

Sistemas Operacionais – Trabalho Prático 1 (19/09/14)

Memória Virtual com Paginação

1 Objetivos

Nas últimas aulas vimos que, através do conceito de memória virtual, é possível fornecer aos processos um espaço de endereçamento virtual e dar a cada um deles a ilusão de que eles detêm toda a memória, quando na verdade, através do mecanismo de paginação, trazemos para a memória apenas pedaços de cada processo. O objetivo deste trabalho prático é amadurecer o conceito de memória virtual com paginação.

2 Entrada de Dados

Seu programa deve receber as seguintes informações do teclado:

1. Esquema de paginação:
 - Se 0, o esquema de paginação desejado é paginação de dois níveis;
 - Se 1, o esquema de paginação escolhido é paginação invertida.
2. Quantidade de bits do espaço de endereçamento virtual;
3. Quantidade de bits do espaço de endereçamento físico;
4. Tamanho da página (em Kb);
5. Algoritmo de substituição de páginas:
 - Se 1, utilizar algoritmo FIFO;
 - Se 2, utilizar algoritmo LRU (implementado utilizando uma pilha).
6. Quantidade de processos em execução. Por exemplo, se o valor recebido nesse campo for 3, existirão processos com id 0, 1 e 2;
7. Quantidade de quadros disponíveis.

Você pode exibir mensagens amigáveis para o usuário, que indiquem a próxima informação que deve ser digitada. Entretanto, as informações solicitadas devem ser pedidas uma após a outra, e **exatamente** na ordem acima.

3 Funcionamento Básico

Após receber as entradas de dados descritas na seção anterior, seu programa deve exibir um menu com as seguintes opções:

- (1) Ler um endereço lógico de um processo;
 - Se o usuário escolher essa opção, você deve requisitar o id do processo, o endereço lógico desejado e ler esse acesso utilizando paginação. Logo após, você deve imprimir o conteúdo desse endereço lógico/físico.
- (2) Escrever um valor em um endereço lógico de um processo;
 - Se o usuário escolher essa opção, você deve requisitar o id do processo, o endereço lógico a ser escrito, o valor a ser escrito nesse endereço e só então realizar a escrita desse valor no endereço requisitado. Não há necessidade de realizar nenhuma impressão após a realização dessa operação.

- (3) Remover da memória uma página de um processo;
 - Se o usuário escolher essa opção, você deve requisitar o id do processo, o número da página a ser removida e executar a operação de remoção.
- (4) Imprimir a(s) tabela(s) de página;
 - Note que, se o mecanismo de paginação utilizado for paginação multinível, você deve imprimir as tabelas de página de todos os processos, e se o mecanismo de paginação utilizado for paginação invertida, você deve imprimir apenas uma tabela.
- (5) Imprimir o conteúdo do espaço de endereçamento virtual de cada processo, lendo diretamente do disco;
- (6) Imprimir o conteúdo da memória física (byte por byte);
- (7) Imprimir a quantidade de acessos realizados até o momento;
- (8) Imprimir a quantidade de falhas de página existentes até o momento;
- (9) Sair do programa.

4 Estruturas de Dados

Você tem total liberdade para modelar as estruturas de dados do seu programa, desde que nele existam pelo menos as seguintes estruturas com as seguintes características:

- *Tabela(s) de página*, onde cada linha da tabela de página deve ter pelo menos o campo *número do quadro*, *bit de validade* e *bit de modificação*. Fique a vontade para adicionar outros campos que você achar conveniente;
- *Lista de quadros livres*, que informa, para cada quadro da memória, se ele está livre ou ocupado;
- Vetor *memória*, que é um vetor de bytes que armazenará de fato as páginas de cada processo;
- Vetor ou matriz *disco*, que deve simular o conteúdo do disco de cada processo. Para simplificar a programação, assuma que todo o espaço de endereçamento virtual de cada processo está preenchido. O conteúdo do espaço de endereçamento virtual pode ser gerado de maneira aleatória.

5 Critérios de correção

- **Compilação:** até 1 ponto

Neste critério, o resultado da compilação do seu programa será avaliado. Para que você obtenha o máximo de pontos deste critério, mensagens de alerta (*warnings*) e erros não podem ocorrer. O valor de 0,5 ponto é descontado do valor máximo a cada mensagem de alerta que ocorrer, com limite de 2 mensagens.

Caso haja uma ou mais mensagens de erro no seu programa, você obtém o valor mínimo neste critério, isto é, 0 (zero). O mesmo ocorre caso seu programa não faça nada.

- **Execução:** até 7 pontos

Caso haja um ou mais erros de compilação no seu programa, você obtém o valor mínimo neste critério, ou seja, 0 (zero). Caso contrário, diversas entradas serão fornecidas ao seu programa e as saídas resultantes serão avaliadas.

- **Código:** até 2 pontos

Caso haja um ou mais erros de compilação no seu programa, você obtém o valor mínimo neste critério, ou seja, 0 (zero). Caso contrário, a organização do seu código será avaliada, tal como indentação, declaração de variáveis, estruturas de dados, estruturas de programação, funções, documentação, etc.

6 Entrega

Instruções para entrega do seu trabalho:

1. Cabeçalho

Seu trabalho deve ter um cabeçalho com o seguinte formato:

```
/*  
 *  
 *  Nomes:  
 *  Trabalho 1 Sistemas Operacionais  
 *  Professor(a): Samuel Ferraz  
 *  
 */
```

2. Compilador

O professor utiliza o `gcc` em um ambiente Linux para corrigir os programas. Se você utilizar algum outro compilador para desenvolver seu programa, antes de entregar verifique se o seu programa compila sem *warnings* e executa corretamente com o `gcc` em um ambiente Linux.

3. Grupos e Forma de entrega

O trabalho deve ser feito em grupos de no máximo 2 pessoas e apenas um envio deve ser feito por grupo, na conta de um dos integrantes. A entrega será realizada diretamente no moodle da FACOM (<http://ead.facom.ufms.br>). Um fórum de discussão deste trabalho no moodle já se encontra aberto. Para entrega do trabalho, você deve estar cadastrado no moodle na disciplina disciplina de Sistemas Operacionais T02. Após abrir uma sessão digitando seu *login* e sua senha, procure um tópico chamado “Entrega do trabalho prático 1”. Você pode entregar o trabalho quantas vezes quiser até às **23 horas e 55 minutos** do dia **17 de outubro de 2014**. A última versão entregue é aquela que será corrigida. Encerrado o prazo, não serão mais aceitos trabalhos.

4. Atrasos

Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

5. Erros

Trabalhos com erros de compilação receberão nota ZERO. Faça todos os testes necessários para garantir que seu programa está livre de erros de compilação.

6. O que entregar?

Você deve entregar um arquivo zip contendo o código do seu programa e um arquivo Makefile capaz de compilá-lo. O arquivo que contem a sua função *main* deve ter como nome o seu *login* e a extensão `c/cpp`. Por exemplo, `joao_silva.c`. Se o seu grupo tiver duas pessoas, o nome desse arquivo deve ser formado pelos *logins* no moodle em ordem lexicográfica. Por exemplo se uma dupla é formada pelo João Silva e pelo Zé Augusto, o nome do arquivo da dupla deve ser `joao_silva_ze_augusto.c`.

7. Verificação dos dados de entrada

Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10 e o usuário digita um número negativo, uma letra, um cifrão, etc, o seu programa pode fazer qualquer coisa, como travar o computador ou encerrar a sua execução abruptamente com respostas erradas.

8. Arquivo fonte

Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso.

9. Conduta ética

O trabalho deve ser feito em DUPLAS. Cada dupla tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho mais de duas pessoas e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir ideias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas NÃO copie o programa!

Trabalhos considerados plagiados terão nota ZERO. Estudante que se envolver em DOIS CASOS DE PLÁGIO estará automaticamente REPROVADO na disciplina.