

# Sistemas Operacionais

André Luiz da Costa Carvalho

andre@icomp.ufam.edu.br

Aula 10 - Paginação

# Aula de Hoje

- 1 Introdução
- 2 Paginação Básica
  - Tabela de Paginação
  - Tradução de Endereços
- 3 Mecanismos de Paginação
  - Armazenamento
  - Entradas da tabela de páginas
- 4 Performance

## Na aula anterior...

Vimos o conceito de gerenciamento de memória via **segmentação**.

Vimos que a segmentação, por quebrar o espaço de endereçamento em blocos de tamanho diferentes, pode levar a **fragmentação externa**.

Para resolver este problema, foi criada a **paginação**.

# Paginação

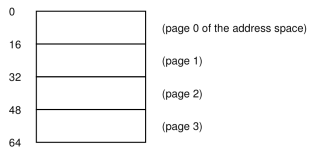
Na paginação, Ao invés do processo ser quebrado em blocos lógicos (code, heap, stack), todo o espaço de endereçamento virtual é dividido em blocos de tamanho fixo, as **páginas**.

A memória também é quebrada em blocos de tamanho fixo, chamadas de **quadros**(ou frames).

Virtualização de memória por paginação.

# Um exemplo simples

## Processo

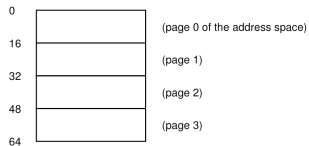


### Processo:

Endereçamento virtual de 64 bytes, 4 páginas de 16 bytes.

# Um exemplo simples

## Processo



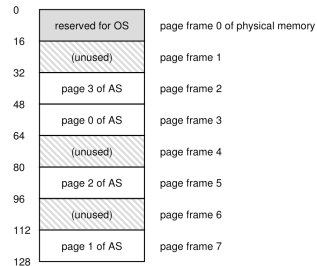
### Processo:

Endereçamento virtual de 64 bytes, 4 páginas de 16 bytes.

### Memória:

8 quadros de 16 bytes, 128 bytes de memória

## Memoria



## Vantagens da Paginação

**Flexibilidade:** S.O. não precisa se preocupar com detalhes do espaço de endereçamento do processo, pra que direção cresce o heap/stack, como usa cada um e etc.

**Simplicidade no gerenciamento de memória livre:** Basta uma lista de páginas livres e first fit.

# Aula de Hoje

- 1 Introdução
- 2 **Paginação Básica**
  - Tabela de Paginação
  - Tradução de Endereços
- 3 Mecanismos de Paginação
  - Armazenamento
  - Entradas da tabela de páginas
- 4 Performance

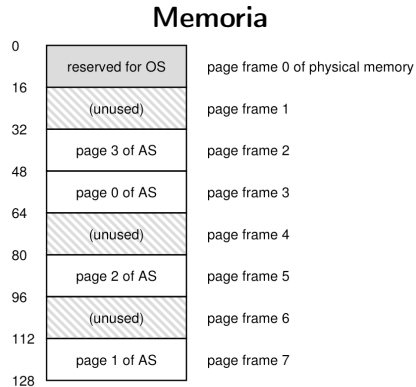
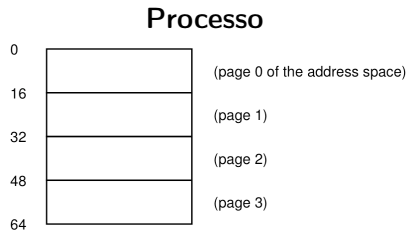


# Tabela de paginação

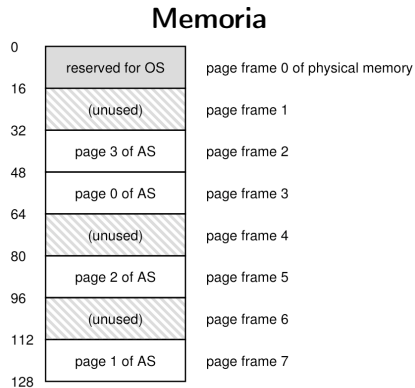
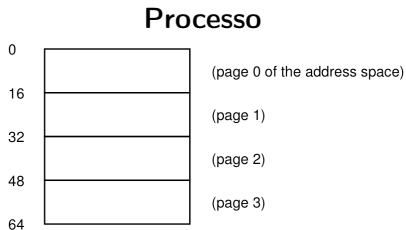
Como saber onde está na memória real as páginas de cada processo? Tabela de Páginas.

Objetivo: Traduzir os endereços virtuais em reais.

# Tabela de paginação



# Tabela de paginação



Uma tabela por processo.

# Aula de Hoje

## 1 Introdução

## 2 Paginação Básica

- Tabela de Paginação
- Tradução de Endereços

## 3 Mecanismos de Paginação

- Armazenamento
- Entradas da tabela de páginas

## 4 Performance

# Tradução de endereço

```
movl <endvirtual>, %eax
```

# Tradução de endereço

```
movl <endvirtual>, %eax
```

Este `<endvirtual>` deve ser traduzido. Para isto, precisamos do número da página virtual (VPN) e do offset da memória dentro desta página.

- Neste exemplo, como temos **64 bytes** de espaço de endereçamento virtual, precisamos de **6 bits para um endereço** ( $2^6 = 64$ ):

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

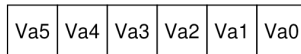
Quantos destes bits devem representar a página virtual e quantos o offset?

# Tradução de endereço

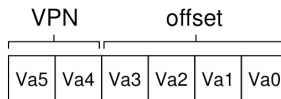
```
movl <endvirtual>, %eax
```

Este `<endvirtual>` deve ser traduzido. Para isto, precisamos do número da página virtual (VPN) e do offset da memória dentro desta página.

- Neste exemplo, como temos **64 bytes** de espaço de endereçamento virtual, precisamos de **6 bits para um endereço** ( $2^6 = 64$ ):



Quantos destes bits devem representar a página virtual e quantos o offset?



# Tradução de Endereços

```
movl 21, %eax
```

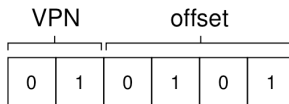
Qual seria a página e o offset?



# Tradução de Endereços

```
movl 21, %eax
```

Qual seria a página e o offset?

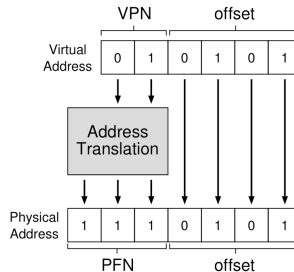


# Tradução de Endereços

Após localizar a página e o offset, o próximo passo é descobrir em qual quadro físico está a página um.

- No caso, na página 7.

De posse disto, basta traduzir a parte virtual que corresponde à página pelo quadro físico:



# Resumindo

De posse desse conhecimento básico, vamos agora ver um pouco dos detalhes sujos sobre como a paginação funciona.

Vamos também resolver algumas dúvidas que vocês podem estar tendo:

- Onde as páginas são armazenadas?
- Qual o conteúdo da tabela de páginas e qual o tamanho delas?
- Esse monte de tradução não deixa o sistema lento?

# Aula de Hoje

- 1 Introdução
- 2 Paginação Básica
  - Tabela de Paginação
  - Tradução de Endereços
- 3 Mecanismos de Paginação**
  - Armazenamento**
  - Entradas da tabela de páginas
- 4 Performance

## Onde as páginas são armazenadas

Em espaços de endereçamento reais, a tabela de páginas pode ficar muito grande, ordens de magnitude maior que tabelas de segmentos.

**Por exemplo:** em um espaço de 32bits, com páginas de 4KB, os endereços seriam 20 bits de número de páginas virtuais e 12 de offset.

20 bits de páginas virtuais significam  $2^{20}$  traduções na tabela de páginas (+ de um milhão). Se cada entrada na tabela de páginas for apenas 1 inteiro (4 bytes), cada tabela teria **4 MEGAS**.

Parece pouco? Imagine que tem 100 processos rodando no sistema, isso daria **400 MEGAS SÓ DE TABELAS DE TRADUÇÃO**.

## Onde as páginas são armazenadas

Em espaços de endereçamento reais, a tabela de páginas pode ficar muito grande, ordens de magnitude maior que tabelas de segmentos.

**Por exemplo:** em um espaço de 32bits, com páginas de 4KB, os endereços seriam 20 bits de número de páginas virtuais e 12 de offset.

20 bits de páginas virtuais significam  $2^{20}$  traduções na tabela de páginas (+ de um milhão). Se cada entrada na tabela de páginas for apenas 1 inteiro (4 bytes), cada tabela teria **4 MEGAS**.

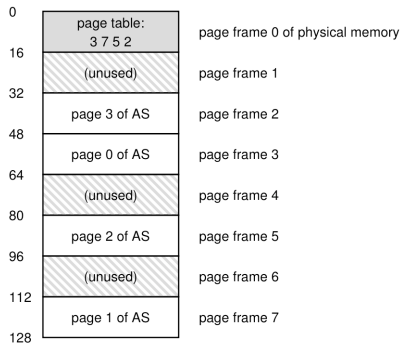
Parece pouco? Imagine que tem 100 processos rodando no sistema, isso daria **400 MEGAS SÓ DE TABELAS DE TRADUÇÃO**.

Agora imagine isso num **espaço de 64 bits!**

# Onde as páginas são armazenadas

Devido aos tamanhos da tabela, não tem condição de ter um hardware próprio na MMU para guardar a tabela do processo atual.

Ao invés disso, as tabelas são guardadas na própria memória:



# Aula de Hoje

## 1 Introdução

## 2 Paginação Básica

- Tabela de Paginação
- Tradução de Endereços

## 3 Mecanismos de Paginação

- Armazenamento
- Entradas da tabela de páginas

## 4 Performance



## Qual o conteúdo da tabela de páginas?

A tabela de páginas é apenas uma estrutura de dados para fazer a tradução de página (virtual) para (quadro).

- Então qualquer estrutura de dados poderia servir.

Por enquanto vamos focar nas tabelas de páginas lineares: Vetores onde o índice é a página virtual e o conteúdo indica, entre outras coisas, o quadro correspondente.

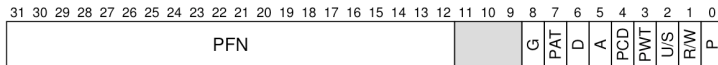
No futuro veremos estruturas mais avançadas.

Mas afinal, o que exatamente é guardado no conteúdo da tabela?

## Conteúdo de entrada na tabela:

Temos, além do número do quadro para a tradução, uma série de bits de flag:

- **Bit de validade:** Diz se esta página já foi alocada ou não.
- **Bits de proteção:** Indicam se a página pode ser lida, escrita ou executada.
- **Bit de presença**
- **Bit dirty**
- **Bit de referência**



## Traduções podem deixar o sistema lento?

Além das tabelas serem grandes, elas também podem levar a atrasos no sistema. Vamos voltar a nossa instrução:

```
movl 21, %eax
```

**Por alto**, para fazer esta instrução (ignorando seu fetch), o sistema deve traduzir o endereço virtual (21) para o real (117).

Para isto, o sistema deve primeiro encontrar a entrada correspondente na tabela de páginas, fazer a tradução e carregar os dados da memória para o registrador.

We have to go deeper.

## Traduções de endereços

Antes de tudo, o hardware tem que saber onde está a tabela de páginas do processo atual.

- Vamos assumir um registrador base para a tabela (como um ponteiro para o início do vetor).

De posse dele, é só calcular a página virtual do endereço e usar ela para achar na tabela:

```
PagVirtual = (EndVirtual & MASKPAG) >> SHIFTPAG
```

```
EntradaPag = *(RegBaseTabelaPag + (PagVirtual * sizeof(entradaTabPag)))
```

Com MASKPAG=110000b pois os dois últimos bits são a página, SHIFTPAG=4 pois são quatro bits de offset

## Tradução de endereços

De posse do endereço na tabela de páginas, basta extrair o número da página física e concatenar com o offset para gerar o endereço físico:

```
offset = (EndVirtual & MASKOFF)  
EndFisico = EntradaPag<<SHIFTPAG | offset
```

Depois de todo esse trabalho que finalmente dá para pegar o conteúdo do endereço físico e colocar no registrador `eax`.

## Tradução de endereços

Que trabalho hein? Mas não acabou!

S.O. ainda precisa checar todos os bits de controle, como permissão de acesso e se a página é válida.

Isto para **CADA USO DA MEMÓRIA**.

Se não for feita corretamente, a paginação pode:

- Gastar muita memória com as tabelas.
- Ser computacionalmente caras de computar.

## Concluindo

Na aula de hoje introduzimos o conceito de paginação, que é a solução de fato utilizada nos sistemas operacionais.

Vimos que ela tem várias vantagens sobre a segmentação, como não ter fragmentação externa e a simplicidade para utilizar endereços esparsos.

Contudo, a implementação de paginação não é simples, e erros de projeto podem levar a desperdícios de recursos no sistema.

Nas próximas aulas duas aulas, veremos como estes problemas podem ser solucionados.