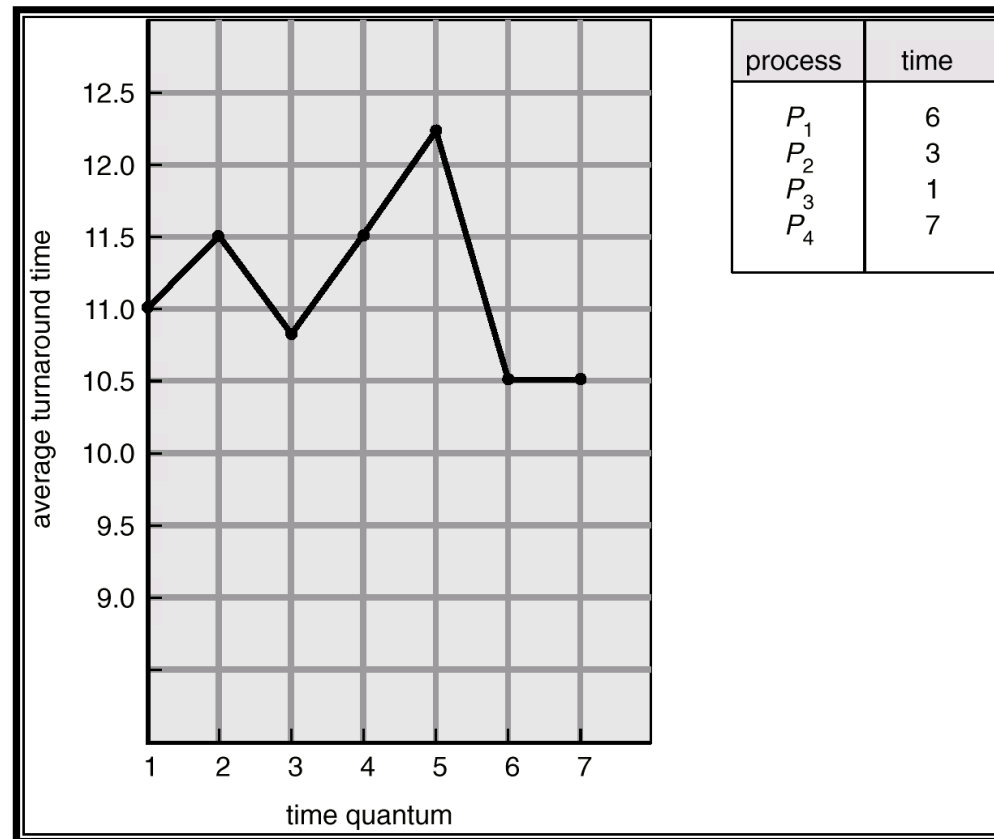
Exercício

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

Escalonamento Round Robin



Escalonamento por Prioridade

- ▶ Cada processo tem uma prioridade
 - ▶ Número inteiro dentro de limites
 - ▶ Faixas 0 a 7 ou 0 a 4095
 - ▶ Menor (ou maior) número \Rightarrow maior prioridade
 - ▶ Empate \Rightarrow FCFS
 - ▶ SJF é um caso especial de prioridade

Escalonamento por Prioridade

- ▶ Prioridade definida interna ou externamente
- ▶ Preemptivo ou não preemptivo
- ▶ Starvation – Estagnação
 - ▶ Bloqueio por tempo indefinido
 - ▶ Solução: aging (envelhecimento)

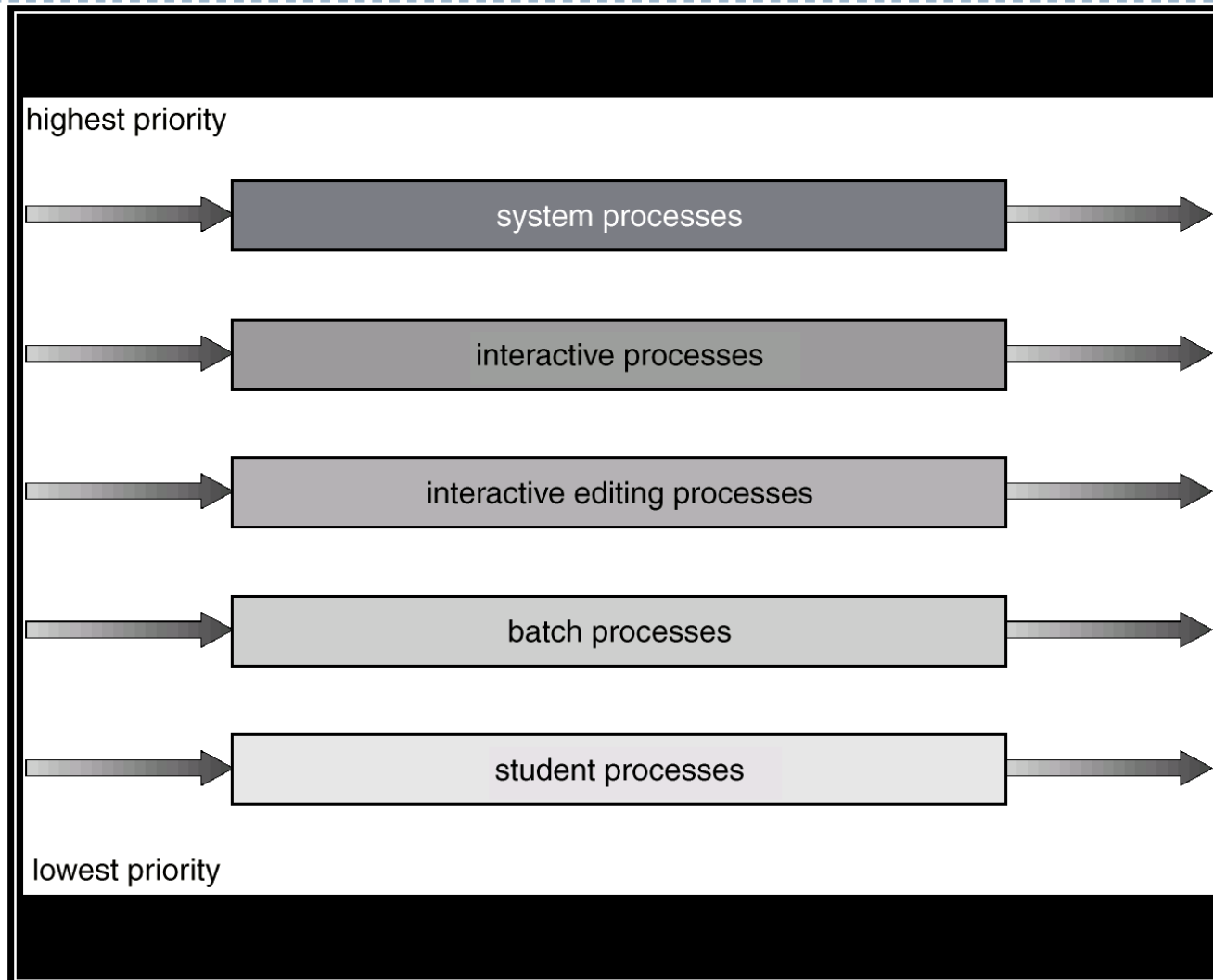
Escalonamento por Múltiplas Filas

- ▶ **Classificação dos processos em grupos**
 - ▶ Primeiro plano (foreground): interativos
 - ▶ Segundo plano (background): batch
- ▶ **Filas separadas para processos prontos**
 - ▶ Cada fila tem seu algoritmo
 - ▶ Foreground – RR
 - ▶ Background – FIFO

Escalonamento por Múltiplas Filas

- ▶ **Escalonamento preemptivo entre filas**
 - ▶ Prioridade fixa: só atende filas menos prioritárias se as demais estiverem vazias
 - ▶ Time slice 80% para foreground com RR e 20% para background com FIFO

Escalonamento por Múltiplas Filas



Escalonamento por Múltiplas Filas

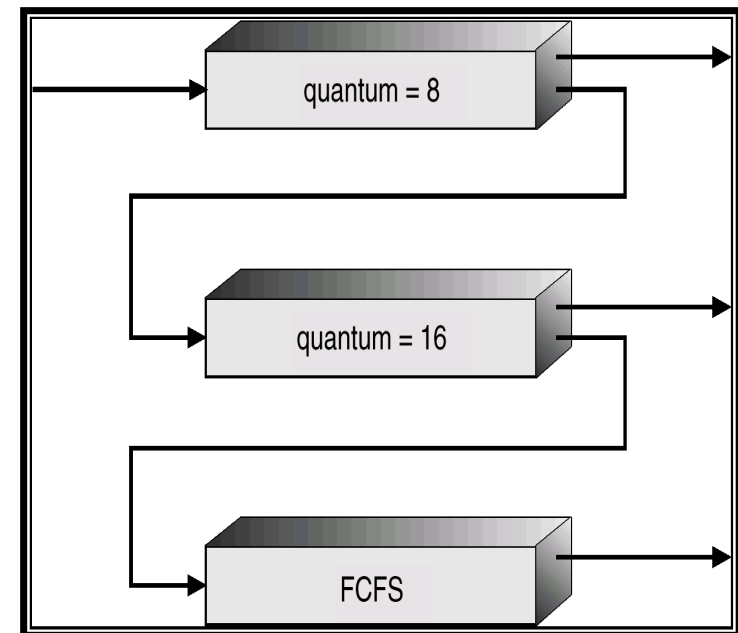
- ▶ **Filas caracterizadas pelos surtos de CPU dos processos**
 - ▶ I/O bound e interativos com mais prioridade
 - ▶ Passam a maior parte do tempo Bloqueados
 - ▶ Processos podem mudar de fila
 - ▶ Aging pode ser facilmente implementado
- ▶ **Algoritmo preemptivo**

Escalonamento por Múltiplas Filas

▶ Exemplo

▶ Três filas

- ▶ Q_0 – quantum 8 ms
- ▶ Q_1 – quantum 16 ms
- ▶ Q_2 – FIFO (FCFS)



Escalonamento com Múltiplos Processadores

- ▶ **Escalonamento de CPU mais complexo**
 - ▶ Existem sistemas com barramento de E/S privativo de determinado processador
- ▶ **Várias filas de processos prontos**
 - ▶ Possibilidade de desperdício de recursos

Escalonamento com Múltiplos Processadores

- ▶ Única fila de processos prontos
 - ▶ Symmetric Multiprocessing (SMP)
 - ▶ Cada processador faz seu escalonamento
 - ▶ Compartilhamento de estruturas de dados do SO
 - Sincronização
 - ▶ Assymmetric Multiprocessing
 - ▶ Escalonamento no processador mestre

Escalonamento de Tempo Real

- ▶ **Sistemas de tempo real crítico**
 - ▶ Limites rígidos de tempo
 - ▶ SO garante execução no tempo ou rejeita
 - ▶ Exige software especial e hardware dedicado

Escalonamento de Tempo Real

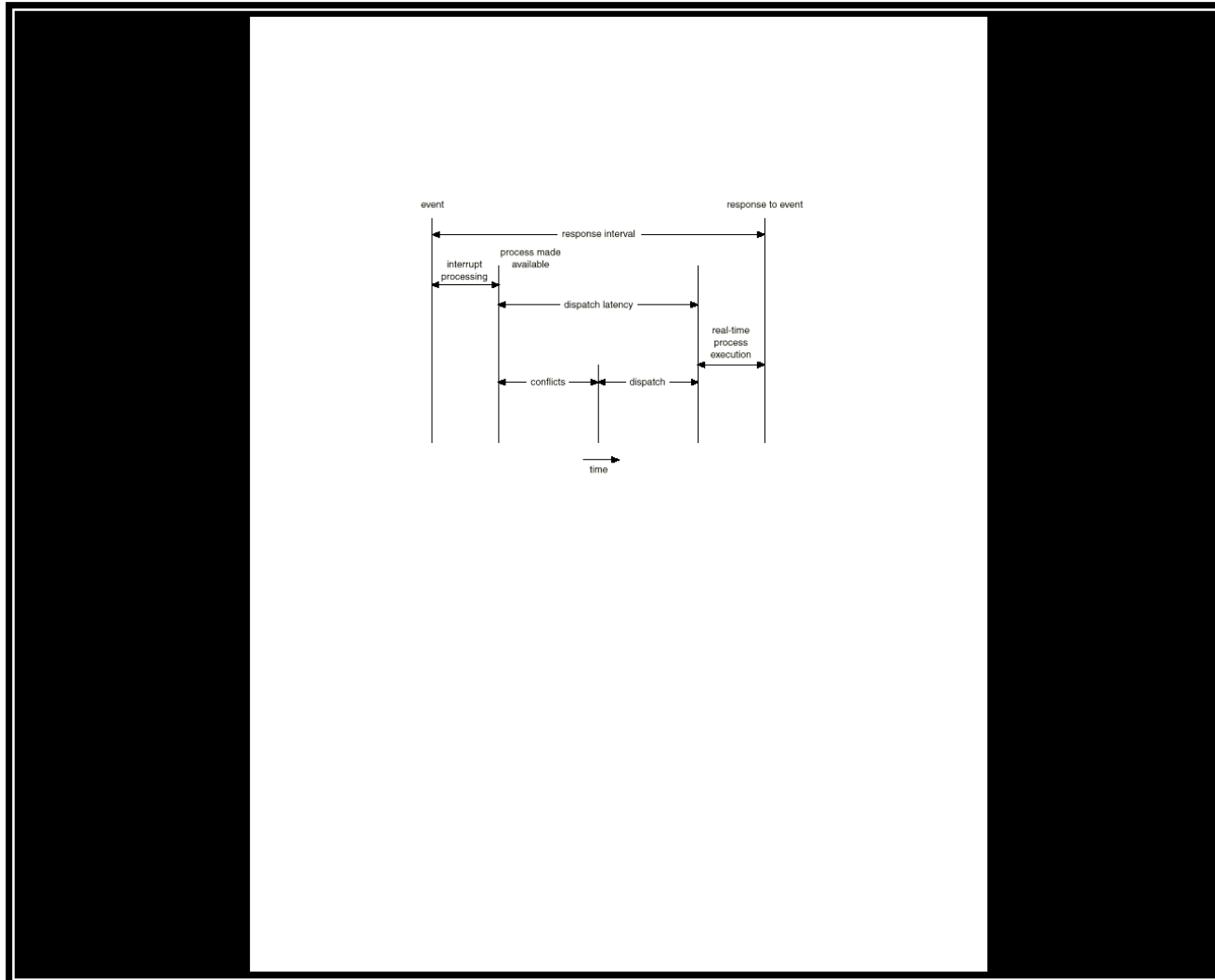
- ▶ **Sistemas de tempo real não-crítico**
 - ▶ Processos críticos com prioridade
 - ▶ Gera desbalanceamento do sistema
 - ▶ Suporte do SO
 - ▶ Escalonamento com prioridade
 - ▶ Não degradação da prioridade dos processos críticos
 - ▶ Latência de carga pequena
 - Chamadas ao sistema e operações de E/S
 - Pontos de preempção seguros
 - Todo Kernel preemptível (sincronização)
 - Protocolo de herança de prioridade

Escalonamento de Tempo Real

▶ Dispatch Latency

- ▶ Descreve a quantidade de tempo que um sistema gasta para responder a requisição de um processo
- ▶ O tempo de resposta (TR) total consiste em:
 - ▶ TR de Interrupção
 - ▶ Dispatch latency
 - ▶ TR da aplicação

Escalonamento de Tempo Real





Solaris

SunOS

Escalonamento no Solaris (SunOS 5.9)

- ▶ Preemptivo por prioridade
- ▶ Fatia de tempo (quantum)
- ▶ Várias classes com prioridades e algoritmos
 - ▶ Maior prioridade \Rightarrow menor fatia de tempo (e vice-versa)
 - ▶ Classes de Prioridade
 - ▶ Real-time, system, interactive (IA), fixed-priority (FX), fair-share (FSS) e time-sharing (TS)

Escalonamento no Solaris

▶ Escalonamento Padrão

- ▶ Política Time-sharing
- ▶ Ajuste dinâmico de prioridades de processos para balancear o tempo de resposta (processos interativos) e o *throughput* de processos *CPU bound*.
- ▶ O *scheduler* troca processos *CPU bound* frequentemente para prover bom tempo de resposta, mas não tão frequentemente que gere overhead demasiado ao sistema.

Escalonamento no Solaris

- ▶ A política time-sharing:
 - ▶ Diminui a prioridade de processos que usam a CPU por longos períodos de tempo sem bloqueio (*sleep*).
 - ▶ Atribui largas fatias de tempo para processos com baixa prioridade (CPU Bound). Se um processo de maior prioridade torna-se pronto durante esse quantum, ocorre preempção por prioridade do processo executando.

Escalonamento no Solaris

- ▶ **Prioridade Real-time**

- ▶ Maior prioridade

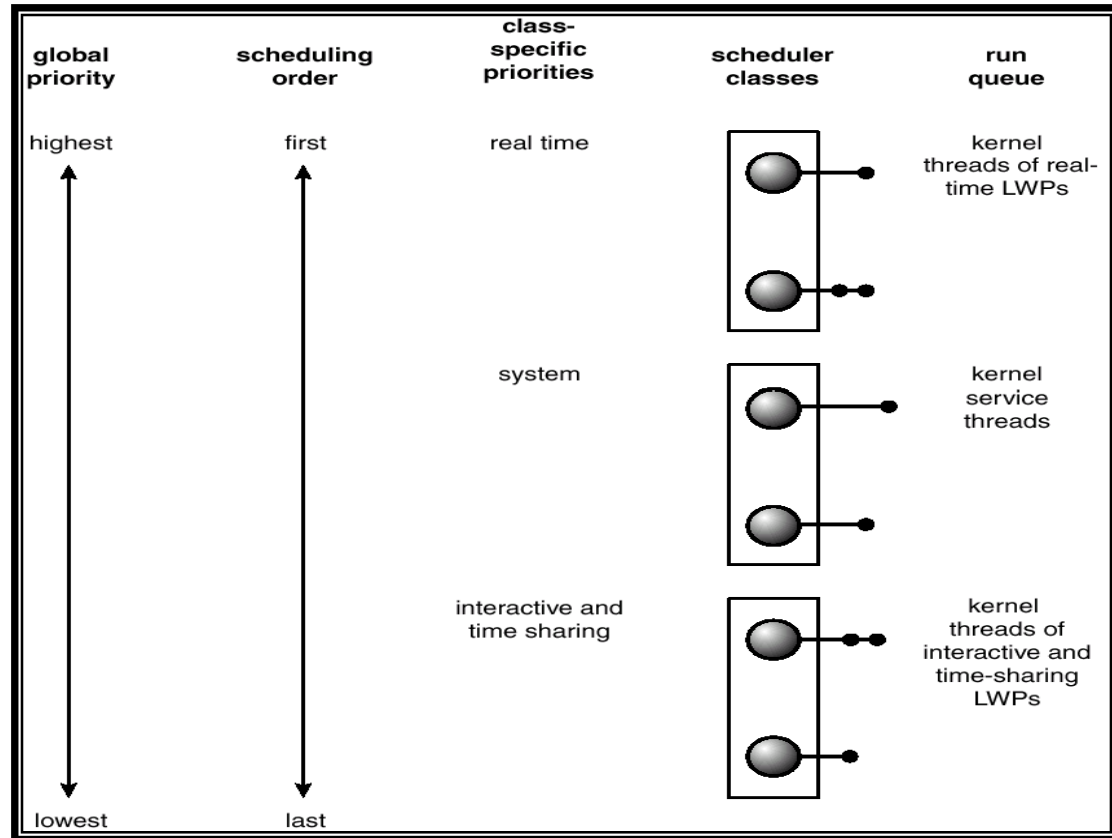
- ▶ Processos sempre retornam à CPU tão logo estejam prontos.

- ▶ Alerta no SunOS:

- ▶ “Careless use of real-time processes can have a dramatic negative effect on the performance of time-sharing processes.”

Escalonamento no Solaris

Process Priorities (Programmer's View)

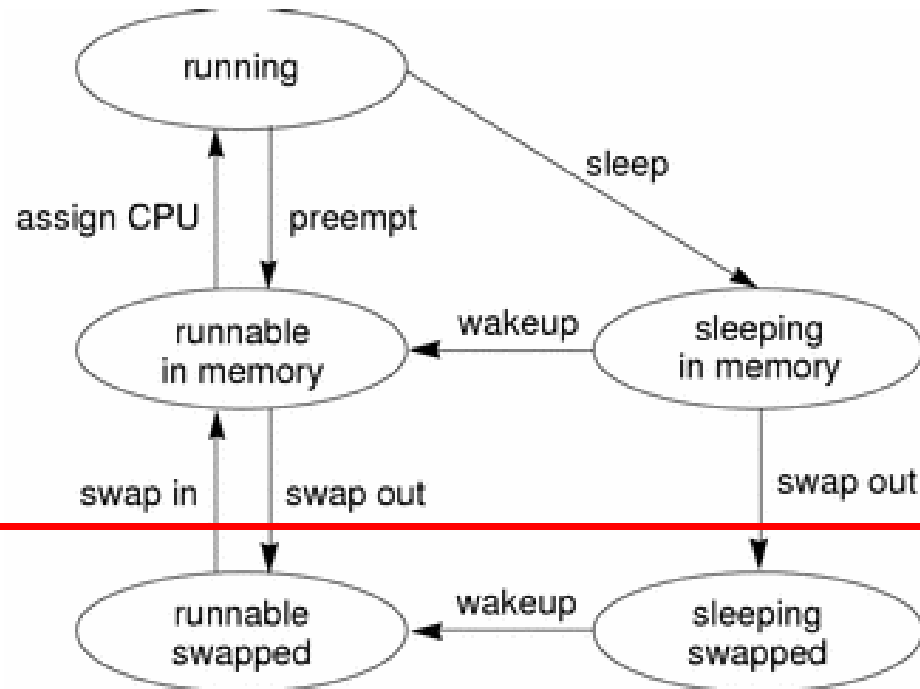


<http://docs.sun.com/db/doc/806-4125/6jd7pe6ak?a=view>

Escalonamento no Solaris

Estados dos Processos

Baixo Nível ou Curto Prazo



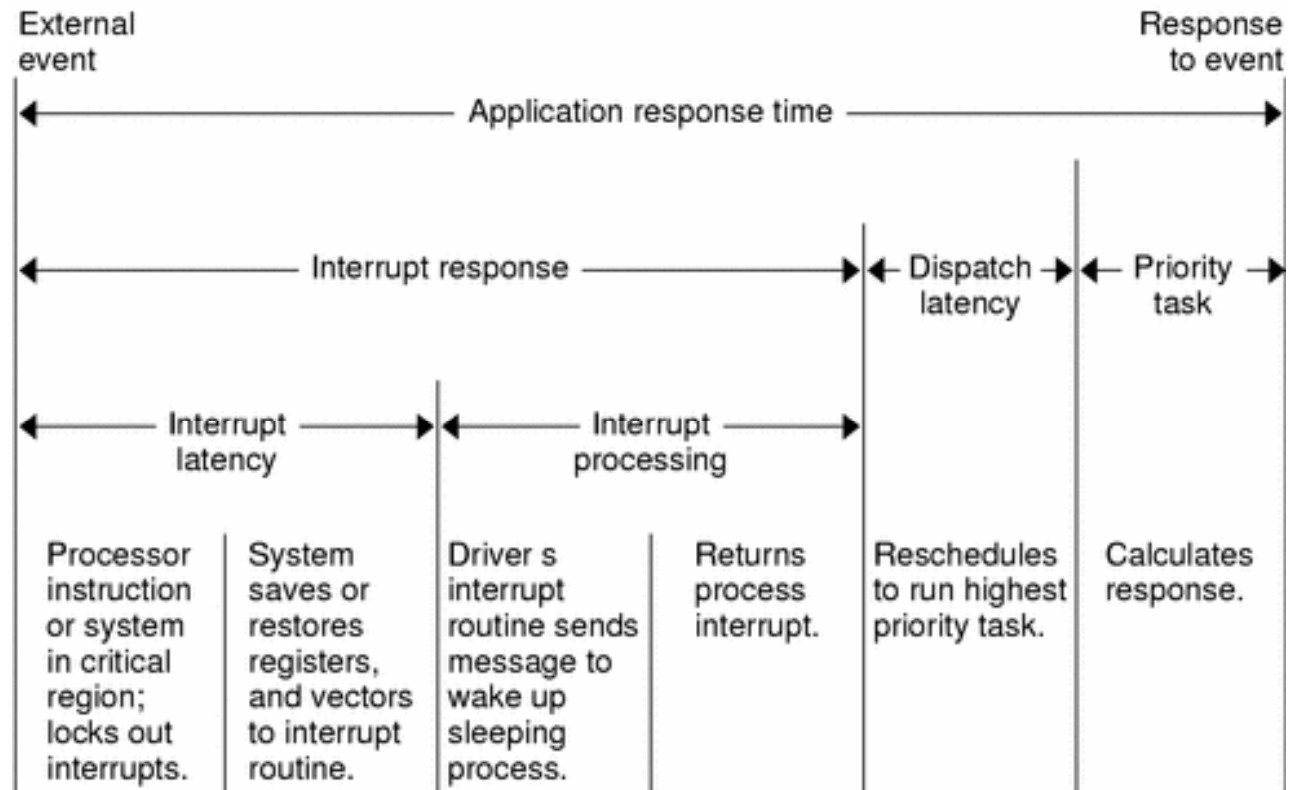
Alto Nível ou Longo Prazo

<http://docs.sun.com/db/doc/806-4125/6jd7pe6ak?a=view>

Escalonamento no Solaris

Escalonamento
Em Tempo
Real

Dispatch
Latency



<http://docs.sun.com/db/doc/806-4125/6jd7pe6ak?a=view>



Microsoft

Windows

Prioridades no W2000

| | 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 |

Estados de Threads no W2000

| Estado | Descrição | Comentário |
|---------------|------------------|---|
| 0 | Initialized | |
| 1 | Ready | The thread is prepared to run on the next available processor. |
| 2 | Running | |
| 3 | Standby | The thread is about to use the processor. |
| 4 | Terminated | |
| 5 | Waiting | The thread is not ready to run, typically because another operation (for example, involving I/O) must finish before the thread can run. |
| 6 | Transition | The thread is not ready to run because it is waiting for a resource (such as code being paged in from disk). |
| 7 | Unknown | The thread is in an unknown state. |

Escalonamento de Threads Java

- ▶ **Escalonamento da JVM**
 - ▶ Baseado em algoritmo com prioridade e preempção
 - ▶ Fatia de tempo depende da implementação da JVM
- ▶ **Acionamento do escalonador**
 - ▶ Thread em execução sai do estado Executável
 - ▶ I/O, suspend ou stop
 - ▶ Thread com prioridade maior entra no estado Executável

Escalonamento de Threads Java

▶ Métodos

▶ yield()

- ▶ Passagem do controle – Multitarefa cooperativa

▶ setPriority()

- ▶ Thread.NORM_PRIORITY = 5
- ▶ Thread.MIN_PRIORITY = 1
- ▶ Thread.MAX_PRIORITY = 10

Escalonamento de Threads Java

- ▶ **Exemplo**
 - ▶ Escalonador Round-Robin com base em Java
 - ▶ Livro Silberschatz et al

Avaliação de Algoritmos

- ▶ **Seleção do melhor algoritmo**
 - ▶ Específico para determinado sistema
 - ▶ Definição dos parâmetros importantes
 - ▶ Normalmente mais de um parâmetro
 - Maximizar uso de CPU e limitar tempo de resposta

Avaliação de Algoritmos

- ▶ Métodos de Avaliação

- ▶ Avaliação Analítica

- ▶ Modelagem determinística

- ▶ Para determinada situação constrói diagramas de Gantt

- ▶ Muito específica para ser útil

- Apenas indica tendências

Avaliação de Algoritmos

▶ Métodos de Avaliação

▶ Modelo de Filas

- ▶ Análise de redes de filas
- ▶ Necessita curvas de distribuição
 - Surtos de CPU e I/O
 - Tempos de chegada dos processos
- ▶ Sistema de computação é uma rede de servidores
- ▶ Exige simplificação para tratamento matemático
 - Precisão dos resultados questionável

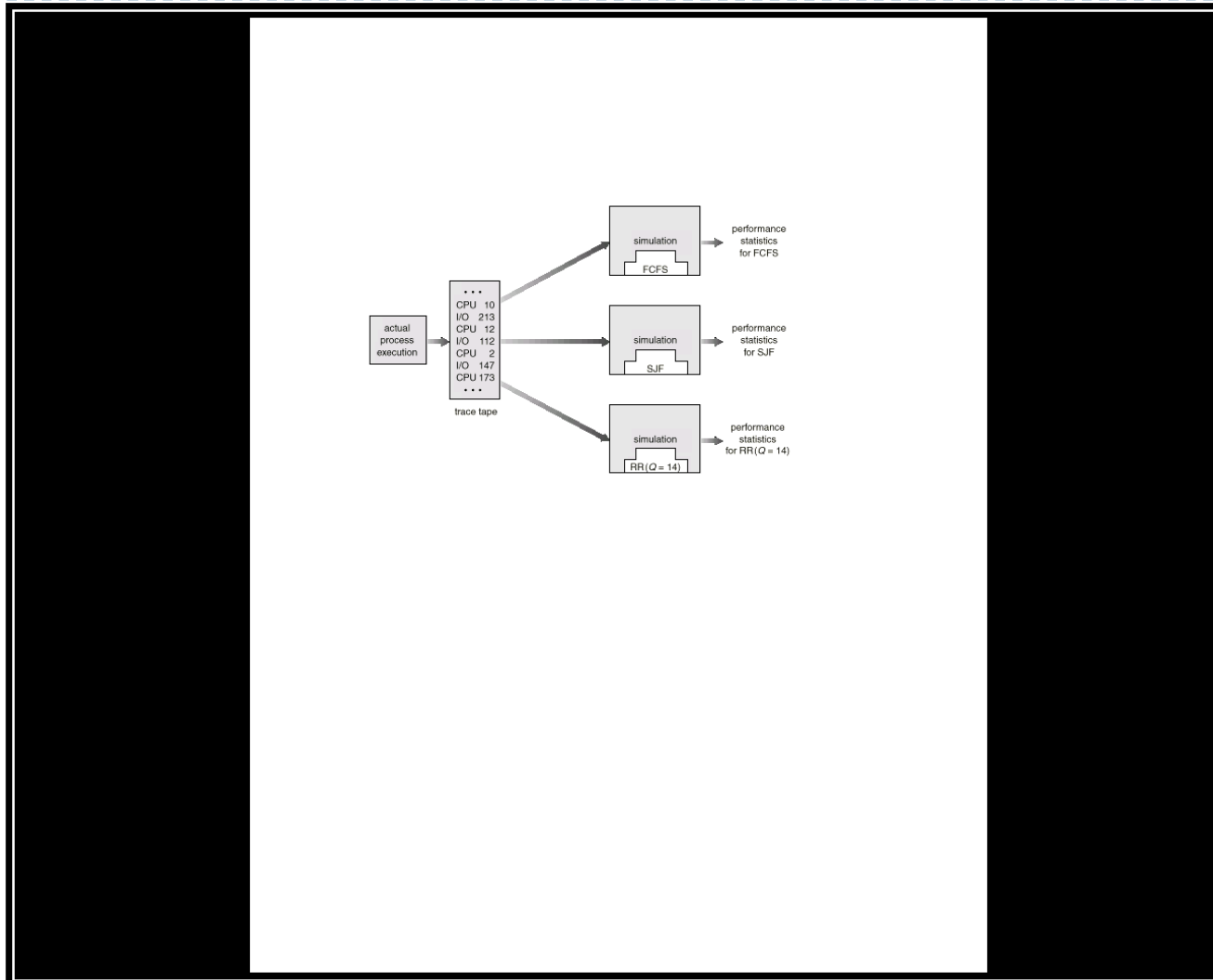
Avaliação de Algoritmos

▶ Métodos de Avaliação

▶ Simulações

- ▶ Modelo do sistema de computador
- ▶ Dados para simulação podem ser estimados ou coletados de sistemas reais
- ▶ Custo alto, uso intensivo da máquina

Avaliação de Algoritmos



Avaliação de Algoritmos

- ▶ **Métodos de Avaliação**
 - ▶ Implementação
 - ▶ Teste real do algoritmo
 - ▶ Problemas com usuários
 - ▶ Mudanças do ambiente original
 - Normais
 - Específicas para o novo algoritmo