

UNIVALI, Universidade do Vale do Itajaí

Memória

Halry Henry Fernandes do Nascimento

SC - Itajaí/2023

Enunciado do projeto

Suponha que um sistema tenha um endereço virtual de tamanho entre 16 bits à 32 bits com deslocamento na página de 256 b à 4 Kb. Escreva um programa que receba um endereço virtual (em decimal) na linha de comando ou leitura do arquivo `addresses.txt` faça com que ele produza o número da página e o deslocamento do endereço fornecido, sendo que essa posição indica qual a posição que será lido do arquivo `data_memory.txt`. Você irá encontrar esses arquivos no github da disciplina, mais especificamente na pasta Memory.

Por exemplo, seu programa seria executado da seguinte forma:

```
./virtual_memory_translate.exe 19986
```

ou

```
./virtual_memory_translate.exe addresses.txt
```

Seu programa produzirá:

- O endereço 19986 contém:
 - o número da página = 4
 - o deslocamento = 3602
 - o Valor lido: 50 (exemplo)

No caso, o número em binário é 0100 1110 0001 0010, sendo que 0100 diz respeito à página e 1110 0001 0010 diz respeito ao deslocamento na página. Você consegue conferir isso com a calculadora do Windows/Linux no modo programador. Para manipular os números em nível de bit, é recomendado usar os operadores bitwise (bit-a-bit) da linguagem escolhida. No caso o exemplo apresentado é para 16 bits. No caso de 32 bits, haveria mais 16 bits a esquerda (mais significativo) referentes ao número de páginas, 0000 0000 0000 0000 0100 1110 0001 0010, porém, ainda será traduzido para página 4 e deslocamento 3602.

Escrever este programa exigirá o uso do tipo de dados apropriado para armazenar 16 à 32 bits (short ou int). É recomendado que você também use tipos de dados sem sinal. Além disso, para endereços de 32 bits deve ser possível usar paginação hierárquica de 2 níveis mantendo 4 Kb, com cada nível tendo 10 bits de tamanho.

Explicando e contexto da aplicação

https://github.com/Halryy/memoryOS_M2

Este programa é projetado para simular a tradução de endereços virtuais em sistemas de gerenciamento de memória. Ele é útil para entender como os endereços virtuais são mapeados em endereços físicos.

O programa aceita várias opções de entrada, permitindo ao usuário fornecer um arquivo de endereços virtuais, um número de página virtual específico ou simplesmente executar o programa para gerar números de página virtual aleatórios. Ele também verifica se o usuário forneceu entradas válidas.

Códigos importantes da implementação

- **getsPhysicalPageNumberFromTable:** Recebe um número de página virtual e um nome de arquivo como entrada. Ele lê o arquivo para obter o número da página física correspondente ao número de página virtual fornecido.
- **getValueFromMemory:** Recebe um número de página física e um deslocamento como entrada. Ele lê o arquivo de memória de dados para obter o valor armazenado na posição correspondente.
- **createsVirtualPageNumber:** Gera um número de página virtual aleatório com base na quantidade de bits especificada.
- **trim:** é usada para remover espaços em branco no início e no final de uma string.

Resultados obtidos com as simulações

O programa é capaz de traduzir endereços virtuais em números de página física e deslocamento com sucesso. Ele fornece informações detalhadas, incluindo o tamanho da memória virtual, o número da página física, o deslocamento e o valor lido na memória de dados.

Análise e discussão sobre os resultados finais

O programa cumpre seu objetivo de traduzir endereços virtuais em números de página física e deslocamento. Ele fornece flexibilidade ao usuário, permitindo que diferentes cenários sejam simulados. A verificação de entrada garante que o programa lide com entradas inválidas de maneira apropriada.