

Capítulo 2

Processos e Threads

2.1 Processos

2.2 Threads

2.3 Comunicação interprocesso

2.4 Problemas clássicos de IPC

2.5 Escalonamento

Processos

O Modelo de Processo

- Todos os computadores modernos podem executar várias tarefas ao mesmo tempo. Enquanto estiver rodando um programa de usuário, o computador pode ler um disco e imprimir em uma impressora.
- Em sistemas multiprogramados, o processador pode ser chaveado entre diversos programas, dando a cada um dezenas ou centenas de milissegundos de processamento. Apesar de só haver um programa executando em determinado instante de tempo, no curso de um segundo, o processador pode ter trabalhado para diversos programas, dando a ilusão do paralelismo de execução (pseudo-paralelismo).
- O tratamento rigoroso de múltiplas atividades realizadas em paralelo pelo sistema operacional não é tarefa fácil. Para um melhor entendimento da função de um Sistema Operacional, é necessário entender alguns conceitos básicos sobre: Processos, Arquivos, Chamadas de Sistema e Interpretador de Comandos

Processos

O Modelo de Processo

- Processo é um programa em execução, sendo constituído do código executável, dos dados referentes ao código, da pilha de execução, do valor do contador de programa, do valor do apontador de pilha, dos demais registradores do hardware, além de um conjunto de outras informações necessárias à execução do programa.
- Um processador é compartilhado por vários processos, estes poderão ser interrompidos pelo sistema operacional, por exemplo, por ter excedido seu tempo de uso de processador compartilhado ou por uma solicitação de um dispositivo de E/S do próprio processo, sendo este processo temporariamente suspenso, tendo a necessidade de restaurá-lo mais tarde, no exato ponto em que foi interrompido, para isso, todas as informações relativas ao processo precisam ser salvas em algum lugar durante o tempo que durar a suspensão de sua execução.
- Todas as informações sobre cada um dos processos são armazenadas em uma tabela do S.O. denominada Tabela de Processos, portanto, um processo suspenso é composto por seu endereçamento e sua entrada na Tabela de Processo.

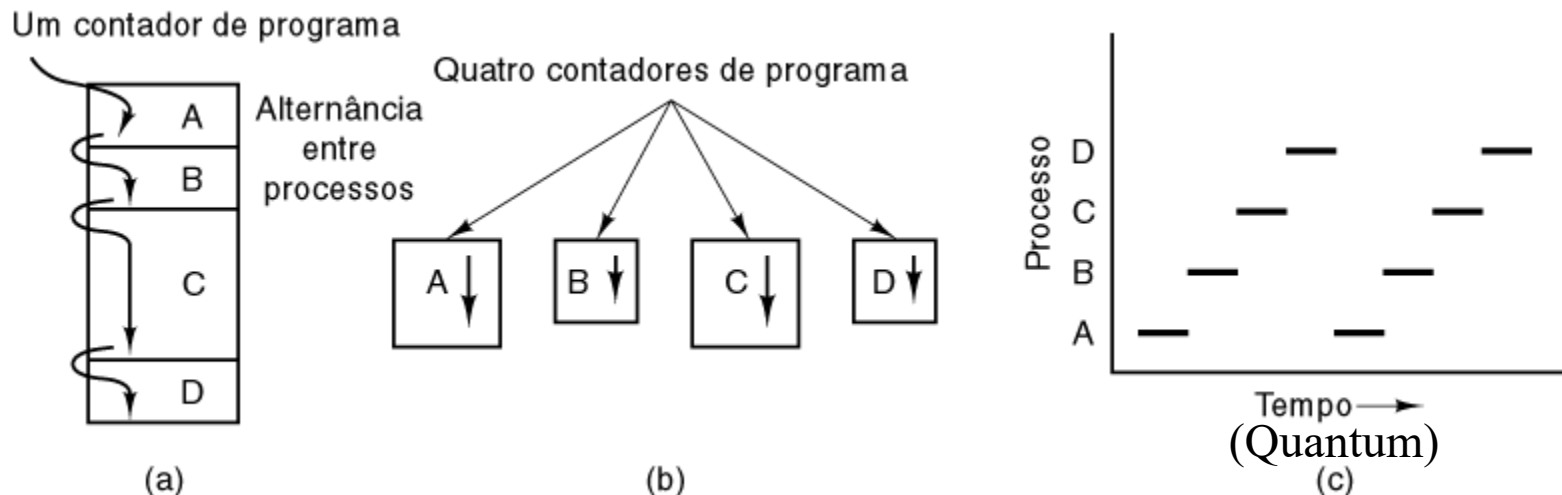
Processos

O Modelo de Processo

- Todo software passível de rodar em um computador, muitas vezes incluindo o próprio sistema operacional, é organizado como um conjunto de processos sequenciais ou simplesmente processos. Um processo nada mais é que um programa em execução, incluindo os valores correntes de todos os registradores do hardware e das variáveis manipuladas por ele no curso de sua execução. O rápido chaveamento do processador entre diversos programas em execução (processos) é denominado de multiprogramação.
- A ideia central é que o processo constitui-se de um certo tipo de atividade, possuindo um programa, uma entrada, uma saída e um estado. Um único processador pode ser compartilhado entre vários processos, com o uso de algum algoritmo de escalonamento (Scheduler) para determinar quando o trabalho em um deles deve ser interrompido e qual o próximo processo a ser servido.

Processos

O Modelo de Processo



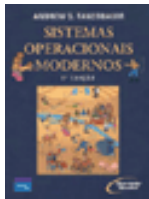
- Multiprogramação de quatro programas
- Modelo conceitual de 4 processos sequenciais, independentes
- Somente um programa está ativo a cada momento

Criação de Processos



- Principais eventos que levam à criação de processos:
 1. Início do sistema
 2. Execução de chamada ao sistema de criação de processos
 3. Solicitação do usuário para criar um novo processo
 4. Início de um job em lote (Batch)

Criação de Processos



```
File Edit Search Options AUTOEXEC.BAT Help
C:\SBCD\DRV\MSCDEX.EXE /D:MSCD001 /V /M:8
C:\WINDOWS\net start
C:\DOS\SMARTDRV.EXE /X
@ECHO OFF
PROMPT $p$g
PATH C:\WINDOWS;C:\DOS
SET TEMP=C:\DOS
MODE CON CODEPAGE PREPARE=((850) C:\DOS\EGA.CPI)
MODE CON CODEPAGE SELECT=850
KEYB BR,,C:\DOS\KEYBOARD.SYS

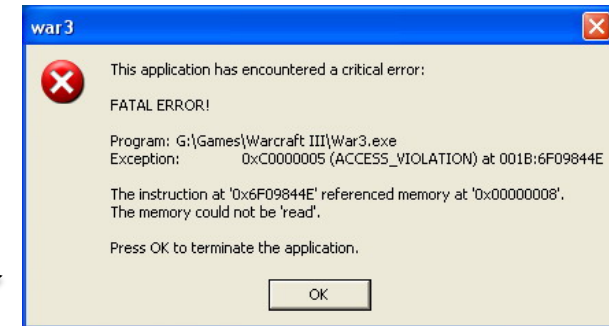
MS-DOS Editor <F1=Help> Press ALT to activate menus CN 00001:001
```

Término de Processos

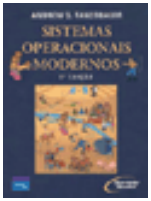


- Condições que levam ao término de processos

1. Saída normal (voluntária)
2. Saída por erro (voluntária)
3. Erro fatal (involuntário) →
4. Cancelamento por um outro processo (involuntário)

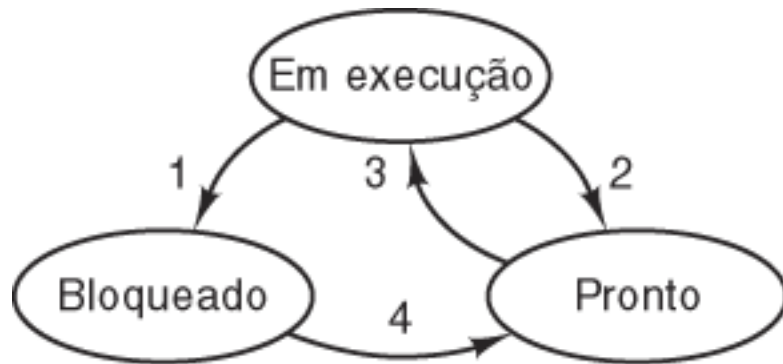


Hierarquias de Processos



- Pai cria um processo filho, processo filho pode criar seu próprio processo
- Formam uma hierarquia
 - UNIX chama isso de “grupo de processos”
- Windows não possui o conceito de hierarquia de processos
 - Todos os processos são criados iguais

Estados de Processos (1)



1. O processo bloqueia aguardando uma entrada
2. O escalonador seleciona outro processo
3. O escalonador seleciona esse processo
4. A entrada torna-se disponível

- Possíveis estados de processos
 - em execução
 - bloqueado
 - pronto
- Mostradas as transições entre os estados

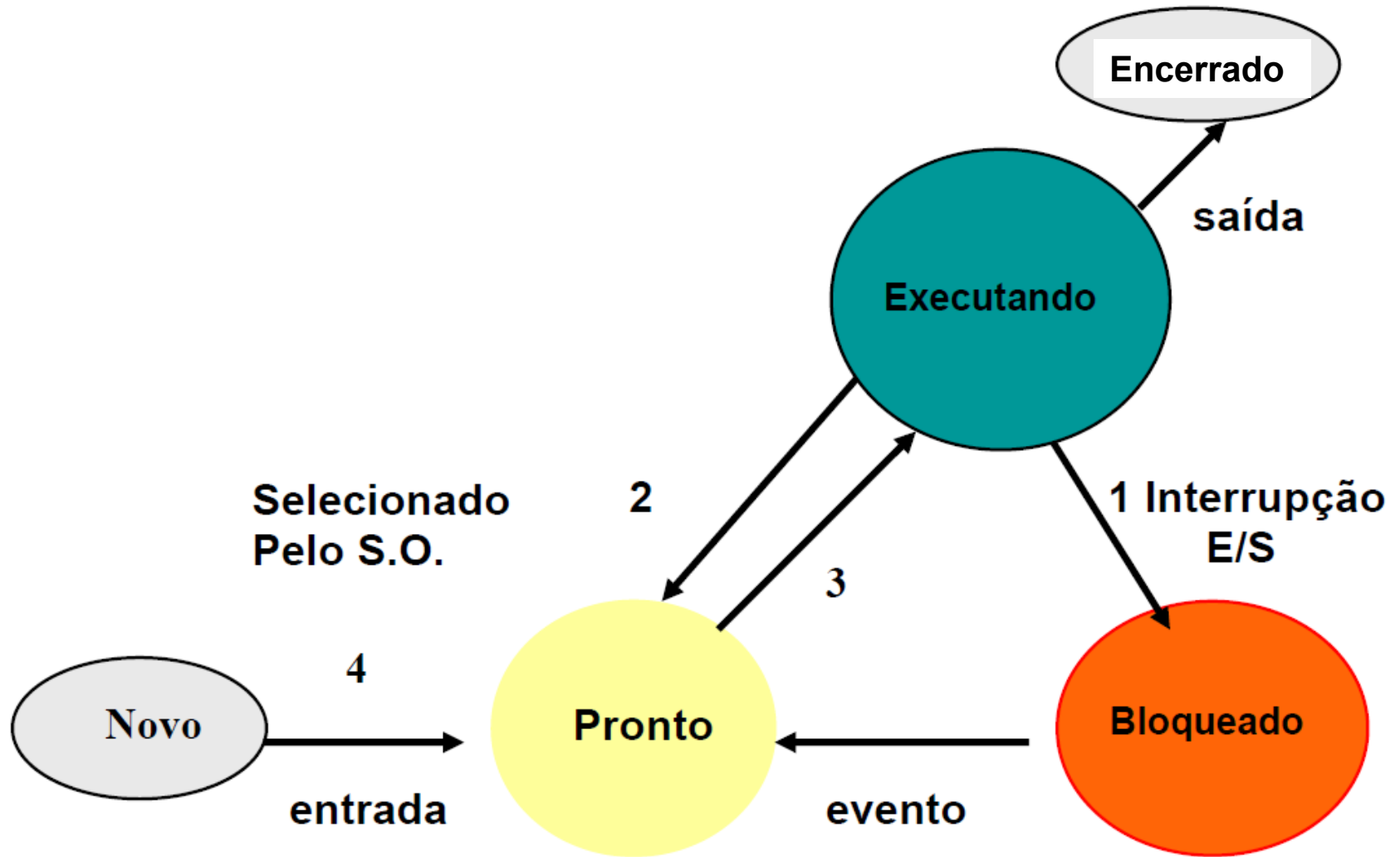
Estados de Processos (1)



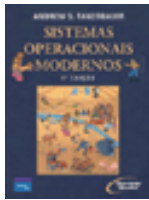
- Os estados de um processo são:
 - Running (Rodando) – Usando o processador neste instante
 - Blocked (Bloqueado) – Impedido de rodar até que ocorra um determinado evento externo ao processo (bloqueado por uma solicitação de E/S)
 - Ready (Pronto) – Em condições de rodar, mas bloqueado temporariamente pelo sistema operacional, para dar vez a outro processo

- As transições de um processo são:
 - (1) Ocorre quando o processo descobre que não tem mais condições lógicas de prosseguir em seu processamento (em alguns sistemas, tal processo executa uma chamada de sistema, BLOCK, indo para estado de bloqueado, quando o processo precisa ler ou gravar).
 - (2) (3) São causadas pelo escalonador de processos, uma rotina do sistema operacional, sem que o processo tome conhecimento delas. (2) Ocorre quando o escalonador decide que o processo corrente já ocupou o processador por tempo suficiente, sendo portanto o momento de deixar que outro processo execute. (3) Ocorre quando todos os demais processos já tiveram oportunidade de rodar, sendo então a vez de o primeiro processo pronto rodar novamente. O escalonador de processos decide qual processo vai rodar e por quanto tempo.
 - (4) Ocorre quando acontece um evento externo pelo qual o processo estava aguardando, por exemplo a conclusão de uma operação de E/S. Se nenhum outro processo estiver rodando neste momento, a transição (3) é acionada imediatamente, e o processo que aguardava a condição vai rodar. Caso contrário, ele vai para o estado de pronto até que o processador fique disponível e o escalonador escolha para rodar.

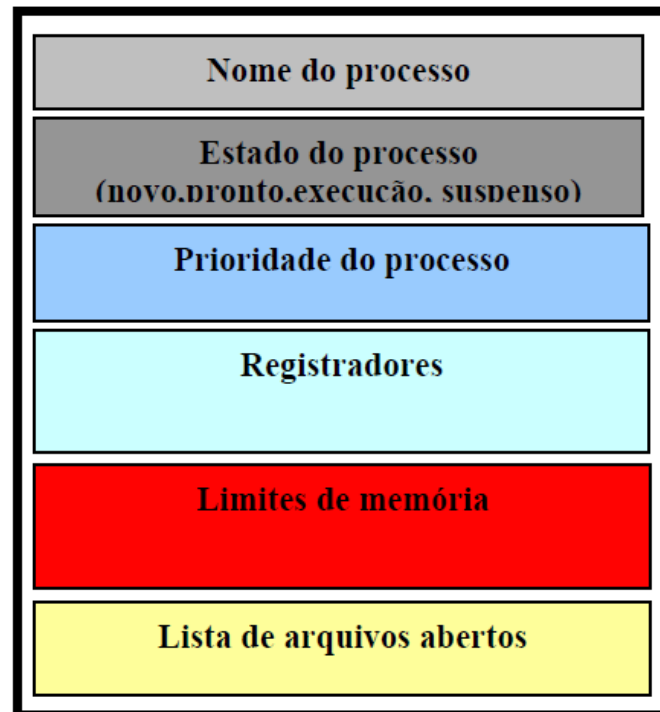
Estados de Processos (1)



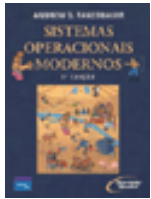
Estados de Processos (1)



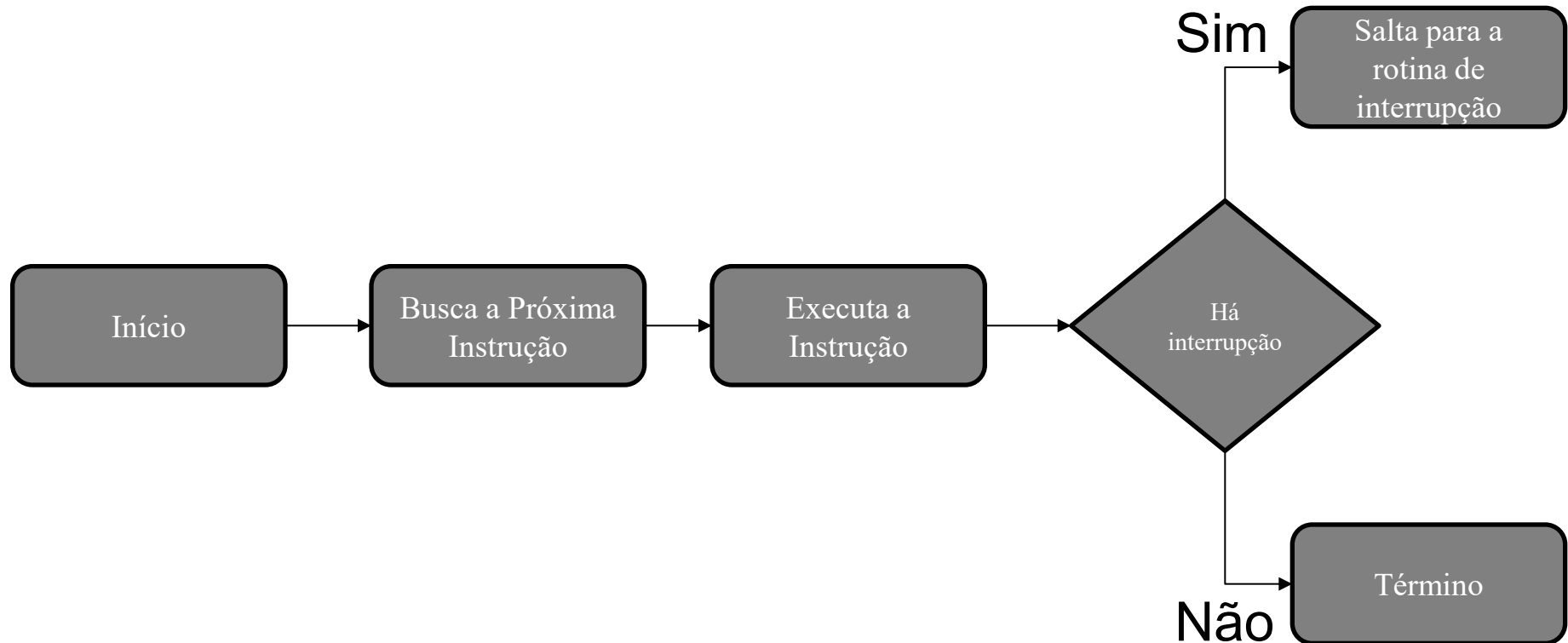
- O Sistema Operacional materializa o processo através de uma estrutura chamada de Bloco de Controle de Processo – BCP (Process Control Block – PCB). A partir do BCP, o sistema mantém todas as informações sobre o processo, como sua identificação, prioridade, estado corrente, recursos alocados e informações sobre o programa em execução.



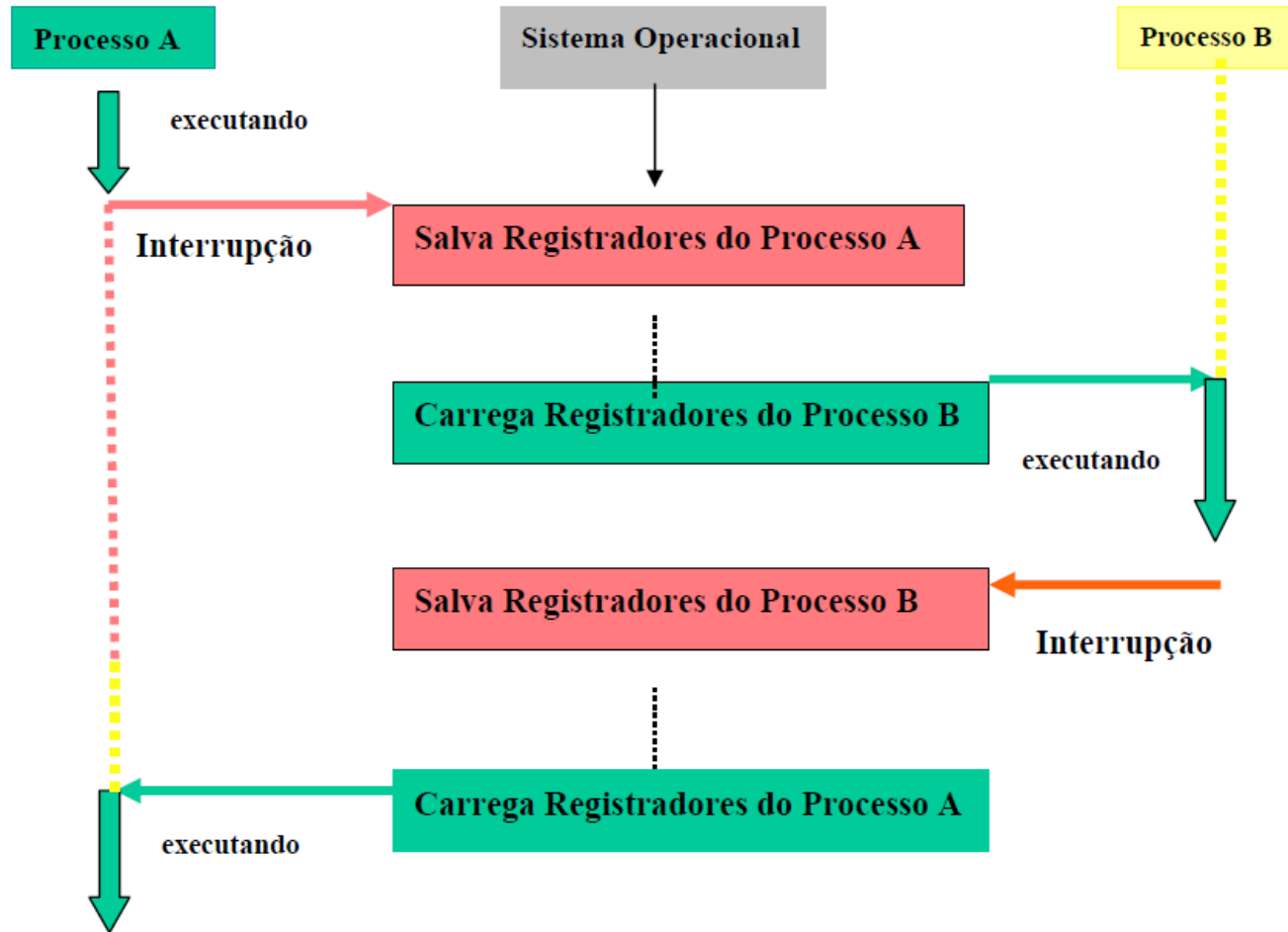
Estados de Processos (1)



- A troca de um contexto por outro na CPU, realizada pelo Sistema Operacional, é denominada mudança de contexto, que consiste em salvar o conteúdo dos registradores da CPU e carregá-los com os valores referentes aos do processo que esteja ganhando a utilização do processador.

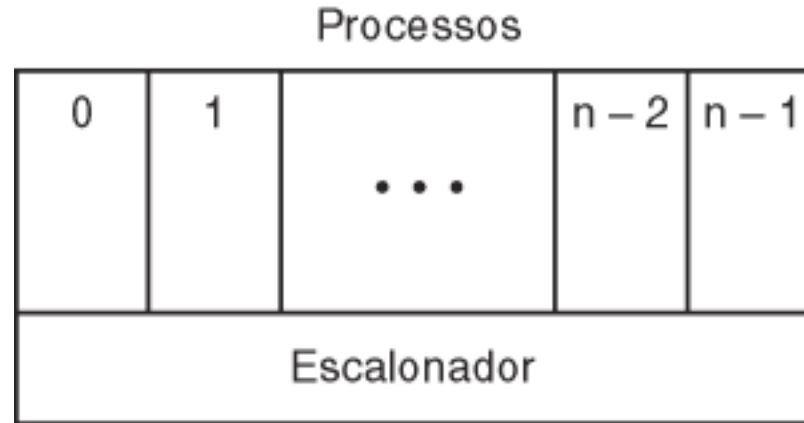


Estados de Processos (1)



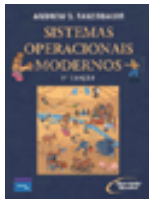
BCP na troca de processos em execução pela CPU

Estados de Processos (2)



- Camada mais inferior de um SO estruturado por processos
 - trata interrupções, escalonamento
- Acima daquela camada estão os processos sequenciais

Escalonamento de Processos



- Os sistemas multiprogramáveis foram implementados pela possibilidade da CPU ser compartilhada entre diversos processos, possuem um critério para determinar qual a ordem na escolha dos processos para execução, entre os vários que concorrem pela utilização do processador. O procedimento de seleção dos processos para que use o processador (CPU) é uma das principais funções realizadas por um S.O., sendo conhecido como escalonamento (troca da CPU entre os vários programas na memória prontos para executar). O responsável pelo escalonamento é chamado de escalonador.
- Os principais objetivos do escalonamento são, basicamente:
 - Manter a CPU ocupada a maior parte do tempo
 - Balancear a utilização do processador entre diversos processos
 - Maximizar o número de processos ativos ao mesmo tempo
 - Oferecer tempos de resposta razoável para os usuários interativos

Algoritmos de Escalonamento

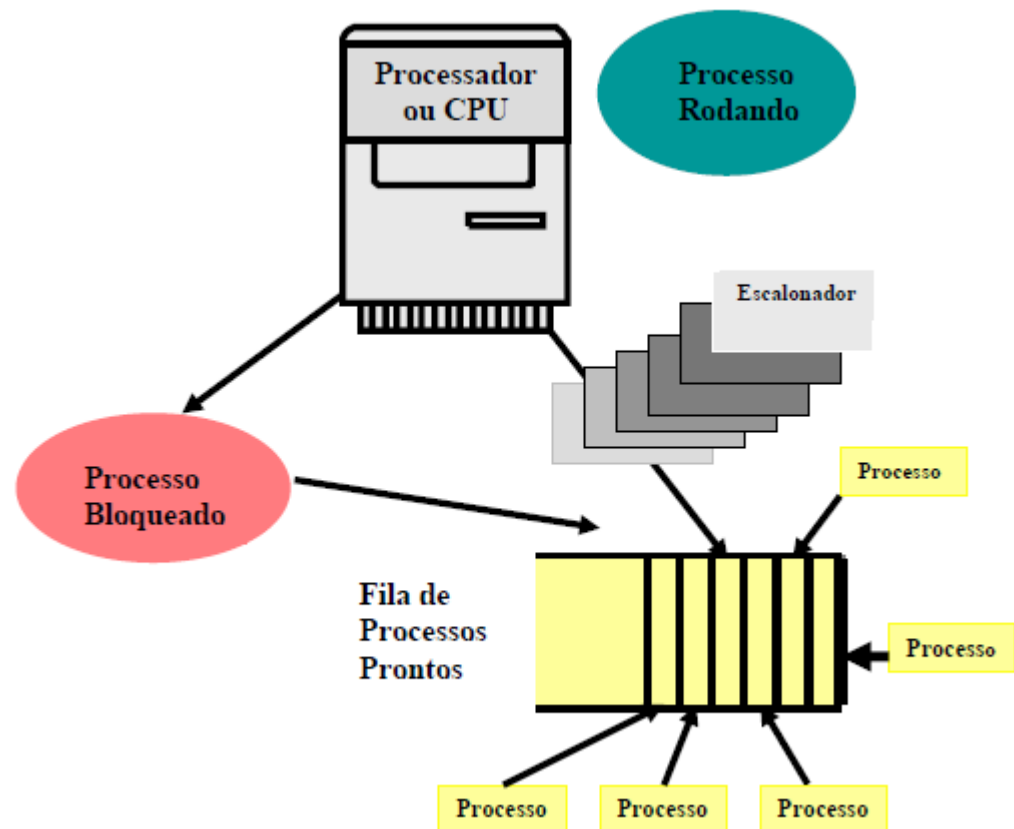


- Um algoritmo de escalonamento tem como função decidir qual dos processos prontos para execução deve ser alocado à CPU. Cada Sistema Operacional necessita de um algoritmo de escalonamento adequado a seu tipo de processamento.
- Os algoritmos podem ser:
 - Não Preemptivos
 - Preemptivos

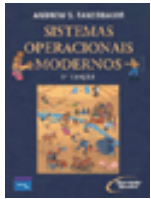
Algoritmos Não-Preemptivos de Escalonamento



- Nos primeiros Sistemas Operacionais multiprogramáveis, onde predominava tipicamente o processamento em lote (batch), o escalonamento implementado era do tipo Não Preemptivo. Nesse tipo de escalonamento, quando um processo ganha o direito de utilizar a CPU, nenhum outro processo pode lhe tirar esse recurso.



Algoritmos Não-Preemptivos de Escalonamento



- **Escalonamento com Algoritmo FCFS (First Come First Served)**
 - Primeiro processo que requisita a CPU é o primeiro que a recebe, isto é, o processo que chegar primeiro (first come) é o primeiro a ser selecionado para execução (first served), esta situação pode ser representada como apenas uma fila. Isto é:
 - Processo pronto para executar entra no final da lista ready (fila de prontos)
 - Quando a CPU é liberada, passa a ser alocada para o primeiro da lista (primeiro da fila)
 - O desempenho do sistema pode ser comprometido de acordo com a ordem de chegada dos processos, o tempo de processamento, bem como o tempo de espera na fila.

Algoritmos Não-Preemptivos de Escalonamento



- Exemplo: Dados 3 processos, com seus tempos de execução

- Caso 1:

Processo	Tempo de Execução
P1	24 s.
P2	3 s.
P3	3 s.

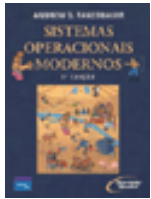
- Tempo Gasto pelos Processos (Acumulado):

P1 = 24 s. P2 = 27 s. P3 = 30 s.



- Tempo Médio (Acumulado): Média = $(24 + 27 + 30) / 3 = 27$ s.

Algoritmos Não-Preemptivos de Escalonamento



- Exemplo: Dados 3 processos, com seus tempos de execução

- Caso 2:

Processo	Tempo de Execução
P1	3 s.
P2	3 s.
P3	24 s.

- Tempo Gasto pelos Processos (Acumulado):
- P1 = 3 s. P2 = 6 s. P3 = 30 s.



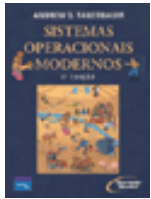
- Tempo Médio (Acumulado): Média = $(3 + 6 + 30) / 3 = 13$ s.

Algoritmos Não-Preemptivos de Escalonamento



- **Escalonamento com Algoritmo SJF (Shortest Job First)**
- O algoritmo é apropriado para sistemas que rodam Jobs em batch, nos quais o tempo de processamento de cada job é conhecido com antecedência. No escalonamento com algoritmo SJF, é selecionado para execução, em primeiro lugar, o processo que precisar de menos tempo de CPU para terminar o seu processamento
- Características:
 - Média pequena do tempo gasto para processar os processos
 - Diminui tempo de espera dos processos pequenos
 - Usado pelo escalonador de longo prazo em sistemas batch
 - Para tempo de execução iguais, utiliza-se FCFS

Algoritmos Não-Preemptivos de Escalonamento



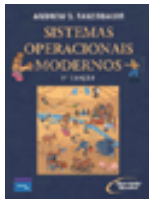
- Exemplo: Dados 4 processos, com seus tempos de execução

- Caso 1:

Processo	Tempo de Execução
P1	6 s.
P2	3 s.
P3	8 s.
P4	7 s.

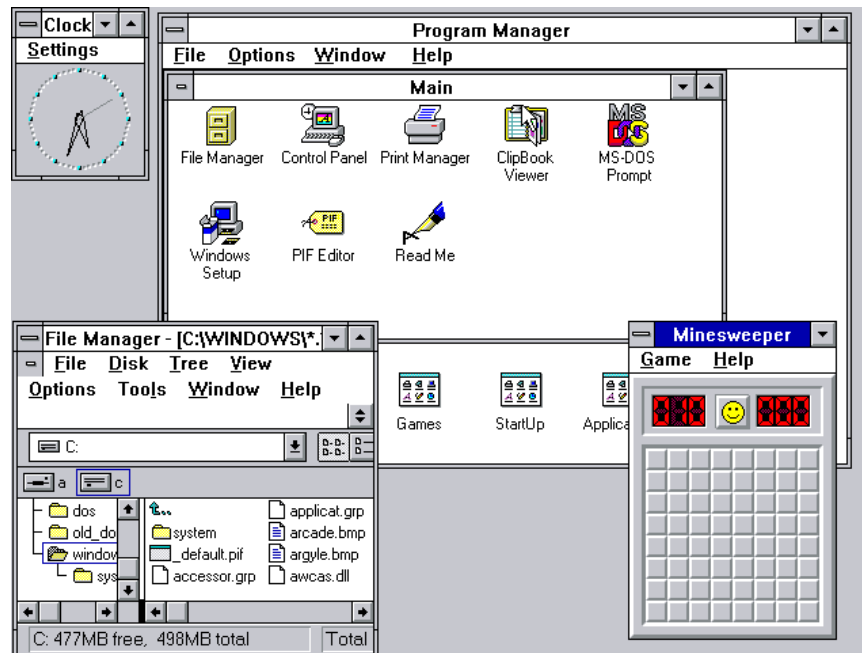
- Tempo Gasto pelos Processos (Acumulado):
P2 = 3 s. P1 = 9 s. P4 = 16 s. P3 = 24 s.
- Tempo Médio: Média = $(3 + 9 + 16 + 24) / 4 = 13$ s.

Algoritmos Não-Preemptivos de Escalonamento

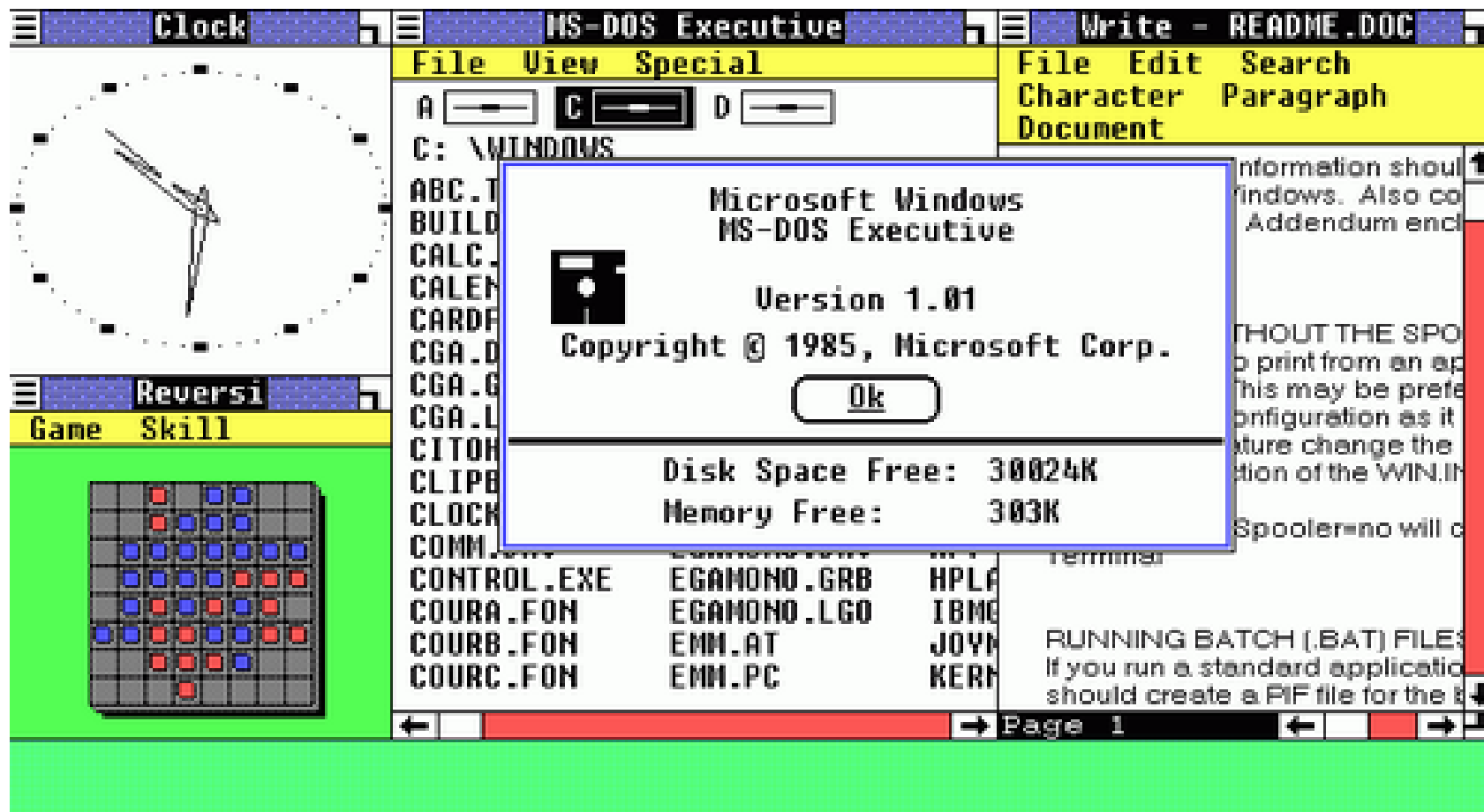


■ Escalonamento Cooperativo

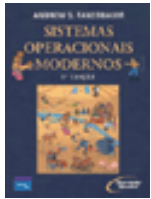
- Este tipo de escalonamento permite a implantação de sistemas multiprogramáveis com uma melhor distribuição do uso de processador entre processos. Sua principal característica está no fato de a liberação do processador ser uma tarefa realizada exclusivamente pelo processo em execução que libera a CPU para um outro processo.
- Um exemplo deste tipo de escalonamento pode ser encontrado em sistemas Microsoft® Windows 3.1 e Windows 3.11, sendo conhecidos como multitarefas cooperativas.



Algoritmos Não-Preemptivos de Escalonamento

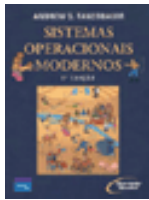


Algoritmos Preemptivos de Escalonamento



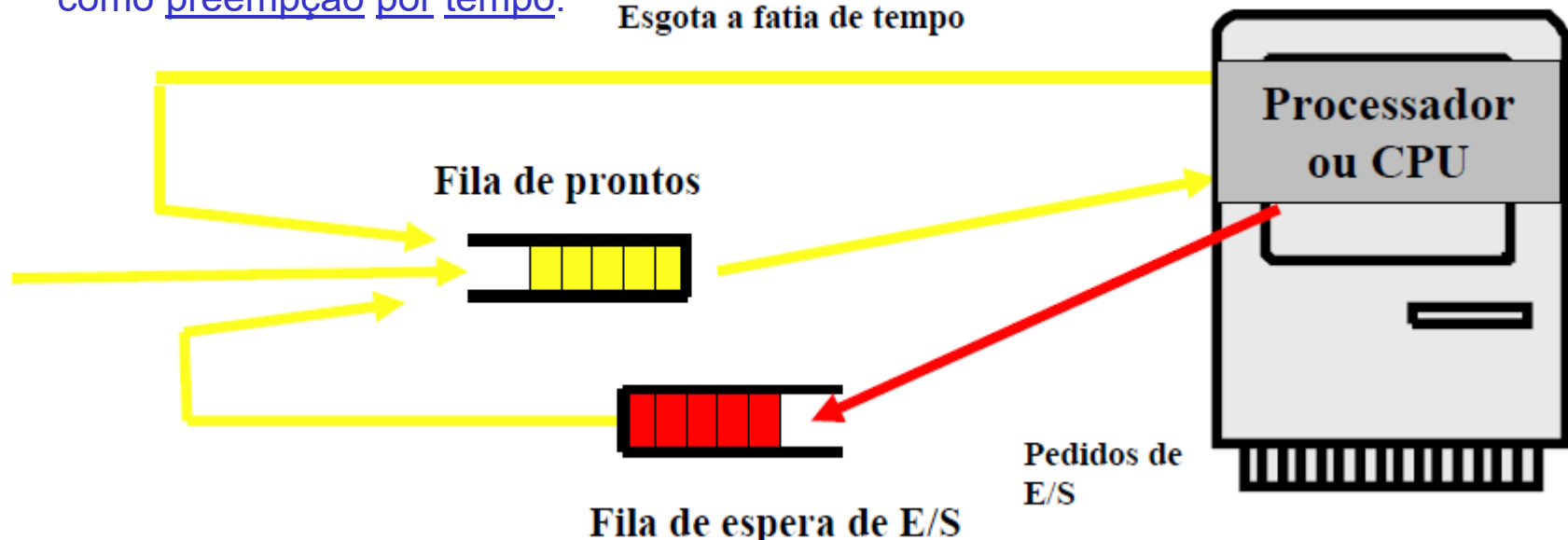
- Um algoritmo de escalonamento é dito preemptivo quando o Sistema Operacional pode interromper um processo em execução para que outro processo utilize o processador. Em Sistemas Operacionais que não implementam preempção, um processo pode utilizar o processador enquanto for necessário.
- O escalonamento preemptivo permite que o Sistema Operacional dê atenção imediata a processos mais prioritários, como no caso de sistemas de tempo real, além de proporcionar melhores tempos de resposta em sistemas de tempo compartilhado. Outro benefício decorrente deste tipo de escalonamento é o compartilhamento do processador de maneira uniforme entre os processos.

Algoritmos Preemptivos de Escalonamento



- **Escalonamento Circular ou Round Robin (RR)**
- O escalonamento circular é implementado através de um algoritmo projetado especialmente para sistemas de tempo compartilhado (time sharing). Nesse algoritmo, quando um processo passa para o estado de execução, ou seja, ganha a CPU, existe um tempo limite para sua utilização de forma contínua. Quando esse tempo, denominado Quantum ou Time-Slice expira, sem que antes a CPU seja liberada pelo processo, este volta ao estado de pronto dando a vez para outro processo. Esse mecanismo é definido como preempção por tempo.

Esgota a fatia de tempo



Algoritmos Preemptivos de Escalonamento



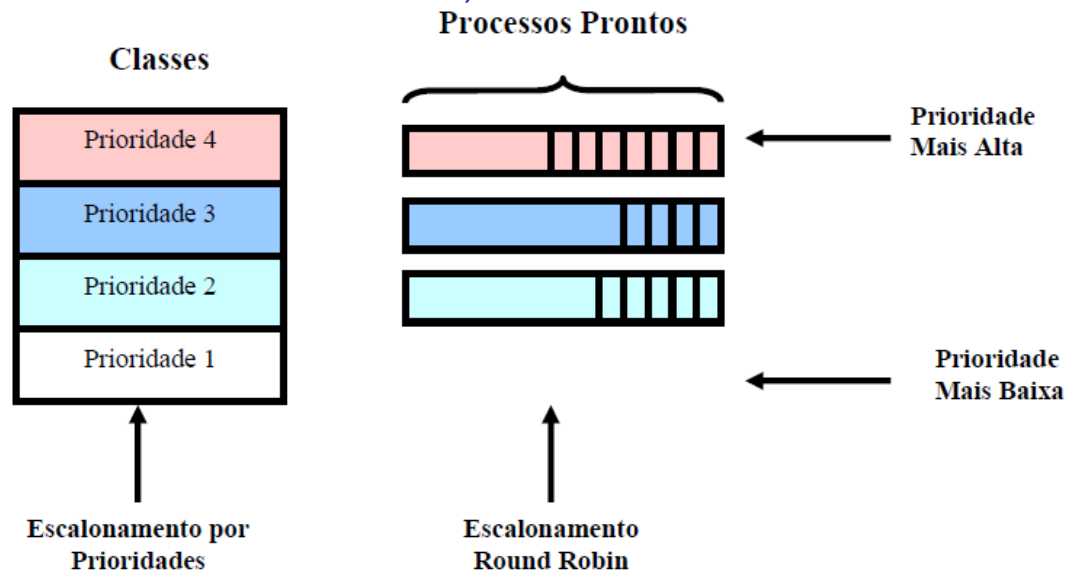
- **Escalonamento por Prioridade**
- O escalonamento circular consegue a distribuição melhor do tempo de CPU em relação aos escalonamentos não preemptivos, porém não consegue implementar um compartilhamento equitativo entre os diferentes tipos de processos. Isso acontece em razão de o escalonamento circular tratar todos os processos de maneira igual.
- Como alguns processos devem ser tratados de maneira diferente dos outros, é preciso associar a cada um deles uma prioridade de execução. Nesse esquema, processos de maior prioridade são escalonados preferencialmente. Toda vez que um processo for para a fila de pronto com prioridade superior ao do processo em execução, o sistema deve interromper o processo corrente, coloca-lo no estado de pronto e selecionar o de maior prioridade para ser executado. Esse mecanismo é definido como preempção por prioridade.
- As prioridades podem ser atribuídas a processos de forma estática ou dinâmica. Em forma estática, por exemplo, em uma universidade, Jobs de professores terão maior prioridade (100), os Jobs de secretárias (50), os monitores (25) e Jobs de alunos terão menor prioridade (10).

Algoritmos Preemptivos de Escalonamento



▪ Escalonamento por Prioridade

- No caso da prioridade dinâmica, Jobs que solicitam um número maior de vezes de dispositivos de E/S terão maior prioridade e Jobs que solicitam um número menor de dispositivos de E/S terão menor prioridade. Algumas vezes, pode ser conveniente agrupar processos em classes de prioridade e executar um escalonamento Round Robin dentro de cada classe. Enquanto houver processos prontos na classe de maior prioridade, tais processos serão disponibilizados segundo o algoritmo RR, não podendo ser escolhido nenhum processo pertencente às classes de menor prioridade. A não atenção com as classes de prioridades pode levar ao Starvation (Quando um processo fica aguardando indefinidamente).



Algoritmos Preemptivos de Escalonamento



Gerenciador de Tarefas

Digite um nome, editor ou PID para pesquisar

Executar nova tarefa Finalizar tarefa

Detalhes

Nome	PID	Status	Nome de u...	CPU	Memória (...)	Arquite...	Descrição
acrotay.exe	35188	Em execução	lecol	00	2.096 K	x86	AcroTray
AggregatorHost.exe	5368	Em execução	SISTEMA	00	2.700 K	x64	Microsoft (R) Aggregator Host
ai.exe	27880	Em execução	lecol	00	23.068 K	x64	Artificial Intelligence (AI) Host for the Microsoft® Windows® Operating System and Platform
AnyDeskMSI.exe	5136	Em execução	SISTEMA	00	18.008 K	x86	AnyDesk
AnyDeskMSI.exe	10984	Em execução	lecol	00	17.320 K	x86	AnyDesk
ApplicationFrameHo...	32712	Em execução	lecol	00	19.524 K	x64	Application Frame Host
ArcControlService.exe	5456	Em execução	SISTEMA	00	3.784 K	x64	Intel(R) Arc(TM) Control Service
avp.exe	5148	Em execução	SISTEMA	00	145.276 K	x86	Kaspersky Lab launcher
avpui.exe	12948	Em execução	lecol	00	3.428 K	x86	Kaspersky
backgroundTaskHos...	24412	Suspense	lecol	00	0 K	x64	Background Task Host
backgroundTaskHos...	25952	Suspense	lecol	00	0 K	x64	Background Task Host
browser_assistant.exe	34984	Em execução	lecol	00	4.204 K	x86	Opera GX Browser Assistant
browser_assistant.exe	27320	Em execução	lecol	00	1.956 K	x86	
BulletService.exe	5676	Em execução	SISTEMA	00	1.296 K	x64	
ColorEngine.exe	37276	Em execução	lecol	00	1.112 K	x64	
conhost.exe	7048	Em execução	SISTEMA	00	4.904 K	x64	
conhost.exe	34888	Em execução	lecol	00	5.608 K	x64	
conhost.exe	11400	Em execução	SISTEMA	00	5.056 K	x64	
conhost.exe	36464	Em execução	SISTEMA	00	5.224 K	x64	
crashpad_handler.exe	7820	Em execução	lecol	00	1.048 K	x64	
csrss.exe	884	Em execução	SISTEMA	00	1.416 K	x64	
csrss.exe	32380	Em execução	SISTEMA	00	2.028 K	x64	
ctfmon.exe	18768	Em execução	lecol	00	69.716 K	x64	
dasHost.exe	3392	Em execução	SERVIÇO L...	00	7.448 K	x64	
dilhost.exe	21088	Em execução	SISTEMA	00	1.596 K	x86	
dilhost.exe	27356	Em execução	lecol	00	7.180 K	x64	
dpft_helper.exe	10916	Em execução	lecol	00	1.128 K	x64	
DSAService.exe	5320	Em execução	SISTEMA	00	49.868 K	x86	
DSALUpdateService.exe	8100	Em execução	SISTEMA	00	18.408 K	x64	
dwm.exe	36216	Em execução	DWM-7	00	224.288 K	x64	Gerenciador de Janelas da Área de Trabalho
esif_of.exe	5392	Em execução	SISTEMA	00	1.572 K	x64	Intel(R) Dynamic Tuning Service
esrv.exe	26584	Em execução	lecol	00	128.452 K	x64	Intel(R) System Usage Report
esrv_svc.exe	7116	Em execução	SISTEMA	00	192.084 K	x64	Intel(R) System Usage Report
explorer.exe	13544	Em execução	lecol	00	259.108 K	x64	Windows Explorer
fontdrvhost.exe	1364	Em execução	UMFD-0	00	1.600 K	x64	Usermode Font Driver Host
fontdrvhost.exe	36856	Em execução	UMFD-7	00	2.868 K	x64	Usermode Font Driver Host

Finalizar tarefa
Finalizar árvore de processos
Enviar comentários

Modo de eficiência
Definir prioridade
Definir afinidade

Analisar cadeia de espera
Virtualização do UAC
Criar arquivo de despejo de memória
Abrir local do arquivo
Pesquisar online
Propriedades
Ir para serviço(s)

Tempo real
Alta
Acima do normal
Normal
Abaixo do normal
Baixa

System Monitor

Monitor Edit View Help

System Processes Resources File Systems

Load averages for the last 1, 5, 15 minutes: 0.38, 0.57, 0.49

Process Name	Status	% CPU	Nice ^	ID	Memory	W
tracker-indexer	Sleeping	0	19	4491	6.1 MiB	dc
firefox	Sleeping	4	0	5191	92.7 MiB	dc
bonobo-activation-server	Sleeping	0	0	4399	800.0 KiB	dc
compiz	Sleeping	0	0	4328	96.0 KiB	dc
compiz-decorator	Sleeping	0	0	4487	80.0 KiB	dc
compiz.real	Sleeping	0	0	4391	11.8 MiB	dc
dbus-daemon	Sleeping	0	0	4165	632.0 KiB	dc
dbus-launch	Sleeping	0	0	4164	240.0 KiB	dc
evolution-alarm-notify	Sleeping	0	0	4404	2.2 MiB	dc
gconfd-2	Sleeping	0	0	4173	2.3 MiB	dc
gtk-window-decorator	Sleeping	0	0	4489	3.2 MiB	dc
x-session-manage	Sleeping	0	0	3952	1.5 MiB	dc

End Process