



ESTGF

ESCOLA SUPERIOR DE TECNOLOGIA
E GESTÃO DE FELGUEIRAS
POLITÉCNICO DO PORTO

Memória Virtual

Sistemas Operativos

Licenciatura em Engenharia Informática
2007/2008

Sumário

- Memória virtual
- *Overlay*
- *Paging*
- MMU

Contexto

- Memória virtual surge da necessidade de executar programas maiores do que a memória física disponível
- Conceito surgiu em 1961
- Funcionava de modo bastante simples
- Tamanho total do programa, dados e pilha poderia superar a memória física existente
- Desde de que se seleccione as partes, do programa, necessárias para a sua execução
- Carregando-as para memória e mantendo o resto em disco

Contexto

- Exemplo

- Considerando um programa que ocupe 1 Mb de espaço
- Poder-se-á, eventualmente, executar o programa utilizando apenas um segmento de memória de 256K
- Basta que se seleccione correctamente, quais os 256K a manter em memória em cada instante de tempo
- Desta forma é também possível suportar multiprogramação
- Segundo este exemplo, um computador com 1 Mb de memória, poderá manter 4 programas constantemente em memória física
- Possibilitando a sua execução concorrente

Contexto

- Técnicas mais comuns de memória virtual
 - *Overlay* (sobreposição)
 - *Paging* (paginação)

Overlay

- Primeira solução encontrada
- Consiste na divisão do programa em segmentos de menor tamanho
- Cada um dos segmentos tem de ser capaz de ser contido em memória física
- Vários segmentos devem ter tamanhos muito aproximados
- Início da execução do programa, carrega-se o primeiro segmento para memória e é executado até ao fim
- Concluindo-se a execução de um segmento, este é substituído na sua totalidade pelo próximo (*overlay*)

Overlay

- Permite a execução de programas de tamanho superior ao da memória existente
- Troca de *overlay* é da responsabilidade do sistema operativo
- Divisão do programa em *overlays* compete ao programador

Paging

- A mais adoptada pelos sistemas de memória virtual
- Assenta na capacidade de utilização de um conjunto limitado de endereços de memória, por parte dos programas
- Recorre para tal à MMU (*Memory Managment Unit*)

```
MOVE  BX, 1000
```

- Linha de código *assembly* leva o CPU a transferir para a zona de memória 1000 o conteúdo do registo BX
- Ao endereço de memória 1000, dá-se o nome de endereço virtual

Paging

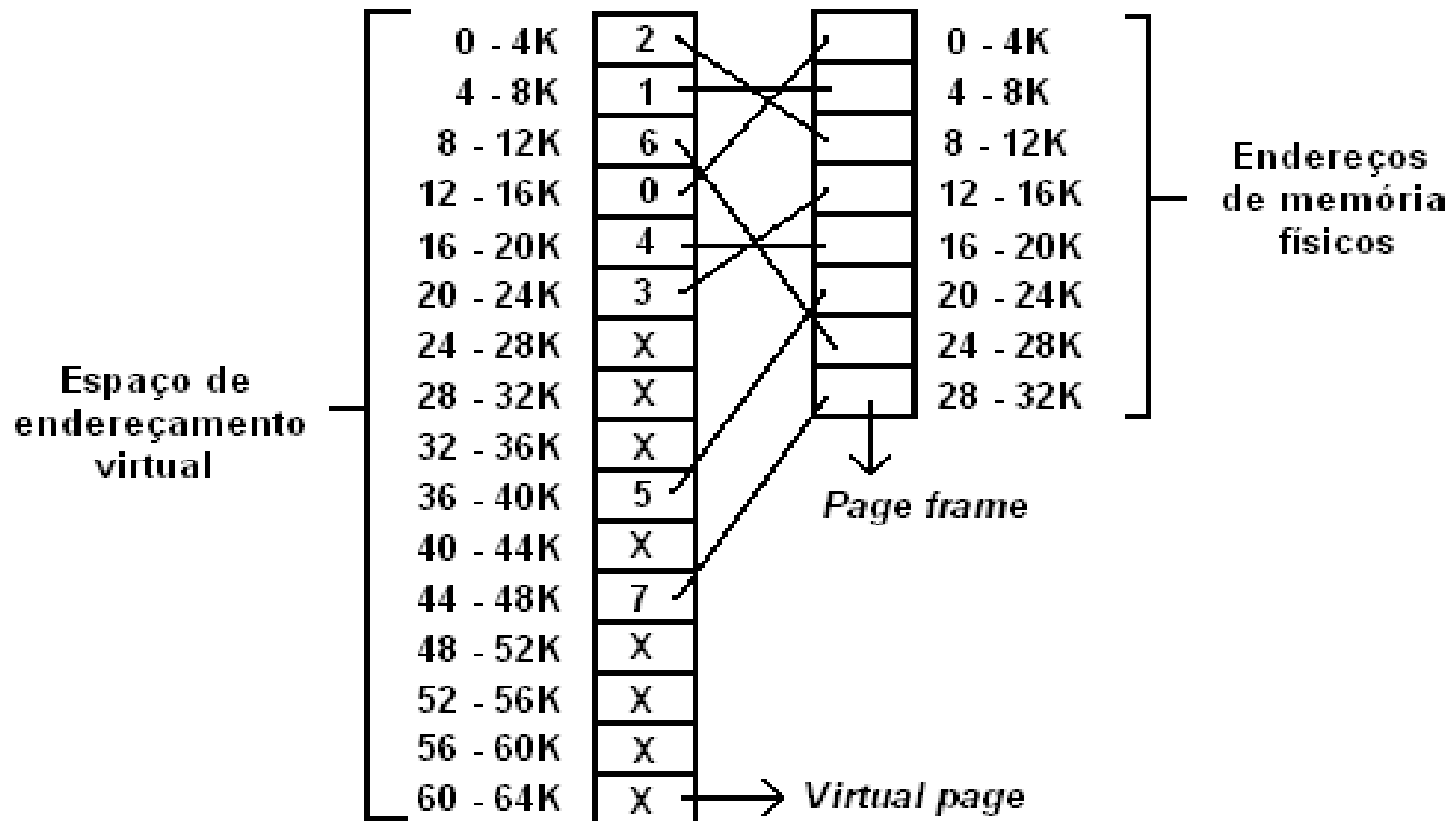
- Conjunto de endereços virtuais dá-se o nome de espaço de endereçamento virtual
- Se não se usar memória virtual, então o endereço virtual vai ser utilizado directamente, pelo barramento de memória, para aceder à informação
- Se se usar Memória Virtual, o endereço virtual será processado pelo MMU, obtendo-se assim o endereço físico correspondente, para enviar para o barramento de memória

Paging

- Espaço de endereçamento virtual é dividido em páginas (*pages*)
- Cada *page* corresponde a um segmento de memória física chamado de *page frame*
- *Pages* e *pages frames* têm obrigatoriamente o mesmo tamanho
- Tamanho varia entre 512b e 8KB

Paging

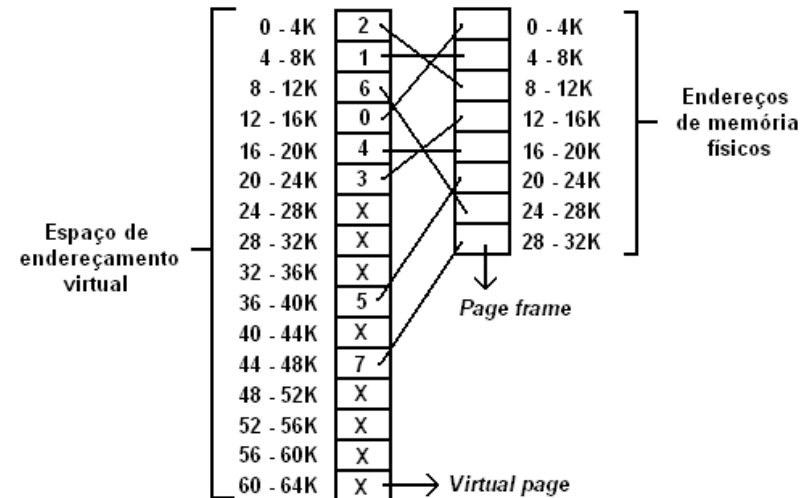
- Exemplo (64K memória virtual; 32K memória física)



Paging

- Exemplo (64K memória virtual; 32K memória física)

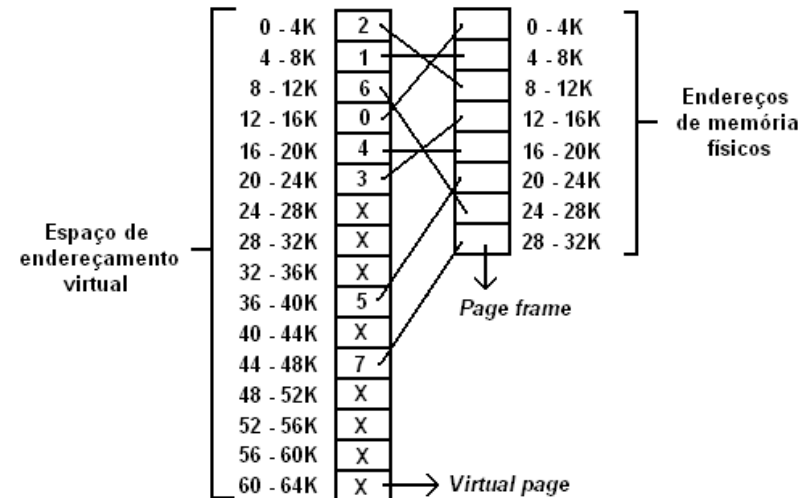
- Quando um programa tenta aceder ao endereço de memória 0 (zero), este endereço é enviado para o MMU
- MMU sabe que o endereço virtual 0 (zero) pertence à *virtual page 0* (0-4095)
- Corresponde à *page frame 2* (8192-12287)



Paging

- Exemplo (64K memória virtual; 32K memória física)

- MMU transforma o endereço de memória virtual 0 (zero) no endereço de memória física 8192
- Sendo enviado de seguida para o barramento de memória
- Assim, o barramento de memória, não chega a aperceber-se da existência de memória virtual



Paging

- Exemplo (64K memória virtual; 32K memória física)

- *Virtual pages* representadas com um “X”, são as que não têm correlação na memória física
- Quando um programa tenta aceder a uma zona destas, CPU gera um *interrupt* de software chamado de *page fault*
- *Page fault* serve para alertar o sistema operativo de tal acontecimento
- Sistema operativo envia a *page frame* menos utilizada para disco e carrega para a *page frame* libertada com a referenciada pelo programa

Paging

- Exemplo (64K memória virtual; 32K memória física)

- Ajustando o espaço de endereçamento virtual
- Assim o programa pode continuar a executar normalmente do ponto onde parou

Paging

- MMU

- *Memory Managment Unit*
- Chip ou conjunto de chips
- Faz(em) parte do processador (CPU)
- Têm a responsabilidade de relacionar endereços de memória virtuais em endereços físicos
- Para um determinado endereço virtual, devolve um determinado endereço físico

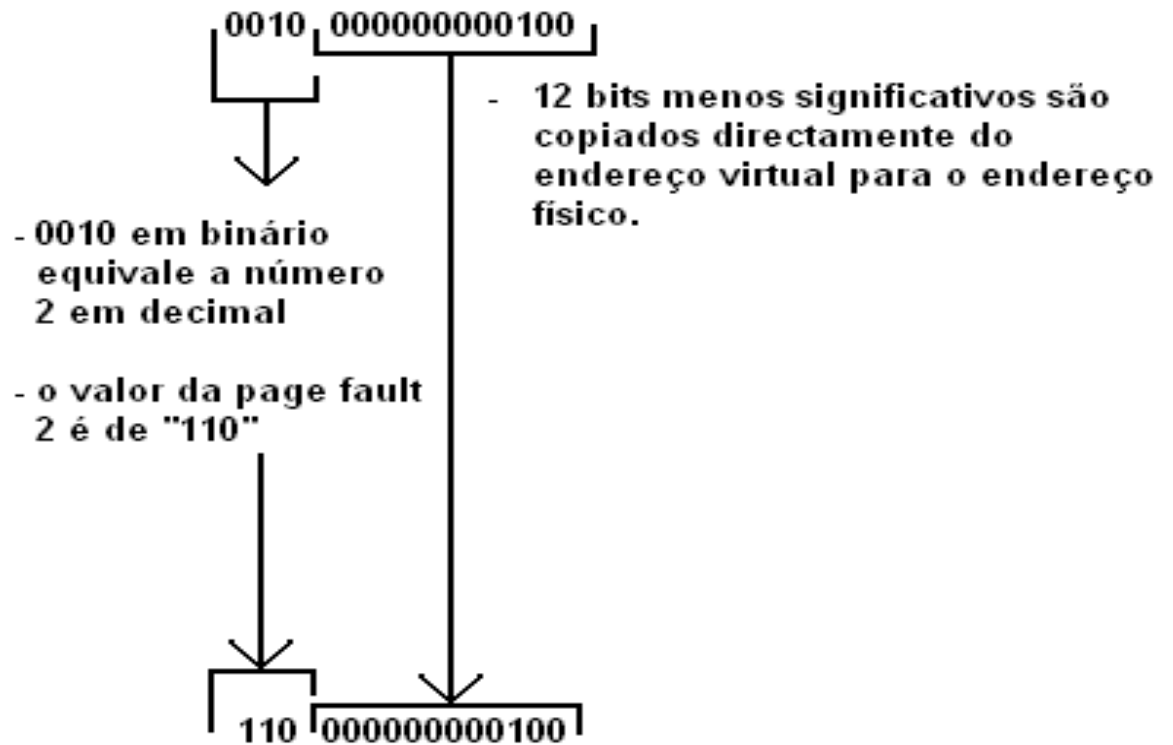
Paging

- MMU

0	010	1
1	001	1
2	110	1
3	000	1
4	100	1
5	011	1
6	000	0
7	000	0
8	000	0
9	101	1
10	000	0
11	111	1
12	000	0
13	000	0
14	000	0
15	000	0



- bit que indica se a page frame existe (1) ou não (0) em memória física.



Paging

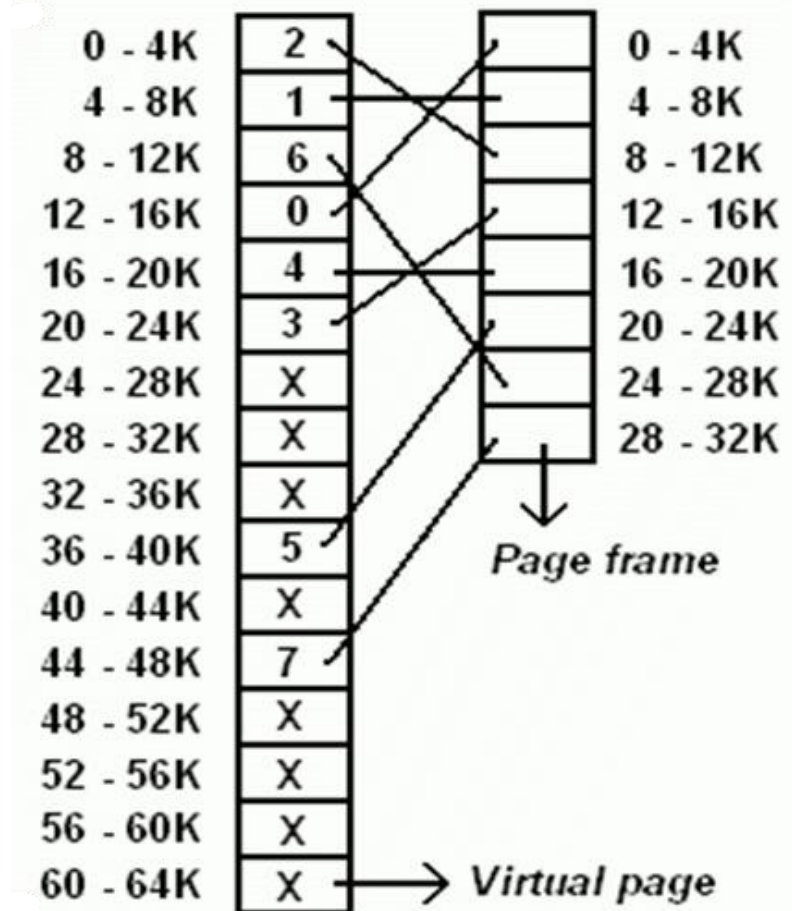
- MMU

- Endereço virtual 8196 (0010000000000100)
- É transformado num endereço físico de 15 bits
110000000000100 (24580)
- Sendo este enviado de seguida para o barramento de memória

Exercícios

- Paging

- Assumindo o cenário representado na figura lateral, indique:
 - Qual o endereço físico do endereço virtual 16884?
 - O endereço 25600 gera um *page fault*?
 - O endereço físico 30720 corresponde a que endereço virtual?



Exercícios

- MMU

- Assumindo o cenário representado na figura lateral, converta os seguintes endereços virtuais em endereços físicos:

a) 16884

b) 25600

c) 30720

0	010	1
1	001	1
2	110	1
3	000	1
4	100	1
5	011	1
6	000	0
7	000	0
8	000	0
9	101	1
10	000	0
11	111	1
12	000	0
13	000	0
14	000	0
15	000	0