

# Sistemas Operacionais

## Módulo 3 – Escalonamentos

# Gustavo Vilar

- Mini – CV
  - PPF / DPF – Papiloscopista Policial Federal
  - Pós-Graduado em Docência do Ensino Superior – UFRJ
  - Graduado em Ciência da Computação e Processamento de Dados – ASPER/PB
  - Aprovações: PRF 2002, PF 2004, MPU 2010, ABIN 2010

# Gustavo Vilar

- Contatos:

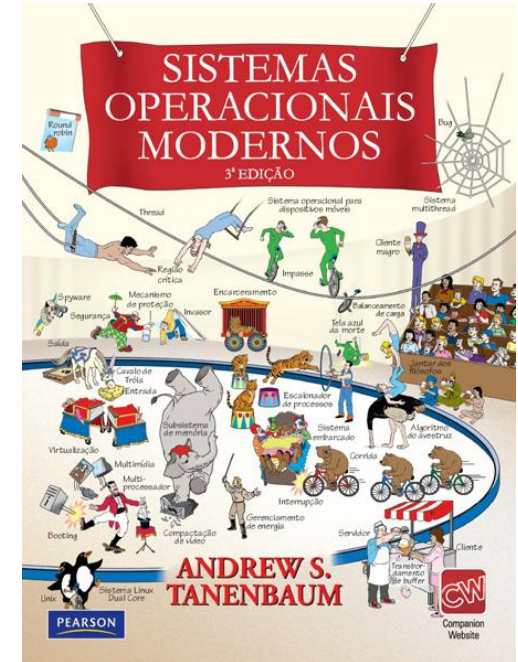
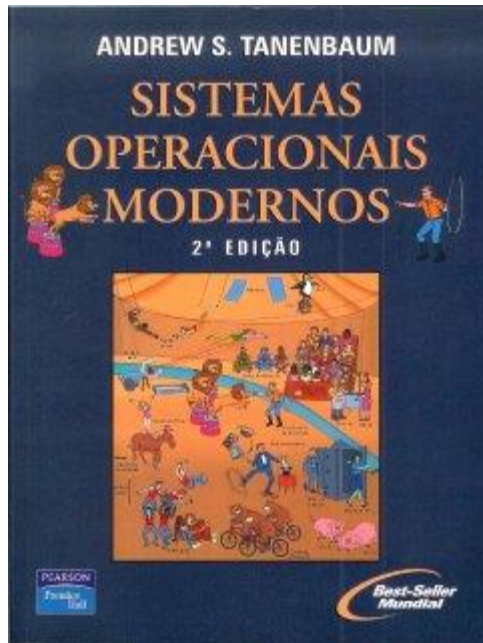
- [gustavopintovilar@gmail.com](mailto:gustavopintovilar@gmail.com)
- [p3r1t0f3d3r4l@yahoo.com.br](mailto:p3r1t0f3d3r4l@yahoo.com.br)



# Escopo

- Abordar os assuntos mais recorrentes e com fortes tendências para concursos atuais
- Familiarizar o concursando com os tipos de questões mais freqüentes.
- Abordar as metodologias de resolução de questões das principais bancas

# Bibliografia



# Sistemas Operacionais 03 de 03 – Carga Horária

- 10 vídeo aulas (03h55m / 00h23m32s)
  - Estratégias Básicas de Escalonamento
    - Lote
      - FCFS/FIFO, SJF, SRTN
    - Tempo Real
    - Interativos
      - Round robin, prioridades, filas múltiplas, SPN, EG, FS, EL
  - Primeira bateria de questões de aprendizagem
  - Substituição de páginas
    - OPT, RAND, FIFO, FIFO com buffer, SC, relógio, NUR, LFU, LRU, WS, WS clock
  - Segunda bateria de questões de aprendizagem



# Sistemas Operacionais

## Escalonamentos de CPU

# Esquema visual





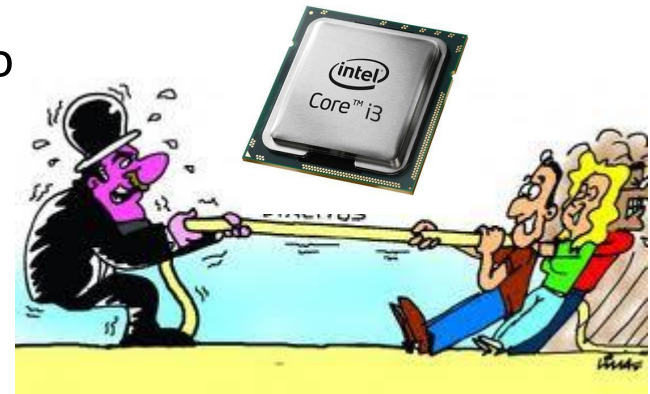
# Escalonamentos de CPU

- **Não preemptivo**

- Processo executa até bloquear, até que voluntariamente libere a CPU
- Estratégia de rodar o processo até o fim. (batch). (não são adequados aos sistemas propósito geral)
- Nunca será compulsoriamente suspenso

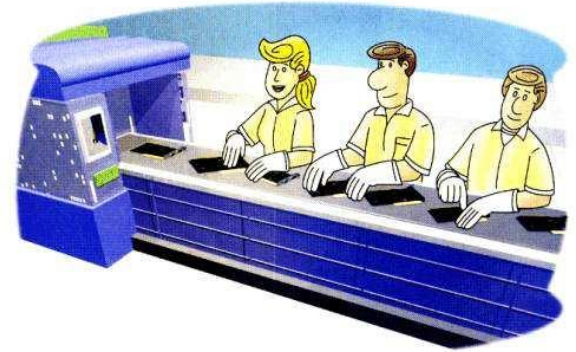
- **Preemptivo**

- Processo executa por um quantum de tempo
- Estratégia de permitir a suspensão temporária dos processos que poderiam continuar rodando
- SO interrompe um processo a qq instante
- Chaveamento de contexto
- Requer interrupção de relógio



# Estratégias de Escalonamento

- **Em lote**
- **Tempo real**
- **Ambientes interativos**



# Escalonamentos de CPU – Em Lote

- Não existem usuários impacientes esperando resposta
- Não preemptivos ou preemptivos com longo intervalo de tempo
- Reduz chaveamentos entre processos, aumentando o desempenho
- Métricas
  - Vazão
    - Número de jobs por hora
  - Tempo de retorno
    - tempo médio do momento que uma tarefa em lote é submetida até o momento em que é terminada
    - Quanto menor, melhor
  - Utilização de CPU
    - Não é um excelente parâmetro



# Escalonamentos de CPU – Em Lote

- FCFS - First come, first served / FIFO\*
  - Não preemptivo
  - Desbloqueados e novos vão para o fim da fila
  - Jobs IO bounds entram várias vezes no fim da fila
  - Não há regras para a saída
  - O primeiro a entrar pode sair em outra posição
  - É justo - Impede starvation
  - Jobs curtos esperam muito
  - Importância da tarefa não é considerada
- Shortest Job/process first
  - Não preemptivo
  - Adequado somente quando todos jobs disponíveis simultaneamente
  - Conhecimento prévio do tempo de execução
  - Favorece os processos mais curtos
- Shortest remaining time next
  - Favorece menor tempo médio de espera
  - Ordem de acordo com tempo de execução, onde o menor vem primeiro
  - Produz maior variância no tempo de espera
  - Reduz o tempo de retorno
  - Produz o menor tempo de retorno para Lote
  - Proporciona Starvation

# Escalonamentos de CPU – Tempo Real

- Preempção, ALGUMAS vezes, desnecessária
  - processos cooperam para a aplicação
- Dá prioridade ao processo em detrimento do sistema operacional
- Produzir resultados em determinados instantes de tempo
- Segue as restrições de tempo real
- Processos sabem que não podem executar por muito tempo e bloqueiam
- Executam apenas programas que visam o progresso da aplicação
- O que importa
  - Cumprimento dos prazos
  - Previsibilidade



# Escalonamentos de CPU – Tempo Real

- Tempo real crítico / Hard real-time
  - Prazos são absolutos
  - Prazos sempre atendidos
- Tempo real não crítico / Soft real-time
  - Descumprimento ocasional é indesejável, entretanto, tolerável
  - Não garante o cumprimento de suas restrições de tempo
- Eventos
  - Periódicos: Garante-se o escalonamento
  - Aperiódicos: Imprevisíveis
- Algoritmos
  - Estáticos: Decisões de escalonamento ANTES do sistema executar
  - Dinâmicos: Decisões de escalonamento EM TEMPO DE EXECUÇÃO



# Escalonamentos de CPU – Interativos

- Preempção é fundamental
- Executam programas de propósito geral / arbitrários / mal intencionados
- O que importa
  - Tempo de resposta
  - Proporcionalidade (expectativa)



# Escalonamentos de CPU – Interativos

- **Escalonamento por alternância circular (round-robin)**

- Preemptivo
- Quantum é o que importa
- Bom para processos interativos
- Premissa: Processos são igualmente importantes
- Não permite monopólio do processador

- Tamanho do quantum:
  - muito pequeno: troca de contexto cara.
  - muito grande: tempo de resposta alto.
- Adiciona sobrecarga no chaveamento de contexto
- Menor throughput





# Escalonamentos de CPU – Interativos

## Escalonamento por alternância circular (round-robin)



# Escalonamentos de CPU – Interativos

- **Escalonamento por prioridades**

- Preemptivo
- Cada processo possui uma prioridade (classes)

- Fim do processo = quantum para a classe de maior numero, ou seja, de menor prioridade

- Alocação de prioridades

- Prioridades atribuídas estática ou dinamicamente
  - Estática: e.g. processos que monitoram a temperatura do urânio rodam primeiro.
  - Dinâmica: e.g. processos que estão

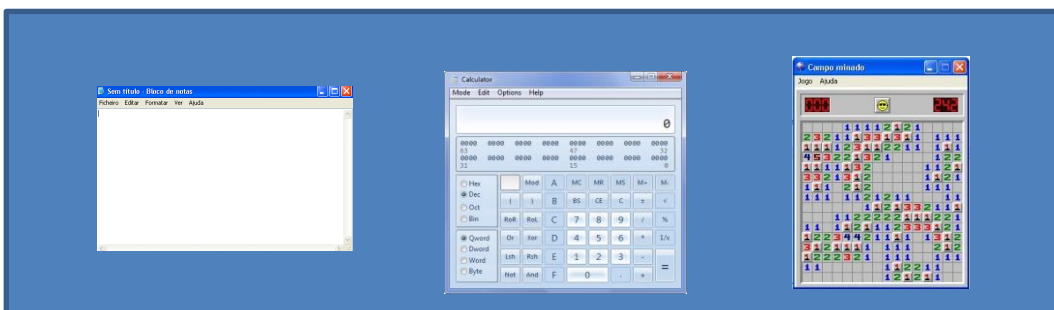
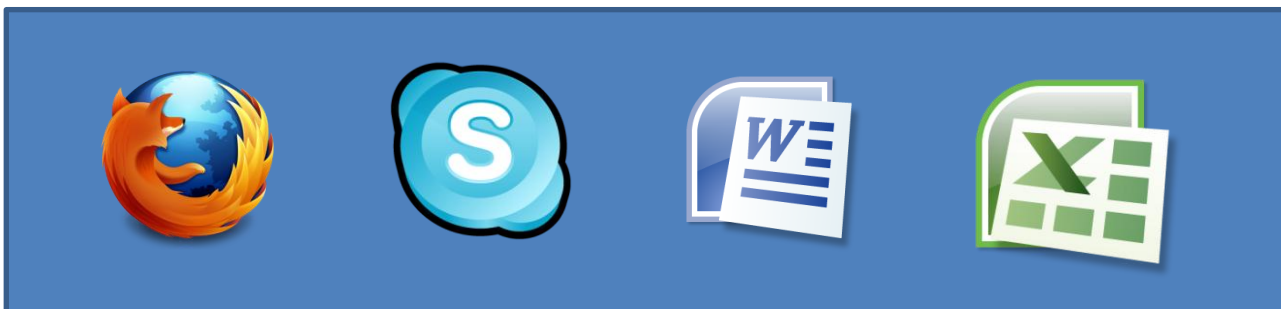
mais perto do seu prazo rodam primeiro

- É comum o uso de classes e dentro das classes uso do Round Robin
  - Pode ocorrer Inanição nas classes mais baixas
    - aging para contornar o problema



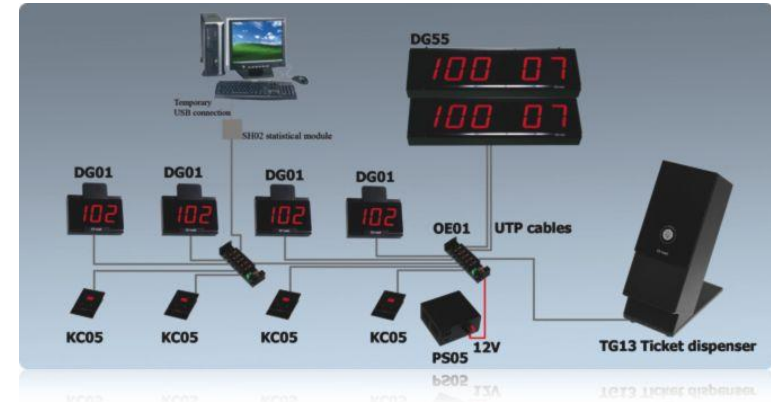
# Escalonamentos de CPU – Interativos

## Escalonamento por prioridades



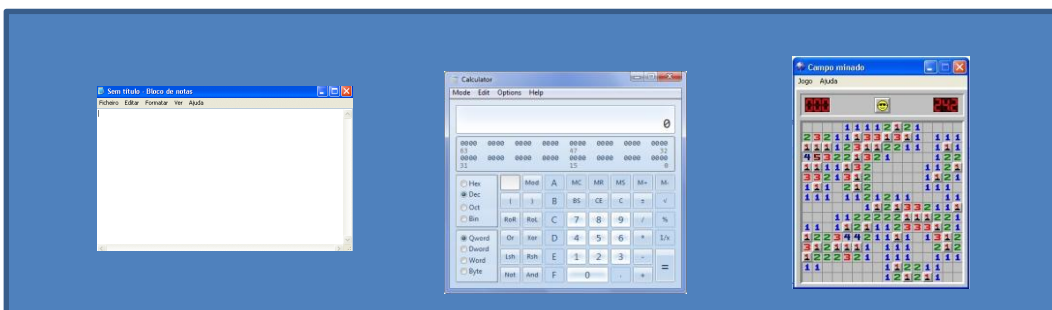
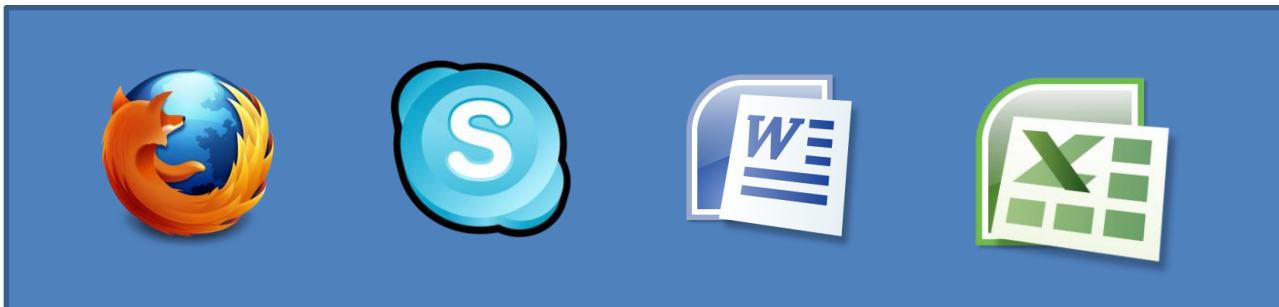
# Escalonamentos de CPU – Interativos

- **Filas múltiplas**
  - Preemptivo
  - Cada fila possui quantuns diferentes, quanto mais prioritária for a fila, maior o quantum
  - Troca os processos entre elas
    - Esgotamento do quantum
    - Mudança da orientação ( IO ou CPU bound)
    - Prioriza os processos IO bound
  - processo novo com prioridade mais alta roda primeiro que o um processo antigo com prioridade mais baixa



# Escalonamentos de CPU – Interativos

## Escalonamento por filas múltiplas



# Escalonamentos de CPU – Interativos

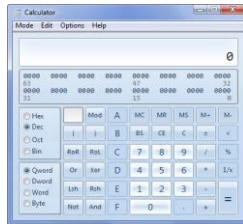
## **Shortest Process Next**

- Preemptivo
- Problema: Qual é o mais curto?
- Saída: Estimativa no passado
  - Tempo da próxima execução é calculado com base no histórico anterior
  - Última execução é fundamental no cálculo da média
- É a versão interativa de algoritmo em lote SRTN



# Escalonamentos de CPU – Interativos

## Shortest Process Next



# Escalonamentos de CPU – Interativos

## Escalonamento Garantido

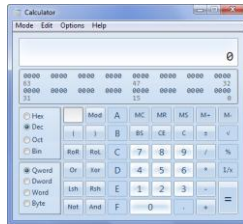
- $1/n$
- $n$  = usuários ou processos
- Foco no usuário para sistemas com muitos usuários
- Foco nos processos para sistemas monousuários





# Escalonamentos de CPU – Interativos

## Escalonamento Garantido



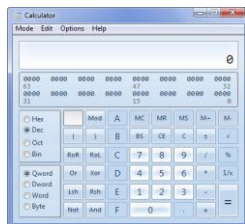
# Escalonamentos de CPU – Interativos

## Fair-Share

- Preemptivo
- Fração alocada por usuário e depois por processo
- Escalonamento de 2 níveis
  - 1 nível – Estático
    - Grupos de processos
    - Cada grupo recebe um quantum diferente
    - Grupos podem ser formados por usuário, prioridade de tarefas, etc...
  - 2 nível - Dinâmico
    - Dentro de cada grupo é implementada um RR, considerando o uso do processador
    - Prioridade de CPU aumenta com o histórico de uso de CPU
- Mais justo
- Mais complexo

# Escalonamentos de CPU – Interativos

## Fair-Share



# Escalonamentos de CPU – Interativos

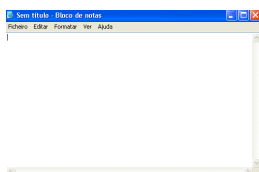
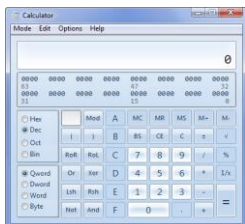
## Escalonamento por loteria

- Processos + importantes recebem bilhetes extras
- Processos cooperativos trocam bilhetes entre si
- Prêmio são recursos do sistema

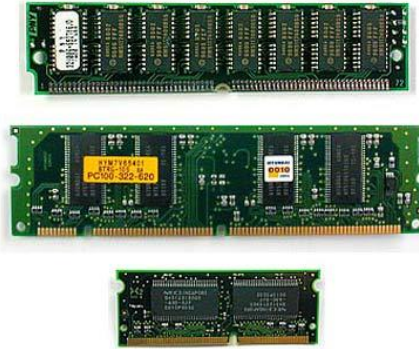


# Escalonamentos de CPU – Interativos

## Escalonamento por loteria



# Esquema visual



# Bateria de questões de aprendizagem 1

Sistemas Operacionais

1. Alguns dos objetivos dos algoritmos de escalonamento de processos são comuns a todos os tipos de sistemas operacionais. Outros, entretanto, variam de acordo com o tipo de sistema. Qual dos objetivos abaixo **NÃO** se aplica a algoritmos de escalonamento de processos utilizados em sistemas voltados para o processamento em lote (batch)?
- A. Atender às requisições dos usuários o mais rápido possível.
  - B. Manter a CPU ocupada o tempo todo.
  - C. Manter os dispositivos de E/S ocupados o máximo de tempo possível.
  - D. Maximizar o número de jobs processados por unidade de tempo.
  - E. Minimizar o tempo entre a submissão e o término de um job.



2. A gerência do processador é uma das principais atividades de um sistema operacional, na qual são estabelecidos critérios para a escolha do processo que fará uso do processador com base em uma política de escalonamento. Um exemplo de escalonamento preemptivo é o escalonamento

- A. first come, first served
- B. first-in, first-out
- C. shortest job first
- D. shortest process first
- E. round robin

## **BNDES – CESGRANRIO 2011 – Profissional Básico – Análise e Desenvolvimento de Sistemas**

3. A gerência do processador é uma das atividades mais importantes em um sistema multiprogramável. Uma política de escalonamento deve ser estabelecida para determinar qual processo será escolhido para fazer uso do processador.

Com relação a essa política, considere as afirmações abaixo.

- I - O escalonamento preemptivo é caracterizado pela possibilidade de o sistema operacional interromper um processo em execução e passá-lo para o estado de espera, com o objetivo de colocar outro processo em execução.
- II - Com o uso da preempção, é possível ao sistema priorizar a execução de processos, como no caso de aplicações de tempo real onde o fator tempo é crítico.
- III - No escalonamento não preemptivo, quando um processo está em execução nenhum evento externo pode ocasionar a perda do uso do processador.
- IV - O escalonamento FIFO (First-In First-Out) é um exemplo de escalonamento não preemptivo no qual o processo que chega primeiro ao estado de pronto é colocado em execução e só perde o uso do processador quando termina seu processamento ou quando executa instruções do próprio código que ocasionam uma mudança para o estado de pronto.

É correto **APENAS o que se afirma em**

- A. I e II
- B. II e III
- C. III e IV
- D. I, II, III
- E. II, III e IV

## **Pref Ibiporã – AOCP 2011 – Analista de Sistemas**

4. Sobre Gerência do Processador em sistemas operacionais, analise as assertivas e assinale a alternativa que aponta as corretas.
- I. Um algoritmo de escalonamento tem como principal função decidir qual dos processos prontos para execução deve ser alocado à UCP.
  - II. Em sistemas multiprogramáveis não existem algoritmos de escalonamento de processos.
  - III. Na maioria dos sistemas é desejável que o processador permaneça a maior parte do seu tempo ocupado.
  - IV. Cada sistema operacional necessita de um algoritmo de escalonamento adequado ao seu tipo de processamento.
- A. Apenas I e II.
  - B. Apenas II e III.
  - C. Apenas I, II e IV.
  - D. Apenas I, III e IV.
  - E. I, II, III e IV.

## INMETRO – CESPE 2010 – Pesq TMQ – Infraestrutura e Redes de TI

5. Assinale a opção correta a respeito de gerenciamento de processador.

- A. Em um sistema computacional, os processos podem estar em execução, livres ou prontos para serem executados. Quando um ou mais processos estão livres, o sistema operacional decide qual deles vai ser executado primeiro, por meio do algoritmo de definição de concorrentes ou assíncronos.
- B. O algoritmo de escalonamento round robin considera a prioridade dos processos, de modo que o processo com maior prioridade associada tem preferência de uso do processador para execução.
- C. Todas as operações que envolvem processos são controladas pelo núcleo, core ou kernel do sistema operacional. O núcleo, normalmente, representa somente uma pequena parte do código, que, em geral, é tratado como sendo todo o sistema operacional, mas é a parte de código mais intensivamente utilizada.
- D. Em sistemas multiprogramados, o compartilhamento de recursos pode ser garantido com a utilização de deadlock, que permite a definição dos processos que serão executados.
- E. Em um ambiente de multiprogramação, quando existe apenas um processador na instalação, cada processo é executado de cada vez, de forma seriada. O sistema operacional aloca a CPU para cada processo em uma ordem que é previsível, considerando fatores externos aos processos, que variam no tempo de acordo com as demandas.

6. Se os processos A, B e C, que requerem 7, 12 e 16 segundos, respectivamente, para serem completados, começarem sua execução ao mesmo tempo em um ambiente de multiprocessamento, o tempo médio de execução ( average turnaround time ) dos processos será, julgue itens.

[87] de 21 s, se for usado o escalonamento preemptivo round robin.

[88] de 20,33 s, se for usado o escalonamento não-preemptivo shortest job first.

7. O escalonamento de tarefas é uma atividade de processamento realizada pela CPU de um computador. Esta atividade permite executar de forma mais eficiente os processos considerados prioritários para o sistema operacional. Assinale a alternativa que apresenta o escalonamento de tarefas em um computador, utilizado como servidor de arquivos de uma rede.
- A. O escalonamento garantido busca atender a demanda da rede, priorizando ações de leitura e escrita em arquivos e banco de dados.
  - B. O algoritmo de escalonamento FIFO (First In, First Out) atua na gravação de arquivos em disco, implementando o conceito de pilha de escalonamento.
  - C. Os algoritmos de escalonamento preemptivos devem permitir que um processo seja interrompido durante sua execução.
  - D. O algoritmo de escalonamento de múltiplas filas permite o acesso simultâneo a arquivos e banco de dados disponibilizados na rede.
  - E. O escalonador de longo prazo seleciona os processos na interface de rede, dando prioridade às ações de I/O (Input/Output).

**EMBASA – CESPE 2010 – Técnico em Programação e Suporte de TI - Produção**

8. A respeito dos conceitos relativos a sistemas operacionais, aritmética computacional, comandos Linux e software livre, julgue os itens a seguir.

[78] O algoritmo alternância circular é usado em um SO para realizar escalonamento de processos utilizando intervalo de tempo (quantum) no qual ele é permitido executar. Essa característica associada à preempção torna esse algoritmo imune a situação de starvation.

# Gabarito

1. A

2. E

3. B

4. D

5. C

6. E,C

7. C

8. C



# Algoritmos de Substituição de Páginas

- Quando existe espaço disponível na memória principal, não existem muitos problemas para trazermos um processo do disco para a memória. A única preocupação é escolher o melhor endereço para inserção da página.
- Mas quando não temos espaço disponível...o que fazer?
- **R: Substituir**



# Esquema visual



# Algoritmos de Substituição de páginas

- Quando existe espaço disponível na memória principal, não existem muitos problemas para trazermos um processo do disco para a memória. A única preocupação é escolher o melhor endereço para inserção da página.
- Mas quando não temos espaço disponível...o que fazer?
- **R: Substituir**

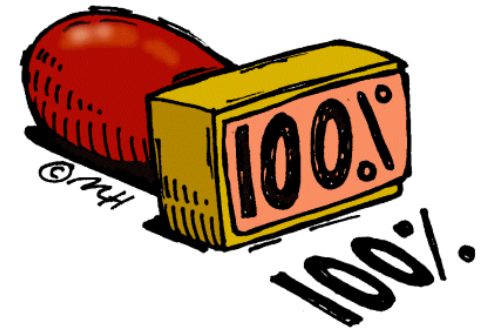


# Algoritmos de Substituição de páginas

## SOLUÇÃO IDEAL

- **Ótimo (impossível)**

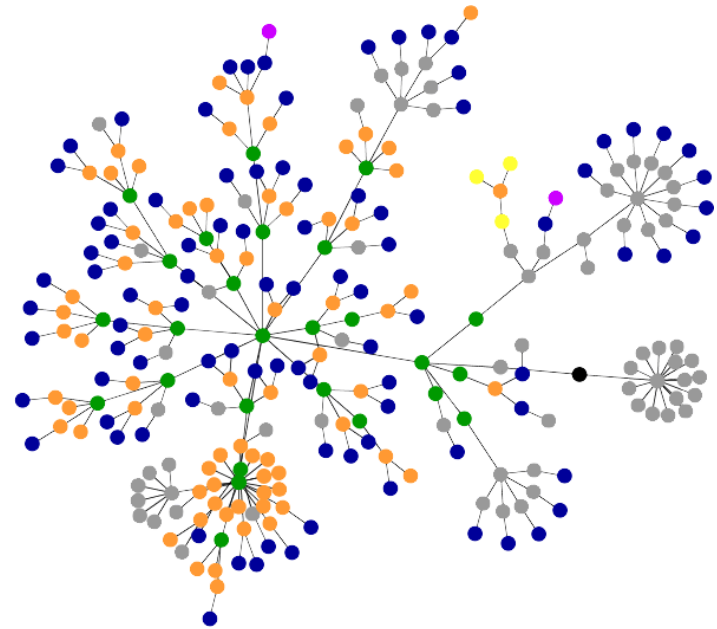
- Cada página é rotulada com o número de instruções que faltam para ela ser acessada
- Remover página com maior rótulo
- Quando determinada página será usada? daqui a 8 milhões de instruções? daqui a 10 instruções?
- Adia a ocorrência da falta de página
- Impossível de implementar
- Referências futuras são imprevisíveis
- Só pode ser implementado depois da execução
- Serve como baliza para as demais estratégias



# Algoritmos de Substituição de páginas

## Página Longínqua

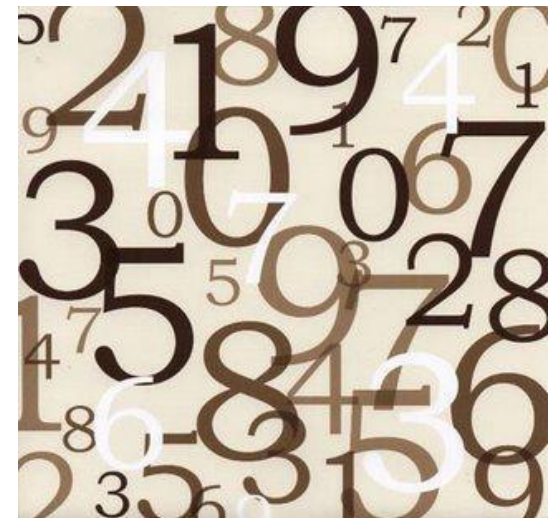
- Uso de grafos
- mais se aproxima da OPT
- Escolhe sempre a mais longe de ser referenciada
- Alta sobrecarga
  - Não é implementada no mercado



# Algoritmos de Substituição de páginas

## Rand / Randômico / Aleatório

- Descarrega uma página qualquer
  - Imprevisibilidade
  - Desconsidera o princípio da localidade



# Algoritmos de Substituição de páginas

## FIFO – First In First out

- Uma página muito referenciada será removida só porque chegou primeiro à memória
- Não leva em conta o grau de utilização da página
- Descarrega o frame mais antigo
- Desconsidera o princípio da localidade
- Baixo custo



# Algoritmos de Substituição de páginas

## FIFO – First In First out com buffer de páginas (Linux)

- FIFO com buffer de páginas
  - 2 listas de páginas
    - Ativas (LPA)
    - Inativas (LPI)

1	Atlético-MG●	28	11	9	1	1	23	8	15	84.8
2	Vasco●	26	11	8	2	1	19	11	8	78.8
3	Fluminense●	25	11	7	4	0	22	7	15	75.8
4	Grêmio●	21	11	7	0	4	16	10	6	63.6
5	Cruzeiro●	20	11	6	2	3	15	11	4	60.6
6	São Paulo●	19	11	6	1	4	14	11	3	57.6
7	Internacional●	19	11	5	4	2	17	11	6	57.6
8	Botafogo●	17	11	5	2	4	21	16	5	51.5
9	Ponte Preta●	15	11	4	3	4	14	14	0	45.5
10	Flamengo●	15	11	4	3	4	15	17	-2	45.5
11	Náutico●	13	11	4	1	6	13	20	-7	39.4
12	Corinthians●	12	11	3	3	5	11	13	-2	36.4
13	Sport●	12	11	3	3	5	12	18	-6	36.4
14	Palmeiras●	10	11	2	4	5	12	12	0	30.3
15	Santos●	10	11	1	7	3	7	9	-2	30.3
16	Coritiba●	9	11	2	3	6	17	23	-6	27.3
17	Portuguesa●	9	11	2	3	6	7	15	-8	27.3
18	Figueirense●	8	11	1	5	5	13	19	-6	24.2
19	Bahia●	8	11	1	5	5	9	18	-9	24.2
20	Atlético-GO●	5	11	1	2	8	8	22	-14	15.2

● Muito difícil de cair    ● Plausível    ● Concorrentes diretos



# Algoritmos de Substituição de páginas

## SC – Second Chance / Segunda Chance

- Alteração no FIFO, examina o bit R da página mais antiga
  - Se  $R = 0$ ; descarte
  - Se  $R = 1$ ;  $R = 0$ ; e insere no final da lista
- Procura página antiga que não tenha sido referenciada
- Ineficaz, pois permanece constantemente inserindo páginas no final da fila
- Se todas as páginas forem referenciadas, o Segunda Chance vira um FIFO puro



# Algoritmos de Substituição de páginas

## Relógio “SC Circular”

- Difere do Segunda Chance apenas na implementação; é uma estratégia melhor
  - Funcionamento
    - Um ponteiro aponta para a mais antiga (cabeça) da lista
    - Na falta de página, o bit R da página mais antiga é examinado
    - Se  $R = 0$ ; a página é substituída
    - Se  $R = 1$ ;  $R = 0$  e o ponteiro avança para a próxima mais antiga
    - O processo é repetido até uma página antiga com  $R = 0$  ser encontrada.



# Algoritmos de Substituição de páginas

## Não Usada Recentemente - NUR/NRU

- Utiliza os bits R (referenciada) e M(modificada)
  - BR periodicamente zerado
  - Bit R possui maior prioridade que o bit M
- Fácil de entender e implementar
- Mais eficiente que LRU e LFU
- O bit R é limpo periodicamente de modo que diferencie páginas que não foram usadas recentemente das que foram
- Classes
  - Classe 0: R=0 e M=0
    - Remove aleatoriamente uma página da classe de ordem mais baixa
  - Classe 1: R=0 e M=1
  - Classe 2: R=1 e M=0
  - Classe 3: R=1 e M=1

# Algoritmos de Substituição de páginas

## Menos Frequentemente Usada / MFU / LFU

- Descarrega aquela página menos referenciada,
- Considera as referências tenham ocorrido no passado distante
- Princípio da localidade temporal não é levado em conta
- Problema: “Nunca esquece de nada”



# Algoritmos de Substituição de páginas

## Menos Recentemente Usada / MRU / LRU

- Descarrega aquela que está a mais tempo sem ser referenciada, mesmo que muito referenciada no passado distante
- Princípio da localidade temporal
- Difícil, ou quase impossível implementação
  - É necessário manter uma lista encadeada de todas as páginas na memória e essa lista deve ser atualizada a cada referência na memória
  - Existem maneiras de se implementar utilizando hardware especial (encontrado em poucas máquinas, talvez nenhuma)



# Algoritmos de Substituição de páginas

## Menos Recentemente Usada / MRU / LRU

- Solução implementável em software
  - Emprega o algoritmo de substituição de página não usada frequentemente (NUF)
  - Utiliza contadores em software, cada um associado à uma página
    - Página com menor contagem é substituída
    - Problema: nunca esquece de nada
  - Transforma o NFU numa aproximação do LRU
  - Considera uma contagem nos últimos X ciclos



# Algoritmos de Substituição de páginas

## WS / Working Set / Conjunto de Trabalho

- Princípio da localidade de referência temporal e espacial
  - Fornece baixa taxa de thrashing
  - Durante a execução, os processos referenciam uma quantidade pequena de páginas
- Pré-paginação em vez de demanda
  - Carregar as páginas de um processo na memória antes de ele ser posto em execução
- A ideia principal é encontrar uma página que não esteja presente no conjunto de trabalho e removê-la da memória
- O conjunto de trabalho de um processo pode ser visto como o conjunto de páginas que ele referenciou durante os últimos  $x$  segundos do tempo virtual
- Varia no tempo e possui uma janela de tamanho  $W$  (quantidade) -  $WS = f(t, w)$
- É o conjunto das páginas usadas nas  $k$  mais recentes referências à página
- Utiliza o bit  $R$  e a idade para descartar páginas



# Algoritmos de Substituição de páginas

## WS Clock

- Melhoramento do Working Set (conjunto de trabalho), baseado no algoritmo do relógio
- Amplamente utilizado devido à simplicidade e desempenho
- Utiliza uma lista circular de molduras de páginas
- Cada entrada contém: instante da última referência, bit R e bit M





# Bateria de questões de aprendizagem 2

Sistemas Operacionais

## PETROBRÁS – CESGRANRIO 2012 – Analista de Sistemas Júnior – Engenharia de Software

1. Seja o seguinte algoritmo de substituição de página:

Todas as páginas são mantidas em uma lista circular, e um ponteiro (H) referencia a página mais antiga.

Quando uma falta de página ocorre, a página apontada por H é inspecionada:

Caso o seu bit de referência seja igual a 0, a página é retirada da lista e uma nova é inserida no seu lugar. O ponteiro H passa a apontar para a próxima página da lista.

Caso o seu bit de referência seja igual a 1, ele será zerado, e H passará a apontar para a próxima página da lista.

O processo é repetido até que uma página onde  $R=0$  seja encontrada.

Qual é o nome desse algoritmo de substituição de página?

- A. Ótimo
- B. Relógio
- C. (FIFO) Primeira a Entrar, Primeira a Sair
- D. (NUR) Não Usada Recentemente
- E. (MRU) Menos Recentemente Usada

**TRANSPETRO – CESGRANRIO 2011 – Analista de Sistemas Júnior – Software**

2. Um processo referencia 5 páginas identificadas por p1, p2, p3, p4 e p5, na seguinte ordem: P1, P2, P3, P1, P4, P2, P5, P1, P2, P5, P2, P1

Considerando-se que o algoritmo de substituição de página seja LRU e que a memória principal encontra-se inicialmente vazia, qual é o número de transferências de páginas em um sistema com 3 quadros em memória principal?

- A. 6
- B. 7
- C. 8
- D. 9
- E. 10

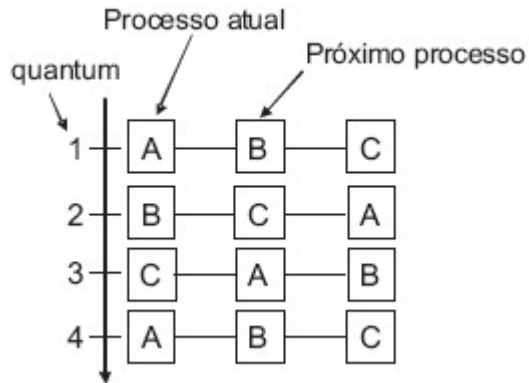
**CEF– CESGRANRIO 2008 – Técnico bancário - TI**

3. Observe, a seguir, o conjunto de processos prontos para execução e a duração de surto de CPU, expressa em milissegundos.

Processo	Duração de Surto
R	5
S	8
T	9
X	3

Considerando-se que o primeiro processo será escalonado no instante 0 ( zero ), qual o tempo de espera, em milissegundos, para o processo T, de acordo com o algoritmo de escalonamento job mais curto primeiro (Shortest Job First - SJF)?

- A. 0
- B. 6,25
- C. 9
- D. 16
- E. 25



4. Qual algoritmo de escalonamento é ilustrado pela figura acima?

- A. Escalonamento por loteria
- B. Escalonamento por prioridades
- C. Primeiro a chegar, primeiro a ser servido
- D. Próximo processo mais curto
- E. Round-robin

SEFAZ-AM – NCE 2005 – Analista de TI

5. Suponha que o esquema adotado para gerenciamento de memória de um determinado computador seja baseado na estratégia de *working sets* -  $W(t, \delta)$ , com política de re-alocação de página do tipo LRU - Least Recently Used e  $\Delta = 3$ . Nessas condições, se um determinado processo apresentar a seguinte sequência de referências a páginas virtuais: 24, 15, 18, 23, 24, 18, 17, 18, 24, 17, 17, 15, 24, 17, 24 e 18, o número de ocorrência de *page faults* será de:

- A. 09
- B. 19
- C. 11
- D. 13
- E. 10

# Gabarito

1. B
2. B
3. D
4. E
5. E