

5. Um computador tem 8 molduras de páginas. Na tabela de páginas existe a seguinte informação relativa às páginas que aí residem no momento

Moldura	Ciclo de transferência	Ciclo da última referência	R	M
0	129	156	1	1
1	132	143	1	0
2	111	111	0	0
3	96	116	0	1
4	152	162	1	1
5	126	146	1	1
6	90	164	1	0
7	138	154	1	1

Na ocorrência de uma falha no acesso a uma página, qual moldura seria substituída pelos seguintes algoritmos:

- a) Substituição de página não usada recentemente;
- b) Primeira a entrar primeira a sair;
- c) Substituição com segunda tentativa I;
- d) Substituição da menos utilizada recentemente.

Resolução:

Para responder às perguntas, precisamos entender os algoritmos de substituição de página e analisar as informações fornecidas na tabela. Os algoritmos mencionados são:

a) Substituição de página não usada recentemente (NRU - Not Recently Used):

Neste algoritmo, as páginas são classificadas em quatro classes diferentes com base em seus bits de referência (R) e modificação (M): (0, 0), (0, 1), (1, 0) e (1, 1). A página escolhida para substituição será da classe com menor prioridade.

b) Primeira a entrar primeira a sair (FIFO - First-In-First-Out):

Nesse algoritmo, a página que entrou primeiro na memória será substituída.

c) Substituição com segunda tentativa I (Second Chance):

Nesse algoritmo, as páginas são percorridas em ordem circular. Se uma página tem R=0, ela é substituída. Caso contrário, seu bit R é zerado e ela é movida para o final da lista. Esse processo é repetido até encontrar uma página com R=0 para substituição.

d) Substituição da menos utilizada recentemente (LRU - Least Recently Used):

Nesse algoritmo, a página que foi menos referenciada (ou seja, a que possui o maior ciclo de transferência ou ciclo da última referência) será substituída.

Agora, vamos aplicar cada algoritmo às informações fornecidas na tabela de páginas:

a) Substituição de página não usada recentemente (NRU):

De acordo com a tabela, podemos classificar as páginas nas seguintes classes:

Página 0: (1, 1)

Página 1: (1, 0)

Página 2: (0, 0)

Página 3: (0, 1)

Página 4: (1, 1)

Página 5: (1, 1)

Página 6: (1, 0)

Página 7: (1, 1)

A página escolhida para substituição será da classe com menor prioridade. Neste caso, a classe com menor prioridade é (0, 0), que corresponde à página 2. Portanto, a moldura a ser substituída pelo algoritmo NRU seria a moldura 2.

b) Primeira a entrar primeira a sair (FIFO):

Neste algoritmo, a página que entrou primeiro na memória será substituída. Com base na tabela, a moldura que entrou primeiro é a moldura 0. Portanto, a moldura a ser substituída pelo algoritmo FIFO seria a moldura 0.

c) Substituição com segunda tentativa I (Second Chance):

Neste algoritmo, começaremos a busca pela moldura 0. Ao percorrer a lista circularmente, a página 0 tem $R=1$, então seu bit R é zerado e ela é movida para o final da lista. A próxima página é a 1, com $R=1$. O mesmo processo é repetido até encontrar uma página com $R=0$. A página 2 tem $R=0$, então ela será substituída.

d) Substituição da menos utilizada recentemente (LRU):

Neste algoritmo, a página que foi menos referenciada (ou seja, a que possui o maior ciclo de transferência ou ciclo da última referência) será substituída

. Analisando a tabela, a página com o maior ciclo de transferência é a página 4. Portanto, a moldura a ser substituída pelo algoritmo LRU seria a moldura 4.

Resumo das molduras a serem substituídas:

- a) Substituição de página não usada recentemente (NRU): Moldura 2
- b) Primeira a entrar primeira a sair (FIFO): Moldura 0
- c) Substituição com segunda tentativa I (Second Chance): Moldura 2
- d) Substituição da menos utilizada recentemente (LRU): Moldura 4

2.O sistema operacional XYZ utiliza paginação como mecanismo de gerência de memória. São utilizadas páginas de 1KByte. Um endereço lógico utiliza 20 bits. Um endereço físico ocupa 24 bits. Cada entrada na tabela de páginas contém, além do número da página física, um bit de válido/inválido e um bit que indica apenas leitura (read only). Mostre como podem ser calculados os seguintes valores:

- a) Tamanho máximo para a memória física
- b) Maior programa que o sistema suporta.
- c) Quantas entradas possui a tabela de páginas.
- d) Quantos bits são necessários para a tabela de páginas (cálculo exato)

Resolução:

Para calcular os valores solicitados, vamos utilizar as informações fornecidas:

Tamanho da página: 1 KByte = 2^{10} bytes

Endereço lógico: 20 bits

Endereço físico: 24 bits

a) Tamanho máximo para a memória física:

O tamanho máximo para a memória física pode ser obtido a partir do número de bits do endereço físico. Neste caso, o endereço físico ocupa 24 bits, o que significa que o sistema operacional XYZ pode suportar uma memória física de tamanho máximo de 2^{24} bytes, ou seja, 16 MB.

b) Maior programa que o sistema suporta:

O tamanho máximo para um programa é limitado pelo tamanho da memória lógica disponível. Como o endereço lógico utiliza 20 bits, podemos representar até 2^{20} endereços diferentes. Considerando que cada endereço corresponde a uma página de 1 KByte, o maior programa

que o sistema operacional XYZ suporta é de $2^{20} * 2^{10}$ bytes, ou seja, 2^{30} bytes, que é igual a 1 GB.

c) Quantas entradas possui a tabela de páginas:

A tabela de páginas armazena informações sobre as páginas presentes na memória física. Cada entrada na tabela de páginas contém o número da página física, um bit de válido/inválido e um bit que indica apenas leitura. Portanto, o número de entradas na tabela de páginas é igual ao número de páginas lógicas.

O número de páginas lógicas pode ser calculado dividindo o espaço de endereçamento lógico pelo tamanho da página:

Número de páginas lógicas = 2^{20} (espaço de endereçamento lógico) / 2^{10} (tamanho da página) = $2^{10} = 1024$

Portanto, a tabela de páginas possui 1024 entradas.

d) Quantos bits são necessários para a tabela de páginas (cálculo exato):

Cada entrada na tabela de páginas contém o número da página física, um bit de válido/inválido e um bit que indica apenas leitura. Portanto, para calcular quantos bits são necessários para a tabela de páginas, devemos somar o número de bits necessários para representar cada um desses elementos.

Número da página física: Precisamos de bits suficientes para representar 1024 páginas (2^{10}). Portanto, são necessários 10 bits.

Bit de válido/inválido: É necessário 1 bit para representar o estado válido/inválido.

Bit que indica apenas leitura: É necessário 1 bit para representar o estado de apenas leitura.

Total de bits necessários por entrada: 10 (número da página física) + 1 (válido/inválido) + 1 (apenas leitura) = 12 bits.

Como a tabela de páginas possui 1024 entradas, o número total de bits necessários para a tabela de páginas é:

Total de bits = 1024 (número de entradas) * 12 (bits por entrada) = 12,288 bits.

Portanto, são necessários 12,288 bits para a tabela de páginas.

- 1) Considere um espaço de endereçamento lógico de 8 páginas de 1K cada, mapeados em uma memória física de 32 frames.

- a) Quantos bits tem o endereço lógico?
- b) Quantos bits tem o endereço físico?

Resolução:

- a) Quantos bits tem o endereço lógico:

Se temos 8 páginas e cada página tem 1K de tamanho, precisamos de $\log_2(8) + \log_2(1024) = 3 + 10 = 13$ bits para endereçar cada página. Portanto, o endereço lógico tem 13 bits.

- b) Quantos bits tem o endereço físico:

Se temos 32 frames na memória física, precisamos de $\log_2(32) + \log_2(1024) = 5 + 10 = 15$ bits para endereçar cada frame. Portanto, o endereço físico tem 15 bits.