

Nome: Davi Augusto Neves Leite

RA: 191027383

Resolução – Lista 2 de Sistemas Operacionais II – Segmentação

- 1) Primeiramente, verifica-se se o segmento 1 está na memória e, de acordo com o desenho esquemático, ele está ausente. Dessa forma, ocorre uma falta de segmento, e é necessário alocar o segmento para, então, converter o endereço virtual em físico.

Seguindo o processo de alocação, verifica-se se há espaço de alocação para o segmento. Neste caso, o segmento 1 pode ser posto **após o segmento 2**, iniciando no endereço $SEG1_INICIO = 10KB + 32KB + 256KB = 298KB$ e terminando em $SEG1_FIM = 298KB + 64KB = 362KB$.

Sendo assim, com um deslocamento $D = 1024B$, o endereço físico é dado por:

$$EF = SEG1_INICIO + D = 298KB + 1024B = 306176B$$

Repetindo o processo, porém levando em conta o segmento 5, é necessário alocar na memória principal com base no desenho esquemático. Dessa forma, seleciona-se o espaço antes do segmento 3 para a alocação do segmento 5, composto do intervalo de $[0, 10KB]$. Em outras palavras, $SEG5_INICIO = 0KB$ e $SEG5_FIM = 10KB$.

Sendo assim, com um deslocamento 1024, o endereço físico é dado por:

$$EF = SEG5_INICIO + D = 0KB + 1024B = 1024B$$

2) Conversão de Endereço Virtual para Físico (Segmentação com Paginação!)

$$\Rightarrow EF = (TAM_PAG * MOLDURA) + DESLOCAMENTO$$

Tamanho Total das Páginas: $4KB = 2^{12} \Rightarrow 12$ bits

A) EV: 1-0-1023

$$\Rightarrow EF = SEG1_PV0_INICIO + D = (4KB * 86) + 1023 = 352256B + 1023B$$

$$\Rightarrow EF = 353279B$$

B) EV: 2-3-4000

$$\Rightarrow EF = SEG2_PV3_INICIO + D = (4KB * 22) + 4000 = 90112B + 4000B$$

$$\Rightarrow EF = 94112B$$

3) Tamanho Total das Páginas: $2^{10} = 1\text{KB}$

A) Buscar do Segmento 1 = Leitura -> possível. Página Virtual 1: presente na moldura 14.

Dessa forma:

$EV = 1-1-3$

=> $EF = \text{SEG1_PAG1_INICIO} + D = (1\text{KB} * 14) + 3 = 14336\text{B} + 3\text{B}$

=> Endereço Físico = 14339B

B) Armazenar no Segmento 0 = Escrita -> impossível

Dessa forma: violação de proteção!

C) Buscar do Segmento 1 = Leitura -> possível. Página Virtual 4: ausente da memória principal (presente no disco) -> Falta de Página!

Dessa forma: falta de página!

D) Saltar para o Segmento 1 = Execução -> impossível.

Dessa forma: violação de proteção!

4) Em ambas estruturas, isto é, tanto na paginação pura quanto na segmentação com paginação ocorre a chamada **fragmentação interna**. Neste tipo de fragmentação, a **última página virtual** apresenta um armazenamento **a mais** do que, teoricamente, o necessário pelo programa (por exemplo: suponha um sistema que tenha páginas de 4KB. Caso um programa necessita-se de 201KB para executar, seriam alocadas 51 páginas com total de 204KB, resultando em uma fragmentação interna de 3KB na última página).

E aqui está a principal diferença da fragmentação entre as duas abordagens: na paginação pura, esse excesso de espaço do programa, de fato, não é usado pelo programa (na maioria das vezes); mas na segmentação com paginação, esse espaço extra pode ser utilizado (na maioria das vezes) para aumentar a estrutura do segmento e ocupar espaço reservado a dados/pilha (haja visto que os segmentos são blocos que separam as estruturas por importância, como segmento para código e segmento para dados). Em outras palavras, na segmentação com paginação, a fragmentação interna será mais eficiente e “pequena” do que na paginação pura.

Portanto, pode-se dizer que, em média, a perda de memória na segmentação com paginação é bem menor do que na paginação pura, sendo um mecanismo mais eficiente nos computadores.