

# SISTEMAS OPERACIONAIS

## AULA II – ESCALONAMENTO DE PROCESSOS

Prof.<sup>a</sup> Sandra Cossul, Ma.



# TÓPICOS DA AULA

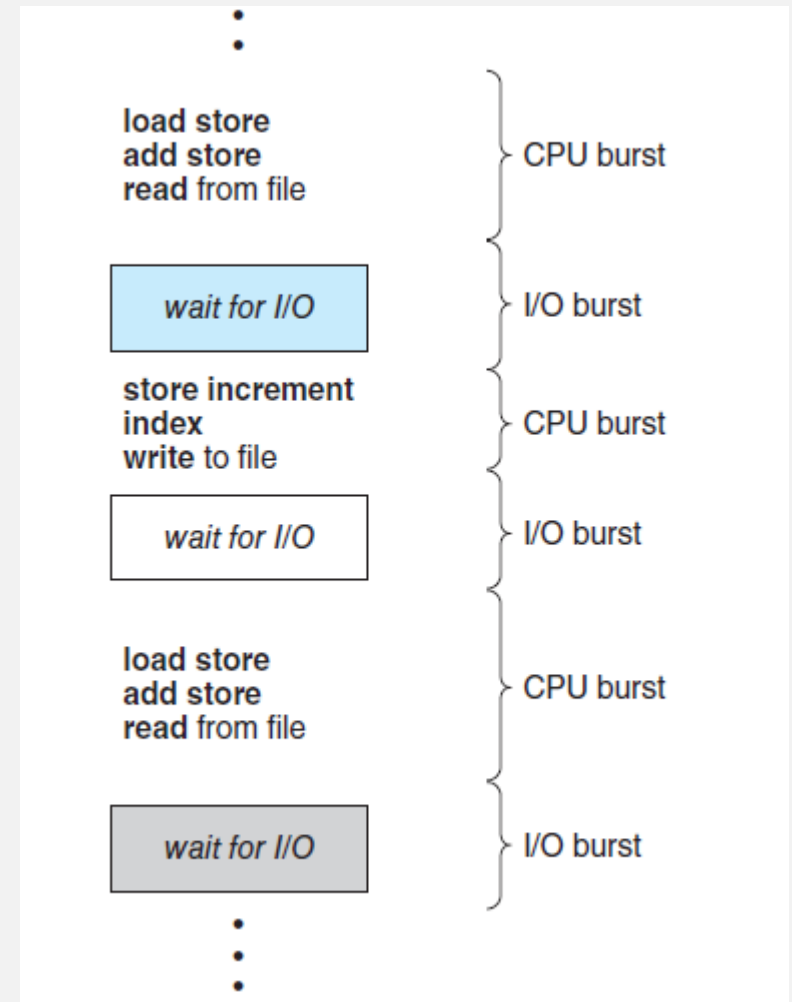
- Escalonador do SO
- Tipos de Processos
- Critérios de Escalonamento
- Algoritmos de Escalonamento

# INTRODUÇÃO

- O escalonamento do processador é a **base** para **SOs multiprogramados**
- A partir da **troca de processos** no CPU, o SO consegue tornar o computador **mais eficiente**
- **Objetivo:** ter algum processo rodando sempre, maximizando o uso do CPU!
- Um processo é executado até que ele tenha que esperar (normalmente até completar uma requisição de E/S)
- Vários processos são mantidos na memória, facilitando a troca de processos
- **single core** – um processo; **multicore** – vários processos

# PROPRIEDADE DE EXECUÇÃO DOS PROCESSOS

- A execução de um processo consiste em:
  - **Ciclo de execução (uso do CPU)**
  - **Espera por E/S**



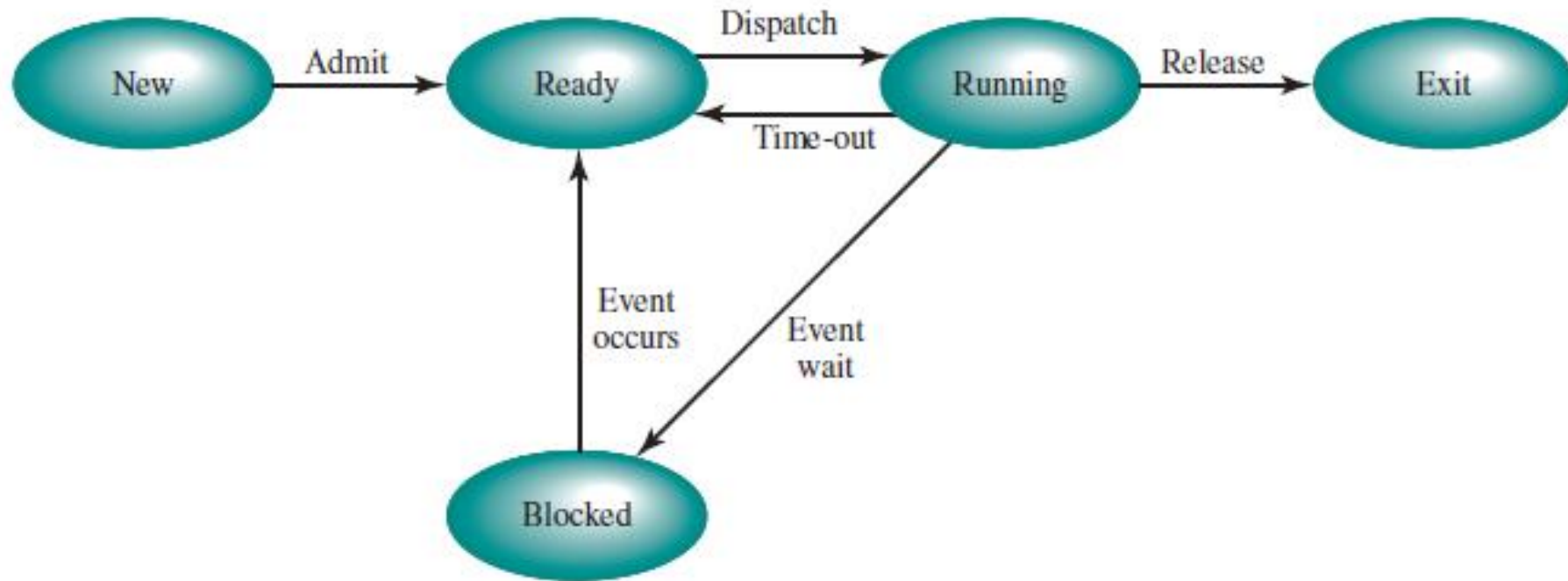
## ESCALONAMENTO DE PROCESSOS

- Quando o CPU fica ocioso, o SO deve selecionar um dos processos em memória na fila de “prontos” e alocar CPU para executar aquele processo;
- O processo de escolha é feito pelo **escalonador de processos (scheduler)**, que decide a ordem de execução dos processos em estado de pronto.

# ESCALONAMENTO DE PROCESSOS

- Circunstâncias que exigem decisões de escalonamento:
  1. Estado do processo muda de **executando** para estado de **espera** (requisição de E/S)
  2. Estado do processo muda de **executando** para estado de **pronto** (ocorrência de interrupção)
  3. Estado do processo muda de **espera** para **pronto** (terminou requisição de E/S)
  4. Quando um processo **termina**
- **1 e 4 – escalonamento não preemptivo** (depois que o CPU foi alocado, o processo só libera ao terminar ou entrar em estado de espera)
- **2 e 3 – escalonamento preemptivo** (pode resultar em condições de corrida quando dados são compartilhados entre os processos)

# ESTADOS DE PROCESOS



## ESCALONADOR (DISPATCHER)

- **Toma a decisão de qual será o próximo processo a executar**
- Isso envolve:
  - Troca de contexto de um processo para outro
  - Troca para modo usuário
  - Saltar para a localização correta no programa de usuário para retomar esse programa
- **Deve ser o mais rápido possível** (já que é chamado a cada troca de contexto)
- **Dispatch latency** – tempo para parar um processo e iniciar a execução de outro



# TIPOS DE PROCESSOS

- Em relação ao comportamento temporal:
  - **Processos de tempo real** – exigem previsibilidade em seus tempos de resposta aos eventos externos (associados ao controle de sistemas críticos)
  - **Processos interativos** – processos que recebem eventos externos (do usuário ou através da rede) e devem respondê-los rapidamente. Inclui a maior parte das aplicações dos sistemas (editores de texto, navegadores web, jogos, etc.) e servidores de rede (email, bancos de dados, etc.)
  - **Processos em lote (batch)** – processos sem requisitos temporais explícitos que, normalmente, executam sem intervenção do usuário (procedimentos de backup, cálculos numéricos longos, renderização de animações, etc.)

# TIPOS DE PROCESSOS

- Em relação ao comportamento no uso do CPU:
  - **Processos orientados a processamento (CPU bound)** – processos que usam intensivamente o processador (maior parte do tempo nos estados de pronto ou executando). Ex.: processamentos longos
  - **Processos orientados a entrada/saída (I/O bound)** – processos que dependem muito mais dos dispositivos de E/S que do processador (maior parte no estado bloqueado aguardando resposta às suas solicitações de leitura ou escrita de dados de E/S). Ex.: servidores de rede, editores...

## TIPOS DE ESCALONAMENTO

- **Sistemas preemptivos** - nestes sistemas um processo pode perder o CPU caso termine seu quantum de tempo, caso execute uma chamada de sistema ou caso ocorra uma interrupção que acorde um processo mais prioritário (que estava bloqueado aguardando um evento).
- A cada interrupção, exceção ou chamada de sistema, o escalonador reavalia todos os processos no estado de pronto e decide se mantém ou substitui o processo atualmente em execução.
- Maioria dos SOs atuais é preemptivo.

## TIPOS DE ESCALONAMENTO

- **Sistemas cooperativos (não-preemptivos)** – o processo em execução permanece no CPU tanto quanto possível, só liberando o mesmo caso termine de executar, solicite uma operação de E/S ou libere explicitamente o CPU, voltando à fila de processos prontos.
- Esses sistemas são chamados de cooperativos por exigir a cooperação dos processo entre si na gestão do processador, para que todos possam executar.

# CRITÉRIOS DE ESCALONAMENTO

- Cada **algoritmo de escalonamento** tem **propriedades** diferentes e a escolha de um **deve considerar os tipos de processos**.
- Critérios para escolha:
  - **Uso de CPU (eficiência)** - manter o CPU o mais ocupado possível
  - **Justiça** – distribuição do CPU entre os processos prontos: dois processos de comportamento e prioridade similares devem ter durações de execução similares
  - **Taxa de transferência** - número de processos completos por unidade de tempo

# CRITÉRIOS DE ESCALONAMENTO

- CrITÉrios para escolha:
  - **Tempo de vida/conclusão (*turnaround time*)** - tempo de submissão do processo até ele estar completo, ou seja, o tempo decorrido desde a criação do processo e seu encerramento, computando todos os tempos de processamento e espera
  - **Tempo de espera (*waiting time*)** - tempo utilizado na fila de prontos, aguardando o processador
  - **Tempo de resposta (*response time*)** - tempo de submissão do processo até a obtenção da primeira resposta, ou seja, tempo decorrido entre a chegada de um evento ao sistema e o resultado imediato de seu processamento.

## USO DE PRIORIDADES

- Em muitos sistemas, a cada processo é associada uma **prioridade**
- O escalonador sempre escolhe um processo de **maior prioridade**
- Lembrar que pode levar a starvation!
- O escalonamento por prioridade pode ser **cooperativo** ou **preemptivo**.

## USO DE PRIORIDADES

### ALGORITMO POR PRIORIDADE COOPERATIVO

- Os valores de prioridade são considerados em uma escala de prioridade positiva, ou seja, valores numéricos maiores indicam maior prioridade.
- A tarefa com maior prioridade executa primeiro.



## USO DE PRIORIDADES

### ALGORITMO POR PRIORIDADE PREEMPTIVO

- Quando um processo de maior prioridade se torna disponível para execução, o escalonador entrega o processador a ela, trazendo o processo atualmente em execução de volta para a fila de prontos.
- Em outras palavras, o processo em execução pode ser “preemptado” por um novo processo mais prioritário.

# ALGORITMOS DE ESCALONAMENTO

- **Função:** escolher qual ou quais processos que estão na fila de prontos devem ser alocados para executar no CPU

# ALGORITMOS DE ESCALONAMENTO

## **FIFO: FIRST-IN FIRST-OUT**

- Algoritmo **simples** (mesma ideia de uma **fila**)
- O processo que pede o CPU primeiro é alocado primeiro;
- Conhecido também como FCFS: First Come – First Served

# ALGORITMOS DE ESCALONAMENTO

## **FIFO: FIRST-IN FIRST-OUT**

- Algoritmo **simples** (mesma ideia de uma **fila**)
- O processo que pede o CPU primeiro é alocado primeiro;
- Desvantagem: tempo de espera geralmente é longo
  - Efeito comboio: todos processos esperam o grande processo terminar
  - Solução: permitir processos pequenos executar primeiro
- Algoritmo não preemptivo: uma vez que o CPU foi alocado para um processo, ele permanece utilizando o CPU até terminar ou requisitar um dado de E/S
  - Problemático para sistemas de tempo compartilhado

# ALGORITMOS DE ESCALONAMENTO

## FIFO: FIRST-IN FIRST-OUT

**Exemplo 1:**

Processo	Tempo de uso do CPU (ms)
P1	24
P2	3
P3	3

- **Ordem de chegada:** P1, P2 e P3

P1		P2	P3
0	24	27	30

Tempo de espera  
médio = 17 ms

- **Ordem de chegada:** P2, P3 e P1

P2	P3	P1
0 3	6	30

Tempo de espera  
médio = 3 ms

## CONJUNTO DE PROCESSOS

Processo	Ingresso	Duração	Prioridade
P1	0	5	2
P2	0	2	3
P3	1	4	1
P4	3	1	4
P5	5	2	5

**Processos na fila de prontos**  
Unidade de tempo: segundos (s)

# ALGORITMOS DE ESCALONAMENTO

## **FIFO: FIRST-IN FIRST-OUT**

- **Exercício I**

- Represente o escalonamento dos processos ao longo do tempo usando o algoritmo FIFO e calcule:
  - *Tempo médio de conclusão/vida*
  - *Tempo médio de espera*
- Situação 1 – sem considerar a prioridade
- Situação 2 – levando em conta a prioridade dos processos (prioridade cooperativa)
- Situação 3 - levando em conta a prioridade dos processos (prioridade preemptiva)

## ALGORITMOS DE ESCALONAMENTO **ROUND-ROBIN**

- **Round-robin (escalonamento circular)** – parecido com FIFO, mas preempção é adicionada para permitir a troca de processos
- Uma **unidade de tempo** (time quantum) é definida – 10 a 100 ms
- A fila de processos prontos é tratada como uma **fila circular**
- O **algoritmo de escalonamento** percorre a fila alocando o CPU para cada processo por um intervalo de tempo de no máximo **1 unidade de tempo**



## ALGORITMOS DE ESCALONAMENTO **ROUND-ROBIN**

- Duas situações:
  - Se o processo utilizar menos que 1 *quantum* de tempo, libera o CPU
  - Se o processo precisar mais que 1 *quantum* de tempo, é interrompido e colocado no final da fila
    - O escalonador escolhe o próximo processo na fila de prontos

# ALGORITMOS DE ESCALONAMENTO

## ROUND-ROBIN

**Exemplo 2:**

Processo	Tempo de uso do CPU (ms)
P1	24
P2	3
P3	3

Unidade de  
tempo/quantum: **4 ms**

- **Ordem de chegada:** P1, P2 e P3

P1	P2	P3	P1	P1	P1	P1	P1
0 4	7	10	14	18	22	26	30

Tempo de espera  
médio = 5,66 ms

- Algoritmo preemptivo!
- Performance depende do tamanho da unidade de tempo

## CONJUNTO DE PROCESSOS

Processo	Ingresso	Duração	Prioridade
P1	0	5	2
P2	0	2	3
P3	1	4	1
P4	3	1	4
P5	5	2	5

**Processos na fila de prontos**  
Unidade de tempo: segundos (s)

# ALGORITMOS DE ESCALONAMENTO

## **ROUND-ROBIN**

- **Exercício 2**
  - Represente o escalonamento dos processos ao longo do tempo usando o algoritmo Round-Robin e calcule:
    - *Tempo médio de conclusão/vida*
    - *Tempo médio de espera*
  - Considere o quantum de tempo = 2 s
  - Desconsidere as prioridades

## ALGORITMOS DE ESCALONAMENTO

### **MENOR PROCESSO PRIMEIRO**

- **Shortest-job-first (SJF)** – o algoritmo associa a cada processo o tempo de uso de CPU. Quando o CPU fica disponível, é alocado o processo com o menor tempo de uso de CPU.
  - se dois processos tiverem o mesmo tempo, é utilizado o algoritmo FIFO
- Reduz o tempo médio de espera
- O próximo tempo de uso do CPU pelo processo deve ser previsto
- Uma vez que o processo recebe o CPU, ele executa até encerrar (cooperativo)

# ALGORITMOS DE ESCALONAMENTO

## MENOR PROCESSO PRIMEIRO

**Exemplo 3:**

Processo	Tempo de uso do CPU (ms)
P1	6
P2	8
P3	7
P4	3

P4	P1	P3	P2
0 3	9	16	24

Tempo de espera  
médio = 7 ms

## CONJUNTO DE PROCESSOS

Processo	Ingresso	Duração	Prioridade
P1	0	5	2
P2	0	2	3
P3	1	4	1
P4	3	1	4
P5	5	2	5

**Processos na fila de prontos**  
Unidade de tempo: segundos (s)

# ALGORITMOS DE ESCALONAMENTO

## **MENOR PROCESSO PRIMEIRO**

- **Exercício 3**
  - Represente o escalonamento dos processos ao longo do tempo usando o algoritmo **Menor processo primeiro (SJF)** e calcule:
    - *Tempo médio de conclusão/vida*
    - *Tempo médio de espera*
    - Desconsidere as prioridades



## ALGORITMOS DE ESCALONAMENTO

### **MENOR TEMPO REMANESCENTE PRIMEIRO**

- **Shortest-remaining-time-first (SRTF)** – versão preemptiva do algoritmo anterior, menor processo primeiro (SJF)
- Em uma variante preemptiva, o escalonador deve comparar a duração prevista de cada novo processo que ingressa no sistema com o tempo de processamento restante dos demais processos presentes, inclusive aquele que está executando no momento
- **Caso o novo processo tenha um tempo restante menor, ele recebe o processador.**

# ALGORITMOS DE ESCALONAMENTO

## MENOR TEMPO REMANESCENTE PRIMEIRO

**Exemplo 4:**

Processo	Tempo de chegada (ms)	Tempo de uso do CPU (ms)
P1	0	8
P2	1	4
P3	2	9
P4	3	5

P1	P2	P2	P2	P4	P1	P3
0 1	2	3	5	10	17	26

Tempo de espera  
médio = 6.5 ms

**Obs.:** P2 chega em 1ms (o tempo remanescente de P1 é 7ms > tempo P2 que é de 4ms  
(P1 é substituído)

## CONJUNTO DE PROCESSOS

Processo	Ingresso	Duração	Prioridade
P1	0	5	2
P2	0	2	3
P3	1	4	1
P4	3	1	4
P5	5	2	5

**Processos na fila de prontos**  
Unidade de tempo: segundos (s)

# ALGORITMOS DE ESCALONAMENTO

## **MENOR TEMPO REMANESCENTE PRIMEIRO**

- **Exercício 4**

- Represente o escalonamento dos processos ao longo do tempo usando o algoritmo **Menor tempo remanescente primeiro (SRTF)** e calcule:
  - *Tempo médio de conclusão/vida*
  - *Tempo médio de espera*
  - Desconsidere as prioridades

- **Exercício 5** - Fazer uma comparação dos algoritmos de escalonamento vistos (tempo médio de execução e espera e número de trocas de contexto)

# ALGORITMOS DE ESCALONAMENTO

## **PRIORIDADE**

- **Escalonamento por prioridade** – uma prioridade é associada a cada processo
  - CPU é alocado para o processo com a maior prioridade
  - Prioridade iguais são resolvidas como uma fila (FIFO)
  - Prioridade são indicadas por números inteiros (0 a 7, por exemplo)
    - **Números altos → prioridade alta**

# ALGORITMOS DE ESCALONAMENTO

## PRIORIDADE

**Exemplo 5:**

Processo	Tempo de uso do CPU (ms)	Prioridade
P1	10	3
P2	1	5
P3	2	2
P4	1	1
P5	5	4

P2	P5	P1	P3	P4
0 1	6	16	18	19

Tempo de espera  
médio = 8,2 ms

# ALGORITMOS DE ESCALONAMENTO

## **PRIORIDADE**

- Prioridades podem ser:
  - **Internas (medidas quantitativas):** limites de tempo, requerimento de memória, número de arquivos abertos, uso de E/S em relação ao uso de CPU
  - **Externas (definidas fora do SO):** importância do processo, custo de execução, etc.
- Escalonamento por prioridade pode ser **preemptivo ou não-preemptivo**.
  - Se for preemptivo, ao chegar um processo com maior prioridade do atual, este é substituído
  - Se for não-preemptivo (ou cooperativo), ao chegar um processo com maior prioridade, este é colocado no topo da fila dos processos prontos para executar

## ALGORITMOS DE ESCALONAMENTO **PRIORIDADE**

- Problema deste algoritmo é **starvation** (processos de baixa prioridade podem ficar esperando para sempre)
- Solução 1: **envelhecimento dos processos** (um processo que está esperando por muito tempo têm sua prioridade aumentada)
- Solução 2: **combinar round-robin com prioridade** (sistema roda os processos de alta prioridade e roda processos com a mesma prioridade utilizando escalonamento round-robin – ver exemplo 6)



# ALGORITMOS DE ESCALONAMENTO

## PRIORIDADE

**Exemplo 6:**

Processo	Tempo de uso do CPU (ms)	Prioridade
P1	4	1
P2	5	2
P3	8	2
P4	7	3
P5	3	1

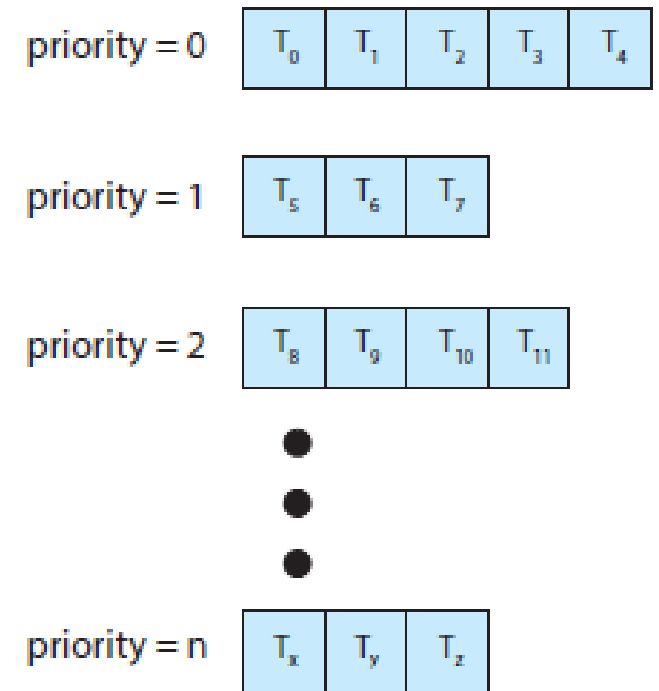
Unidade de tempo: **2 ms**

P4		P2	P3	P2	P3	P2	P3	P1	P5	P1	P5
0	7	9	11	13	15	16	20	22	24	26	27

# ALGORITMOS DE ESCALONAMENTO

## FILA MULTINÍVEL

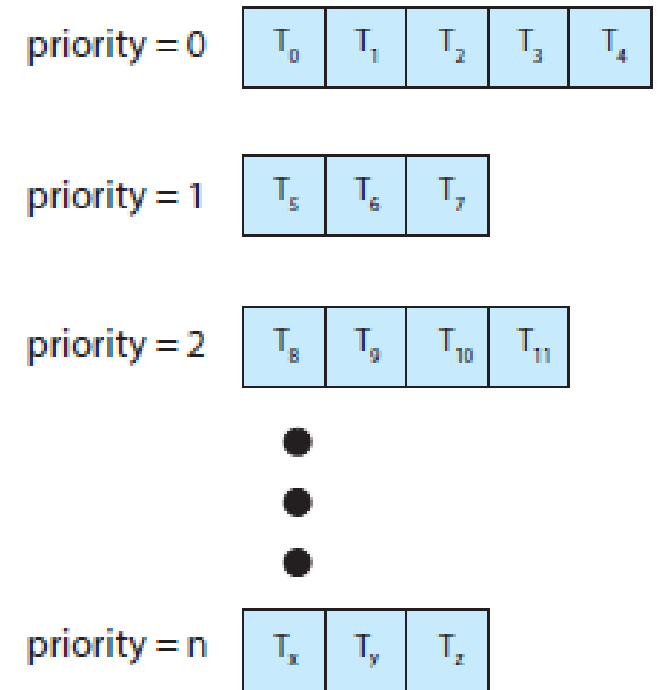
- Nos algoritmos de escalonamento por prioridade e round-robin, os processos são colocados em uma única fila, e o escalonador seleciona o processo com a maior prioridade.
- Problema:** percorrer toda fila para encontrar o processo de maior prioridade (operação custosa)
- Solução:** separar os processos com mesma prioridade em filas distintas



# ALGORITMOS DE ESCALONAMENTO

## **FILA MULTINÍVEL**

- A abordagem de fila multinível pode ser combinada com round-robin:
- Dentro de cada fila de prioridades, se tiver mais de um processo, estes são executados na ordem round-robin.



# ALGORITMOS DE ESCALONAMENTO

## **FILA MULTINÍVEL**

- **O algoritmo de Fila multinível** pode ser utilizado para particionar processos em diferentes filas baseado no tipo de processos.
  - **Foreground processes** (processos interativos)
  - **Background processes** (processos batch)
- Esses dois tipos de processos tem **requisitos de tempo de resposta diferentes** e, portanto, tem **necessidades de escalonamento diferentes**.
- Cada fila (de processos) pode ter o seu próprio algoritmo de escalonamento.

## EXERCÍCIOS DE REVISÃO

# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO I

- Considere o tempo de chegada e o tempo de uso de CPU dos processos descritos na tabela abaixo. Utilize escalonamento não-preemptivo em A e B.

Processo	Tempo de chegada (ms)	Tempo de uso do CPU (ms)
P1	0	8
P2	0,4	4
P3	1	1

- **A)** Qual o tempo médio de conclusão para esses processos considerando o algoritmo de escalonamento FIFO? **10,53 ms**

# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO I

- Considere o tempo de chegada e o tempo de execução dos processos descritos na tabela abaixo. Utilize escalonamento não-preemptivo em A e B..

Processo	Tempo de chegada (ms)	Tempo de uso do CPU (ms)
P1	0	8
P2	0,4	4
P3	1	1

- **B)** Qual o tempo médio de conclusão para esses processos considerando o algoritmo de escalonamento SJF (menor processo primeiro)? **9,53 ms**

# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO I

- Considere o tempo de chegada e o tempo de execução dos processos descritos na tabela abaixo. Utilize escalonamento não-preemptivo em A e B.

Processo	Tempo de chegada (ms)	Tempo de uso do CPU (ms)
P1	0	8
P2	0,4	4
P3	1	1

- **C)** Qual o tempo médio de conclusão para esses processos considerando o algoritmo de escalonamento SJF (menor processo primeiro) e escalonamento preemptivo (ou SRTF)? **6,33 ms**



# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO 2

- Considere a seguinte tabela de processos:

Processo	Tempo de execução	Prioridade	Tempo de chegada
P1	50	1	0
P2	20	4	20
P3	100	2	40
P4	40	3	60

- Faça o diagrama de tempo e determine o tempo médio de vida e o tempo médio de espera para os seguintes algoritmos de escalonamento:
  - **FIFO:** 42,5
  - **SJF (menor processo primeiro)** 27,5
  - **Round-robin (quantum de tempo = 30)** 42,5
  - **Prioridade (preemptivo)** 50

# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO 3

- A tabela a seguir representa um conjunto de processos prontos para utilizar um processador.
- Desenhe um diagrama de tempo da execução dos processos e calcule o **tempo médio de vida**, o **tempo médio de espera** e o **nº de trocas de contexto** para as políticas de escalonamento a seguir: **a)** FCFS cooperativa; **b)** SJF cooperativa; **c)** SJF preemptiva (SRTF); **d)** Prioridade cooperativa; **e)** Prioridade preemptiva e **f)** RR com quantum = 2.

ID do Processo	Prioridade	Ingresso	Duração (tempo de uso CPU)
P1	2	0	5
P2	3	0	4
P3	5	3	5
P4	9	5	6
P5	6	7	4

# ALGORITMOS DE ESCALONAMENTO

## EXERCÍCIO 3 - RESPOSTAS

Algoritmo	Média Tempo de Vida	Média Tempo de Espera	Nº de trocas de contexto
a) FCFS	11,4	6,6	4
b) SJF	10,6	5,8	4
c) SRTF	10,6	5,8	4
d) Prior. Coop.	11,2	6,4	4
e) Prior. Preemp.	14,4	9,6	6
f) RR qt = 2	15,0	10,2	12

## BIBLIOGRAFIA

- Tanenbaum, A. S. **Sistemas Operacionais Modernos**. Pearson Prentice Hall. 3<sup>rd</sup> Ed., 2009.
- Silberschatz, A; Galvin, P. B.; Gagne G.; **Fundamentos de Sistemas Operacionais**. LTC. 9<sup>th</sup> Ed., 2015.
- Stallings, W.; **Operating Systems: Internals and Design Principles**. Prentice Hall. 5th Ed., 2005.
- Oliveira, Rômulo, S. et al. **Sistemas Operacionais - VII - UFRGS**. Disponível em: Minha Biblioteca, Grupo A, 2010.