

MÚLTIPLA ESCOLHA

Acerca de processos dentro de um sistema operacional, assinale a alternativa correta.

A O espaço de endereçamento de um processo contém a pilha de execução de um programa executável.

B O sistema operacional encerra definitivamente a execução de um processo quando este consome seu tempo de CPU permitido.

C Processos podem se dividir em subprocessos, mas não podem invocar a execução de novos processos.

D O contador de programa é um valor reservado na CPU que informa o tamanho da fila de processos.

E Processos maliciosos não detectados por programas antivírus são denominados daemons.

MÚLTIPLA ESCOLHA - (INPE)

No contexto da informática, considerando que um processo pode estar em um dos três estados possíveis (executando, pronto ou bloqueado), analise as afirmativas a seguir e assinale (V) para a verdadeira e (F) para a falsa.

- () Um processo no estado “executando” está fisicamente utilizando a CPU (Unidade de Processamento Central).
- () Um processo no estado “pronto” significa que o mesmo foi executado com sucesso.
- () Um processo no estado “bloqueado” não pode mais ser executado.

As afirmativas são, respectivamente,

- A F – V – V.
- B F – V – F.
- C V – F – V.
- D V – F – F.**
- E V – V – F.

MÚLTIPLA ESCOLHA - (UERJ)

Em relação aos estados possíveis de um processo em um sistema operacional moderno, quando um determinado processo é interrompido por uma solicitação de disco, esse processo entrará no estado de:

- A pronto**
- B terminado
- C em espera
- D em execução

Escalonamento de CPU

Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

ANTERIORMENTE...

- **PROCESSO** como **unidade de execução**
- **PCB** (Bloco de Controle do Processo) representa o processo para o **S0**
 - *Estado, escalonamento, gerenciamento de memória, etc*
- NOVO | PRONTO | EM ESPERA | EXECUTANDO | FINALIZADO
- Em um processador único (**1 CPU**), existe no máximo **1 processo rodando por vez.**
- O programa que está executando na CPU muda com **TROCA DE CONTEXTO.**
- Processos se comunicam por **MENSAGENS** (send/receive) ou **MAPEAMENTO DE MEMÓRIA** (mmap)

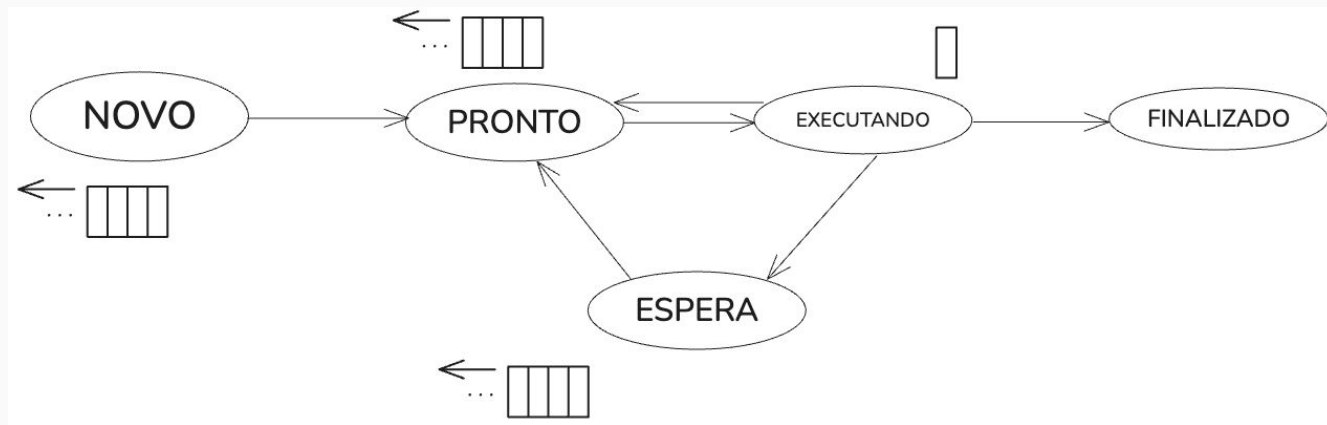
NESTA AULA: ALGORITMOS DE ESCALONAMENTO

- Qual é o **objetivo** de escalonar a CPU?
- Algoritmos **FCFS** (primeiro a chegar, primeiro a ser servido) e **Round Robin**
- **SJF** (menor trabalho primeiro)
- Filas de realimentação (**feedback queues**)
- Escalonamento por **loteria**

OBJETIVO DO ESCALONAMENTO: MULTIPROGRAMAÇÃO

MULTIPROGRAMAÇÃO: Executar mais de um processo ao mesmo tempo.

- Aumenta utilização do sistema
- Sobreposição de operações de I/O e atividade na CPU



- Todos os processos que o S0 está atualmente gerenciando estão em somente 1 das filas de estado.

ESCALONAMENTO DE PROCESSOS

- ESCALONAMENTO DE LONGO PRAZO

Como o S0 vai determinar o grau de multiprogramação, isto é, o número de tarefas executando ao mesmo tempo em memória principal?

- ESCALONAMENTO DE CURTO PRAZO

Como o S0 deve selecionar um processo na fila de “PRONTO” para executar?

- *Políticas: objetivos e opções*
- *Considerações sobre as implementações*

ESCALONAMENTO DE PROCESSOS

- ESCALONAMENTO DE LONGO PRAZO

Como o SO vai determinar o grau de multiprogramação, isto é, o número de tarefas executando ao mesmo tempo em memória principal?

O QUE ACONTECE SE O SISTEMA TIVER PROCESSOS DE MAIS EXECUTANDO?

DEFINE 1 LIMITE, A PARTIR DO QUAL PROCESSOS NÃO SÃO INICIALIZADOS E VÃO PARA UMA FILA DE "ESPERA PARA INICIALIZAR"

- ESCALONAMENTO DE CURTO PRAZO

Como o SO deve selecionar um processo na fila de "PRONTO" para executar?

- *Políticas: objetivos e opções*
- *Considerações sobre as implementações*

TODO NOVO PROCESSO É INICIALIZADO E ENTRA NO PIPELINE DE ESTADOS DE PROCESSO COMO "NOVO"

ESCALONAMENTO DE CURTO PRAZO

- *Quando o **KERNEL** vai rodar o escalonador?*
 - Quando um processo:
 - Troca de “executando” para “em espera”,
 - é interrompido,
 - é criado ou finalizado.
- **Sistema não-preemptivo**: o escalonador precisa esperar por um desses eventos acima.
- **Sistema preemptivo**: o escalonador pode interromper um processo em execução.

CRITÉRIOS PARA COMPARAR ALGORITMOS DE ESCALONAMENTO

- **UTILIZAÇÃO DA CPU** - % de tempo que a CPU está ocupada realizando trabalho útil.
- **VAZÃO (*throughput*)** - O número de processos executados (completos) em 1 unidade de tempo.
- **TEMPO DE RETORNO (*turnaround*)** - A duração de tempo TOTAL que demora para executar um processo desde a inicialização até a finalização, incluindo tempo em todas as filas.
- **TEMPO DE ESPERA** - Tempo total de espera de um processo na fila "PRONTO".
- **TEMPO DE RESPOSTA** - Tempo entre o processo estar pronto para executar, e a próxima (ou anterior) requisição I/O

CRITÉRIOS PARA COMPARAR ALGORITMOS DE ESCALONAMENTO

- **UTILIZAÇÃO DA CPU** - % de tempo que a CPU está ocupada realizando trabalho útil.

- **VAZÃO (*throughput*)** - O número de processos executados (completo)

- **TEMPO DE ATENDIMENTO** - O tempo TOTAL que um processo leva desde a submissão até a finalização.
...*tradeoffs*:

- **TEMPO DE ESPERA** - O tempo que um processo na fila "Pronto para executar" leva para ser executado.
- I/O vs Utilização de CPU,*
Sistemas interativos VS Sistemas em batch

- **TEMPO DE RESPOSTA** - O tempo que um processo leva desde a submissão até a finalização.
...*tradeoffs*:
I/O vs Utilização de CPU,
Sistemas interativos VS Sistemas em batch
etc...

POLÍTICAS DE ESCALONAMENTO

- O ideal seria otimizar todos os critérios, mas é impossível. Algoritmos são escolhidos pela habilidade de satisfazer uma política:

Minimizar média de tempo de resposta - **OUTPUT** para o usuário o mais **rápido** possível e processar o **INPUT** o mais **cedo** possível após recebimento.

Minimizar variação de tempo de resposta - Em sistemas **interativos**, porém, **previsibilidade** é **melhor** do que média baixa e alta variação.

POLÍTICAS DE ESCALONAMENTO

Maximizar vazão (throughput) - 2 aspectos:

Minimizar o **OVERHEAD** (custo adicional do S0 em **mudança de contexto**)

Utilizar de forma eficiente os recursos (CPU, I/O)

Minimizar tempo de espera - Dar a cada processo a **mesma quantidade de tempo** no processador. **ISSO PODE AUMENTAR TEMPO DE RESPOSTA!**

POLÍTICAS DE ESCALONAMENTO

SUPOSIÇÕES:

- UM PROCESSO POR USUÁRIO
- UMA THREAD POR PROCESSO
- PROCESSOS SÃO INDEPENDENTES

Essas suposições eram **realísticas** em 1970, mas hoje não mais.

Os algoritmos foram criados baseado nessas suposições, e ainda hoje é um problema em aberto como “relaxar” essas suposições em algoritmos atuais.

PRINCIPAIS ALGORITMOS

FCFS (First Come, First Served) | Primeiro a chegar, primeiro a ser servido. (FIFO)

Round Robin: Usa uma fatia de tempo por tarefa e preempção para alternar entre tarefas.

SJF (Shortest Job First) | Tarefas Menores Primeiro

Filas de Retroalimentação Multiníveis: Round Robin específico para cada fila de prioridade.

Escalonamento por loteria: Tarefas ganham bilhetes e o escalonador aleatoriamente sorteia um bilhete premiado.

FCFS (OU FIFO)

FCFS: First Come, First Served | Primeiro a chegar, primeiro a ser servido

ou FIFO - First In, First Out (Primeiro a entrar, primeiro a sair)

- O escalonador executa tarefas até o final na ordem em que elas chegam.
- No começo, o FCFS não liberava a CPU mesmo se estivesse bloqueado durante uma operação I/O.
- Vamos considerar um FCFS sem preempção, ou seja, se um processo precisa sair da CPU para fazer I/O, outro processo entra no seu lugar até o final (sem ser interrompido).

FCFS: EXEMPLO

Se processos chegam a cada 1 unidade de tempo, qual é o tempo médio de ESPERA desses três casos?

tempo →

ORDEM DE CHEGADA: B, C, A (SEM I/O)



ORDEM DE CHEGADA: A, B, C (SEM I/O)



ORDEM DE CHEGADA: A, B, C (A FAZ I/O)



A FAZ REQUISIÇÃO DE I/O

FCFS - VANTAGENS E DESVANTAGENS

Vantagem: é **simples**.

Desvantagens:

- **Tempo médio de espera** é VARIÁVEL, uma vez que **tarefas pequenas tem que esperar por tarefas maiores na fila**.
- **Sobreposição I/O + CPU é pobre:** os processos que estão vinculados à CPU fazem os processos vinculados a I/O esperar pela CPU, deixando os dispositivos I/O ociosos.

ROUND ROBIN

- Variantes de “Round Robin” são utilizadas na maior parte de sistemas que envolvem **compartilhamento de tempo**.
- Adiciona um **TIMER** e usa uma política **preemptiva**.
- Após cada **FATIA DE TEMPO**, move a thread em execução para o **FINAL DA FILA**.

Evita **INANIÇÃO** (Starvation).

Qual é o tempo MÁXIMO de espera garantido pelo Round Robin?

Como selecionar a fatia de tempo?

ROUND ROBIN

FATIA DE TEMPO

- Muito grande: tempo de espera aumenta, degenera para FCFS se processos não tiverem preempção.
- Muito pequeno: vazão diminui, pois muito tempo é gasto em troca de contexto.

Como equilibrar esses tradeoffs: *selecionar uma fatia de tempo em que TROCA DE CONTEXTO seja aprox. 1% dessa fatia de tempo.*

ROUND ROBIN

FATIA DE TEMPO

- Muito grande: tempo de espera aumenta, degenera para FCFS se processos não tiverem preempção.
- Muito pequeno: vazão diminui, pois muito tempo é gasto em troca de contexto.

Como equilibrar esses tradeoffs: *selecionar uma fatia de tempo em que TROCA DE CONTEXTO seja aprox. 1% dessa fatia de tempo.*

ATUALMENTE: fatia: 10-100 ms e context-switch: 0.1 - 1ms

VANTAGEM: é justo! CADA TAREFA TEM UMA OPORTUNIDADE IGUAL DE FAZER USO DA CPU.

DESVANTAGEM: tempo médio de retorno (completude) pode ser ruim.

ROUND ROBIN - EXEMPLOS

- 5 tarefas
 - 100 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	100				
2	100				
3	100				
4	100				
5	100				
Média					

ROUND ROBIN - EXEMPLOS

- 5 tarefas
 - 100 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	100	100			
2	100	200			
3	100	300			
4	100	400			
5	100	500			
Média		300			

ROUND ROBIN - EXEMPLOS

- 5 tarefas
 - 100 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	100	100	496		
2	100	200	497		
3	100	300	498		
4	100	400	499		
5	100	500	500		
Média		300	498		

ROUND ROBIN - EXEMPLOS

- 5 tarefas
 - 100 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	100	100	496	0	
2	100	200	497	100	
3	100	300	498	200	
4	100	400	499	300	
5	100	500	500	400	
Média		300	498	200	

ROUND ROBIN - EXEMPLOS

- 5 tarefas
 - 100 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	100	100	496	0	396
2	100	200	497	100	397
3	100	300	498	200	398
4	100	400	499	300	399
5	100	500	500	400	400
Média		300	498	200	398

ROUND ROBIN - EXEMPLO 2

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	50				
2	40				
3	30				
4	20				
5	10				
Média					

ROUND ROBIN - EXEMPLO 2

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	50	50			
2	40	90			
3	30	120			
4	20	140			
5	10	150			
Média		110			

ROUND ROBIN - EXEMPLO 2

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	50	50	150		
2	40	90	140		
3	30	120	120		
4	20	140	90		
5	10	150	50		
Média		110	110		

ROUND ROBIN - EXEMPLO 2

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		FCFS	Round Robin	FCFS	Round Robin
1	50	50	150	0	100
2	40	90	140	50	100
3	30	120	120	90	90
4	20	140	90	120	70
5	10	150	50	140	40
Média		110	110	80	80

SJF/SRTF: Tarefa Menor Primeiro (*Shortest Job First*)

- Escalonar a tarefa que tem a menor quantidade de trabalho (tempo de CPU) esperada até sua próxima requisição I/O ou término.

VANTAGENS:

- É provadamente **ótimo** para minimizar o tempo médio de espera
- Funciona **com preempção** ou **sem preempção**
- SJF preemptivo é chamado de **SRTF** (Shortest Remaining Time First) (**Primeiro a Terminar o Tempo Restante**)
 - *Exemplo: Processos chegam A(10s), B(8s). Depois de 4s, chega o processo C (2s). Com preempção: C tem o menor tempo restante.*

SJF/SRTF: Tarefa Menor Primeiro (*Shortest Job First*)

- Escalonar a tarefa que tem a menor quantidade de trabalho (tempo de CPU) esperada até sua próxima requisição I/O ou término.

VANTAGENS:

- É provadamente **ótimo** para minimizar o tempo médio de espera
- Funciona **com preempção ou sem preempção**
- SJF preemptivo é chamado de **SRTF** (Shortest Remaining Time First) (**Primeiro a Terminar o Tempo Restante**)

Consequência: Tarefas limitadas por I/O tem prioridade sobre tarefas limitadas por CPU. Porquê?

SJF/SRTF: Tarefa Menor Primeiro (*Shortest Job First*)

- Escalonar a tarefa que tem a menor quantidade de trabalho (tempo de CPU) esperada até sua próxima requisição I/O ou término.

VANTAGENS:

- É provadamente **ótimo** para minimizar o tempo médio de espera
- Funciona **com preempção ou sem preempção**
- SJF preemptivo é chamado de **SRTF** (Shortest Remaining Time First) (**Primeiro a Terminar o Tempo Restante**)

DESVANTAGENS:

SJF/SRTF: Tarefa Menor Primeiro (*Shortest Job First*)

- Escalonar a tarefa que tem a menor quantidade de trabalho (tempo de CPU) esperada até sua próxima requisição I/O ou término.

VANTAGENS:

- É provadamente **ótimo** para minimizar o tempo médio de espera
- Funciona **com preempção ou sem preempção**
- SJF preemptivo é chamado de **SRTF** (Shortest Remaining Time First) (**Primeiro a Terminar o Tempo Restante**)

DESVANTAGENS:

- Não é possível implementar dessa forma ideal, pois é impossível ter certeza sobre o tamanho das tarefas!
- Tarefas maiores podem entrar em INANIÇÃO (*starvation*)

SJF - EXEMPLO

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE			TEMPO DE ESPERA		
		FCFS	Round Robin	SJF	FCFS	Round Robin	SJF
1	50	50	150		0	100	
2	40	90	140		50	100	
3	30	120	120		90	90	
4	20	140	90		120	70	
5	10	150	50		140	40	
Média		110	110		80	80	

SJF - EXEMPLO

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE			TEMPO DE ESPERA		
		FCFS	Round Robin	SJF	FCFS	Round Robin	SJF
1	50	50	150	150	0	100	
2	40	90	140	100	50	100	
3	30	120	120	60	90	90	
4	20	140	90	30	120	70	
5	10	150	50	10	140	40	
Média		110	110	70	80	80	

SJF - EXEMPLO

- 5 tarefas
 - 50, 40, 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos

Tarefa	Duração	TEMPO DE COMPLETUDE			TEMPO DE ESPERA		
		FCFS	Round Robin	SJF	FCFS	Round Robin	SJF
1	50	50	150	150	0	100	100
2	40	90	140	100	50	100	60
3	30	120	120	60	90	90	30
4	20	140	90	30	120	70	10
5	10	150	50	10	140	40	0
Média		110	110	70	80	80	40

FILAS DE RETORNO EM MÚLTIPLOS NÍVEIS

- FCFS é **injusto** e **aumenta o tempo ocioso de I/O**
- SJF otimiza o tempo de espera, mas incorre em perigo de INANIÇÃO. É impossível prever com acurácia.
- Round Robin é **justo** e **evita inanição**, mas **aumenta o tempo de espera.**

⇒ **FILAS DE RETORNO EM MÚLTIPLOS NÍVEIS** usam comportamento passado para prever o futuro e atribuir prioridades.

também conhecido como **FILAS MULTINÍVEL (MLQ)**

FILAS MULTINÍVEL (MLQ)

- **Objetivo: superar o problema de “prever” o tamanho de uma tarefa para o SJF**
- As menores tarefas são **geralmente** vinculadas a I/O
- Esse é um efeito colateral do SJF: favorecer tarefas I/O
- Otimizar isso também nos leva a uso ótimo de CPU e I/O

ENTÃO COMO PREVER O TAMANHO DE UMA TAREFA? =>
PREVER SE ELA FAZ MUITO I/O

FILAS MULTINÍVEL (MLQ)

- Se um processo **faz I/O no passado, é provável que faça I/O no futuro** (*programas não são tão aleatórios assim*)
- Para explorar esse comportamento, o escalonador pode favorecer tarefas que usam pouca CPU, **se aproximando do SJF**
- Essa política é **ADAPTATIVA**, ou seja, se baseia em comportamento anterior - logo, mudanças no comportamento ao longo de muitas execuções podem levar a mudanças no escalonador.

FILAS MULTINÍVEL (MLQ)

- MÚLTIPLAS FILAS com prioridades diferentes!
- ROUND ROBIN em cada nível de prioridade, executando tarefas em prioridades maiores primeiro.
- Depois que as tarefas de maior prioridade acabam, **ROUND ROBIN no próximo nível de prioridade, e assim em diante.** Pode levar à **INANIÇÃO**. (PORQUÊ?)
- A **FATIA DE TEMPO** cresce **EXPONENCIALMENTE EM PRIORIDADES MENORES**. (PORQUÊ?)

FILAS MULTINÍVEL (MLQ)

	PRIORIDADE			FATIA DE TEMPO		
<table><tr><td></td><td>G</td><td>F</td><td>A</td></tr></table>		G	F	A	1	1
	G	F	A			
<table><tr><td></td><td></td><td></td><td>E</td></tr></table>				E	2	2
			E			
<table><tr><td></td><td></td><td>D</td><td>B</td></tr></table>			D	B	3	4
		D	B			
<table><tr><td></td><td></td><td></td><td>C</td></tr></table>				C	4	8
			C			

FILAS MULTINÍVEL (MLQ)

- Se a fila de prioridades altas estiver sempre cheia \Rightarrow **INANIÇÃO das prioridades baixas.**
- Tarefas vinculadas à CPU têm fatias exponencialmente maiores.

• COMO AJUSTAR ESSAS PRIORIDADES AO LONGO DO TEMPO?

FILAS MULTINÍVEL (MLQ)

- Todas as tarefas começam na maior prioridade.
- Se a fatia de tempo da tarefa expira, ela cai para uma prioridade abaixo.
- Se a fatia de tempo não expira (a mudança de contexto vem de uma interrupção, por exemplo), então aumente a prioridade em 1 nível, sem ultrapassar o máximo.

FILAS MULTINÍVEL (MLQ)

- Todas as tarefas começam na maior prioridade.
- Se a fatia de tempo da tarefa expira, ela cai para uma prioridade abaixo.
- Se a fatia de tempo não expira (a mudança de contexto vem de uma interrupção, por exemplo), então aumente a prioridade em 1 nível, sem ultrapassar o máximo.

FILAS MULTINÍVEL (MLQ)

- Todas as tarefas começam na maior prioridade.
- Se a fatia de tempo da tarefa expira, ela cai para uma prioridade abaixo.
- Se a fatia de tempo não expira (a mudança de contexto vem de uma interrupção, por exemplo), então aumente a prioridade em 1 nível, sem ultrapassar o máximo.

FILAS MULTINÍVEL (MLQ)

- Todas as tarefas começam na maior prioridade.
- Se a fatia de tempo da tarefa expira, ela cai para uma prioridade abaixo.
- Se a fatia de tempo não expira (a mudança de contexto vem de uma interrupção, por exemplo), então aumente a prioridade em 1 nível, sem ultrapassar o máximo.

DESSA FORMA, TAREFAS-CPU CAEM RÁPIDO DE NÍVEL (PRIORIDADE) E TAREFAS-I/O SE MANTÉM EM PRIORIDADES ALTAS.

FILAS MULTINÍVEL (MLQ) - EXEMPLO

- 3 tarefas
 - 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos
 - todas CPU, nenhuma I/O
 - 3 filas

FILA	FATIA DE TEMPO	TAREFA
1	1	
2	2	
3	4	

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILAS MULTINÍVEL (MLQ) - EXEMPLO

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	
3	4	

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	A_5^3 , B_7^3 , C_9^3
3	4	

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	A_5^3 , B_7^3 , C_9^3
3	4	A_{13}^7 , B_{17}^7 , C_{21}^7

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	A_5^3 , B_7^3 , C_9^3
3	4	A_{13}^7 , B_{17}^7 , C_{21}^7 , A_{25}^{11} , B_{29}^{11} , C_{32}^{10} ...

TAREFA_{TEMPO}^{TEMPO_EXEC}

FILAS MULTINÍVEL (MLQ) - EXEMPLO

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60	60	30	
B	20	50	53	30	
C	10	30	32	20	
Média		46 2/3	48 1/3	26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	A_5^3 , B_7^3 , C_9^3
3	4	A_{13}^7 , B_{17}^7 , C_{21}^7 , A_{25}^{11} , B_{29}^{11} , C_{32}^{10} ...

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

- 3 tarefas
 - 30, 20 e 10 segundos cada
 - fatia de tempo 1 segundo
 - mudança de contexto 0 segundos
 - tarefa C faz I/O de duração 1s a cada 1s
 - 2 filas

FILA	FATIA DE TEMPO	TAREFA
1	1	
2	2	

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	
2	2	

TAREFA_{TEMPO}TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1
2	2	

TAREFA_{TEMPO} ^{TEMPO_EXEC}

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A ₁ ¹ , B ₂ ¹ , C ₃ ¹
2	2	A ₅ ³

TAREFA

TEMPO

TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3
2	2	A_5^3

$$TAREFA_{TEMPO}^{TEMPO_EXEC}$$

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3
2	2	A_5^3 , B_8^3

$TAREFA_{TEMPO}^{TEMPO_EXEC}$

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3 , C_9^4
2	2	A_5^3 , B_8^3

$TAREFA_{TEMPO}^{TEMPO_EXEC}$

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A ₁ ¹ , B ₂ ¹ , C ₃ ¹ , C ₆ ³ , C ₉ ⁴
2	2	A ₅ ³ , B ₈ ³ , A ₁₁ ⁵

TAREFA

TEMPO

TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3 , C_9^4 , C_{12}^7
2	2	A_5^3 , B_8^3 , A_{11}^5

TAREFA_{TEMPO} ^{TEMPO_EXEC}

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3 , C_9^4 , C_{12}^7
2	2	A_5^3 , B_8^3 , A_{11}^5 , B_{14}^5

$TAREFA_{TEMPO}^{TEMPO_EXEC}$

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3 , C_9^4 , C_{12}^7 , C_{15}^9
2	2	A_5^3 , B_8^3 , A_{11}^5 , B_{14}^5

TAREFA_{TEMPO} ^{TEMPO_EXEC}

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A ₁ ¹ , B ₂ ¹ , C ₃ ¹ , C ₆ ³ , C ₉ ⁴ , C ₁₂ ⁷ , C ₁₅ ⁹
2	2	A ₅ ³ , B ₈ ³ , A ₁₁ ⁵ , B ₁₄ ⁵ , A ₁₇ ⁷

TAREFA_{TEMPO}
TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	A_1^1 , B_2^1 , C_3^1 , C_6^3 , C_9^4 , C_{12}^7 , C_{15}^9 , C_{18}^{10}
2	2	A_5^3 , B_8^3 , A_{11}^5 , B_{14}^5 , A_{17}^7

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ) - EXEMPLO 2

Tarefa	Duração	TEMPO DE COMPLETUDE		TEMPO DE ESPERA	
		RR	MLQ	RR	MLQ
A	30	60		30	
B	20	50		30	
C	10	30		20	
Média		46 2/3		26 2/3	

FILA	FATIA DE TEMPO	TAREFA
1	1	$A_1^1, B_2^1, C_3^1, C_6^3, C_9^4, C_{12}^7, C_{15}^9, C_{18}^{10}$
2	2	$A_5^3, B_8^3, A_{11}^5, B_{14}^5, A_{17}^7, B_{20}^7, A_{22}^9, B_{20}^9, A_{11}^{11}, B_{11}^{11}, A_{13}^{13}, B_{13}^{13}, A_{15}^{15}, B_{15}^{20}, A_{24}^{24}, B_{26}^{19}, A_{28}^{19}, B_{30}^{19}, A_{32}^{32}, B_{34}^{34}, A_{36}^{36}, B_{38}^{38}, A_{40}^{40}, B_{42}^{42}$

TAREFA_{TEMPO} TEMPO_EXEC

FILAS MULTINÍVEL (MLQ)

- Ainda tem problemas: é **injusto**

MLQ favorece explicitamente tarefas I/O. Tarefas de muita CPU podem sofrer inanição.

- Como tornar o MLQ + justo?

FILAS MULTINÍVEL (MLQ)

- Ainda tem problemas: é **injusto**

MLQ favorece explicitamente tarefas I/O. Tarefas de muita CPU podem sofrer inanição.

- Como tornar o MLQ + justo?

DIVIDIR A CPU.

EX: 80% DA CPU PARA PRIORIDADE 1, 20% PARA AS OUTRAS PRIORIDADES.

FILAS MULTINÍVEL (MLQ)

- Ainda tem problemas: é **injusto**

MLQ favorece explicitamente tarefas I/O. Tarefas de muita CPU podem sofrer inanição.

- Como tornar o MLQ + justo?

DIVIDIR A CPU.

EX: 80% DA CPU PARA PRIORIDADE 1, 20% PARA AS OUTRAS PRIORIDADES.

Só funciona se a distribuição for apropriada. Não queremos ter 80% de CPU ociosa, queremos?

FILAS MULTINÍVEL (MLQ): MAIS IMPARCIALIDADE

PROBLEMA: *Já que SJF é injusto, qualquer aumento em imparcialidade ao dar mais tempo para tarefas maiores enquanto tarefas menores estão disponíveis, vai diminuir o tempo de espera.*

- Como evitar **inanição** provocada pelo SJF?

Possíveis soluções:

1. **Dar para cada fila uma fração da CPU.** Só é viável se a distribuição das tarefas pelas filas for apropriada.
2. **Ajustar a PRIORIDADE de tarefas que estão esperando há muito tempo (UNIX original fazia isso)**

Evita inanição, mas aumenta tempo de espera pois quando o sistema está sobrecarregado todas as tarefas ficam com prioridade máxima.

ESCALONAMENTO POR LOTERIA

- Dá a cada tarefa um número de um bilhete.

ESCALONAMENTO POR LOTERIA

- Dá a cada tarefa um número de um bilhete.
- A cada fatia de tempo, aleatoriamente escolhe um bilhete vencedor.

ESCALONAMENTO POR LOTERIA

- Dá a cada tarefa um número de um bilhete.
- A cada fatia de tempo, aleatoriamente escolhe um bilhete vencedor.
- Em média, tempo de CPU é proporcional ao número de bilhetes dado para cada tarefa.

ESCALONAMENTO POR LOTERIA

- Dá a cada tarefa um número de um bilhete.
- A cada fatia de tempo, aleatoriamente escolhe um bilhete vencedor.
- Em média, tempo de CPU é proporcional ao número de bilhetes dado para cada tarefa.
- Atribui o maior número de bilhetes possível para as menores tarefas, e menos bilhetes para tarefas maiores (Aproximando do SJF). Para evitar INANIÇÃO, cada tarefa ganha pelo menos 1 bilhete.

ESCALONAMENTO POR LOTERIA

- Dá a cada tarefa um número de um bilhete.
- A cada fatia de tempo, aleatoriamente escolhe um bilhete vencedor.
- Em média, tempo de CPU é proporcional ao número de bilhetes dado para cada tarefa.
- Atribui o maior número de bilhetes possível para as menores tarefas, e menos bilhetes para tarefas maiores (Aproximando do SJF). Para evitar INANIÇÃO, cada tarefa ganha pelo menos 1 bilhete.

Degrada rapidamente. Adicionar ou deletar uma tarefa afeta todas as outras proporcionalmente, independente do número de bilhetes que uma tarefa tem.

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1		
0/2	-	
2/0		-
10/1		
1/10		

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1	91% (10/11)	9% (1/11)
0/2	-	
2/0		-
10/1		
1/10		

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1	91% (10/11)	9% (1/11)
0/2	-	50% (1/2)
2/0		-
10/1		
1/10		

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1	91% (10/11)	9% (1/11)
0/2	-	50% (1/2)
2/0	50% (10/20)	-
10/1		
1/10		

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1	91% (10/11)	9% (1/11)
0/2	-	50% (1/2)
2/0	50% (10/20)	-
10/1	10% (10/101)	< 1% (1/101)
1/10		

ESCALONAMENTO POR LOTERIA - EXEMPLO

- TAREFAS CURTAS RECEBEM 10 BILHETES CADA.
- TAREFAS LONGAS RECEBEM 1 BILHETE CADA.

# TAREFAS CURTAS / # TAREFAS LONGAS	% DE CPU QUE CADA TAREFA CURTA RECEBE	% DE CPU QUE CADA TAREFA LONGA RECEBE
1/1	91% (10/11)	9% (1/11)
0/2	-	50% (1/2)
2/0	50% (10/20)	-
10/1	10% (10/101)	< 1% (1/101)
1/10	50% (10/20)	5% (1/20)

ESCALONAMENTO - RESUMO

- FCFS - Injusto, tempo médio de espera é ruim.
- Round Robin - Imparcial, mas tempo médio de espera é ruim.
- SJF - Injusto, mas tempo médio de espera é mínimo (ótimo) assumindo que conseguimos prever com acurácia a duração da próxima tarefa na CPU. Inanição é possível.
- MLQ (Filas Multinível) - Aproximação do SJF
- Loteria - Não determinístico

NOSSA MODELAGEM ASSUMIU QUE MUDANÇAS DE CONTEXTO TOMAM 0 SEGUNDOS, O QUE NÃO É REALISTA.

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.