

Alguns detalhes sobre o projeto de Gerenciamento de Memória

Este projeto consiste em escrever um programa que traduza endereços lógicos para endereços físicos de um espaço de endereçamento virtual de $2^{16}=65.536$ bytes. Seu programa lerá de um arquivo contendo endereços lógicos e, usando TLB expandida para conter numero de segmento e número de páginas, traduzirá cada endereço lógico para seu correspondente endereço físico, dando saída no valor do byte armazenado no endereço físico resultante. O objetivo deste projeto é a simulação dos passos envolvidos na tradução de endereços lógicos para físicos, desta vez considerando segmentação.

A tradução de endereços ocorre como descrito na figura 9.34.

Manipulando da falta de página

Seu programa implementará a paginação por demanda (só traz a página para a memória quando ela é necessária).

O arquivo **BACKING_STORE.bin** representa o arquivo binário (o código do processo) de 65.536 bytes. Quando ocorrer uma falta de página, você lerá uma página de 256 bytes a partir do arquivo BACKING_STORE.bin e a armazenará em uma moldura disponível na memória física. Por exemplo, se um endereço lógico com o número de página 15 resultar em uma falta de página, seu programa busca a página 15 em BACKING_STORE.bin e armazenará em uma moldura (quadro) na memória física. Uma vez que essa página seja armazenada, acessos subsequentes à página 15 serão resolvidos pela TLB ou pela tabela de páginas.

Você tratará BACKING_STORE.bin como um arquivo de acesso aleatório para que possa pesquisar aleatoriamente certas posições do arquivo para leitura. (Use as funções de C fopen(), fread(), fseek() e fclose()).

Neste primeiro momento, seu projeto apresenta memória física e virtual apresentam o mesmo tamanho, ou seja, não será necessário implementar algoritmos de substituição de página.

Na segunda versão (**que deve ser entregue**) será necessário implementar um algoritmos de substituição de página (como descrito no arquivo tarefa1.pdf)

Turma, uma correção aqui!!!

Foram disponibilizados dois arquivos (BACKING_STORE.bin e BACKING_STORE_sequencial.bin) em formato texto, representando uma instrução por linha com “\n” ao final de cada instrução, esses arquivos apresentam tamanho superior a 65.536 bytes (portanto estão fora da especificação do trabalho, no entanto sua utilização não compromete o objetivo do trabalho). Esses arquivos **podem** ser usados em sua implementação.

Para atender ao enuncialdo, estou disponibilizando um gerador de arquivo binário (geradorBS_C.c), que gera uma sequencia de bytes **sem sinal** (0 a 255), **sem uso de “\n”**, portanto com tamanho exato de 65536 bytes. O arquivo .c apresenta exemplo de leitura byte a byte do arquivo (atenção ao nome do arquivo gerado).

Podem ser utilizado qualquer um dos tipos de arquivo de entrada (texto ou binário), solicito apenas que a dupla deixe claro qual tipo de arquivo de entrada será utilizado.

Arquivo de Texto

O arquivo addresses.txt contém valores inteiros representando endereços lógicos que variam de 0 a 65.535 (o tamanho do espaço de endereçamento virtual de um segmento). Seu programa abrirá

esse arquivo, realiza a leitura de cada endereço lógico, o traduzirá para o físico e exibe o conteúdo armazenado nesse endereço físico (o conteúdo previamente carregado do arquivo BACKING_STORE.bin).

Como Executar Seu Programa

Seu programa deve ser executado como descrito a seguir:

```
./a.out addresses.txt
```

O programa deve dar saída nos seguintes valores:

1. O endereço lógico que está sendo traduzido (obtido de address.txt);
2. O endereço físico correspondente (aquele para o qual seu programa traduziu o endereço lógico, preceda também o número do segmento);
3. O conteúdo armazenado no endereço físico correspondente (carregado de BACKING_STORE.bin).

Exemplo do arquivo que será gerado:

Endereço Virtual: 16916	Endereço Físico: 20	Conteúdo: 0
Endereço Virtual 62493	Endereço Físico: 29	Conteúdo: 0
Endereço Virtual: 30198	Endereço Físico: 246	Conteúdo: 29
Endereço Virtual: 53683	Endereço Físico: 179	Conteúdo: 108

Implemente as estatísticas e a modificação indicada no pdf.