Faculdade de Tecnologia, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Gleba A, Avenida L3 Norte, CEP 70.910-900, Brasília-DF
Caixa Postal 4386, fone +55 61 3307 2300, fone/fax +55 61 3273 8893, ftd@unb.br, www.ft.unb.br

Trabalho 2 – Sistemas Operacionais de Redes

Instruções

O trabalho poderá ser feito em grupos de até 4 integrantes. A entrega será pelo campusvirtual.unb.br até o dia 19 de novembro, com um arquivo zipado contendo:

1. Código fonte C/C++ (baseado no código fonte disponibilizado para a turma no campusvirtual): O ambiente de compilação e execução deverá ser UNIX. Trabalhos feitos para Windows não serão aceitos. Além disso, é preciso que seja criado um Makefile para compilação automática do projeto. Trabalhos que apresentem qualquer erro de compilação ou execução serão automaticamente descartados da avaliação e receberão nota ZERO. Trabalhos copiados (ou plagiados) também receberão nota ZERO.
2. PPT com a apresentação da solução: Os slides deverão detalhar claramente a solução, incluindo a conclusão do grupo a respeito dos resultados obtidos. Além disso, os slides devem apresentar e detalhar cada uma das chamadas de sistema utilizadas no desenvolvimento das soluções.

UM INTEGRANTE DO GRUPO SERÁ SORTEADO PARA APRESENTAR O TRABALHO AO PROFESSOR E DEVERÁ SER CAPAZ DE RESPONDER A QUALQUER PERGUNTA EM NOME DO GRUPO, OU SEJA, RESPONSABILIZANDO-SE PELA NOTA DA EQUIPE.

Motivação & Objetivos

O gerenciador de memória virtual constitui uma parte importante do projeto e funcionamento de sistemas operacionais. É graças ao gerenciador de memória virtual que os processos podem endereçar conjuntos de posições maiores que o disponível em memória física. Sem ele, programas maiores que a quantidade disponíveis de memória física não poderiam existir.

Simulador: Gerenciador de Memória Virtual

Este projeto consiste em escrever um programa que traduz endereços lógicos para endereços físicos, considerando um espaço de endereço virtual de tamanho $2^{16} = 65.536$ bytes. Seu programa lerá de um arquivo os endereços lógicos e, usando uma TLB assim como uma tabela de páginas, irá traduzir cada endereço lógico para o seu correspondente endereço físico, além de armazenar o valor do byte correspondente no endereço físico traduzido. O objetivo por trás deste projeto é simular as etapas envolvidas na tradução de endereços lógicos para físicos.

Detalhes do Trabalho

Seu programa lerá um arquivo contendo vários números inteiros de 32 bits que representam endereços lógicos. No entanto, você só precisa se preocupar com 16 bits de endereço, por isso você deve mascarar os 16 bits mais à direita de cada endereço lógico. Esses 16 bits são divididos em (i) um número de página de 8 bits e (ii) deslocamento de página de 8 bits. Assim, os endereços são estruturados conforme mostrado na Fig.1 abaixo.

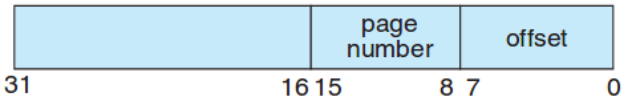

Fig1. Estrutura de endereçamento.

Faculdade de Tecnologia, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Gleba A, Avenida L3 Norte, CEP 70.910-900, Brasília-DF
Caixa Postal 4386, fone +55 61 3307 2300, fone/fax +55 61 3273 8893, ftd@unb.br, www.ft.unb.br

Outros detalhes de implementação incluem:

- 2^8 entradas na tabela de páginas
- Tamanho da página de 2^8 bytes
- 16 entradas na TLB
- Tamanho do quadro de 2^8 bytes
- 256 quadros
- Memória física de 65536 bytes (256 quadros x 256-bytes de tamanho de quadro)

Além disso, o seu programa só precisa se preocupar com endereços lógicos sendo traduzidos para seus endereços físicos correspondentes. Vocênão precisa suportar a gravação no espaço de endereço lógico. O esquema de tradução de páginas é apresentado na Fig.2, conforme visto em sala de aula.

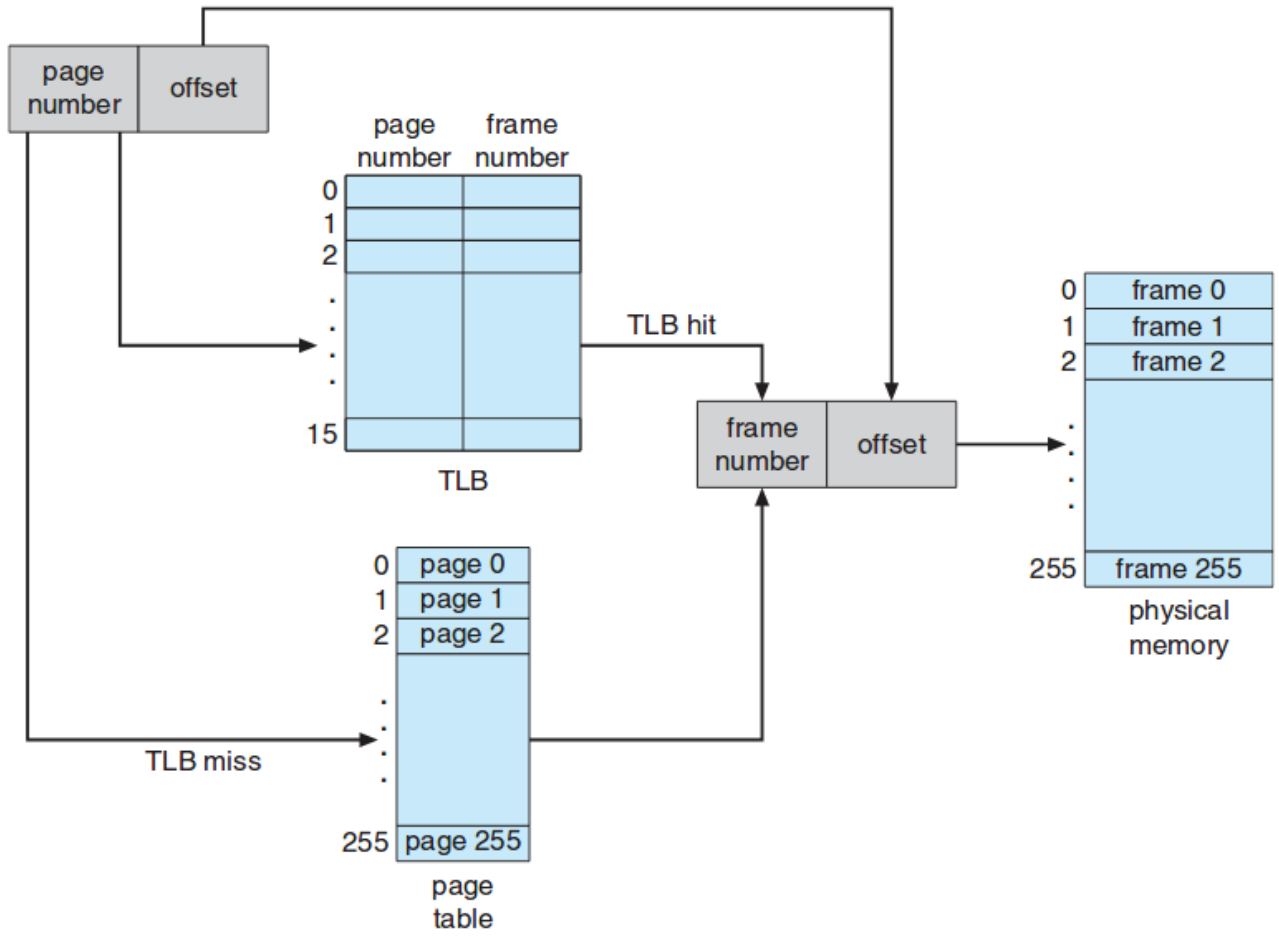

Fig.2: Processo de tradução de endereço lógico-físico.

Seu programa implementará a paginação por demanda conforme descrito em sala de aula. O dispositivo de armazenamento será representado por um arquivo BACKSTORE.bin, que é um arquivo binário de tamanho 65.536 bytes. Quando ocorre uma falha de página, você lerá em uma página de 256 bytes do arquivo BACKSTORE.bin e a armazenará em um quadro de página disponível na memória física. Por exemplo, se um endereço lógico com o número de página 15 resultou em uma falha de página, seu programa iria ler do arquivo BACKSTORE.bin a respectiva página e a gravaria em um quadro da memória física. Lembre-se que as páginas começam em 0 e têm 256 bytes de tamanho). Uma vez que este quadro é armazenado (e a tabela

Faculdade de Tecnologia, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Gleba A, Avenida L3 Norte, CEP 70.910-900, Brasília-DF
Caixa Postal 4386, fone +55 61 3307 2300, fone/fax +55 61 3273 8893, ftd@unb.br, www.ft.unb.br

de páginas e o TLB estão atualizados), os acessos subsequentes à página 15 serão resolvidos pelo TLB ou pela tabela de páginas. Você precisará tratar o arquivo BACKSTORE.bin como um arquivo de acesso aleatório para que você pode procurar aleatoriamente determinadas posições do arquivo para leitura. Sugiro que usem as funções padrão da biblioteca C para executar E/S, incluindo: fopen(), fread(), fseek() e fclose().

Um arquivo de teste (enderecos.txt) que será disponibilizado no campusvirtual deverá ser usado. Ele contém valores inteiros que representam endereços lógicos variando de 0-65535 (o tamanho do espaço de endereço virtual). Seu programa irá abrir este arquivo, ler cada endereço lógico e traduzir ao seu endereço físico correspondente, mostrando o valor do byte designado no endereço físico.

O seu programa deverá fazer as seguintes estatísticas:

1. Taxa de falta de páginas
2. Taxa de acerto da TLB

Considere que o tamanho da memória física é do mesmo tamanho do espaço de endereçamento - 65.536 bytes - para que você não precise se preocupar com substituições de página durante uma falta de página.

Pontuação (Nota)

O trabalho vale 10 e será pontuado em 4 partes:

1. 25% para o grupo que apresentar um programa capaz de ler cada endereço virtual do arquivo de endereços e imprimir na tela seu respectivo número de página e deslocamento.
2. Mais 25% para o grupo que apresentar uma versão já funcional do simulador sem TLB, ou seja, apenas usando tabela de página.
3. Mais 25% para o grupo que apresentar uma versão funcional do simulador com TLB e tabela de página.
4. Mais 25% para o grupo que apresentar uma versão funcional do simulador com TLB e tabela de páginas, apresentando as estatísticas de Taxa de falta de páginas e Taxa de acerto da TLB.

Note que outros fatores também serão avaliados no programa, como: legibilidade, plágio, etc... Portanto, apenas implementar os itens descritos acima não garante 100% de nota.