

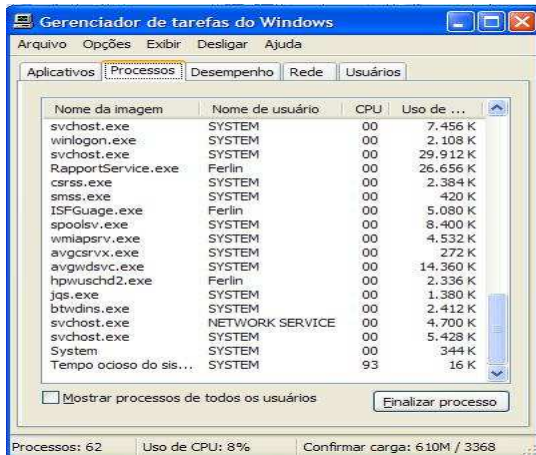
Escalonamento de Processos

Prof. Edson Pedro Ferlin

Agradecimento ao Prof. Osmar Betazzi Dordal

- **Objetivos**
 - Entender os modelos de escalonamento de processos
- **Conteúdos**
 - Escalonamento de processos

Processos

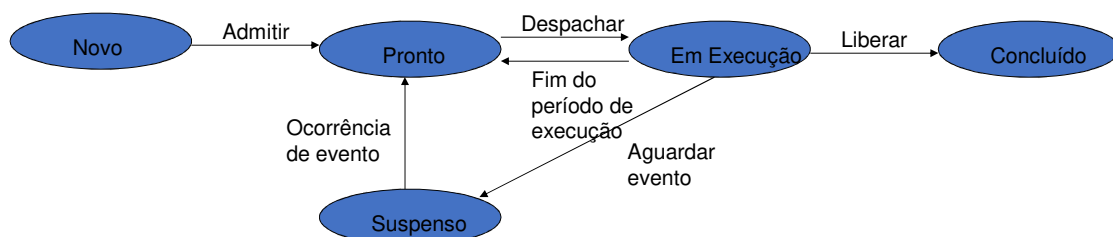


3

Escalonamento de Processos

Prof. Edson Pedro Ferlin

Escalonador



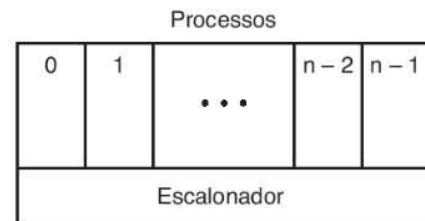
4

Escalonamento de Processos

Prof. Edson Pedro Ferlin

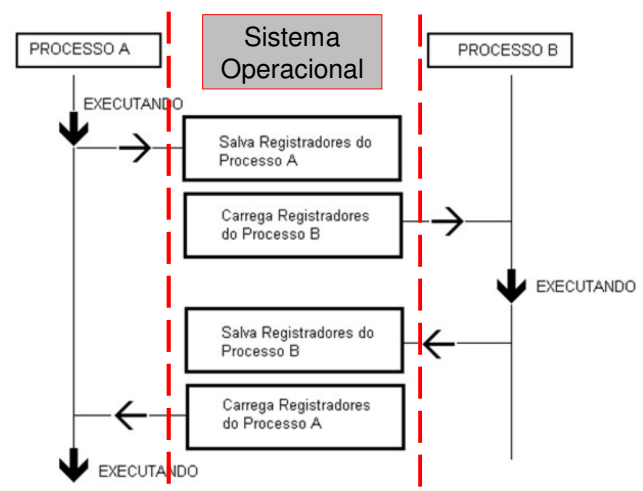
Escalonamento de Processos

- Para existir o escalonamento de processos existe um programa chamado **escalonador**.
- O **escalonador de processos** que escolhe qual será o próximo processo a ser executado.
- São várias técnicas (algoritmos) para o escalonamento de processos.
- Isto é realizado em nível mais baixo do Sistema Operacional.



Mudança de Contexto

- A mudança de contexto leva a um *overhead* de tempo (**tarefa custosa**).
- É preciso salvar todas as informações do processo que está **deixando a CPU** e **entrando em seu BCP – Bloco de Controle de Processos**.
- Salvar o conteúdo dos registradores.



Componentes Envolvidos no Escalonamento

- **Dispatcher** (Despachante)
 - Armazena e recupera o contexto.
 - Atualiza as informações do BCP.
 - Processo relativamente rápido – 0,1 milissegundo.
- **Scheduler** (Escalonador)
 - Escolhe a próxima tarefa a receber o processador.
 - Parte mais demorada.

Chamada do Escalonador

- O Escalonador é chamando quando:
 - Um novo processo é **criado**;
 - Um **processo cria outro**;
 - Execução do Pai ou Filho?
 - Um processo **chegou ao fim** e um processo pronto deve ser executado;
 - Um processo é **bloqueado**, ou deve ser executado
 - Dependência de E/S.
- **Quando E/S ocorre o escalonador deve:**
 - Executar o processo que estava esperando esse evento **OU**
 - Continuar executando o processo que já estava sendo executado **OU**
 - Executar um terceiro processo que esteja pronto para ser executado.

Categorias do Escalonador

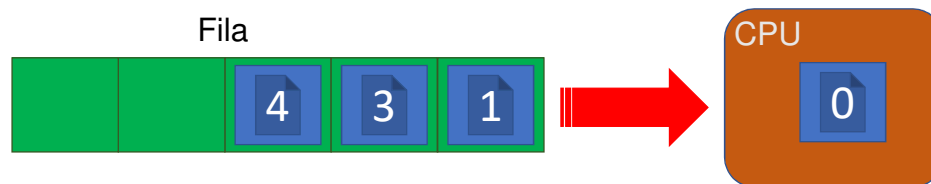
- **Preemptivo (“Para o que está fazendo”):**
 - Quando um processo pode, por algum motivo, perder seu uso da CPU;
 - Provoca uma interrupção forçada de um processo para que outro possa usar a CPU.
- **Não Preemptivo (“Não para o que está fazendo”):**
 - Permite que o processo sendo executado continue executando;
 - As condições de parada são:
 - Termine de executar;
 - Solicite uma operação de E/S;
 - Libere explicitamente o processador, voltando à fila de prontos.

Algoritmos de Escalonamento

- Sistemas Batch – Lote
 - *Fist-Come / Fist-Served – FIFO*
- Sistemas Interativos
 - *Shortest Job Fist – SJF*
- Sistemas de Tempo Real
 - *Shortest Remaining Time Next – SRTN*

Fist-Come / Fist-Served – FIFO

- O processo **executa até o fim** – **Não preemptivo**.
- São executados na CPU seguindo a ordem de requisição.
- Fácil de entender e programar.
- Gera ineficiência quando processos são demorados na sua execução.
Só é interrompido por E/S ou quando é finalizado.



Shortest Job Fist – SJF

- O processo **executa até o fim** – **Não preemptivo**.
- Deve ser previsto o tempo de execução.
- O menor processo da lista é executado primeiro.
- Menor *turnaround* (médio)
 - Tempo total necessário desde a submissão até o final do resultado.
- Todos os *jobs* precisam ser conhecidos de antemão.
- Se *jobs* curtos começarem a chegar, os longos podem demorar para serem executados.

Shortest Remaining Time Next – SRTN

- O processo **pode deixar a CPU sem finalizar – Preemptivo.**
- É a versão do *Shortest Job First*, porém, preemptiva.
- Processos com menor tempo de execução são executados primeiro.
- Se um processo novo chega e seu tempo de execução é menor do que o do processo corrente na CPU, a CPU suspende o processo corrente e executa o processo que acabou de chegar.
- **Processos que consomem mais tempo podem demorar muito para serem finalizado**, caso muitos processos pequenos cheguem (**starvation – inanição**).

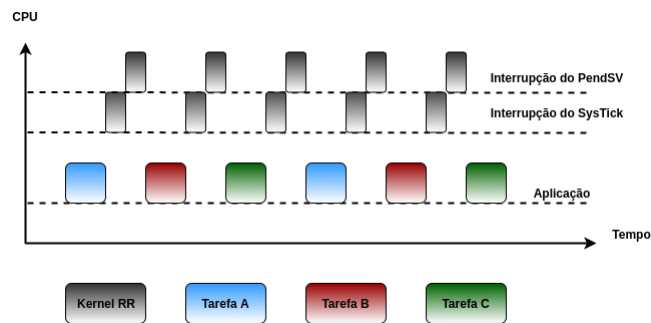
Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- *Lottery*
- *Fair-Share*

Escalonamento em Sistemas Iterativos

• **Round-Robin**

- Gira os processos com fatias de tempo iguais.
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- *Lottery*
- *Fair-Share*



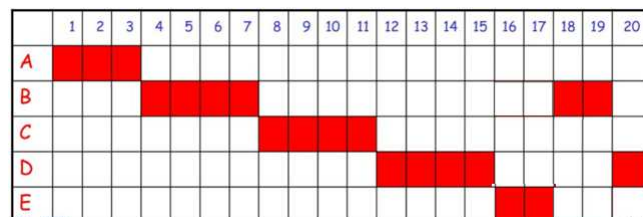
Escalonamento em Sistemas Iterativos

• **Round-Robin**

- Gira os processos com fatias de tempo iguais.
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- *Lottery*
- *Fair-Share*

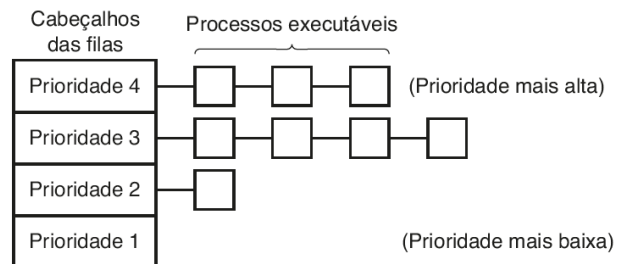
Quantum = 4

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Escalonamento em Sistemas Iterativos

- *Round-Robin*
- **Prioridade**
 - Cada processo possui uma prioridade.
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- *Lottery*
- *Fair-Share*



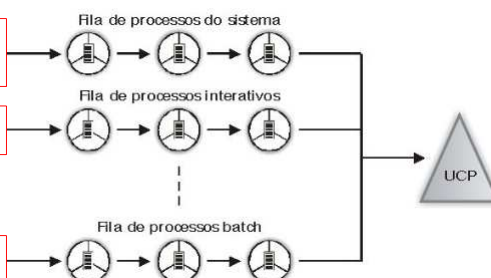
Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- **Múltiplas Filas**
 - Quanto mais próximo de ser finalizado, menos frequente é o processo na CPU.
- *Shortest Process Next*
- Garantido
- *Lottery*
- *Fair-Share*

Existem várias filas de processos em estado de pronto com uma prioridade específica.

Não possui prioridade, fica esta característica associada a fila.

Cada fila possui um mecanismo próprio de escalonamento.



Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- Múltiplas Filas
- ***Shortest Process Next***
 - Não se conhece o tempo, verifica comportamentos passados de execução.
- Garantido
- *Lottery*
- *Fair-Share*

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	■	■	■																	
B				■	■	■	■	■	■	■										
C												■	■	■	■					
D																■	■	■	■	■
E											■	■								

19

Escalonamento de Processos

Prof. Edson Pedro Ferlin

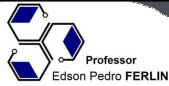
Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- ***Garantido***
 - Aloca determinado tempo específico garantido para cada usuário.
- *Lottery*
- *Fair-Share*

20

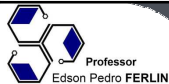
Escalonamento de Processos

Prof. Edson Pedro Ferlin



Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- **Lottery**
 - Cada processo recebe “bilhetes” que lhe dão direito a recursos do sistema (inclusive processador). Processo importantes pode receber mais “bilhetes”.
- *Fair-Share*



Escalonamento em Sistemas Iterativos

- *Round-Robin*
- Prioridade
- Múltiplas Filas
- *Shortest Process Next*
- Garantido
- *Lottery*
- **Fair-Share**
 - O dono do processo é levado em conta conforme a quantidade de processos que cada dono tenha.

Algoritmo de Tempo Real

- Tempo é um fator crítico.
 - *Hard Real Time:*
 - Atrasos não são tolerado;
 - Pilotos automáticos, Hospitais, Usinas, entre outros.
 - *Soft Real Time:*
 - Atrasos são eventualmente tolerado;
 - Bancos, multimídia, entre outros.

Contato



eferlin@live.com



(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) ProfEdsonPedroFerlin