

An abstract network diagram consisting of numerous small grey circular nodes connected by thin grey lines. The nodes are distributed across the frame, with a higher density on the left side, creating a web-like structure that suggests a complex system or data network.

Escalonamento de CPU

Sumário

- Introdução
- Ciclos de CPU e I/O
- Escalonador de CPU
- Escalonamento
- *Dispatcher*
- Critérios de Escalonamento
- Políticas de Escalonamento



Introdução

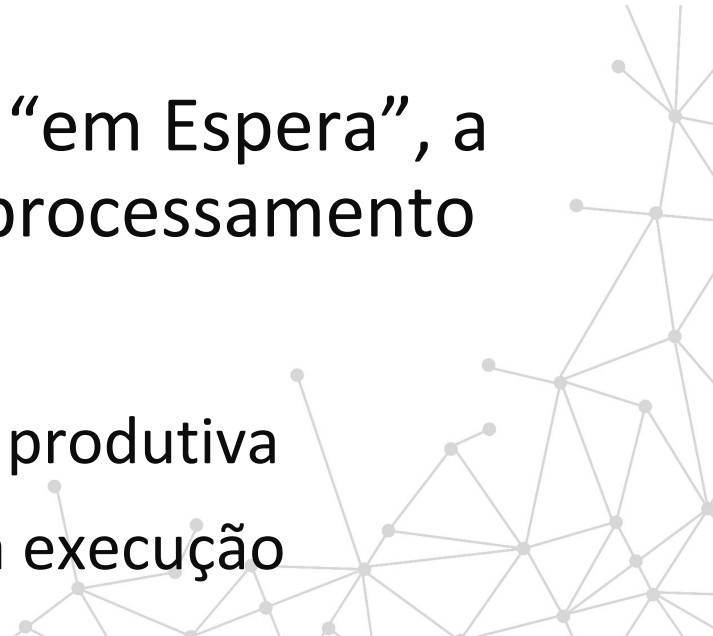
- Escalonamento é a base para Sistemas Operacionais Multiprogramados
- Escalonamento de: Processos e Threads (no nível do Kernel)
- O escalonamento é função fundamental dos Sistemas Operacionais

Quase todos os recursos são escalonados antes do uso. A CPU é o principal recurso → ponto central para o projeto de sistemas operacionais.



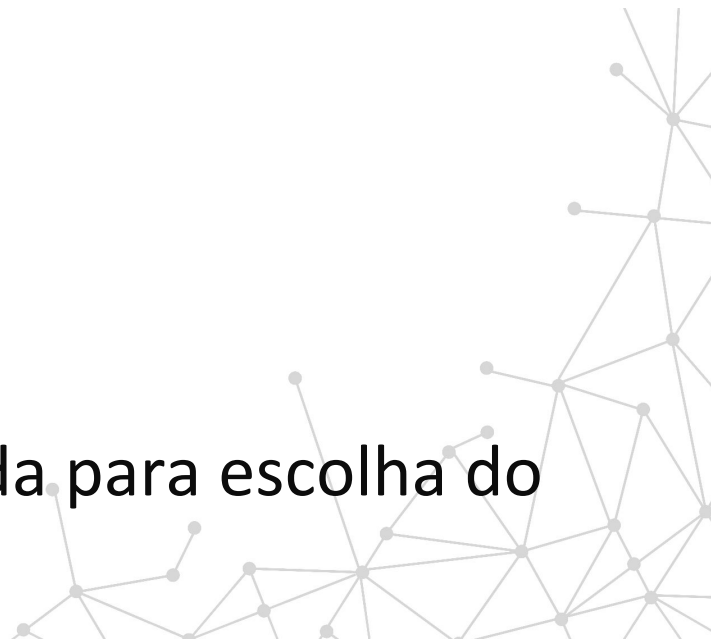
Introdução

- CPU sem trabalho útil □ tempo perdido (ociosidade)
- Processos passam para o estado “em Espera”, a CPU pode ocupar-se com outro processamento
- **Multiprogramação**
 - Procura utilizar o tempo de forma produtiva
 - Vários processos na memória para execução



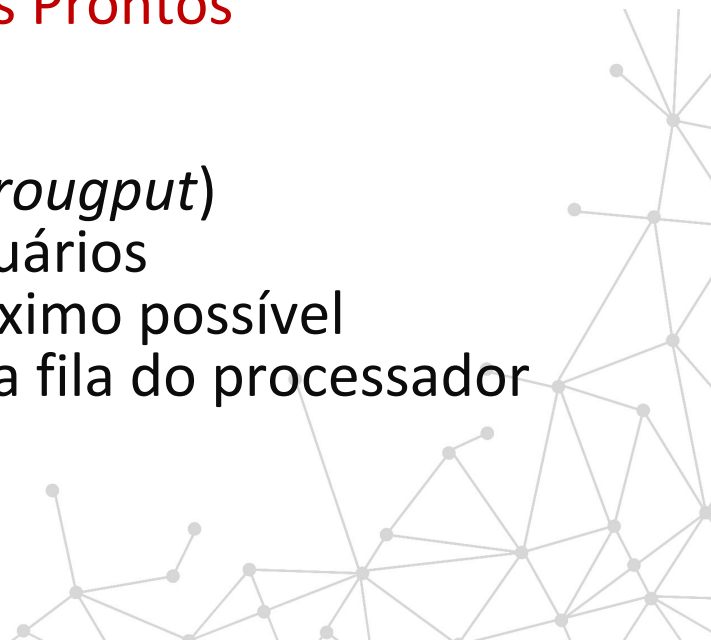
Ciclos de CPU e I/O

- A execução de um processo alterna entre
 - ciclos de CPU
 - ciclos de I/O
- Processo limitado por I/O:
 - Possui muitos ciclos de surtos curtos de CPU
- Processo limitado por CPU:
 - Alguns ciclos de surtos longos de CPU
- Essa distribuição deve ser observada para escolha do algoritmo de escalonamento



Escalonador de CPU

- Escalonamento executado pelo Escalonador de Curto Prazo
- Escolhe um processo na Fila de Processos Prontos
- Objetivos:
 - Aumentar a produção do sistema (*throughput*)
 - Diminuir o tempo de resposta aos usuários
 - Manter o processador ocupado o máximo possível
 - Diminuir o tempo médio de espera na fila do processador (*turnaroundtime*)



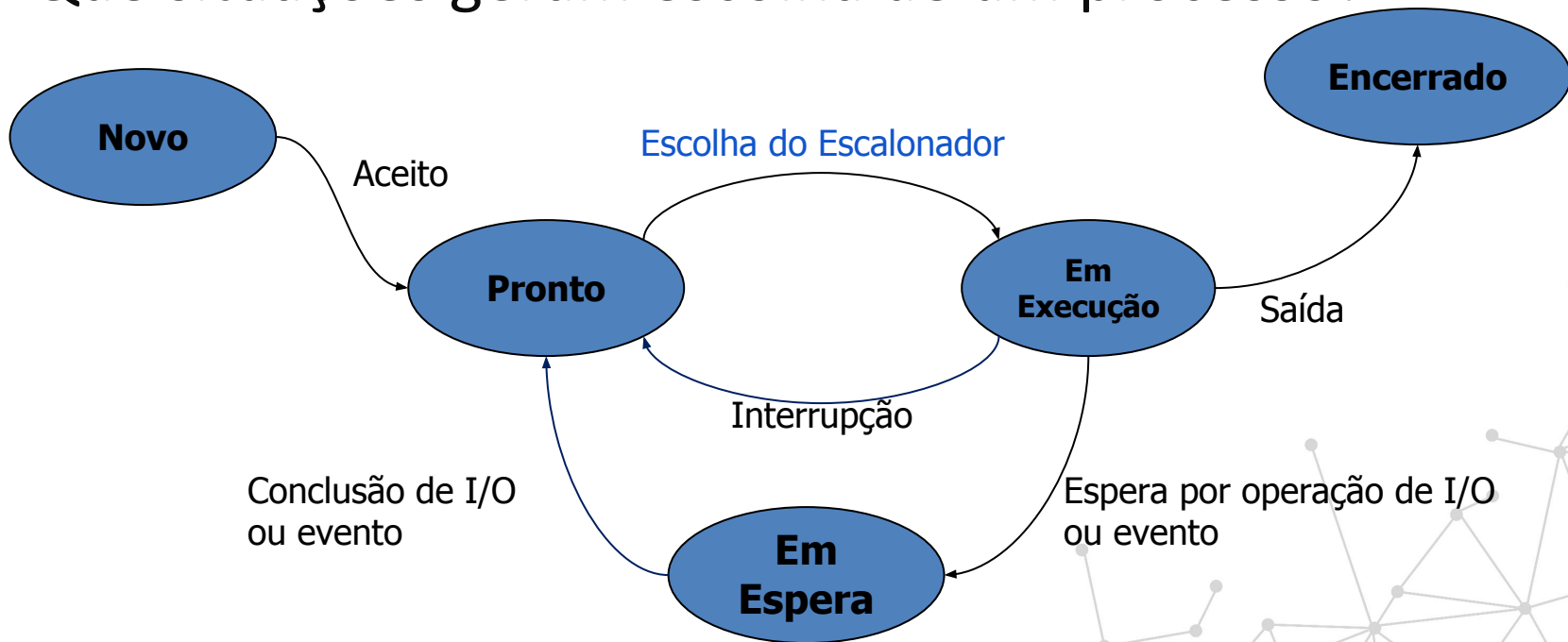
Escalonador de CPU

- Uma fila de escalonamento não necessariamente é implementada como FIFO
- Pode ser:
 - Fila de prioridades
 - Lista encadeada desordenada
 - Árvore



Escalonador de CPU

- Que situações geram escolha de um processo?

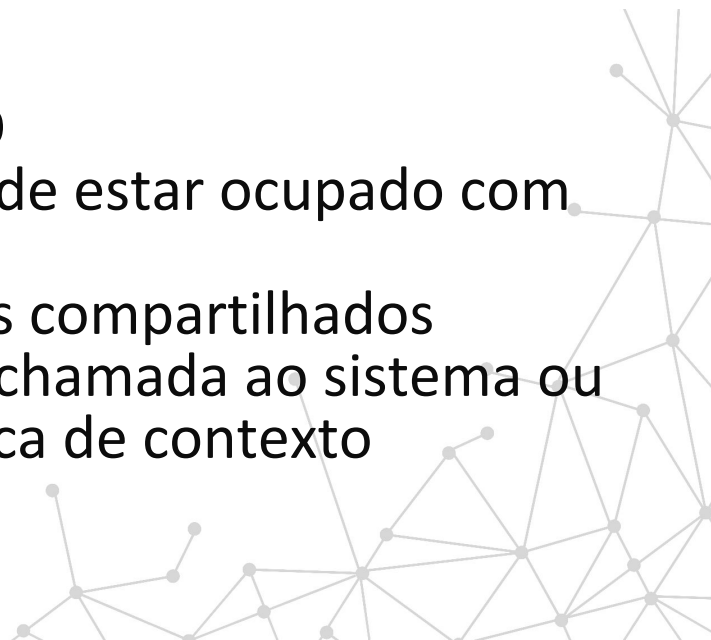


Escalonamento

- O Escalonamento pode ser:
 - **Preemptivo:**
 - As interrupções estão habilitadas
 - A execução pode ser interrompida
 - **Não-preemptivo ou Cooperativo**
 - O processo, após ter a CPU alocada, mantém o uso até liberá-la
 - Término da execução
 - Passagem para o estado Em espera
 - Não é necessário o temporizador de hardware

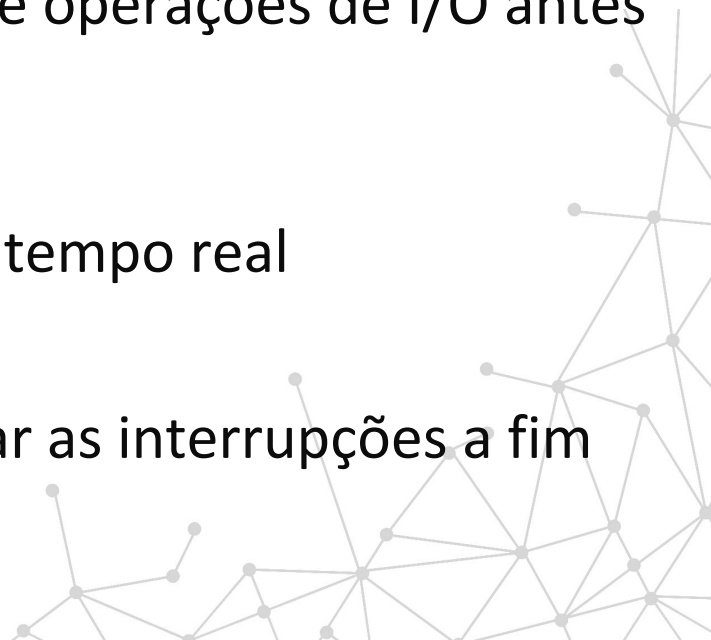
Escalonamento

- No Escalonamento Preemptivo :
 - Há custo associado à coordenação do acesso aos dados compartilhados
- Influencia no projeto do kernel do SO
 - Chamada ao sistema: o kernel pode estar ocupado com outro processo
 - Podem gerar mudanças em dados compartilhados
 - É necessária a conclusão de uma chamada ao sistema ou operação I/O antes de fazer a troca de contexto



Escalonamento

- Devido ao escalonamento preemptivo, alguns SO são implementados de forma a esperar a conclusão de uma chamada ao sistema ou de um bloco de operações de I/O antes de ser efetuada a troca de contexto.
 - Simplicidade do Kernel
 - Versões do Unix
 - Não eficiente para sistemas de tempo real
- Há possibilidade de habilitar/desabilitar as interrupções a fim de proteger uma seção de código

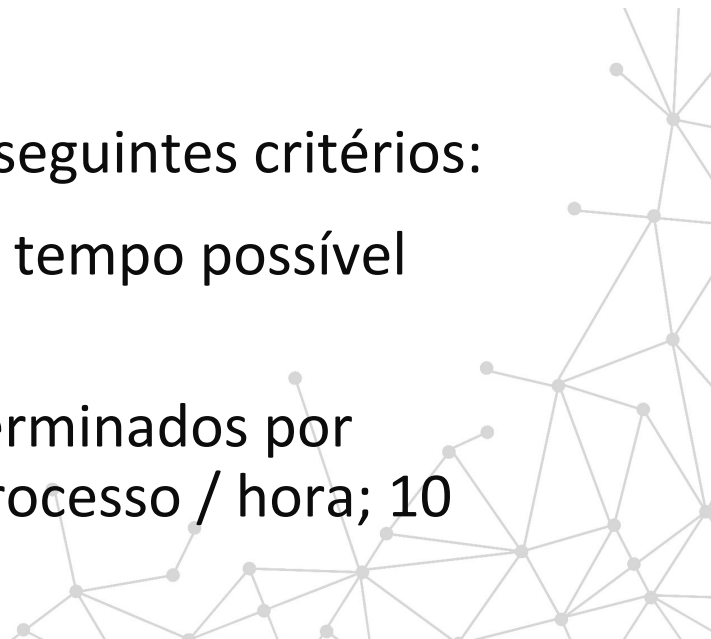


Dispatcher = Executor

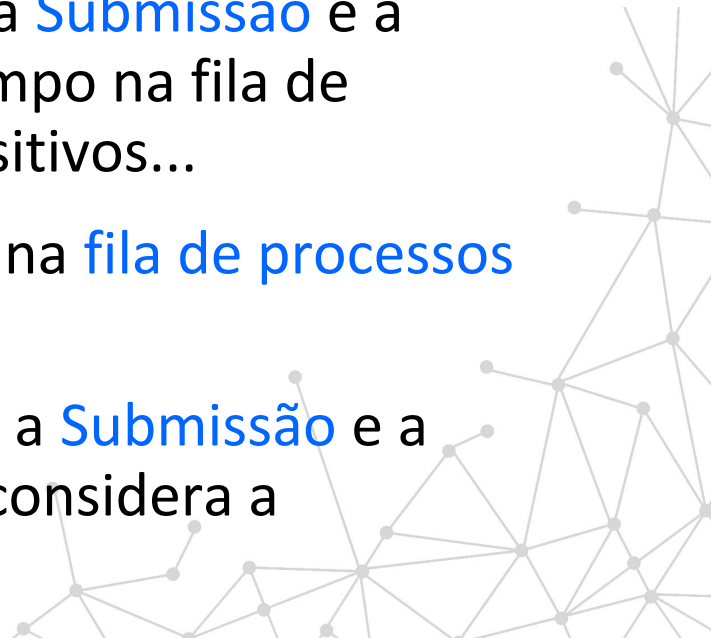
- Módulo que dá o controle da CPU ao processo escalonado. Faz:
 - passagem para o modo de usuário / supervisor;
 - troca de contexto:
 - Salva no descritor de processo ou na pilha do processo
 - Ao ser restaurado, o *Program Counter* (PC) indica o ponto de continuação da execução
 - Mudança para posição adequada no programa do usuário conforme o *Program Counter* (PC)

Critérios de Escalonamento

- São critérios utilizados para comparação de algoritmos / políticas de escalonamento
- O escalonamento procura **maximizar** os seguintes critérios:
 - **Utilização de CPU:** Deve ficar o maior tempo possível ocupada
 - **Throughput:** Número de Processos terminados por unidade de tempo. Por exemplo: 1 processo / hora; 10 processos / segundo



Critérios de Escalonamento

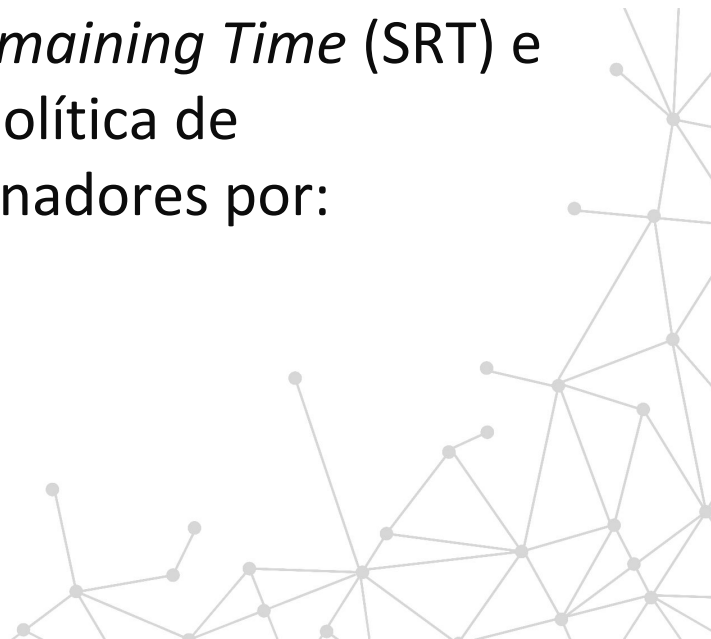
- O escalonamento procura **minimizar** os seguintes critérios:
 - **Tempo de Retorno:** É o tempo entre a **Submissão** e a **Conclusão** do Processo. Envolve o tempo na fila de prontos, em execução, usando dispositivos...
 - **Tempo de Espera:** Tempo esperando na **fila de processos prontos** (não afeta operações de I/O)
 - **Tempo de Resposta:** É o tempo entre a **Submissão** e a **Primeira Resposta** do processo (não considera a velocidade do dispositivo de saída)
- 
- A decorative network diagram is located in the bottom right corner of the slide. It consists of a series of interconnected nodes (small grey circles) and edges (thin grey lines), forming a complex web-like structure that extends towards the right edge of the frame.

Políticas de Escalonamento

- As políticas de escalonamento constituem-se em algoritmos de escalonamento que selecionam quais processos devem utilizar a CPU em determinado instante. As principais políticas são:
 - First-Come, First-Served (FCFS) ou First-In, First-Out (FIFO)
 - Shortest Job First (SJF) ou Shortest Job Next (SPN).
 - Escalonamento por prioridade
 - Round-Robin ou Revezamento Circular ou Fatia de Tempo

Políticas de Escalonamento

- Existem outros algoritmos de escalonamento como *High Response Ratio Next* (HRRN), *Shortest Remaining Time* (SRT) e também a combinação de mais de uma política de escalonamento, caracterizando os escalonadores por:
 - Múltiplas Filas
 - Múltiplas Filas com realimentação



Algoritmos de Escalonamento

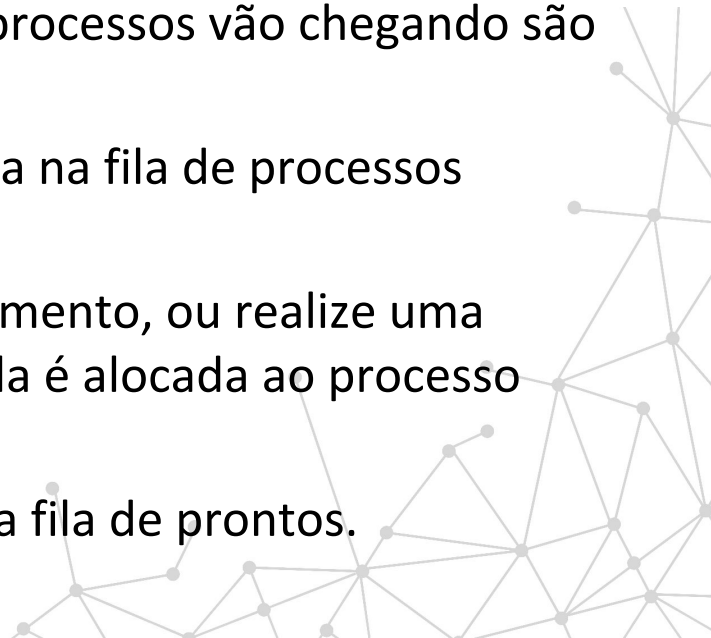
- Escalonamento *First-Come, First-Served* (FCFS) ou *First-In, First-Out* (FIFO)
- Escalonamento *Shortest Job First* (SJF)
- Escalonamento por Prioridade
- Escalonamento Round-Robin (Revezamento)
- Escalonamento por Múltiplas Filas
- Escalonamento por Múltiplas Filas com realimentação



First-Come, First-Served (FCFS) ou First-In, First-Out (FIFO)



First-Come, First-Served (FCFS) ou First-In, First-Out (FIFO)

- É o algoritmo mais simples de implementar: o Processo que solicita a CPU primeiro, recebe primeiro, ou seja, conforme os processos vão chegando são inseridos no fim da fila de prontos.
 - É implementado por uma FIFO: um processo entra na fila de processos prontos e seu PCB é ligado ao final da fila.
 - O processo utiliza CPU até o término do processamento, ou realize uma chamada ao sistema. Quando a CPU é liberada, ela é alocada ao processo que está no início da fila.
 - O processo que entra em execução é removido da fila de prontos.
- 

First-Come, First-Served (FCFS) ou First-In, First-Out (FIFO)

- Geralmente, apresenta tempo de espera longo, pois não favorece os processos I/O bound. Exemplo:

<u>Processo</u>	<u>Tempo de Espera</u>	<u>Duração Surto CPU</u>
P1		20
P2		3
P3		3

Diagrama de Gantt



Tempo de Espera Médio: 14,33 ms

First-Come, First-Served (FCFS) ou First-In, First-Out (FIFO)

Situação Dinâmica:

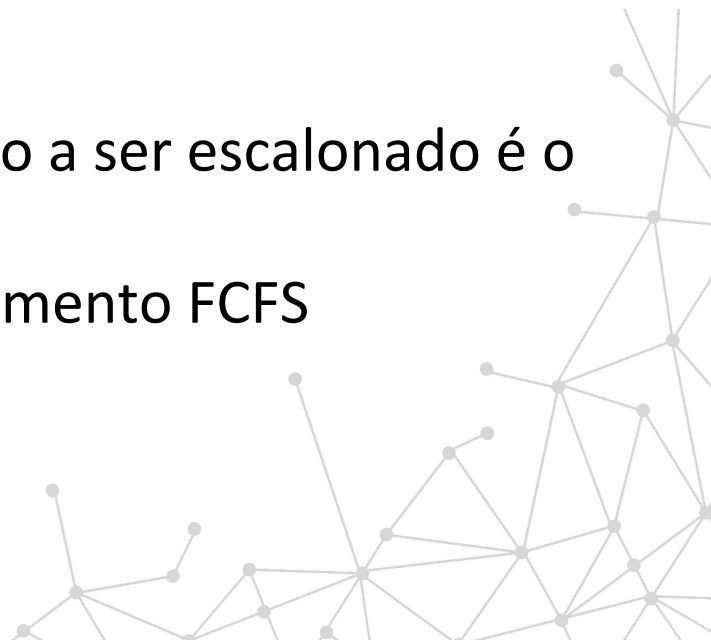
- Um Processo limitado pela CPU: obtém e detém a CPU.
- Muitos Processos limitados por I/O: terminam sua operação de I/O e passam para Prontos
 - Dispositivos de I/O ociosos
- Processo limitado pela CPU termina e passa para operação de I/O
- Todos os Processos limitados I/O são executados pois tem surtos curtos de CPU
 - Executam rapidamente e voltam para a fila de I/O
 - CPU ociosa
- Processo limitado pela CPU mais uma vez vai para fila de prontos e retorna para a CPU
- Processos limitados por I/O esperam na fila de prontos até a CPU estar disponível

Shortest Job First - SJF



Shortest Job First - SJF

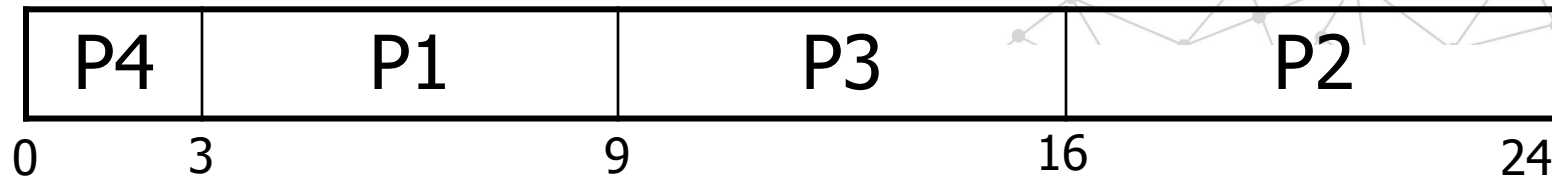
- Para cada processo é associada a duração de seu próximo ciclo de surto de CPU
- Quando a CPU está disponível, o processo a ser escalonado é o que tem surto de menor duração.
 - Se houver empate, usa-se o escalonamento FCFS
- É um algoritmo **ótimo**
 - Tempo de Espera Médio é mínimo



Shortest Job First - SJF

<u>Processo</u>	<u>Tempo de Espera</u>	<u>CPU</u>
P1		6
P2		8
P3		7
P4		3

Diagrama de Gantt



Tempo de Espera Médio : 10,25 ms

Shortest Job First - SJF

- Problema

- Para o escalonamento de CPU (escalonador de curto prazo) não existe como saber a duração do próximo ciclo de surto de CPU
- Em sistemas *batch*, o usuário pode especificar esse tempo quando submete o job
- Pode-se fazer uma previsão: espera-se que o próximo ciclo de surto de CPU seja semelhante em duração aos anteriores

Shortest Job First - SJF

- Essa previsão pode ser feita por uma **média exponencial** dos ciclos de surtos anteriores:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n$$

onde:

τ : valor previsto (histórico)

t_n : duração do n - ésimo surto (informação recente)

α : peso relativo

t_0 : é estimado ou é a média geral do sistema

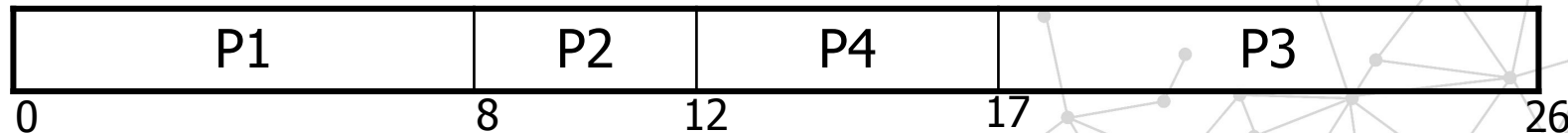
- Geralmente, $\alpha = 0.5$
 - pode variar de 0 a 1: se for 0 não considera o passado, se for 1, considera apenas o último ciclo do processador.

Shortest Job First - SJF

- O algoritmo SJF é não-preemptivo. Alguns autores como SILBERSCHATZ (2004) consideram também uma versão do algoritmo SJF preemptivo.

<u>Processo</u>	<u>Instante da Chegada</u>	<u>Duração Surto CPU</u>
P1	0	8
P2	1	4
P3	2	9
P4	3	5

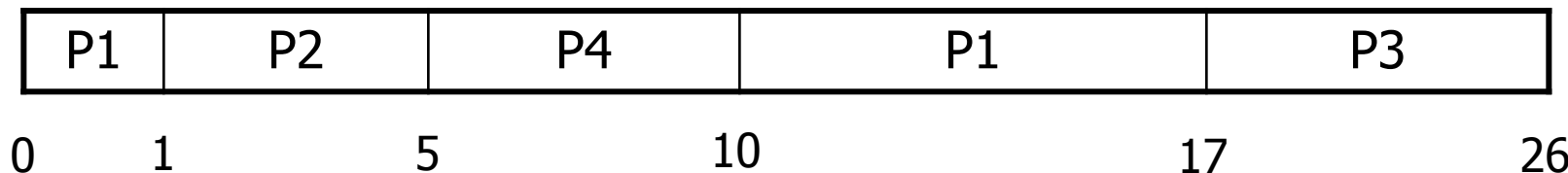
Diagrama de Gantt – Execução Não Preemptiva



Tempo de Espera Médio : 7,75 ms

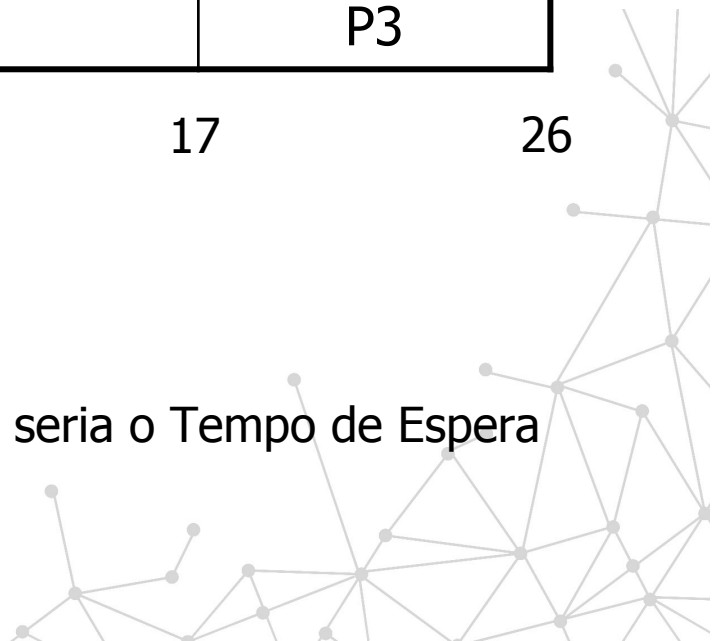
Shortest Job First - SJF

Diagrama de Gantt – Execução Preemptiva



Tempo de Espera Médio : 6,5 ms

E se fosse utilizado o escalonamento FCFS, qual seria o Tempo de Espera Médio, para esse caso?



Escalonamento por Prioridade



Escalonamento por Prioridade

- SJF é um caso de escalonamento por prioridade:
 - A prioridade é dada aos processos de menores ciclos de surtos de CPU
- Na política de escalonamento por Prioridade, uma prioridade é associada a cada processo e a CPU é alocada ao processo de mais alta prioridade.
 - Se houver empate, o escalonamento FCFS é utilizado

Escalonamento por Prioridade

- Prioridades
 - São definidas em faixas de valores:
 - 0..10
 - 0..255
- É pré-definido se a prioridade mais alta é determinada pelo maior ou menor número
 - Consideraremos o menor número como sendo a prioridade mais alta

Escalonamento por Prioridade

- As Prioridades podem ser:

Interna

Representam alguma quantidade mensurável para calcular a prioridade do processo (tempo, memória, arquivos abertos, razão entre surtos I/O e CPU)

Externa

Definida por critérios externos ao Sistema Operacional (importância do processo, tipo, quantidade paga pelo uso no computador, departamento...)

Escalonamento por Prioridade

- As Prioridades podem ser:

Interna

Representam alguma quantidade mensurável para calcular a prioridade do processo (tempo, memória, arquivos abertos, razão entre surtos I/O e CPU)

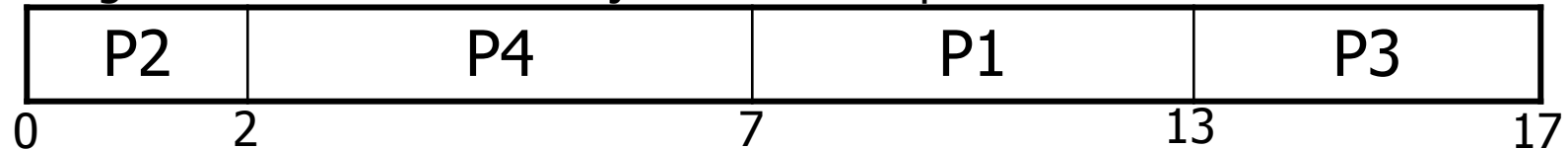
Externa

Definida por critérios externos ao Sistema Operacional (importância do processo, tipo, quantidade paga pelo uso no computador, departamento...)

Escalonamento por Prioridade

<u>Processo</u>	<u>Prioridade</u>	<u>Duração Surto CPU</u>
P1	3	6
P2	1	2
P3	4	4
P4	2	5

Diagrama de Gantt – Execução Não Preemptiva



Tempo de Espera Médio : 5,5 ms

Escalonamento por Prioridade

- Pode ser não-preemptivo ou preemptivo.
- Por exemplo, na seguinte situação: **um processo chegou no sistema durante a execução de outro e o processo que chegou tem prioridade mais alta**
- No escalonamento **não-preemptivo**:
 - o processo é colocado no início da fila, pois tem prioridade mais alta;
- No escalonamento **preemptivo**:
 - o processo em execução é interrompido e a CPU é alocada ao novo processo.

Escalonamento por Prioridade

- **Problema:** Bloqueio por tempo indefinido ou *starvation* ou postergação indefinida
 - Processos de baixa prioridade podem ficar esperando pela CPU indefinidamente.
- **Solução**
 - Técnica de Envelhecimento (conhecida como *aging*) □
consiste em aumentar gradativamente a prioridade dos processos que estão na fila de Prontos durante muito tempo, para favorecer sua execução.

Escalonamento *Round – Robin (RR)* ou Revezamento Circular ou Fatia de Tempo




Escalonamento *Round – Robin (RR)*

- Adequado para Sistemas de Tempo Compartilhado
- É o escalonamento FCFS acrescido da preempção para alternar entre os processos
- O sistema define um **quantum** de tempo (fatia)
 - 10, 100 ms
- O processador percorre a Fila de Prontos, alocando os processos por um intervalo de tempo de até um quantum.
- É mantido por FIFO: novos processos são adicionados ao final da fila

Escalonamento *Round – Robin (RR)*

- **Escalonador**

- a. Seleciona o primeiro processo da fila
- b. Aciona um temporizador para interromper o processamento depois de um quantum
- c. Submete o processo à CPU



Enquanto isso... podem ocorrer duas situações

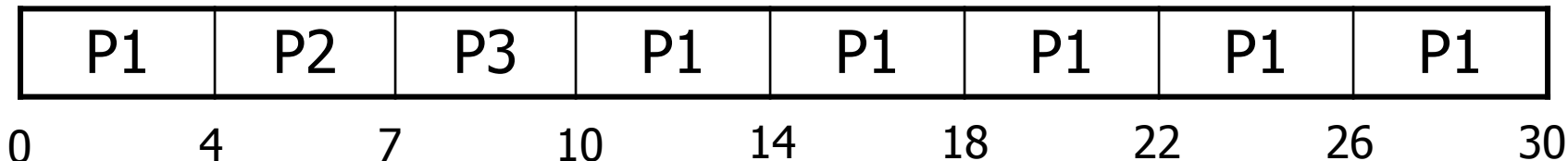
Escalonamento *Round – Robin (RR)*

1. Surto de CPU de menos de um quantum
 - O próprio processo libera CPU
 - O próximo processo é escalonado
2. Surto de CPU excede um quantum
 - Temporizador esgotará o tempo de execução do processo atual
 - Acontece Interrupção do SO
 - Há Troca de contexto dos processos
 - Processo interrompido vai para o final da fila de prontos
 - O próximo processo é escalonado

Escalonamento *Round – Robin (RR)*

<u>Processo</u>	<u>Tempo de Espera</u>	<u>Duração Surto CPU</u>
P1		24
P2		3
P3		3

Diagrama de Gantt – Quantum = 4



Tempo de Espera Médio : 5,66 ms

→ Geralmente é longo!

Compare o tempo de espera com o Algoritmo SJF

Escalonamento *Round – Robin (RR)*

- O Algoritmo Round-Robin é preemptivo
- Se houver n processos na Fila de Prontos e q quantum
- Então
 - Cada processo terá $1/n$ do tempo da CPU em lotes de no máximo q unidades de tempo
 - Cada processo deve esperar no máximo $(n-1)q$ unidades de tempo até o próximo quantum

Escalonamento *Round – Robin (RR)*

- Desempenho depende do tamanho do quantum
 - Se o quantum for muito grande □ ficará semelhante ao FCFS
 - Se o quantum for muito pequeno □ compartilhamento de processador □ como se cada um dos n processos tivesse seu próprio processador executando a $1/n$ da velocidade real do processador (execução com travamentos: executa e para, executa e para...)
- Tempo de Retorno também depende do tamanho do quantum

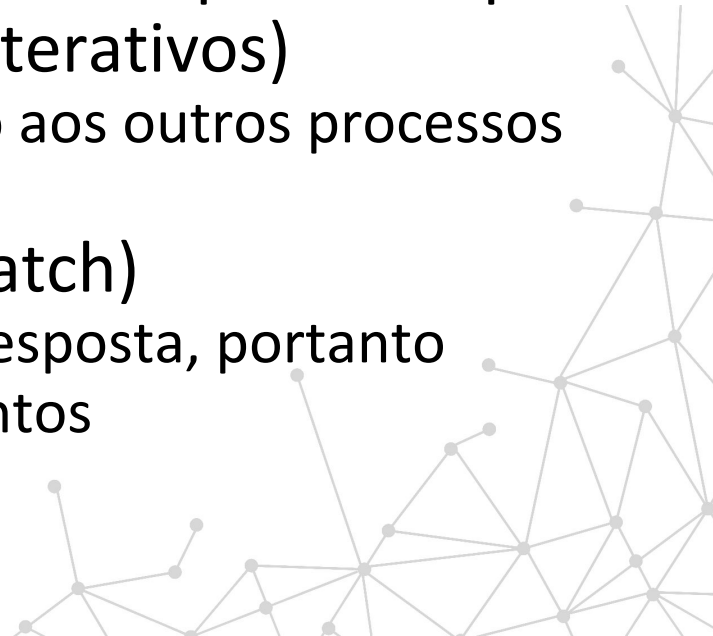
80 % dos surtos de CPU < quantum

Escalonamento por Múltiplas Filas



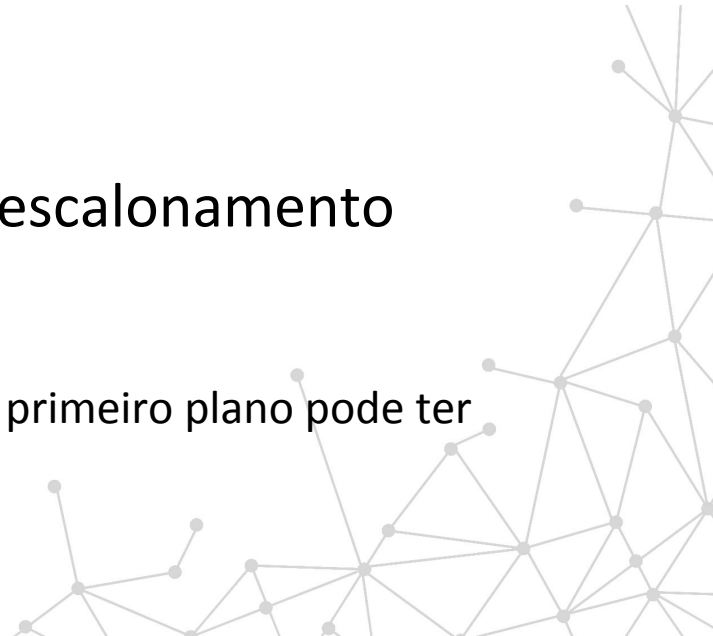
Escalonamento por Múltiplas Filas

- Utilizado em casos onde os processos podem ser classificados em **diferentes grupos** como por exemplo:
 - Processos de Primeiro Plano (interativos)
 - Podem ter prioridade em relação aos outros processos
 - Processos de Segundo Plano (batch)
 - Possuem diferentes tempos de resposta, portanto podem ter escalonamentos distintos



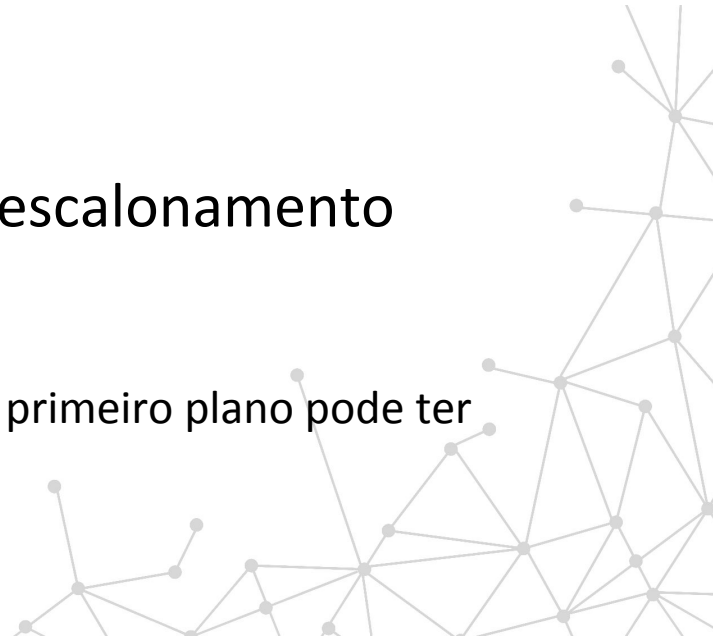
Escalonamento por Múltiplas Filas

- A fila de processos prontos é dividida em várias filas separadas por um dos critérios:
 - Prioridade
 - Tamanho
 - Tipo
- Cada fila possui seu próprio algoritmo de escalonamento
- Deve haver escalonamento entre as filas
 - Geralmente Preemptivo de Prioridade Baixa (fila do primeiro plano pode ter prioridade absoluta sobre a de segundo plano)ou
 - O tempo é fracionado entre as filas



Escalonamento por Múltiplas Filas

- A fila de processos prontos é dividida em várias filas separadas por um dos critérios:
 - Prioridade
 - Tamanho
 - Tipo
- Cada fila possui seu próprio algoritmo de escalonamento
- Deve haver escalonamento entre as filas
 - Geralmente Preemptivo de Prioridade Baixa (fila do primeiro plano pode ter prioridade absoluta sobre a de segundo plano)ou
 - O tempo é fracionado entre as filas



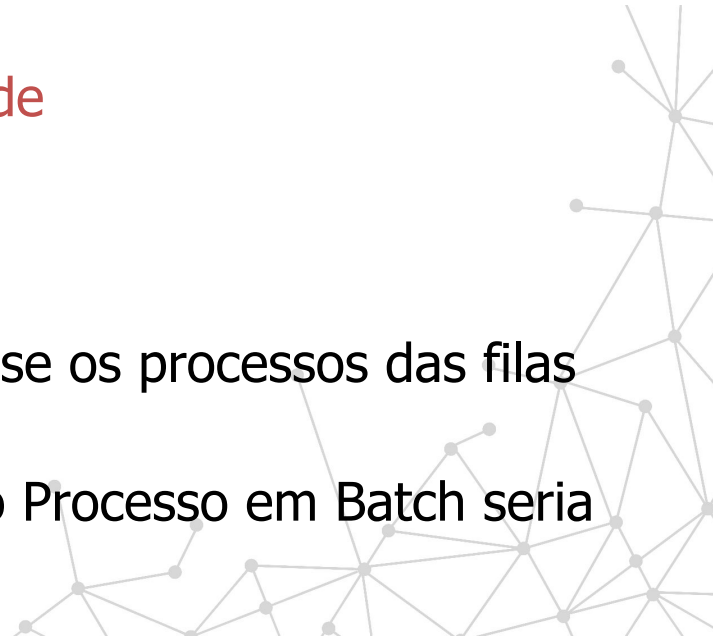
Escalonamento por Múltiplas Filas

- Exemplo com 5 filas
 - Processos do Sistema
 - Processos Interativos
 - Processos em Batch
 - Processos Secundários



Prioridade

- ◆ Processos na fila Batch só poderiam executar se os processos das filas do Sistema e Interativos estivessem vazias.
- ◆ Se um Processo do Sistema entrasse na fila, o Processo em Batch seria interrompido

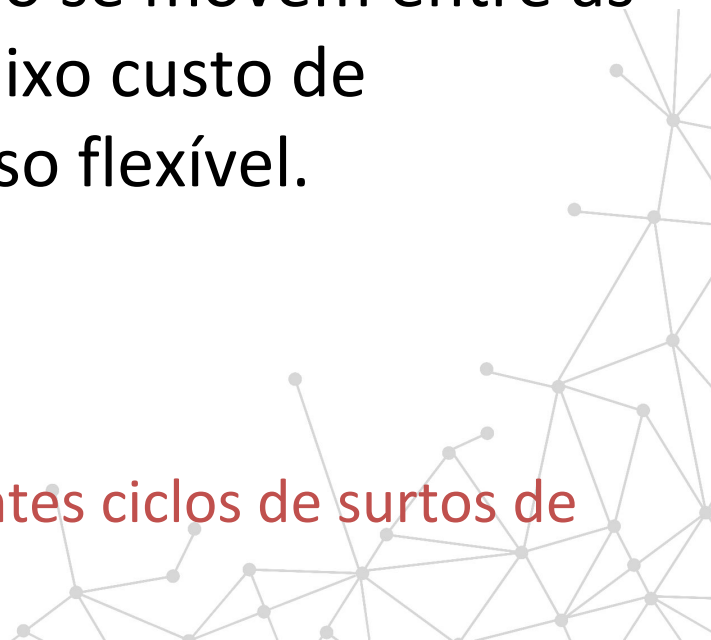


Escalonamento por Múltiplas Filas com realimentação

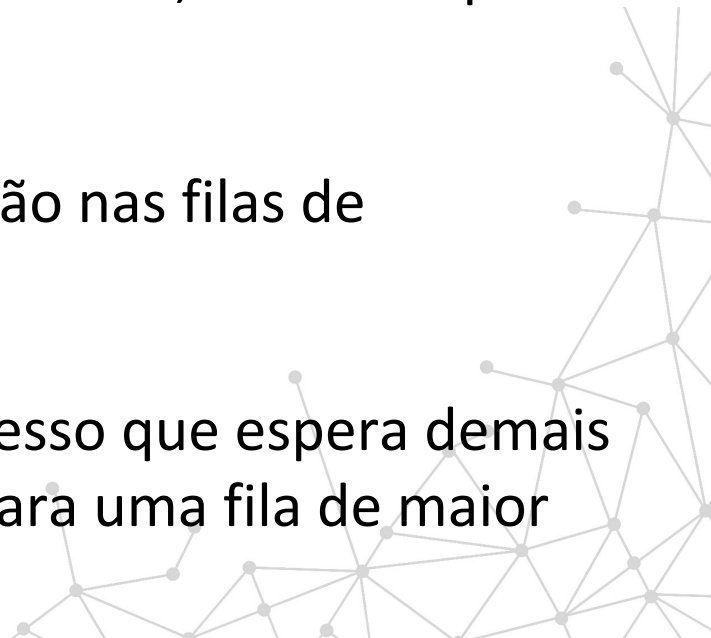


Escalonamento por Múltiplas Filas com realimentação

- No caso anterior, os processos não se movem entre as filas. Apesar de apresentar um baixo custo de escalonamento, não é um processo flexível.
- Nesse escalonamento
 - Processo se move entre as filas
 - Processos são separados por diferentes ciclos de surtos de CPU

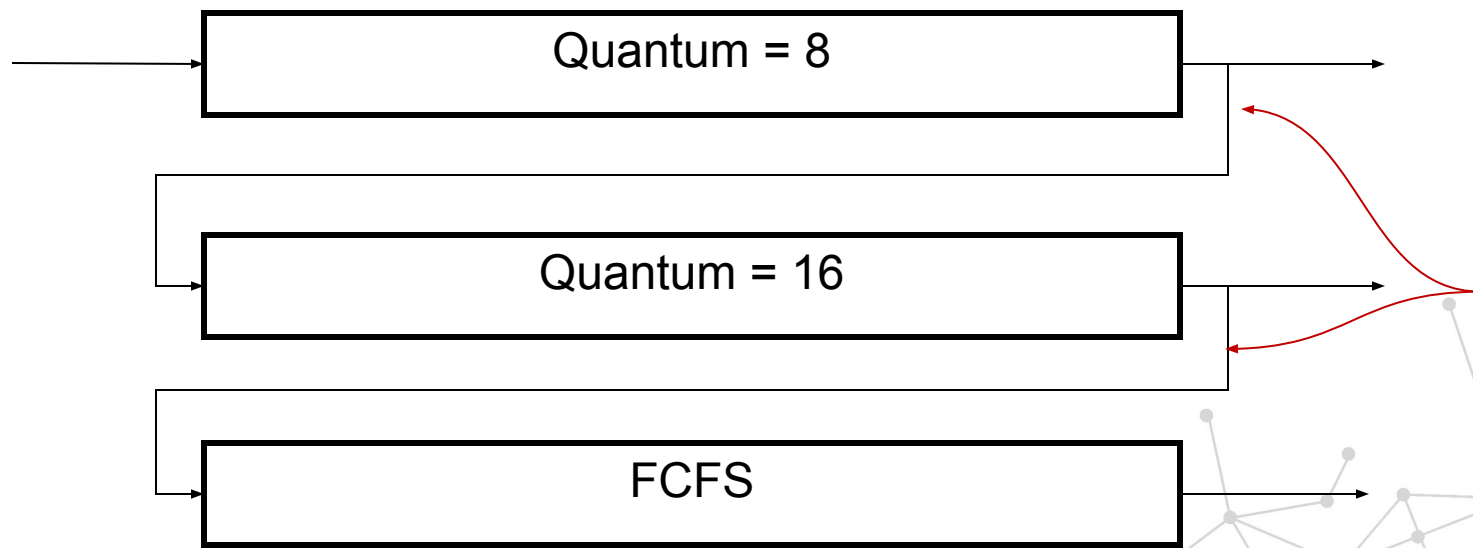


Escalonamento por Múltiplas Filas com realimentação

- Se um processo utilizar surto de CPU excessivo, é movido para uma fila de menor prioridade
 - Logo: os processos limitados por I/O estão nas filas de prioridade mais alta
 - Possui técnica de Envelhecimento: processo que espera demais em uma fila de baixa prioridade passa para uma fila de maior prioridade
- 

Escalonamento por Múltiplas Filas com realimentação

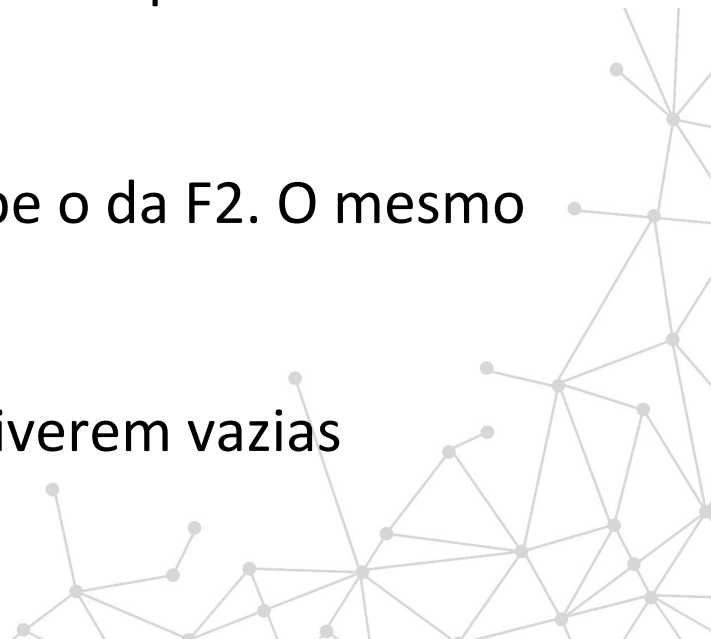
Exemplo: 3 filas (F0, F1, F2)



Se o processo não terminar no tempo especificado, vai para a próxima fila

Escalonamento por Múltiplas Filas com realimentação

- Executa todos os processos das filas nessa sequência $F0 \rightarrow F1 \rightarrow F2$
- Se chegar um processo na F1, interrompe o da F2. O mesmo acontece se chegar um processo na F0
- F2 somente executa quando F0 e F1 estiverem vazias



Escalonamento por Múltiplas Filas com realimentação

- Considerações sobre o exemplo
 - Prioridade mais alta é dada aos processos com Surto de CPU ≤ 8 ms
 - Processos com Surto de CPU entre 8 e 24 ms são rapidamente atendidos
 - Processos com Surto de CPU maior que 24 ms vão para a F2
 - Os processos limitados por I/O estão nas filas de prioridade mais alta

Escalonamento por Múltiplas Filas com realimentação

- Esse escalonador é definido pelos seguintes parâmetros
 - Número de filas
 - Algoritmo de escalonamento para cada fila
 - Método para remover/rebaixar processos
 - Método para determinar em que fila entrará determinado processo

