MAC0422 - Sistemas Operacionais

Daniel Macêdo Batista

IME - USP, 19 de Novembro de 2020

Roteiro

Memória Virtual

Page fault

Tabelas de página

Memória Virtual

Ideia básica da memória virtual

- Fornecer para cada programa seu próprio espaço de endereço (já foi visto isso) e esse espaço é quebrado em **páginas**.
- 1 página = uma faixa contígua de endereços
- As páginas são mapeadas na memória física (cada página vai ficar em uma dada posição da memória física) mas nem todas as páginas precisam estar na memória principal ao mesmo tempo para que um programa possa rodar
- Pode ser vista como uma generalização da ideia de base e limite (a diferença é que agora essa base e limite são valores que não necessariamente precisam ter uma relação de 1 para 1 com a memória principal (física))

Funcionamento da memória virtual

- O programa referencia uma posição do seu espaço de endereço
  $\rightarrow$ O SO traduz automaticamente essa posição para uma posição na memória física
  - Se essa página está na memória física $\rightarrow$:-)
  - Senão $\rightarrow$:-(. A instrução falha porque houve um *miss*, O SO é informado, copia o endereço do disco para a memória principal e re-executa a instrução que falhou (Enquanto a cópia ocorre, a CPU pode ser usada por outro processo)

Paginação

- Uma implementação de memória virtual
- Suponha que um programa referencie uma posição de memória assim:

```
MOV EAX, 1000
```

- O endereço 1000 vai ser mapeado num novo endereço que estará dentro da faixa de endereços virtuais do processo. Similar ao que é feito com a base e o limite mas agora essa base e esse limite estão dentro da memória virtual, que pode ser maior que a memória física (O espaço de endereços de um processo agora é um espaço de endereços virtuais)

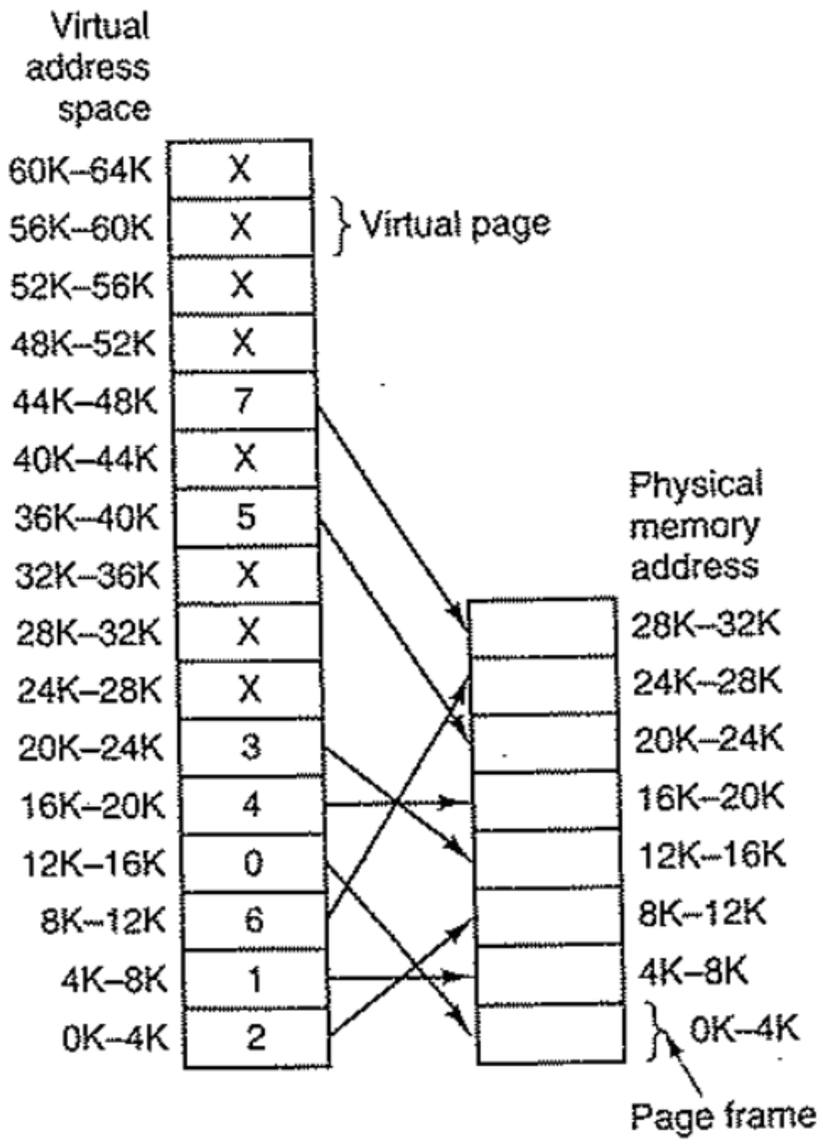
Paginação

- Um endereço virtual é enviado para a MMU (*Memory Management Unit*), que sabe transformar endereços virtuais em endereços físicos
- A ideia é que o espaço de endereços virtuais seja suficiente para carregar o programa inteiro mas não precisa caber na memória física

Exemplo de paginação

- Um computador/SO suporta uma memória virtual com endereços de 16 bits (a memória virtual total tem capacidade para $2^{16} = 64KBytes$)
- O computador tem apenas 32KBytes de memória física. Assim, um programa que ocupe 64KBytes pode ser executado, mas todo o conteúdo dele não consegue ser carregado de uma única vez na memória principal (embora tenha que ser mantido no disco)
- O espaço da memória virtual precisa ser dividido em **páginas** (ideia similar às unidades de alocação). Suponha que nesse caso cada página tem 4KB. O equivalente na memória física chama-se quadro de páginas (*page frame*). As transferências de endereços entre o disco e a memória principal tem a granularidade de 1 página

Exemplo de paginação



Exemplo de paginação

- `MOV EAX, 0`, considerando que a base é 0: a MMU sabe que esse endereço virtual está na página 0 (0-4K), que está mapeada no quadro de página 2 (8K-12K), assim o que é feito na verdade é `MOV EAX, 8192`.
- `MOV EAX, 8192` $\rightarrow$ `MOV EAX, 24576`
- `MOV EAX, 20500` (A página 5 começou em 20480) $\rightarrow$ `MOV EAX, 12308` (20 bytes do início do quadro de página 3 que começou em 12288)

Page fault

Mas o problema não está sendo resolvido!

- **Obs.: usando o exemplo anterior**
- Só fazer esse mapeamento não resolve o problema de fato porque no fim das contas só o que cabe na memória física é 32KB.
- Apenas 8 quadros de páginas na memória física e 16 páginas na memória virtual
- Um bit de **Presente/Ausente** tem que existir em cada página para informar quais delas estão na memória física e quais não estão. Num dado instante de tempo, no máximo 8 páginas podem estar com o bit de **Presente** ligado.

Mas e se a página não está mapeada?

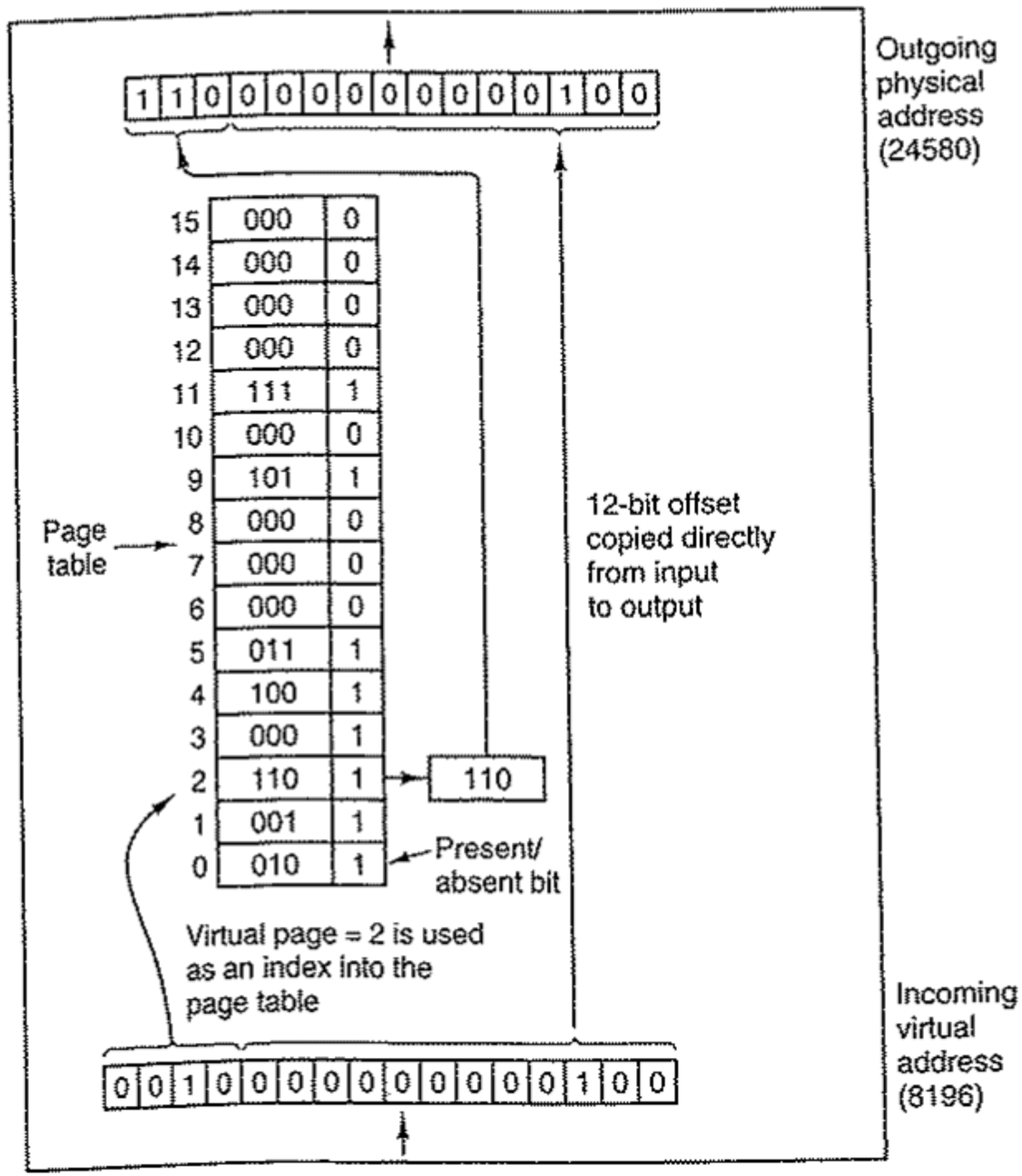
- Um algoritmo bem genérico:
  1. SO escolhe um quadro de página pouco usado
  2. Copia o conteúdo do quadro do passo 1 no disco (caso não esteja lá)
  3. Copia o conteúdo da página que deu falha no quadro do passo 1
  4. Atualiza o mapeamento e os bits de Presente/Ausente para todas as páginas envolvidas
  5. Reexecuta a instrução

Exemplo

- `MOV EAX, 32780`
- 32780 está na página que começa com 32K (32768), ou seja, é a página 32K-36K
- A página 32K-36K estaria com o bit de Ausente ligado, informando que essa página não está na memória real
- Suponha que o algoritmo implementado pelo SO escolha ocupar o quadro de página número 1
- Copia o conteúdo desse quadro 1 no endereço virtual dele
- Marca a página virtual 4K-8K como Ausente
- Marca a página virtual 32K-36K como Presente, copia o conteúdo dessa página para o quadro de página 4K-8K e guarda a informação de que a página virtual 32K-36K está mapeada no quadro de página 1.
- O `MOV EAX, 32780` é reexecutado agora como `MOV EAX, 4108`

Tabelas de página

Como a MMU traduz os endereços?



$8196 \rightarrow 24580$

Resumindo...

- um endereço virtual é dividido no número da página virtual (bits mais significativos) e um *offset* (bits menos significativos)
- ex.: endereços de 16 bits, páginas de 4KB: 4 últimos bits especificam 1 das 16 páginas virtuais e 12 primeiros bits especificam 1 dos 4096 endereços dentro da página
- Número da página virtual = Indice na tabela de páginas para encontrar a entrada daquela página virtual
- Número do quadro de página é encontrado e é colocado no lugar do número da página virtual
- A Tabela de páginas funciona como uma função