

LAB 3: Paginação

OBJETIVO: COM ALOCAR E GERIR MEMÓRIA

Testar dois simuladores em Python para compreender o processo de tradução de memória virtual em memória física utilizando sistema de representação linear de páginas e o sistema de representação multi-tabela.

1. Representação Linear da memória física

Na representação linear da memória física, é criada uma tabela para representar todos os espaços de memória física. Uma vez que a memória física está, à semelhança da memória virtual, dividida em páginas, cada entrada na tabela vai guardar o endereço físico da página (PFN – *page frame number*) que corresponde à página virtual (VPN – *virtual page number*).

Cada endereço virtual que existe no código tem duas componentes: **Virtual Page Number (VPN)** e **offset**.
Por exemplo,

- Um espaço de endereços virtual de 64 bytes;
- Necessitamos de 6 bit por endereço para identificar cada posição ($2^6=64$):

Va5	Va4	Va3	Va2	Va1	Va0
-----	-----	-----	-----	-----	-----

- Cada página vai ter uma dimensão de 16 bytes, o que significa que no espaço de endereçamento vamos usar $64/16 = 4$ páginas (Precisamos de 2 bits para as identificar).

Pode dividir-se o endereço virtual da seguinte forma:

VPN			offset		
Va5	Va4	Va3	Va2	Va1	Va0

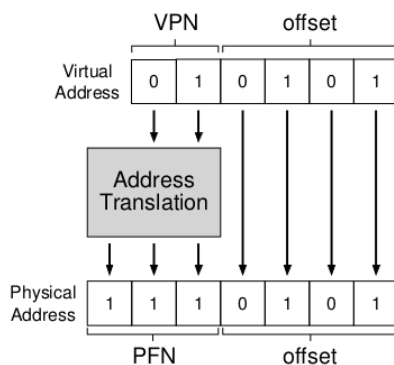
VPN (virtual page number): Os dois bits mais significativos denotam o número da página;

offset: Os restantes bits indicam qual é a posição dentro dessa página;

A tabela vai ter a dimensão de 4 entradas, uma por página.

VPN0	VPN1	VPN2	VPN3
PFN0	PFN1	PFN2	PFN3

O endereço físico obtém-se segundo o esquema:



2. Representação multi-tabela da memória física

Quando a memória é de uma dimensão considerável, a tabela de páginas adquire dimensões proibitivas. Há que reduzir a sua dimensão.

Por exemplo, para um espaço de endereçamento de 32-bit (4GB de memória) e com páginas de dimensão 4KB, vão existir 2^{20} páginas ($4 \times 2^{30} / 4 \times 2^{10} = 2^{20}$) o que corresponde a aproximadamente 1 milhão de páginas!

Se cada entrada na tabela de páginas tiver a dimensão de 4 bytes então o valor sobe para 4 milhões.

A solução? Usar uma estrutura intermédia designada diretoria de páginas.

Vejamos um outro exemplo para testar a ideia:

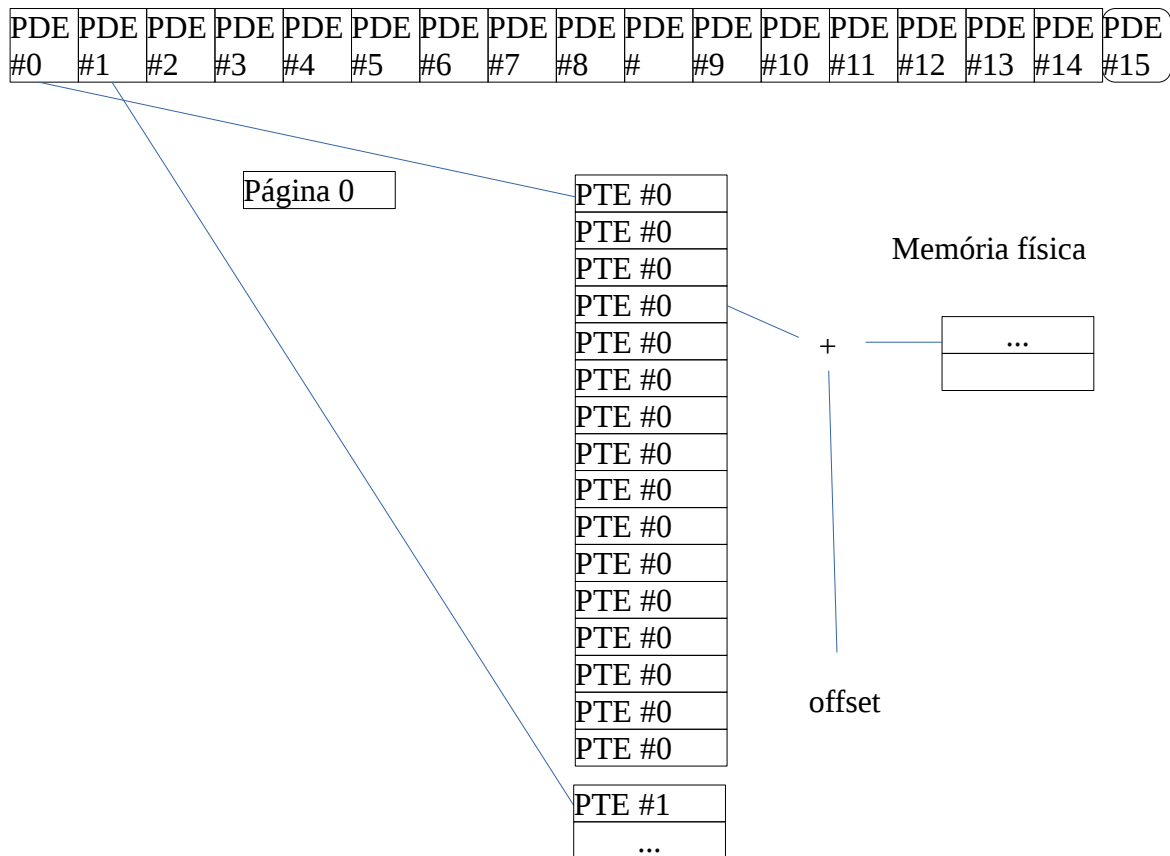
- Espaço de endereços com 16KB com páginas de 64-bytes
- $16KB = 4 \times 2^{10}$ bytes; 64 byte = 2^6 bytes; $4 \times 2^{10} / 2^6 = 2^8 = 256$ páginas
- Como um endereço tem 14 bits, **8 bits serão o VPN** e **6 bits o offset**

A tabela de páginas teria 256 entradas.

Vamos saber qual é o espaço que ocupa esta estrutura com 256 entradas:

- Cada PTE ocupa 4 bytes, logo a dimensão da tabela é $4 \times 2^8 = 2^{10}$ bytes (1024 bytes).
- Como cada página tem a dimensão de 64 bytes, a estrutura ocupa $2^{10} / 2^6 = 2^4 = 16$ páginas.

Criamos uma estrutura intermédia, designada por **tabela de diretoria de páginas** com 16 entradas. Cada entrada guarda a localização física de uma das páginas da tabela de páginas:



O endereçamento de 16K passa a ser representado por um espaço para a entrada na tabela de diretorias, um espaço para identificar qual a entrada da PTE na página respetiva e finalmente o offset para chegar ao valor final da memória física.

3. Questões

3.1 Tradução linear

Nesta tarefa utiliza-se o programa `paging-linear-translate.py` para realizar a tradução de endereços virtuais em endereços físicos. Deve consultar o documento “README” para ter mais informação.

a) Usar o simulador para estudar como as dimensões da tabela de páginas varia com a introdução de parâmetros diferentes. Utilizando a flag `-v`, é possível ver quantas entradas da tabela de páginas foram preenchidas.

i. Execute os seguintes comandos e verifique a alteração da dimensão da tabela à medida que o espaço de endereçamento aumenta:

```
paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0
paging-linear-translate.py -P 1k -a 2m -p 512m -v -n 0
paging-linear-translate.py -P 1k -a 4m -p 512m -v -n 0
```

ii. Execute os seguintes comandos e verifique como a alteração do tamanho de páginas, altera a dimensão da tabela de páginas.

```
paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0
paging-linear-translate.py -P 2k -a 1m -p 512m -v -n 0
paging-linear-translate.py -P 4k -a 1m -p 512m -v -n 0
```

iii. O que esperava? Como deve variar a dimensão da tabela de páginas à medida que o espaço de endereçamento aumenta? E no que respeita ao aumento da dimensão de página? Por que razão não utilizar apenas páginas de grande dimensão?

b) Neste exercício pretende-se realizar algumas traduções.

Utilize um ou mais exemplos, mudando o número de páginas que estão alocadas ao espaço de endereçamento utilizando a flag `-u`. Por exemplo:

```
paging-linear-translate.py -P 1k -a 16k -p 32k -v -u 0
paging-linear-translate.py -P 1k -a 16k -p 32k -v -u 25
paging-linear-translate.py -P 1k -a 16k -p 32k -v -u 50
paging-linear-translate.py -P 1k -a 16k -p 32k -v -u 75
paging-linear-translate.py -P 1k -a 16k -p 32k -v -u 100
```

O que acontece à medida que se aumenta em percentagem o número de páginas que estão alocadas em cada espaço de endereçamento?

c) Experimentamos utilizar “seeds” diferentes e parâmetros (loucos) para os espaços de endereçamento:

```
paging-linear-translate.py -P 8 -a 32 -p 1024 -v -s 1
paging-linear-translate.py -P 8k -a 32k -p 1m -v -s 2
paging-linear-translate.py -P 1m -a 256m -p 521m -v -s 3
```

Qual destas combinações parece ser irrealista? Porquê?

4. Utilize o programa para experimentar outro tipo de problemas. Consegue encontrar os limites para os quais o programa não funciona? Por exemplo, o que acontece se a dimensão de espaço de endereçamento for maior que a memória física?

3.2 Tradução por multi-tabela

a) O programa `paging-multilevel-translate.py` pressupõe

- A dimensão da página é muito pequena (32 bytes).
- O espaço de endereçamento virtual é de 32 KB
- A memória física é constituída por 128 páginas.

i. Indique quantos bits tem um endereço virtual? 15

ii. Indique quantas páginas tem o espaço de endereçamento virtual? 1024

iii. Quantas entradas teria tabela de páginas se se utilizasse a tradução linear? 1024

iv. Quantas páginas ocupa essa tabela de páginas? $2^{10} / 2^5 = 2^5 = 32$ páginas.

- v. Identifique no esquema do endereçamento virtual o **offset**, o **VPN** e o **PDE**
- vi. Desenhe o esquema da tabela de diretoria mais a tabela de páginas e identificando os conteúdos das entradas de cada tabela.
- b) Corra o programa (primeiro sem -c) e traduza os endereços virtuais.