

INSTITUTO FEDERAL DO ESPÍRITO SANTO

Sistemas Operacionais

CAMPUS SERRA

Caderno de Exercícios

Prof. Flávio Giraldeli

v1.0.0

Sumário

Cap. 1: Introdução.....	4
Cap. 2: Estruturas do Sistema Operacional	5
Cap. 3: Processos	6
Cap. 4: Threads	7
Cap. 5: Escalonamento de CPU	8
Cap. 6: Sincronismo de Processos	11
Cap. 7: Deadlocks	12
Cap. 8: Memória Principal	14
Cap. 9: Memória Virtual	17
Cap. 10: Interface do Sistema de Arquivos	21
Cap. 11: Implementação do Sistema de Arquivos	22
Histórico de Versões	23

Notas

- Os exercícios estão classificados segundo o grau de aprendizagem adquirida. Apesar de ser recomendado que todos os exercícios sejam feitos, deve-se dar prioridade aos com mais asteriscos, ou seja...

[***] > [**] > [*] > “nada”

Quanto mais asteriscos, mais “interessantes” eles foram considerados pelo professor. ☺

- Os exercícios puramente conceituais (aqueles cuja resposta pode ser facilmente encontrada no livro texto e/ou slides) são importantes para revisão e estão marcados com **[R]**. Atenção: Isso **não** os torna desprezíveis!

Cap. 1: INTRODUÇÃO

- 1) [R] Quais são as três finalidades principais de um sistema operacional?
- 2) [*] Em um ambiente de multiprogramação e tempo compartilhado, vários usuários compartilham o sistema simultaneamente. Essa situação pode resultar em diversos problemas de segurança.
 - a. Cite dois desses problemas.
 - b. Podemos assegurar o mesmo nível de segurança tanto em uma máquina dedicada como em uma máquina de tempo compartilhado? Explique sua resposta.
- 3) A questão da utilização de recursos assume formas distintas em diferentes tipos de sistemas operacionais. Liste que recursos devem ser gerenciados cuidadosamente nas configurações a seguir:
 - a. Sistemas mainframe ou de minicomputador
 - b. Estações de trabalho conectadas a servidores
 - c. Computadores móveis
- 4) [R] Descreva as diferenças entre os multiprocessamentos simétrico e assimétrico. Cite três vantagens e uma desvantagem de sistemas multiprocessadores.
- 5) [*] Defina as propriedades essenciais dos tipos de sistema operacional a seguir:
 - a. Batch
 - b. Interativo
 - c. De tempo compartilhado
 - d. De tempo real
 - e. De rede
 - f. Paralelo
 - g. Distribuído
 - h. Em Cluster
 - i. Móvel
- 6) [R] Quais são as cinco principais atividades de um sistema operacional relacionadas ao **gerenciamento de processos**?
- 7) [R] Quais são as três principais atividades de um sistema operacional relacionadas ao **gerenciamento de memória**?
- 8) [R] Quais são as três principais atividades de um sistema operacional relacionadas ao **gerenciamento de memória secundária**?

Cap. 2: ESTRUTURAS DO SISTEMA OPERACIONAL

- 1) [R] Qual é a finalidade das chamadas de sistema?
- 2) [R] Qual é a finalidade dos programas de sistema?
- 3) [R] Qual é a principal vantagem da abordagem em camadas para o projeto de sistemas? Quais são as desvantagens do uso da abordagem em camadas?
- 4) [**] Liste cinco serviços fornecidos por um sistema operacional e explique por que cada um deles é conveniente para os usuários. Em que casos seria impossível programas de nível de usuário fornecerem esses serviços? Explique sua resposta.
- 5) [R] Como seria o projeto de um sistema que permitisse a escolha do sistema operacional a partir do qual se dará a inicialização? O que o programa bootstrap teria que fazer?
- 6) [R] Por que a separação entre mecanismo e política é desejável?
- 7) [R] Qual é a principal vantagem da abordagem de microkernel para o projeto de sistemas? Como os programas de usuário e serviços do sistema interagem em uma arquitetura de microkernel? Quais são as desvantagens do uso da abordagem de microkernel?
- 8) [R] Qual é a principal vantagem de usar uma arquitetura de máquina virtual para um projetista de sistema operacional? Qual é a principal vantagem para o usuário?

Cap. 3: PROCESSOS

- 1) [**] O Palm OS não fornece um meio de processamento concorrente.
Discuta três grandes complicações que o processamento concorrente adiciona a um sistema operacional.
- 2) [R] Descreva as diferenças entre o escalonamento de curto prazo, de médio prazo e de longo prazo.
- 3) [R] Descreva as ações executadas por um kernel na mudança de contexto entre processos.
- 4) [R] Explique por que programas são ditos serem entidades *passivas*, enquanto processos são entidades *ativas*.
- 5) [R] Cite e descreva os 5 estados básicos que um processo pode assumir num sistema operacional.
- 6) [R] Existem basicamente dois modelos de comunicação interprocesso (IPC). Quais são eles e como funcionam?
- 7) [R] Defina passagem de mensagem com bloqueio/síncrona e sem bloqueio/assíncrona.

Cap. 4: THREADS

- 1) [******] Forneça dois exemplos de programação em que a criação de vários threads proporcione melhor desempenho do que uma solução com um único thread.
- 2) [******] Forneça dois exemplos de programação em que a criação de vários threads não proporcione melhor desempenho do que uma solução com um único thread.
- 3) [R] Cite duas diferenças entre os threads de nível de usuário e os de nível de kernel. Sob que circunstâncias um tipo é melhor do que o outro?
- 4) [R] Descreva as ações executadas por um kernel para mudar o contexto entre threads de nível de kernel.
- 5) Descreva as ações executadas por uma biblioteca de threads para mudar o contexto entre threads de nível de usuário.
- 6) [R] Que recursos são usados quando um thread é criado? Em que eles diferem dos usados quando um processo é criado?
- 7) [******] Sob que circunstâncias uma solução com vários threads usando múltiplos threads de kernel fornece melhor desempenho do que uma solução com um único thread em um sistema com apenas um processador?
- 8) [R] Qual dos componentes de estado de programa a seguir são compartilhados pelos threads em um processo com vários threads?
 - a. Valores do registrador
 - b. Memória do heap
 - c. Variáveis globais
 - d. Memória da pilha
- 9) [******] Uma solução com vários threads usando múltiplos threads de nível de usuário pode obter melhor desempenho em um sistema multiprocessador do que em um sistema com um único processador? Explique.
- 10) [*******] Considere um sistema multiprocessador e um programa com vários threads escrito com o uso do modelo muitos-para-muitos de criação de threads. Permita que a quantidade de threads de nível de usuário do programa seja maior do que a quantidade de processadores do sistema. Discuta as implicações de desempenho dos cenários a seguir.
 - a. A quantidade de threads do kernel alocada para o programa é menor do que a quantidade de processadores.
 - b. A quantidade de threads do kernel alocada para o programa é igual ao número de processadores.
 - c. A quantidade de threads do kernel alocada para o programa é maior do que a quantidade de processadores, mas menor do que a quantidade de threads de nível de usuário.

Cap. 5: ESCALONAMENTO DE CPU

- 1) [R] Explique a diferença entre escalonamento com e sem preempção.
- 2) [*] Suponhamos que os processos a seguir chegassem para execução nos momentos indicados. Cada processo será executado durante o período de tempo listado. Ao responder às perguntas, use o escalonamento sem preempção e baseie todas as decisões nas informações disponíveis no momento em que a decisão tiver de ser tomada.

Processo	Tempo de Chegada	Duração do Pico
P ₁	0,0	8
P ₂	0,4	4
P ₃	1,0	1

- a. Qual é o tempo médio de turnaround desses processos com o algoritmo de escalonamento FCFS?
 - b. Qual é o tempo médio de turnaround desses processos com o algoritmo de escalonamento SJF?
 - c. O algoritmo SJF deveria melhorar o desempenho, mas observe que optamos por executar o processo P₁ no momento 0 porque não sabíamos que dois processos mais curtos estavam para chegar. Calcule qual será o tempo médio de turnaround se a CPU for deixada ociosa durante a primeira unidade de tempo 1 para então o escalonamento SJF ser usado. Lembre que os processos P₁ e P₂ estão esperando durante esse tempo ocioso e, portanto, seu tempo de espera pode aumentar. Esse algoritmo poderia ser chamado de escalonamento de conhecimento futuro.
- 3) [*] Qual a vantagem de termos tamanhos diferentes para o quantum de tempo em níveis distintos de um sistema de enfileiramento em vários níveis?
 - 4) [**] Muitos algoritmos de escalonamento da CPU são parametrizados. Por exemplo, o algoritmo RR requer um parâmetro que indique a parcela de tempo. Filas com retroalimentação em vários níveis requerem parâmetros que definam a quantidade de filas, o algoritmo de escalonamento para cada fila, os critérios usados para mover processos entre as filas e assim por diante.

Portanto, na verdade, esses algoritmos são conjuntos de algoritmos (por exemplo, o conjunto de algoritmos RR para todas as parcelas de tempo etc.). Um conjunto de algoritmos pode incluir outro (por exemplo, o algoritmo FCFS é o algoritmo RR com um quantum de tempo infinito).

Que relação existe (se existir alguma) entre os conjuntos de pares de algoritmos a seguir?
 - a. Por prioridades e SJF
 - b. Filas com retroalimentação em vários níveis e FCFS
 - c. Por prioridades e FCFS
 - d. RR e SJF
 - 5) [*] Suponhamos que um algoritmo de escalonamento (do nível do escalonamento de CPU de curto prazo) favorecesse os processos que usaram o menor tempo do processador no passado recente. Por que esse algoritmo favorecerá programas limitados por I/O e ao mesmo tempo não deixará os programas limitados por CPU em estado permanente de inanição?

- 6) [*] Suponhamos que um sistema operacional mapeasse threads de nível de usuário para o kernel usando o modelo muitos-para-muitos e que o mapeamento fosse feito através do uso de LWPs. Além disso, o sistema permite que os desenvolvedores de programas criem threads de tempo real. É necessário vincular um thread de tempo real a um LWP?
- 7) [*] Por que é importante o escalonador distinguir programas limitados por I/O de programas limitados pela CPU?
- 8) [**] Explique como os pares de critérios de escalonamento a seguir entram em conflito em certas configurações.
- Utilização da CPU e tempo de resposta
 - Tempo médio de turnaround e tempo máximo de espera
 - Utilização de dispositivos de I/O e utilização da CPU
- 9) [**] Considere o conjunto de processos a seguir, com a duração do pico de CPU dada em milissegundos:

Processo	Duração do Pico	Prioridade
P ₁	10	3
P ₂	1	1
P ₃	2	3
P ₄	1	4
P ₅	5	2

Presume-se que os processos tenham chegado na ordem P₁, P₂, P₃, P₄, P₅, todos no momento 0.

- Desenhe quatro gráficos de Gantt que ilustrem a execução desses processos usando os algoritmos de escalonamento a seguir: FCFS, SJF, por prioridades sem preempção (um número de prioridade menor implica uma prioridade mais alta) e RR (quantum = 1).
 - Qual é o tempo de turnaround de cada processo para cada um dos algoritmos de escalonamento da parte a?
 - Qual é o tempo de espera de cada processo para cada um desses algoritmos de escalonamento?
 - Qual dos algoritmos resulta no menor tempo médio de espera (para todos os processos)?
- 10) Qual dos algoritmos de escalonamento a seguir poderia resultar em inanição?
- “Primeiro a entrar, primeiro a ser atendido”
 - Menor job primeiro
 - Round-Robin
 - Por prioridades
- 11) [*] Considere uma variante do algoritmo de escalonamento RR em que as entradas da fila de prontos são ponteiros para os PCBs.
- Qual seria o efeito da inserção de dois ponteiros para o mesmo processo na fila de prontos?
 - Cite duas grandes vantagens e duas desvantagens desse esquema.
 - Como você modificaria o algoritmo RR básico para obter o mesmo efeito sem os ponteiros duplicados?
- 12) [**] Considere um sistema executando dez tarefas limitadas por I/O e uma tarefa limitada por CPU. Suponhamos que as tarefas limitadas por I/O emitissem uma operação de I/O a cada milissegundo de

processamento da CPU e que cada operação de I/O levasse 10 milissegundos para ser concluída. Suponhamos também que o overhead da mudança de contexto fosse de 0,1 milissegundo e que todos os processos fossem tarefas de execução longa. Descreva a utilização da CPU para um escalonador round-robin quando:

- a. O quantum de tempo é de 1 milissegundo
- b. O quantum de tempo é de 10 milissegundos

13) [*] Explique as diferenças no grau de atuação dos algoritmos de escalonamento a seguir em favor de processos curtos:

- a. FCFS
- b. RR
- c. Filas com retroalimentação em vários níveis

14) Usando o algoritmo de escalonamento do Windows XP, determine a prioridade numérica de cada um dos threads a seguir.

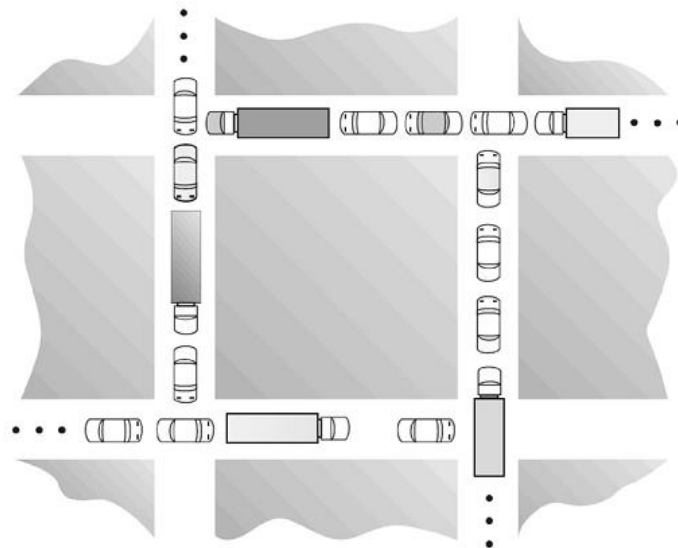
- a. Um thread da classe `REALTIME_PRIORITY_CLASS` com uma prioridade relativa `HIGHEST`
- b. Um thread da classe `NORMAL_PRIORITY_CLASS` com uma prioridade relativa `NORMAL`
- c. Um thread da classe `HIGH_PRIORITY_CLASS` com uma prioridade relativa `ABOVE_NORMAL`

Cap. 6: SINCRONISMO DE PROCESSOS

- 1) [R] Defina o conceito de Região Crítica (ou Seção Crítica) de um processo.
- 2) As condições de corrida podem ocorrer em muitos sistemas de computação. Considere um sistema bancário com duas funções: `deposit(amount)` e `withdraw(amount)`. Essas duas funções recebem a quantia (`amount`) que deve ser depositada em uma conta bancária ou dela retirada. Suponha que exista uma conta bancária conjunta de um marido e sua esposa e, concorrentemente, o marido chamasse a função `withdraw()` e a esposa chamasse `deposit()`. Descreva como uma condição de corrida pode ocorrer e o que pode ser feito para impedir que ela ocorra.
- 3) [R] Quais os requisitos principais de qualquer solução para o problema da seção crítica?
- 4) [R] O que é *busy wait*? Qual a consequência principal da solução que a possui?
- 5) [R] Explique como funciona o mecanismo de inibição/desativação de interrupções.
- 6) [*] Explique por que a implementação de primitivas de sincronização através da desativação de interrupções não é apropriada em um sistema com um único processador se os primitivas de sincronização tiverem de ser usados em programas de nível de usuário.
- 7) [R] Explique por que as interrupções não são apropriadas para a implementação de primitivas de sincronização em sistemas multiprocessadores.
- 8) [**] Os servidores podem ser projetados para limitar a quantidade de conexões abertas. Por exemplo, um servidor pode querer ter apenas N conexões de socket em um determinado momento. Assim que N conexões forem estabelecidas, ele não aceitará outra conexão até que uma conexão existente seja liberada. Explique como os semáforos podem ser usados por um servidor na limitação da quantidade de conexões concorrentes.
- 9) [*] Demonstre que, se as operações de semáforo `wait()` e `signal()` não forem executadas atômicaamente, a exclusão mútua pode ser violada.
- 10) Demonstre como podem ser implementadas as operações de semáforo `wait()` e `signal()` em ambientes multi-processadores com o uso da instrução `TestAndSet()`. A solução deve incorrer em um nível mínimo de espera em ação.
- 11) Discuta a relação entre compartilhamento justo e throughput nas operações do problema dos leitores-escretores. Proponha um método para resolver o problema dos leitores-escretores sem causar inanição.
- 12) [R] Cite e explique alguns problemas inerentes ao uso de semáforos na sincronização de processos.

Cap. 7: DEADLOCKS

- 1) [*] Liste três exemplos de deadlocks que não estejam relacionados a um ambiente de sistema de computação.
- 2) [**] Para pensar: Por que humanos não entram em deadlock?
- 3) [R] Quais os principais métodos gerais para tratamento de Deadlocks?
- 4) Suponha que um sistema esteja em um estado inseguro. Demonstre ser possível que os processos concluam sua execução sem entrar em estado de deadlock.
- 5) [**] Um sistema pode detectar que algum de seus processos está sofrendo de *starvation* (inanição)? Se você responder “sim”, explique como ele pode fazer isso. Se responder “não”, explique como o sistema pode lidar com o problema da inanição.
- 6) [*] É possível ocorrer um deadlock envolvendo apenas um processo (com uma única thread, naturalmente)? Explique sua resposta.
- 7) Considere o deadlock de trânsito mostrado na figura abaixo.
 - a. Demonstre que as quatro condições necessárias para a ocorrência do deadlock estão presentes nesse exemplo.
 - b. Defina uma regra simples para impedir deadlocks nesse sistema



- 8) [*] Considere a situação de deadlock que pode ocorrer no problema dos filósofos glutões quando os filósofos obtêm os pausinhos um de cada vez. Discuta como as quatro condições necessárias para a ocorrência do deadlock estão presentes nesta situação. Discuta como os deadlocks poderiam ser evitados com a eliminação de qualquer uma das quatro condições necessárias.
- 9) [***] Considere um sistema composto por quatro recursos do mesmo tipo que são compartilhados por três processos, cada um deles precisando de no máximo dois recursos. Demonstre que o sistema está livre de deadlocks.

- 10) [***] Considere a versão do problema dos filósofos glutões em que os pausinhos são colocados no centro da mesa e qualquer par deles pode ser usado por um filósofo. Assuma que as solicitações de pausinhos são feitas uma de cada vez. Descreva uma regra simples para determinar se uma solicitação específica pode ser atendida sem causar deadlock dada a presente alocação de pausinhos para os filósofos.
- 11) [**] Considere novamente a situação do exercício anterior. Assuma agora que cada filósofo precise de três pausinhos para comer. As solicitações de recursos ainda são emitidas uma de cada vez. Descreva algumas regras simples para determinar se uma solicitação específica pode ser atendida sem causar um deadlock dada esta alocação de chopsticks para os filósofos.

Cap. 8: MEMÓRIA PRINCIPAL

- 1) [R] Cite duas diferenças entre endereços lógicos e físicos.
- 2) [*] Considere um sistema em que um programa possa ser separado em duas partes: código e dados. A CPU sabe se deseja uma instrução (busca de instrução) ou dados (busca ou armazenamento de dados). Portanto, dois pares de registradores base-limite são fornecidos: um para instruções e outro para dados. O par de registradores base-limite de instruções é automaticamente somente de leitura; logo, os programas podem ser compartilhados entre diferentes usuários. Discuta as vantagens e desvantagens desse esquema.
- 3) [*] Por que os tamanhos de página são sempre potências de 2?
- 4) [*] Considere um espaço de endereçamento lógico de 64 páginas com 1.024 palavras cada, mapeado para uma memória física de 32 quadros.
 - a. Quantos bits há no endereço lógico?
 - b. Quantos bits há no endereço físico?
- 5) [*] Qual é o efeito de permitir que duas entradas em uma tabela de páginas apontem para o mesmo quadro de página na memória? Explique como esse efeito poderia ser usado para diminuir o período de tempo necessário para copiar uma grande quantidade de memória de um local para outro. Que efeito a atualização de algum byte em uma página teria na outra página?
- 6) [*] Descreva um mecanismo pelo qual um segmento poderia pertencer ao espaço de endereçamento de dois processos diferentes.
- 7) [R] Explique a diferença entre fragmentação interna e externa.
- 8) Considere o processo a seguir para a geração de binários. Um compilador é utilizado para gerar o código-objeto para módulos individuais, e um link editor é usado para combinar vários módulos-objeto em um único binário de programa. Como o link editor altera a vinculação de instruções e dados a endereços de memória? Que informações precisam ser passadas do compilador para o link editor para facilitar as tarefas de vinculação da memória do link editor?
- 9) [*] Dadas cinco partições de memória de 100 KB, 500 KB, 200 KB, 300 KB e 600 KB (em ordem), como os algoritmos *first-fit*, *best-fit* e *worst-fit* alocaariam processos de 212 KB, 417 KB, 112 KB e 426 KB (em ordem)? Que algoritmo faz o uso mais eficiente da memória?
- 10) [**] A maioria dos sistemas permite que um programa aloque mais memória a seu espaço de endereçamento durante a execução. A alocação de dados nos segmentos de programa do heap é um exemplo desse tipo de alocação de memória. O que é necessário para suportar a alocação de memória dinâmica nos seguintes esquemas?
 - a. Alocação de memória contígua
 - b. Segmentação pura
 - c. Paginação pura

- 11) [******] Compare os esquemas de alocação de memória contígua, de segmentação pura e de paginação pura para a organização da memória em relação às questões a seguir:
 - a. Fragmentação externa
 - b. Fragmentação interna
 - c. Possibilidade de compartilhar código entre processos
- 12) [*******] Em um sistema com paginação, um processo não pode acessar memória que ele não possui. Por quê? Como o sistema operacional poderia permitir o acesso a outra memória? Por que ele deveria ou não deveria fazer isso?
- 13) [******] Compare a paginação com a segmentação no que diz respeito à quantidade de memória requerida pelas estruturas de tradução de endereços para converter endereços virtuais em endereços físicos.
- 14) [*****] Os programas binários de muitos sistemas são, tipicamente, estruturados como descrito a seguir. O código é armazenado começando com um pequeno endereço virtual fixo, como 0. O segmento de código é seguido pelo segmento de dados que é usado para armazenar as variáveis do programa. Quando o programa começa a ser executado, a pilha é alocada na outra extremidade do espaço de endereços virtuais e pode crescer em direção a endereços virtuais menores. Qual é a importância dessa estrutura para os esquemas a seguir?
 - a. Alocação de memória contígua
 - b. Segmentação pura
 - c. Paginação pura
- 15) [*****] Supondo um tamanho de página de 1 KB, quais são os números de página e deslocamentos para as referências de endereço a seguir (fornecidas como números decimais):
 - a. 2375
 - b. 19366
 - c. 30000
 - d. 256
 - e. 16385
- 16) [******] Considere um sistema de computação com um endereço lógico de 32 bits e um tamanho de página de 4 KB. O sistema suporta até 512 MB de memória física. Quantas entradas existem em cada uma das tabelas de páginas a seguir?
 - a. Uma tabela de páginas convencional com um único nível
 - b. Uma tabela de páginas invertida
- 17) [*****] Considere um sistema de paginação com a tabela de páginas armazenada em memória.
 - a. Se uma referência à memória levar 200 nanossegundos, quanto tempo levará uma referência à memória paginada?
 - b. Se adicionarmos TLBs e 75 por cento de todas as referências à tabela de páginas estiverem nos TLBs, qual é o tempo efetivo de referência à memória? (Suponha que a busca de uma entrada na tabela de páginas nos TLBs leve um período de tempo igual a 0, se a entrada estiver aí.)
- 18) [******] Explique por que é mais fácil compartilhar um módulo reentrante quando é usada a segmentação em vez da paginação pura.
- 19) [*****] Considere a tabela de segmentos a seguir:

Segmento	Base	Tamanho
0	219	600
1	2300	14
2		
90	100	
3	1327	580
4	1952	96

Quais são os endereços físicos para os seguintes endereços lógicos?

- a. <0, 430>
- b. <1, 10>
- c. <2, 500>
- d. <3, 400>
- e. <4, 112>

20) [R] Qual é a finalidade da paginação das tabelas de páginas?

Cap. 9: MEMÓRIA VIRTUAL

- 1) [R] Sob que circunstâncias ocorrem erros de página? Descreva as ações executadas pelo sistema operacional quando ocorre um erro de página.
- 2) [**] Suponha que você tenha uma sequência de referências de página para um processo com m quadros (todos inicialmente vazios). A sequência de referências de página tem tamanho p ; n números de páginas distintas ocorrem nela. Responda a estas perguntas para qualquer algoritmo de substituição de páginas:
 - a. O que é um limite inferior para o número de erros de página?
 - b. O que é um limite superior para o número de erros de página?
- 3) [**] Quais das técnicas e estruturas de programação a seguir são “boas” para um ambiente paginado por demanda? Quais delas são “ruins”? Explique suas respostas.
 - a. Pilha
 - b. Tabela de símbolos com hash
 - c. Busca sequencial
 - d. Busca binária
 - e. Código puro
 - f. Operações em vetores
 - g. Acesso indireto
- 4) [R] Considere os algoritmos de substituição de páginas a seguir. Classifique estes algoritmos do melhor para o pior no que diz respeito a sua taxa de erros de página. Separe os algoritmos afetados pela anomalia de Belady daqueles que não o são.
 - a. Substituição LRU
 - b. Substituição FIFO
 - c. Substituição ótima
 - d. Substituição da segunda chance
- 5) [*] Quando a memória virtual é implementada em um sistema de computação, há certos custos associados à técnica e certos benefícios. Liste os custos e os benefícios. É possível que os custos superem os benefícios? Se for, que medidas podem ser tomadas para garantir que isso não ocorra?
- 6) [***] Considere o array bidimensional A em C/C++:

```
int A[100] [100]
```

onde $A[0][0]$ está na localização 200 em um sistema de memória paginada com páginas de tamanho 200. Um pequeno processo que manipula a matriz reside na página 0 (localizações 0 a 199). Assim, toda busca de instrução ocorrerá a partir da página 0.

Para três quadros de páginas, quantos erros de página serão gerados pelos seguintes loops de inicialização do array, utilizando a substituição LRU e supondo que o quadro de páginas 1 contenha o processo e os outros dois estejam inicialmente vazios?

```
a) for (int i = 0; i < 100; i++)
    for (int j = 0; j < 100; j++)
        A[i] [j] = 0;
```



```
b) for (int j = 0; j < 100; j++)
    for (int i = 0; i < 100; i++)
        A[i] [j] = 0;
```

Dica: Se não já não souber, pesquise a forma como é feita a alocação de matrizes bidimensionais em C. Ou seja, como é feito o mapeamento 2D → 1D (a memória é um como um array unidimensional). Você verá que é uma excelente lição de programação a ser aprendida!

7) [R] Considere a seguinte sequência de referências de página:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Quantos erros de página ocorreriam para os algoritmos de substituição a seguir, considerando um, dois, três, quatro, cinco, seis e sete quadros? Lembre-se de que todos os quadros estão inicialmente vazios, de modo que a primeira página de cada um implicará em um erro de página.

- Substituição LRU
- Substituição FIFO
- Substituição ótima

8) [*] Você imaginou um novo algoritmo de substituição de páginas que acha que pode ser ótimo. Em alguns casos de teste distorcidos, ocorre a anomalia de Belady. O novo algoritmo é ótimo? Explique sua resposta.

9) [*] A segmentação é semelhante à paginação, mas utiliza “páginas” de tamanho variável. Defina dois algoritmos de substituição de segmentos baseados nos esquemas de substituição de páginas FIFO e LRU. Lembre-se de que, como os segmentos não são do mesmo tamanho, o segmento escolhido para ser substituído pode não ser grande o bastante para deixar localizações consecutivas suficientes para o segmento requerido. Considere estratégias para sistemas em que os segmentos não possam ser relocados e estratégias para sistemas onde isso possa ocorrer.

10) [**] Considere um sistema de computação paginado por demanda em que o grau de multiprogramação esteja correntemente fixado em quatro. O sistema foi recentemente medido para determinar a utilização da CPU e o disco de paginação. Os resultados são uma das alternativas apresentadas a seguir. O que está acontecendo em cada caso? É possível aumentar o grau de multiprogramação para aumentar a utilização da CPU? A paginação está ajudando?

- a. Utilização da CPU, 13 por cento; utilização do disco, 97 por cento
- b. Utilização da CPU, 87 por cento; utilização do disco, 3 por cento
- c. Utilização da CPU, 13 por cento; utilização do disco, 3 por cento

11) [**] Suponha que um programa tivesse acabado de referenciar um endereço na memória virtual. Descreva um cenário em que cada uma das situações abaixo possa ocorrer. (Se tal cenário não puder ocorrer, explique por quê.)

- Erro de TLB sem erro de página
- Erro de TLB e erro de página
- Sucesso do TLB sem erro de página
- Sucesso do TLB e erro de página

12) [*] Considere um sistema que utilize paginação por demanda pura.

- a. Quando um processo inicia sua execução pela primeira vez, como você caracterizaria a taxa de erros de página?
 - b. Uma vez que o conjunto ativo para um processo seja carregado em memória, como você caracterizaria a taxa de erros de página?
 - c. Suponha que um processo altere sua localidade e o tamanho do novo conjunto ativo seja grande demais para ser armazenado na memória livre disponível. Identifique algumas opções que os projetistas de sistemas poderiam escolher para manipular essa situação.
- 13) [**] Um determinado computador fornece a seus usuários um espaço de memória virtual de 2^{32} bytes. O computador tem 2^{18} bytes de memória física. A memória virtual é implementada pela paginação e o tamanho da página é de 4.096 bytes. Um processo de usuário gera o endereço virtual 11.123.456. Explique como o sistema estabelece a localização física correspondente. Faça a distinção entre operações de software e operações de hardware.
- 14) Suponha que tenhamos uma memória paginada por demanda. A tabela de páginas é mantida em registradores. São necessários 8 milissegundos para manipular um erro de página se um quadro vazio estiver disponível ou se a página substituída não foi modificada e 20 milissegundos se a página substituída foi modificada. O tempo de acesso à memória é de 100 nanossegundos. Suponha que a página a ser substituída seja modificada 70 por cento das vezes. Qual é a taxa de erros de página máxima aceitável para um tempo de acesso efetivo de não mais do que 200 ns?
- 15) [**] Quando ocorre um erro de página, o processo que está solicitando a página deve ser bloqueado enquanto espera que a página seja transferida do disco para a memória física. Suponha que exista um processo com cinco threads de nível de usuário e que o mapeamento de threads de usuário para threads do kernel seja muitos-para-um. Se um thread de usuário incorrer em um erro de página ao acessar sua pilha, os outros threads de usuário pertencentes ao mesmo processo também seriam afetados pelo erro de página — isto é, eles também teriam que esperar a página em erro ser transferida para a memória? Explique.
- 16) [*] Suponha que você esteja monitorando a taxa segundo a qual o ponteiro do algoritmo do relógio (que indica a página candidata à substituição) se move. O que você pode dizer sobre o sistema se observar o seguinte comportamento:
- a. O ponteiro está se movendo rapidamente.
 - b. O ponteiro está se movendo lentamente.
- 17) Discuta situações (ou seja, crie exemplos hipotéticos) em que o algoritmo de substituição de páginas menos frequentemente utilizadas gere menos erros de página do que o algoritmo de substituição de páginas menos recentemente utilizadas. Discuta também sob que circunstâncias ocorre o oposto.
- 18) Discuta situações (ou seja, crie exemplos hipotéticos) em que o algoritmo de substituição de páginas mais frequentemente utilizadas gere menos erros de página do que o algoritmo de substituição de páginas menos recentemente utilizadas. Discuta também sob que circunstâncias ocorre o oposto.
- 19) O sistema VAX/VMS usa um algoritmo de substituição FIFO para páginas residentes e um pool de quadros livres de páginas recentemente utilizadas. Suponha que o pool de quadros livres seja gerenciado usando a política de substituição de páginas menos recentemente utilizadas. Responda às seguintes perguntas:
- a. Se ocorrer um erro de página e a página não existir no pool de quadros livres, como será gerado espaço livre para a página recém-solicitada?

- b. Se ocorrer um erro de página e a página existir no pool de quadros livres, como o conjunto de páginas residentes e o pool de quadros livres serão gerenciados para gerar espaço para a página solicitada?
- c. Para que o sistema degenera se o número de páginas residentes estiver posicionado em um?
- d. Para que o sistema degenera se o número de páginas do pool de quadros livres for zero?

20) [***] Considere um sistema de paginação por demanda com as seguintes medidas de tempo de utilização:

Utilização da CPU = 20%

Disco de paginação = 97,7%

Outros dispositivos de I/O = 5%

Para cada uma das situações a seguir, diga se ela irá (ou poderá vir a) melhorar a utilização da CPU.

Explique suas respostas.

- a. Instalação de uma CPU mais rápida.
- b. Instalação de um disco de paginação maior.
- c. Aumento do grau de multiprogramação.
- d. Diminuição do grau de multiprogramação.
- e. Instalação de mais memória principal.
- f. Instalação de um disco rígido mais rápido ou de múltiplos controladores com múltiplos discos rígidos.
- g. Inclusão da pré-paginação nos algoritmos de busca de páginas.
- h. Aumento do tamanho da página.

21) [**] Suponha que uma máquina forneça instruções que podem acessar localizações de memória utilizando o esquema de endereçamento indireto de um nível. Que sequência de erros de página ocorrerá se todas as páginas de um programa estiverem não-residentes e a primeira instrução do programa for uma operação de carga de memória indireta? O que acontecerá se o sistema operacional estiver usando uma técnica de alocação de quadros por processo e somente duas páginas forem alocadas a este processo?

22) [*] Suponha que sua política de substituição (em um sistema paginado) seja examinar cada página regularmente e descartar a página se ela não tiver sido usada desde o último exame. O que você ganharia e o que perderia usando esta política em vez da substituição LRU ou da segunda chance?

23) Considere um sistema de paginação por demanda com um disco de paginação que tem um tempo médio de acesso e transferência de 20 milissegundos. Os endereços são traduzidos por intermédio de uma tabela de páginas na memória principal, com tempo de acesso de 1 microssegundo por acesso à memória. Assim, cada referência à memória, por intermédio da tabela de páginas, exige dois acessos. Para melhorar este tempo, adicionamos uma memória associativa que reduz o tempo de acesso para uma referência à memória se a entrada da tabela de páginas está na memória associativa.

Suponha que 80 por cento dos acessos ocorram à memória associativa e que, do restante, 10 por cento (ou 2 por cento do total) provoque erros de página. Qual é o tempo efetivo de acesso à memória?

24) [R] Qual é a causa da atividade improdutiva (*thrashing*)? Como o sistema detecta a atividade improdutiva? Uma vez que ela seja detectada, o que o sistema pode fazer para eliminar este problema?

25) [*] É possível que um processo tenha dois conjuntos ativos, um representando dados e outro representando código? Explique.

Cap. 10: INTERFACE DO SISTEMA DE ARQUIVOS

- 1) [******] Alguns sistemas suportam muitos tipos de estruturas para os dados de um arquivo enquanto outros suportam simplesmente uma cadeia de bytes. Quais são as vantagens e desvantagens de cada abordagem?
- 2) [**R**] Explique o objetivo das operações `open()` e `close()`.
- 3) [*****] Dê um exemplo de uma aplicação em que os dados de um arquivo devam ser acessados na seguinte ordem:
 - a. Sequencialmente
 - b. Aleatoriamente
- 4) Considere um sistema de arquivos em que um arquivo possa ser removido e seu espaço em disco requisitado enquanto ainda existirem links para o arquivo. Que problemas podem ocorrer se um novo arquivo for criado na mesma área de armazenamento ou com o mesmo nome de caminho absoluto? Como esses problemas podem ser evitados?
- 5) [******] A tabela de arquivos abertos é utilizada para manter informações sobre os arquivos que estão correntemente abertos. O sistema operacional deveria manter uma tabela separada para cada usuário ou apenas uma tabela que contenha referências aos arquivos que estão sendo correntemente acessados por todos os usuários? Se o mesmo arquivo estiver sendo acessado por dois programas ou usuários diferentes, deveriam existir entradas separadas na tabela de arquivos abertos?
- 6) [*******] Quais são as vantagens e desvantagens de fornecer locks obrigatórios em vez de locks consultivos cujo uso é deixado a critério dos usuários?
- 7) [******] Quais são as vantagens e desvantagens de registrar o nome do programa criador com os atributos do arquivo (como é feito no sistema operacional Macintosh)?
- 8) [******] Se o sistema operacional soubesse que uma determinada aplicação iria acessar os dados de um arquivo de modo sequencial, como ele poderia usar esta informação para melhorar o desempenho?
- 9) [******] Dê um exemplo de uma aplicação que pudesse se beneficiar do suporte do sistema operacional ao acesso aleatório a arquivos indexados.

Cap. 11: IMPLEMENTAÇÃO DO SISTEMA DE ARQUIVOS

- 1) [*] Por que o mapa de bits para alocação de arquivos deve ser mantido em memória de massa, e não na memória principal?
- 2) [***] Considere um sistema que suporte as estratégias de alocação contígua, encadeada e indexada. Que critérios devem ser levados em conta na decisão de qual estratégia é a melhor para um arquivo em particular?
- 3) [**] Um problema com a alocação contígua é que o usuário deve pré-alocar espaço suficiente para cada arquivo. Se o arquivo crescer a ponto de ficar maior do que o espaço a ele alocado, ações especiais devem ser levadas a efeito. Uma solução para este problema é definir uma estrutura de arquivo consistindo de uma área contígua inicial (de um tamanho especificado). Se essa área for preenchida, o sistema operacional definirá automaticamente uma área excedente que será encadeada na área contígua inicial. Se a área excedente for preenchida, outra área excedente será alocada. Compare esta implementação de um arquivo com as implementações padrão contígua e encadeada.
- 4) [*] Como os caches ajudam a melhorar o desempenho? Por que os sistemas não utilizam mais caches ou caches maiores se eles são tão úteis?
- 5) [***] Explique como a camada VFS permite que um sistema operacional suporte facilmente múltiplos tipos de sistemas de arquivos.
- 6) [*] Considere um sistema de arquivos que utilize um esquema de alocação contígua modificado com suporte a extensões. Um arquivo é uma coleção de extensões, cada extensão correspondendo a um conjunto contíguo de blocos. Uma questão-chave em tais sistemas é o grau de variabilidade no tamanho das extensões. Quais são as vantagens e desvantagens dos seguintes esquemas?
 - a. Todas as extensões são do mesmo tamanho e o tamanho é predeterminado.
 - b. As extensões podem ser de qualquer tamanho e são alocadas dinamicamente.
 - c. As extensões podem ser de poucos tamanhos fixos e esses tamanhos são predeterminados.
- 7) [**] Quais são as vantagens da variante da alocação encadeada que utiliza uma FAT para encadear os blocos de um arquivo?
- 8) [**] Alguns sistemas de arquivos permitem que a memória em disco seja alocada em níveis de granularidade diferentes. Por exemplo, um sistema de arquivos poderia alocar 4 KB de espaço em disco como um único bloco de 4 KB ou como oito blocos de 512 bytes. Como poderíamos tirar vantagem dessa flexibilidade para melhorar o desempenho? Que modificações teriam de ser feitas no esquema de gerenciamento do espaço livre para suportar essa característica?
- 9) [*] Discuta como otimizações de desempenho para sistemas de arquivos podem resultar em dificuldades para manter a consistência dos sistemas em caso de quedas do computador.
- 10) [**] Explique por que o registro em log de atualizações de metadados garante a recuperação de um sistema de arquivos após esse sistema de arquivos cair.

HISTÓRICO DE VERSÕES

[2013-03-22] - v1.0.0

- Adição de exercícios dos Capítulos 10 e 11

[2013-02-14] - v0.9.0

- Adição de exercícios dos Capítulos 8 e 9
- Adicionadas classificações aos exercícios, conforme nota após o Sumário.

[2013-01-22] - v0.7.0

- Adição de exercícios dos Capítulos 6 e 7

[2012-12-08] - v0.5.0

- Versão Inicial. Capítulos 1 a 5