

## Escalonamento

- Conceitos Básicos
- Critérios de Escalonamento
- Algoritmos de Escalonamento
- Escalonamento Multiprocessador
- Escalonamento Tempo-Real
- Avaliação de Algoritmos

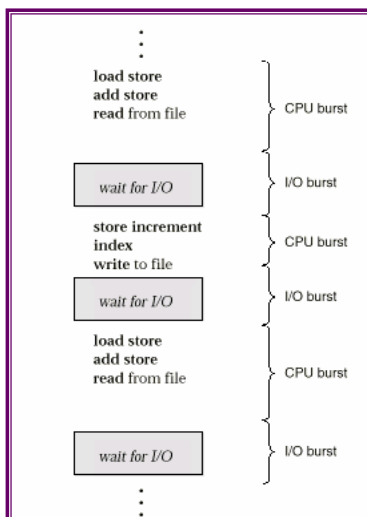
25/4/2007

INE5355 – Sistemas Operacionais I

6.1

## Conceitos Básicos

- Execução de processo consiste de um ciclo de UCP e espera de E/S

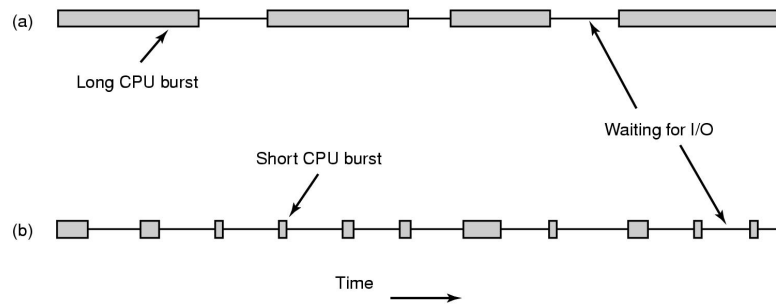


25/4/2007

INE5355 – Sistemas Operacionais I

6.2

## Conceitos Básicos



25/4/2007

INE5355 – Sistemas Operacionais I

6.3

## Tipo de Escalonamento

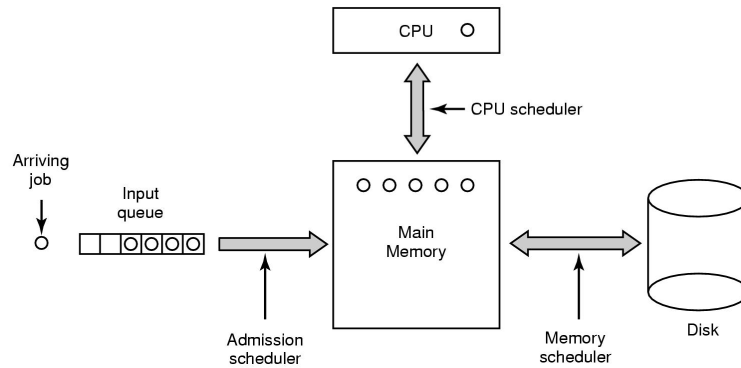
- Long-term – determina que programas são admitidos para execução; controla o grau de multiprogramação; mais processos .....
- Medium-term – determina processos que são trazidos de/para memória;
- Short-term – determina que processo recebe UCP; executa mais frequente; invocado quando eventos ocorrem (int., chamadas de sistema, sinais)

25/4/2007

INE5355 – Sistemas Operacionais I

6.4

## Tipo de Escalonamento



25/4/2007

INE5355 – Sistemas Operacionais I

6.5

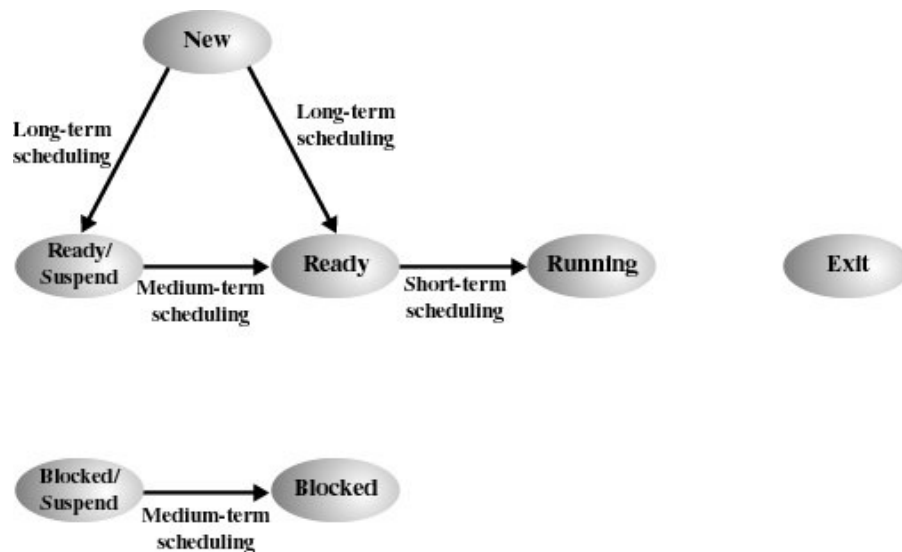
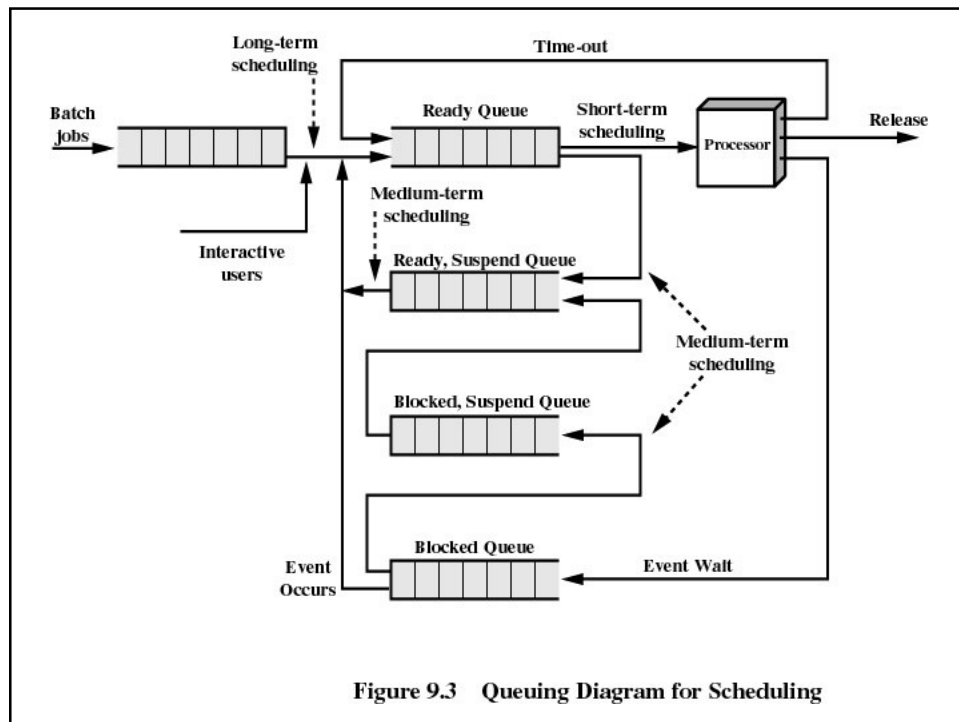


Figure 9.1 Scheduling and Process State Transitions



## Tipo de Escalonamento

- Não preemptivo
  - Estando um processo no estado running, ele continuará até terminar ou bloquear devido a E/S
- Preemptivo
  - Processo running pode ser interrompido e levado para o estado Ready pelo SO
  - Permite melhor serviço, nenhum processo pode monopolizar o processador por muito tempo

## Escalonador da UCP

- Seleciona entre os processos na memória que estão prontos para executar e aloca a UCP para um deles.
- As decisões de escalonamento da UCP podem acontecer quando um processo:
  1. Muda do estado *running* para o estado *waiting*.
  2. Muda do estado *running* para o estado *ready*.
  3. Muda do estado *waiting* para o estado *ready*.
  4. Termina.
- Escalonamento em 1 e 4 é *não preemptivo*.
- Nos outros casos o escalonamento é *preemptivo*.

25/4/2007

INE5355 – Sistemas Operacionais I

6.9

## Despachador

- O módulo Despachador passa o controle da UCP para o processo selecionado pelo escalonador (short-term); isto envolve:
  - Troca de contexto
  - Mudança para modo usuário
  - Salto para a posição apropriada no programa do usuário para reiniciar o programa
- *Latencia de Despacho*— tempo gasto pelo despachador para parar um processo e iniciar outro.

25/4/2007

INE5355 – Sistemas Operacionais I

6.10

## Critérios de Escalonamento

- Utilização da UCP – manter a UCP ocupada
- Desempenho – # de processos que completam sua execução por unidade de tempo
- Tempo de turnaround – quantidade de tempo para executar um processo particular
- Tempo de espera – quantidade de tempo que um processo espera na fila de prontos
- Tempo de resposta – quantidade de tempo gasto desde a submissão da requisição até produzir a primeira resposta

25/4/2007

INE5355 – Sistemas Operacionais I

6.11

## Critérios de Otimização

- Max utilização da UCP
- Max desempenho
- Min turnaround
- Min tempo de espera
- Min tempo de resposta

25/4/2007

INE5355 – Sistemas Operacionais I

6.12

## Metas dos Algoritmos

- Todos os sistemas
  - Justiça – dar a cada processo parcela da UCP
  - Aplicação de política – assegurar que política é adotada
  - Balanceamento – manter todas as partes do sistema ocupadas
- Sistemas Batch
  - Max desempenho
  - Min. Turnaround
  - Max utilização da UCP
- Sistemas Interativos
  - Min tempo de resposta
  - Proporcionalidade – atender expectativas dos usuários
- Sistemas de Tempo-Real
  - Atender prazos (deadlines) – evitar perda de dados
  - Previsibilidade – evitar degradação da qualidade em sistemas multimídia

25/4/2007

INE5355 – Sistemas Operacionais I

6.13

## First-Come, First-Served (FCFS)

<u>Processo</u>	<u>Burst Time</u>
$P_1$	24
$P_2$	3
$P_3$	3

- Supor que os processos chegam na ordem:  $P_1, P_2, P_3$   
diagrama de Gantt para o escalonamento:



- Tempo de espera para  $P_1 = 0$ ;  $P_2 = 24$ ;  $P_3 = 27$
- Tempo de espera médio :  $(0 + 24 + 27)/3 = 17$

25/4/2007

INE5355 – Sistemas Operacionais I

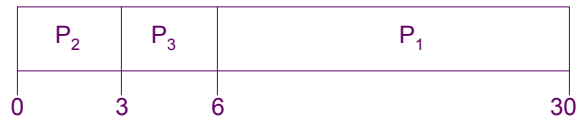
6.14

## FCFS (Cont.)

Supor que os processos chegam na ordem

$P_2, P_3, P_1$ .

- Diagrama de Gantt :



- Tempo de espera  $P_1 = 6$ ;  $P_2 = 0$ ;  $P_3 = 3$
- Tempo de espera médio:  $(6 + 0 + 3)/3 = 3$
- Bem melhor que o caso anterior.
- *Efeito comboio* processos pequenos seguem processos longos

## Shortest-Job-First (SJF)

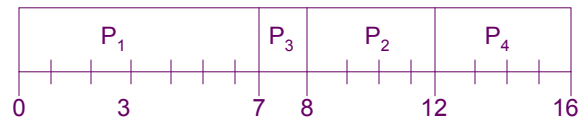
- Associar com cada processo o tamanho de seu próximo ciclo de UCP. Usar estes valores para escalonar o processo com o menor tempo.
- Dois esquemas:
  - não preemptivo
  - preemptivo – se um novo processo chegar com ciclo de UCP menor que o tempo remanescente do processo em execução, preempta. Este esquema é conhecido como Shortest-Remaining-Time-First (SRTF).
- SJF é ótimo – resulta mínimo tempo médio de espera para um dado conjunto de processos.



## Exemplo: SJF Não-Preemptivo

Processo	Chegada	Ciclo de UCP
$P_1$	0.0	7
$P_2$	2.0	4
$P_3$	4.0	1
$P_4$	5.0	4

- SJF (não-preemptivo)



- Tempo médio de espera =  $(0 + 6 + 3 + 7)/4 - 4$

25/4/2007

INE5355 – Sistemas Operacionais I

6.17

## Exemplo: SJF Preemptivo

Process	Arrival Time	Burst Time
$P_1$	0.0	7
$P_2$	2.0	4
$P_3$	4.0	1
$P_4$	5.0	4

- SJF (preemptive)



- Average waiting time =  $(9 + 1 + 0 + 2)/4 - 3$

25/4/2007

INE5355 – Sistemas Operacionais I

6.18

## Determinar tamanho do ciclo da UCP

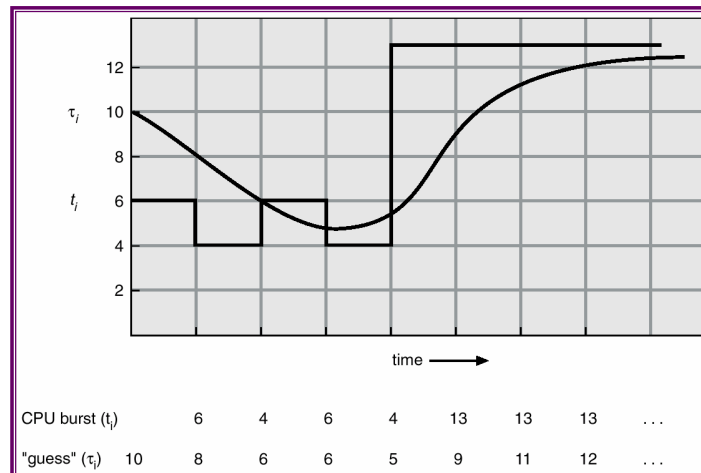
- Pode apenas estimar o tamanho.
- Esta estimativa leva em consideração que o próximo ciclo vai ser parecido com o anterior, assim é possível processar uma aproximação do próximo valor.
- Suponha que o tempo estimado para a execução de comandos em um terminal seja  $T_0$ , e que na próxima rodada o tempo medido seja  $T_1$ . Podemos atualizar nossa estimativa considerando a soma ponderada destes dois valores, a  $T_0 + (1 - a) T_1$ .

25/4/2007

INE5355 – Sistemas Operacionais I

6.19

## Exemplo de estimativa



25/4/2007

INE5355 – Sistemas Operacionais I

6.20

## Estimativa

- Através da escolha de  $a$ , podemos fazer com que o processo cujo tempo estamos estimando esqueça rapidamente as últimas rodadas ou lembre-se delas por um tempo mais longo.
- $a = 0$ 
  - $\tau_{n+1} = \tau_n$
  - História recente não conta.
- $a = 1$ 
  - $\tau_{n+1} = t_n$
  - Apenas o último ciclo conta.
- Com  $a = 1/2$ , teremos as seguintes estimativas sucessivas :

$$T_0, T_0/2 + T_1/2, T_0/4 + T_1/2 + T_2/2, T_0/8 + T_1/8 + T_2/4 + T_3/2$$

- Observe que após 3 novas rodadas o peso de  $T_0$  no tempo estimado caiu para  $1/8$ .
- A técnica da estimativa do próximo valor através de uma série ponderada é chamada de **aging**, sendo aplicável nas situações em que a previsão deve ser baseada em valores anteriores.

25/4/2007

INE5355 – Sistemas Operacionais I

6.21

## Prioridade

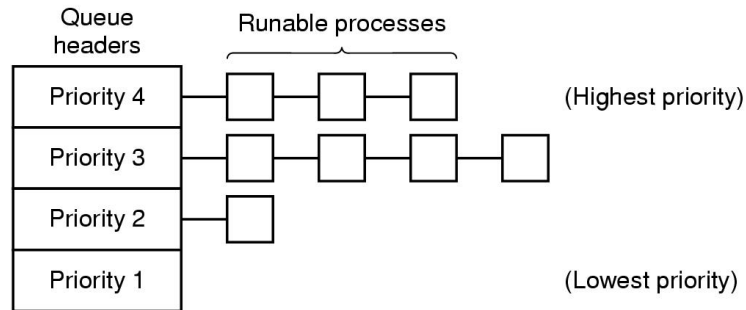
- Uma prioridade (numero inteiro) é associado a cada processo
- A UCP é alocada para o processo com a prioridade mais alta (menor inteiro  $\equiv$  maior prioridade).
  - Preemptivo
  - Não preemptivo
- SJF é um escalonamento com prioridade onde .....
- Problema  $\equiv$  postergação ....
- Solução  $\equiv$  ????

25/4/2007

INE5355 – Sistemas Operacionais I

6.22

## Prioridade



25/4/2007

INE5355 – Sistemas Operacionais I

6.23

## Prioridade

		Win32 process class priorities					
Win32 thread priorities		Realtime	High	Above Normal	Normal	Below Normal	Idle
	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

Mapeamento de prioridades Win32 para prioridades Windows 2000

25/4/2007

INE5355 – Sistemas Operacionais I

6.24

## Round Robin (RR)

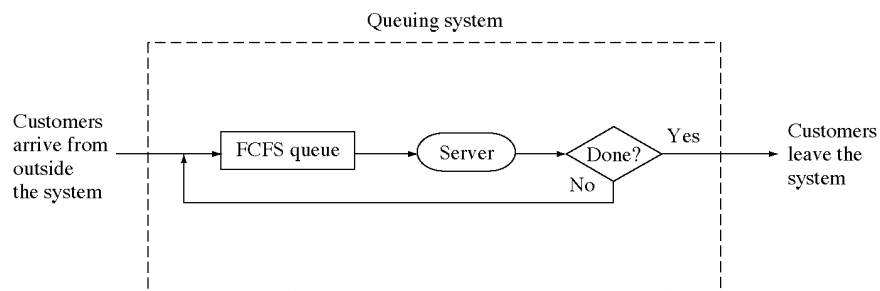
- Cada processo recebe uma pequena unidade de tempo da UCP (*time quantum*), usualmente 10-100 millisegs. Depois de passado este tempo, o processo é preemptado e colocado no final da fila de prontos.
- Se existem  $n$  processos na fila de prontos e o quantum é  $q$ , então cada processo recebe  $1/n$  do tempo da UCP em pedaços de no máximo  $q$  unidades de tempo a cada vez. Nenhum processo espera mais que  $(n-1)q$  unidades de tempo.
- Desempenho
  - $q$  grande  $\Rightarrow$  ????
  - $q$  pequeno  $\Rightarrow$  ????

25/4/2007

INE5355 – Sistemas Operacionais I

6.25

## Round-robin scheduling

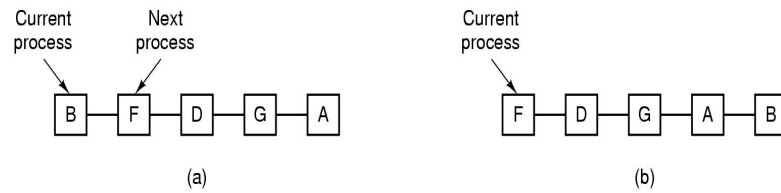


25/4/2007

INE5355 – Sistemas Operacionais I

6.26

## Round Robin (RR)



25/4/2007

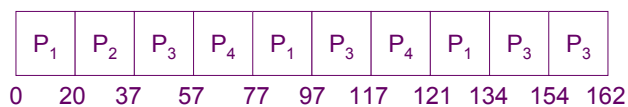
INE5355 – Sistemas Operacionais I

6.27

## Exemplo: RR, Quantum = 20

Processo	ciclo de UCP
$P_1$	53
$P_2$	17
$P_3$	68
$P_4$	24

- Diagrama de Gantt :



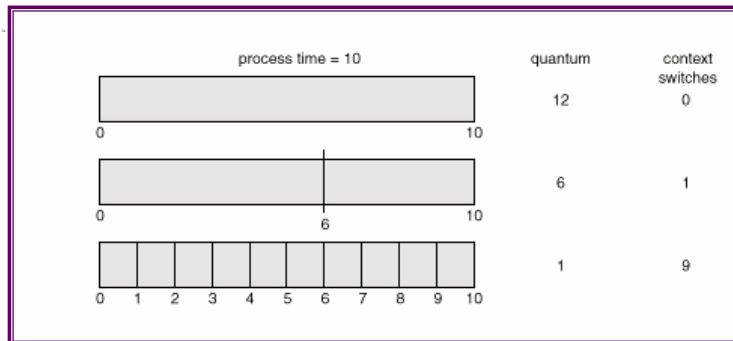
- Tipicamente, turnaround médio maior que SJF, melhor resposta.

25/4/2007

INE5355 – Sistemas Operacionais I

6.28

## Quantum e Chaveamento de contexto

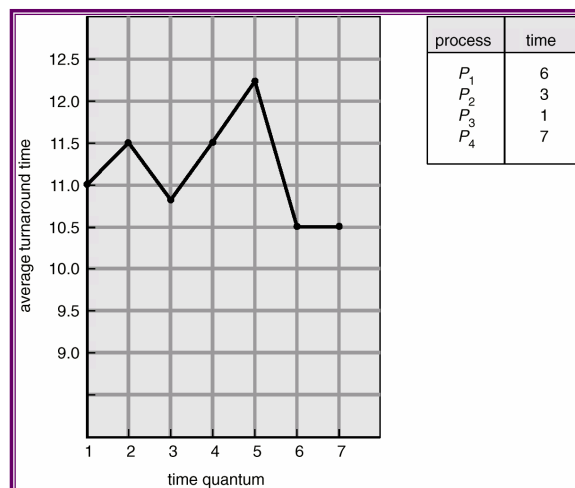


25/4/2007

INE5355 – Sistemas Operacionais I

6.29

## Turnaround varia com Quantum



25/4/2007

INE5355 – Sistemas Operacionais I

6.30

## Filas Multinível

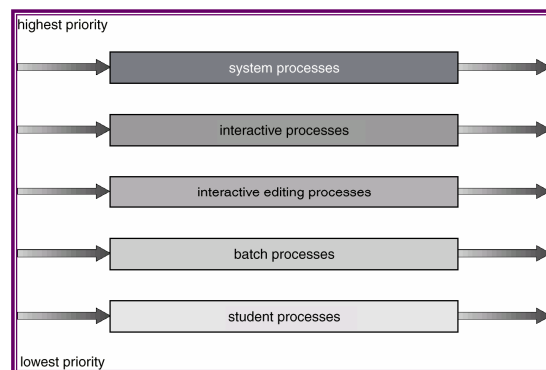
- Fila de prontos é particionada em filas separadas:  
foreground (interativa)  
background (batch)
- Cada fila tem seu próprio algoritmo,  
foreground – RR  
background – FCFS
- Escalonamento deve ser feito entre as filas.
  - Prioridade fixa; (i.e., serve todos da foreground depois da background). Possibilidade de .....
  - Fatia de tempo – cada fila recebe uma quantidade do tempo de UCP o qual escala entre seus processos; i.e., 80% para foreground com RR, 20% para background com FCFS

25/4/2007

INE5355 – Sistemas Operacionais I

6.31

## Filas Multinível



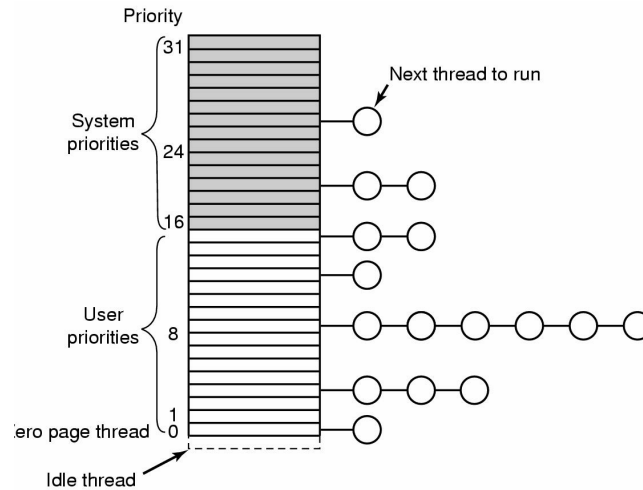
25/4/2007

INE5355 – Sistemas Operacionais I

6.32



## Escalonamento Filas Múltiplas



Windows 2000 suporta 32 prioridades para as threads

25/4/2007

INE5355 – Sistemas Operacionais I

6.33

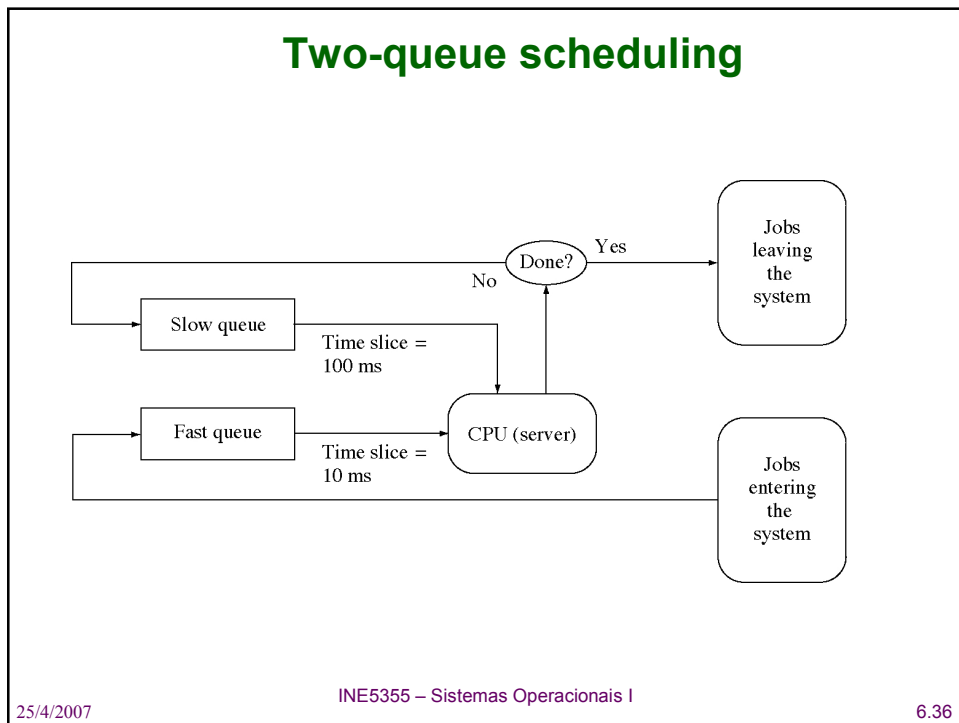
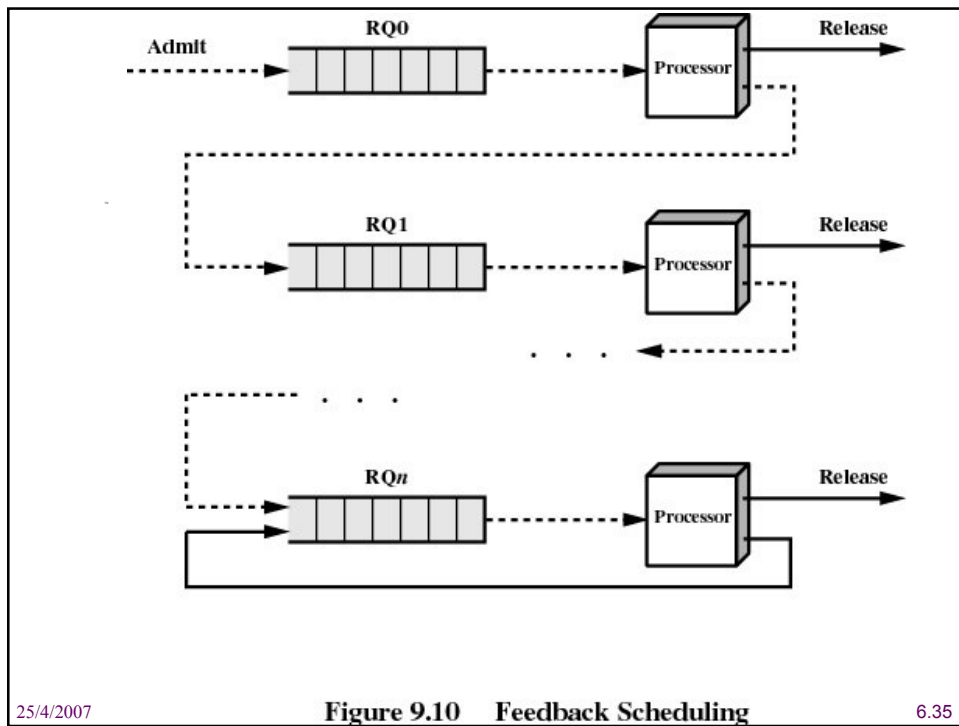
## Filas Multinível c/ Feedback

- Um processo pode mover-se entre as várias filas; envelhecimento pode ser implementado desta forma.
- Escalonador fila-multinível-feedback definido pelos seguintes parâmetros:
  - número de filas
  - algoritmos de escalonamento para cada fila
  - método usado para determinar quando subir um processo
  - método usado para determinar quando baixar um processo
  - método usado para determinar em qual fila um processo entra quando precisa serviço

25/4/2007

INE5355 – Sistemas Operacionais I

6.34



## Exemplo: Filas Multinível c/ Feedback

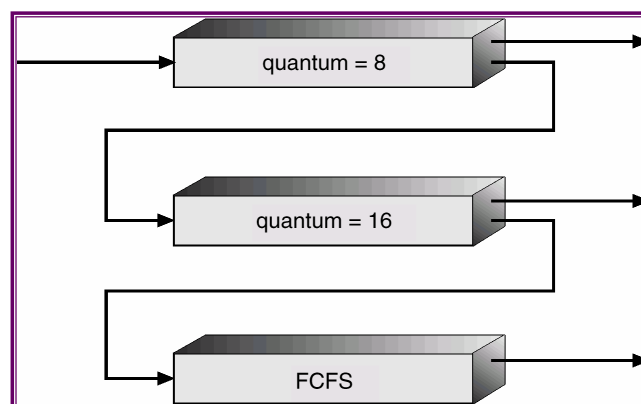
- 3 filas:
  - $Q_0$  – quantum 8 millisegs
  - $Q_1$  – quantum 16 millisegs
  - $Q_2$  – FCFS
- Escalonamento
  - Um novo job entra na fila  $Q_0$  servida por FCFS. Quando ganha UCP, o job recebe 8 millisegs. Se ele não termina em 8 millisegs, o job é movido para fila  $Q_1$ .
  - Em  $Q_1$  o job é servido por FCFS e recebe 16 millisegs. Se ainda não completa, ele é preemptado e movido para fila  $Q_2$ .

25/4/2007

INE5355 – Sistemas Operacionais I

6.37

## Filas Multinível c/ Feedback



25/4/2007

INE5355 – Sistemas Operacionais I

6.38

## Escalonamento UNIX Traditional

- Filas Multinível com feedback usando RR em cada uma das filas
- Prioridades são recalculadas uma vez por segundo
- Prioridade é baseada no tipo de processo e sua história de execução

$$P_j(i) = \text{Base}_j + (\text{CPU}_j / 2) + \text{nice}_j$$

$$\text{CPU}_j(i) = (\text{U}_j(i) / 2) + (\text{CPU}_j(i-1) / 2)$$

- Prioridade Base divide todos processos em faixas
- Fator de ajuste (CPU, nice) é usado para manter o processo na sua faixa (Base)

25/4/2007

INE5355 – Sistemas Operacionais I

6.39

## Faixas

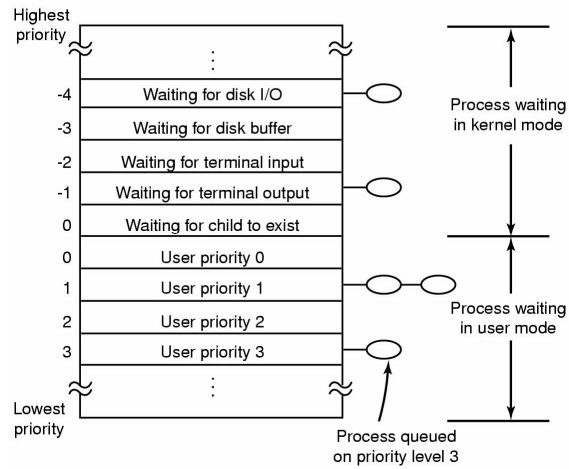
- Ordem de prioridade decrescente
  - Swapper
  - Controle de Dispositivos de E/S orientados a Bloco
  - Manipulação de Arquivos
  - Controle de Dispositivos de E/S orientados a caracter
  - Processos de usuário

25/4/2007

INE5355 – Sistemas Operacionais I

6.40

## UNIX Scheduler



O escalonador UNIX é baseado em uma estrutura de fila multinível

25/4/2007

INE5355 – Sistemas Operacionais I

6.41

Time	Process A		Process B		Process C	
	Priority	CPU Count	Priority	CPU Count	Priority	CPU Count
0	60	0	60	0	60	0
	1					
	2					
	...					
	60					
1	75	30	60	0	60	0
			1			
			2			
			...			
			60			
2	67	15	75	30	60	0
					1	
					2	
					...	
					60	
3	63	7	67	15	75	30
	8					
	9					
	...					
	67					
4	76	33	63	7	67	15
			8			
			9			
			...			
			67			
5	68	16	76	33	63	7

Shaded rectangle represents executing process

Figure 9.17 Example of Traditional UNIX Process Scheduling

## Escalonamento Fair-Share - FSS

- Aplicação do usuário executa como uma coleção de processos (threads)
- Usuário está preocupado com o desempenho da aplicação
- Necessário tomar decisões de escalonamento baseado em conjuntos de processos
  - Grupo de usuários
  - Atribuição de peso para utilização dos recursos
- Monitorar uso
  - Mais para aqueles que usam menos do que o fair share
  - Menos para aqueles que usam mais do que o fair share

25/4/2007

INE5355 – Sistemas Operacionais I

6.43

## Escalonamento FSS - Unix

- O sistema divide a comunidade de usuários em um conjunto de grupos *fair-share* e aloca uma fração do processador para cada grupo.
- Desta forma, se existirem 4 grupos, cada um vai ganhar 25% do tempo do processador.
- O escalonamento é feito com base em prioridades, levando em conta a prioridade do processo, sua utilização de CPU e a utilização de CPU do grupo ao qual o processo pertence.
- Quanto maior o valor numérico menor a prioridade.
- Os cálculos são baseados nas seguintes fórmulas:

25/4/2007

INE5355 – Sistemas Operacionais I

6.44

## Escalonamento FSS - Unix

- $P_j(i) = Base_j + CPU_j/2 + GCPU_k/4 \times W_k$

- $CPU_j(i) = U_j(i-1)/2 + CPU_j(i-1)/2$

- $GCPU_k(i) = GU_k(i-1)/2 + GCPU_k(i-1)/2$

- onde:

$P_j(i)$  = prioridade do processo j no inicio do intervalo i

$Base_j$  = prioridade base do processo j

$U_j(i)$  = utilização da CPU pelo processo j no intervalo i

$GU_k(i)$  = utilização da CPU pelo grupo k durante o intervalo i

$CPU_j(i)$  = utilização (ponderada) da CPU pelo processo j no intervalo i

$GCPU_k(i)$  = utilização (ponderada) da CPU pelo grupo k durante o intervalo i

$W_k$  = peso atribuído ao grupo (0..1)

25/4/2007

INE5355 – Sistemas Operacionais I

6.45

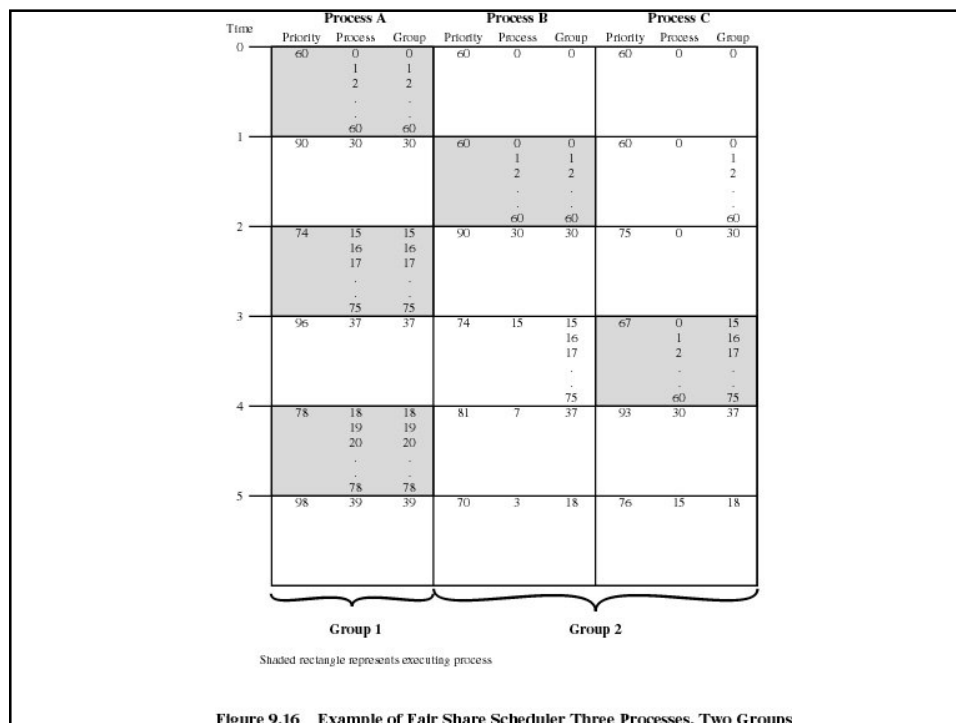
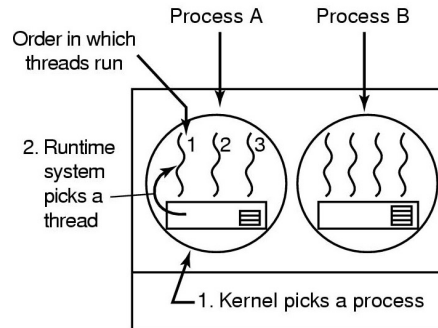


Figure 9.16 Example of Fair Share Scheduler Three Processes, Two Groups

## Escalonamento de Threads



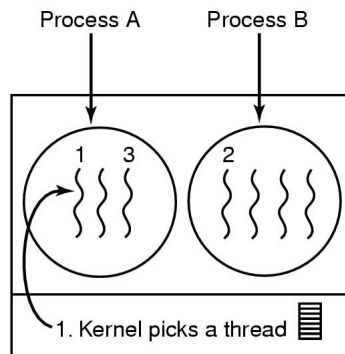
Possible: A1, A2, A3, A1, A2, A3  
 Not possible: A1, B1, A2, B2, A3, B3

25/4/2007

INE5355 – Sistemas Operacionais I

6.47

## Escalonamento de Threads



Possible: A1, A2, A3, A1, A2, A3  
 Also possible: A1, B1, A2, B2, A3, B3

25/4/2007

INE5355 – Sistemas Operacionais I

6.48



## MultiProcessador

- Quando se tem múltiplas UCPs o problema de escalonamento torna-se mais complexo.
- Processadores de um multiprocessador são idênticos (homogêneo)
  - qualquer processador que esteja disponível, pode ser usado para executar qualquer processo da fila. A questão que se deve resolver é se a atribuição deve ser estática ou dinâmica
- *Load sharing*
  - Cada processador é auto escalonado, ou seja, cada processador examina a fila de prontos comum e seleciona um processo para executar, sendo que desta forma devemos assegurar que dois processadores não irão escolher o mesmo processo.
- *Multiprocessamento assimétrico*
  - Apontar um processador como escalonador dos outros criando uma estrutura mestre - escravo. Somente um processador acessa as estruturas de dados do sistema, sem necessidade de compartilhar dados.
- Gang Scheduling
  - Algoritmo de escalonamento para multiprocessadores onde são escalonados threads/processos relacionados para executar em diferentes processadores

25/4/2007

INE5355 – Sistemas Operacionais I

6.49

## Tempo-Real

- *Hard real-time* – *task* é tarefa crítica que deve ser completada dentro de um tempo determinado e isto tem que ser garantido.
- *Soft real-time* – *tasks* são menos restritivos, requerem que os processos críticos recebam as prioridades maiores.
- O importante é que todas as tarefas hard sejam completadas em seus deadlines e que tanto quanto possível as tarefas soft também sejam completadas.
  - *Abordagens estáticas dirigidas por tabelas*
  - *Abordagens estáticas com preempção dirigidas por prioridade*
  - *Abordagens dinâmicas baseadas em planejamento*
  - *Abordagens dinâmicas baseadas no melhor esforço*

25/4/2007

INE5355 – Sistemas Operacionais I

6.50

## Avaliação de algoritmos

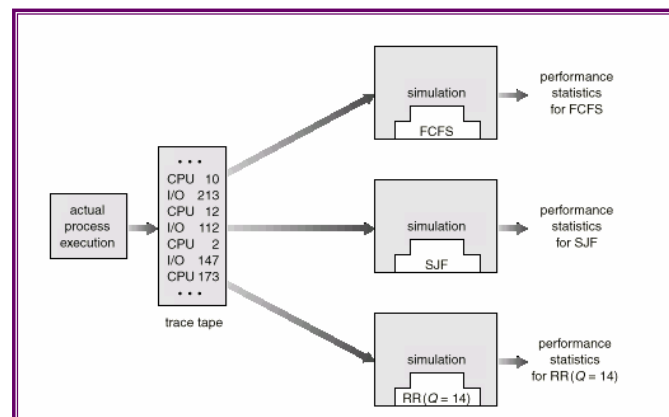
- Como selecionar um algoritmo de escalonamento da UCP para um sistema particular?
- Modelagem determinista – verifica para uma carga particular o desempenho de cada algoritmo.
- Modelos de filas
- Simulação
- Implementação

25/4/2007

INE5355 – Sistemas Operacionais I

6.51

## Simulação



25/4/2007

INE5355 – Sistemas Operacionais I

6.52

## A scheduling example

- Job 1, 10 seconds, priority 3
- Job 2, 2 seconds, priority 2
- Job 3, 5 seconds, priority 1
- Job 4, 3 seconds, priority 4
  
- Calcule o tempo médio de turnaround para as seguintes políticas de escalonamento :
  - FCFS, SJF, Prioridade, RR ( $q=2$ )