



Projeto 02

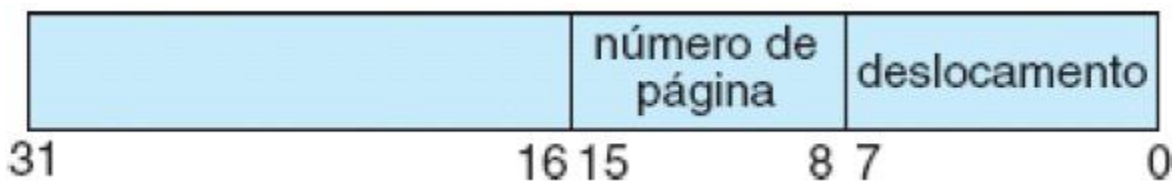
Prof. Mateus Mendelson
mendelson@unb.br

Projetando um Gerenciador de Memória Virtual

Esse 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 tanto um TLB quanto uma tabela de páginas, traduzirá cada endereço lógico para seu correspondente endereço físico, dando saída no valor do byte armazenado no endereço físico resultante. O objetivo por trás desse projeto é a simulação dos passos envolvidos na tradução de endereços lógicos para endereços físicos.

Especificidades

Seu programa lerá um arquivo contendo vários números inteiros sem sinal de 32 bits que representam endereços lógicos. No entanto, você precisa se preocupar apenas com endereços de 16 bits; portanto, você deve mascarar os 16 bits da extrema direita de cada endereço lógico. Esses 16 bits são divididos em (1) um número de página de 8 bits e (2) um deslocamento de página de 8 bits. Assim, os endereços estão estruturados como mostrado na figura abaixo.



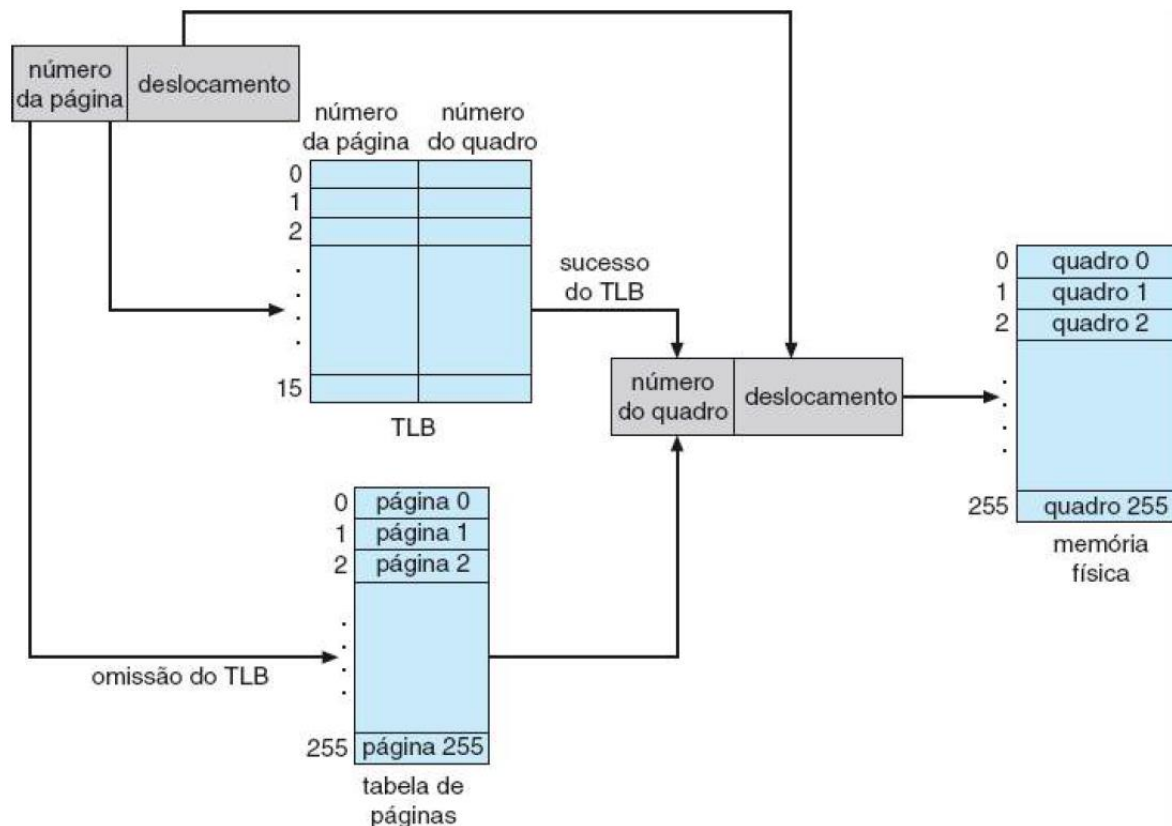
Outras especificidades incluem:

- 2^8 entradas na tabela de páginas;
- Tamanho de página de 2^8 bytes;
- 16 entradas no TLB;
- Tamanho de quadro de 2^8 bytes;
- 256 quadros; e
- Memória física de 65.536 bytes ($256 \text{ quadros} \times \text{tamanho de quadro de } 256 \text{ bytes}$).

Além disso, seu programa precisa se preocupar apenas com a leitura de endereços lógicos e sua tradução para seus endereços físicos correspondentes. Você não precisa dar suporte à gravação no espaço de endereçamento lógico.

Tradução de Endereços

Seu programa traduzirá endereços lógicos para físicos usando um TLB e uma tabela de páginas como descrito na Seção 8.5 do livro em português. Primeiro, o número da página é extraído do endereço lógico e o TLB é consultado. Em caso de sucesso do TLB, o número do quadro é obtido a partir do TLB. Em caso de omissão do TLB, a tabela de páginas deve ser consultada. No último caso, o número do quadro é obtido na tabela de páginas ou ocorre um erro de página. Uma representação visual do processo de tradução de endereços aparece na figura abaixo.



Manipulando Erros de Página

Seu programa implementará a paginação por demanda como descrito na Seção 9.2 do livro em português. A memória de retaguarda é representada pelo arquivo `BACKING_STORE.bin`, um arquivo binário de 65.536 bytes. Quando ocorrer um erro de página, você lerá uma página de 256 bytes a partir do arquivo `BACKING_STORE` e a armazenará em um quadro de página disponível na memória física. Por exemplo, se um

endereço lógico com o número de página 15 resultar em um erro de página, seu programa lerá a página 15 em `BACKING_STORE` (lembre-se de que as páginas começam em 0 e têm 256 bytes de tamanho) e a armazenará em um quadro de página na memória física. Uma vez que esse quadro seja armazenado (e a tabela de páginas e o TLB forem atualizados), acessos subsequentes à página 15 serão resolvidos pelo TLB ou pela tabela de páginas.

Você deverá tratar `BACKING_STORE.bin` como um arquivo de acesso aleatório para que possa pesquisar aleatoriamente certas posições do arquivo para leitura. É sugerido o uso das funções-padrão da biblioteca de C para execução de I/O, incluindo `fopen ()`, `fread ()`, `fseek ()` e `fclose ()`.

O tamanho da memória física é igual ao tamanho do espaço de endereçamento virtual: 65.536 bytes; logo, você não precisa se preocupar com substituições de página durante um erro de página.

Arquivo de Texto

É fornecido um arquivo `addresses.txt`, que contém valores inteiros representando endereços lógicos que variam de 0 a 65535 (o tamanho do espaço de endereçamento virtual). Seu programa abrirá esse arquivo, lerá cada endereço lógico e o traduzirá para seu endereço físico correspondente, e dará saída no valor do byte sinalizado contido no endereço físico.

Como Começar

Primeiro, escreva um programa simples que extraia o número e o deslocamento da página a partir dos números inteiros a seguir:

```
1 256 32768 32769 128 65534 33153
```

Talvez a maneira mais fácil de fazer isso seja usando os operadores de mascaramento de bits e de deslocamento de bits. Uma vez que você possa estabelecer corretamente o número e o deslocamento da página a partir do número inteiro, estará pronto para começar.

Inicialmente, é sugerido que você ignore o TLB e use apenas a tabela de páginas. Você pode integrar o TLB, uma vez que sua tabela de páginas esteja funcionando apropriadamente. Lembre-se de que a tradução de endereços pode funcionar sem um TLB; o TLB apenas a torna mais rápida. Quando você estiver pronto para implementar o TLB, não se esqueça de que ele tem somente 16 entradas; assim, você terá que usar uma estratégia de substituição quando atualizar um TLB cheio. Você pode usar uma política FIFO ou LRU para atualizar seu TLB.

Como Executar Seu Programa

Seu programa deve ser executado como descrito a seguir:

```
$ ./proj02.out addresses.txt
```

Seu programa lerá o arquivo `addresses.txt`, que contém 10.000 endereços lógicos variando de 0 a 65535. Seu programa deve traduzir cada endereço lógico para um endereço físico e determinar o conteúdo do byte sinalizado armazenado no endereço físico correto.

O programa deve dar saída nos seguintes valores:

1. O endereço lógico que está sendo traduzido (o valor inteiro que está sendo lido em `addresses.txt`);
2. O valor do quadro traduzido a partir do endereço virtual;
3. O valor do offset traduzido a partir do endereço virtual;
4. O endereço físico correspondente (aquele para o qual seu programa traduziu o endereço lógico); e
5. O valor do byte sinalizado armazenado no endereço físico resultante.

Estatísticas

Após concluído, seu programa deve relatar as seguintes estatísticas:

1. Taxa de erros de página – o percentual de referências de endereços que resultaram em erros de página; e
2. Taxa de sucesso do TLB – o percentual de referências de endereços que foram resolvidas no TLB.

Já que os endereços lógicos em `addresses.txt` foram gerados aleatoriamente e não refletem quaisquer localidades de acesso à memória, não espere obter uma alta taxa de sucesso do TLB.