

# Sistemas Operacionais I

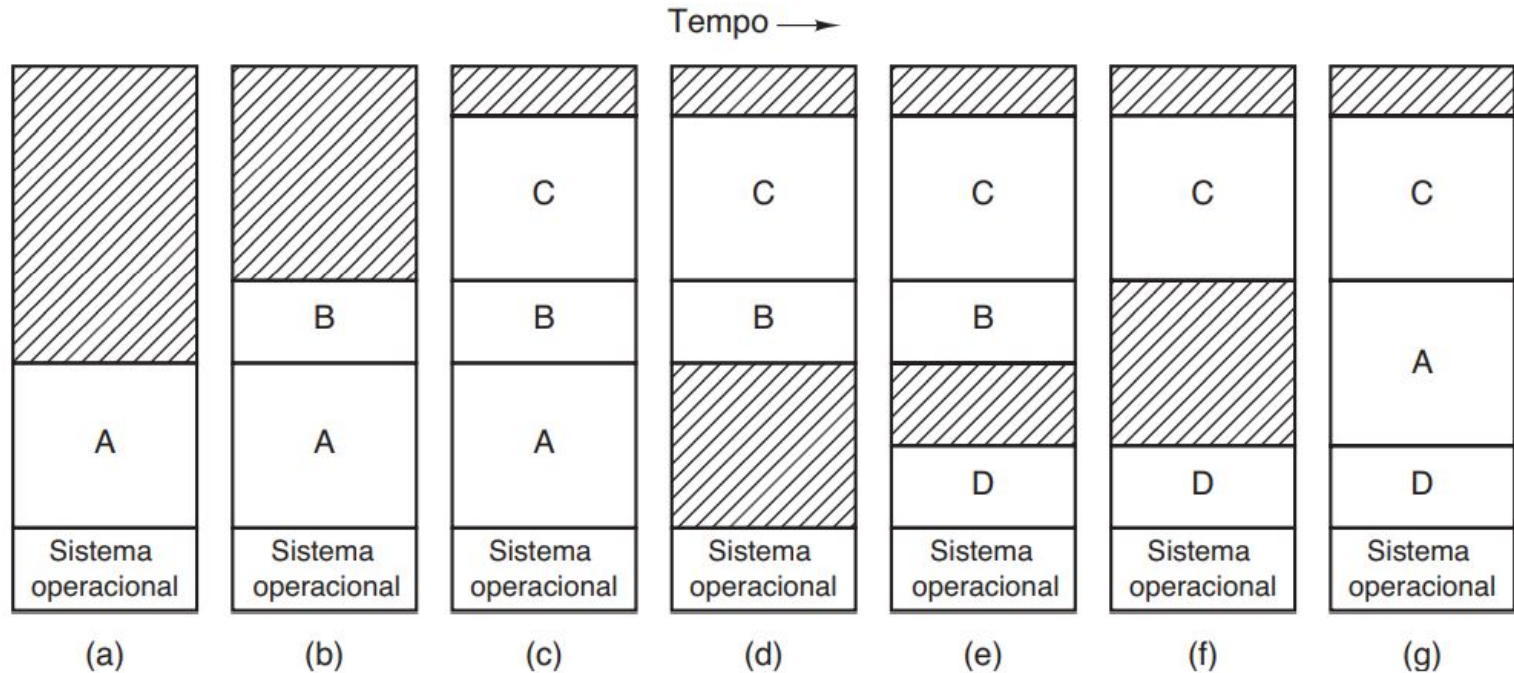




# Aula passada

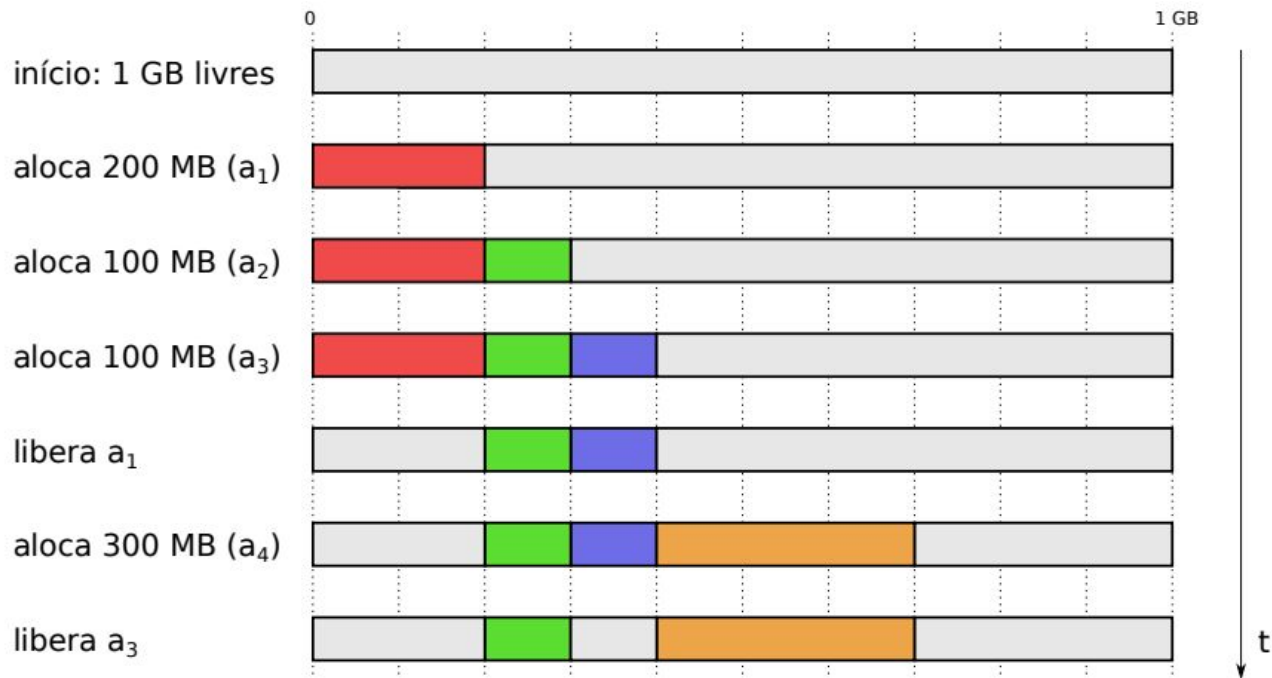
- Alocação de Memória Contígua;
  - Proteção da Memória;
  - Alocação de Memória;
  - Fragmentação;
  - Fragmentação.

# Alocação de Memória Contígua





# Alocação Básica





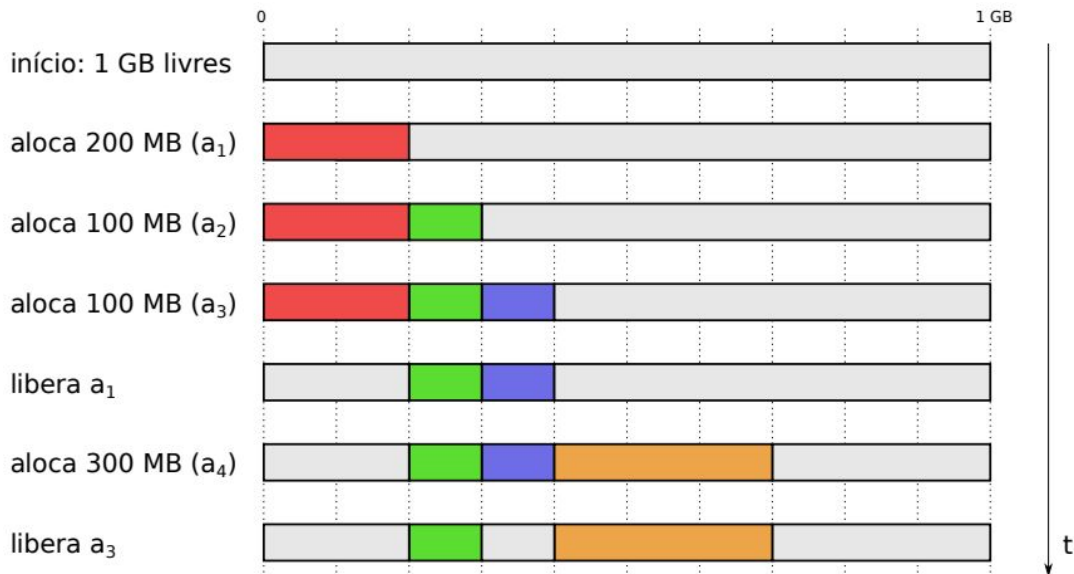
# Estratégias de Alocação

Há muitas soluções para esse problema de fragmentação externa.

As estratégias do:

- **Primeiro-apto** (*first-fit*);
- **Mais-apto** (*best-fit*);
- **Menos-apto** (*worst-fit*).

São as mais usadas para selecionar uma **brecha** livre no conjunto de **brechas** disponíveis.

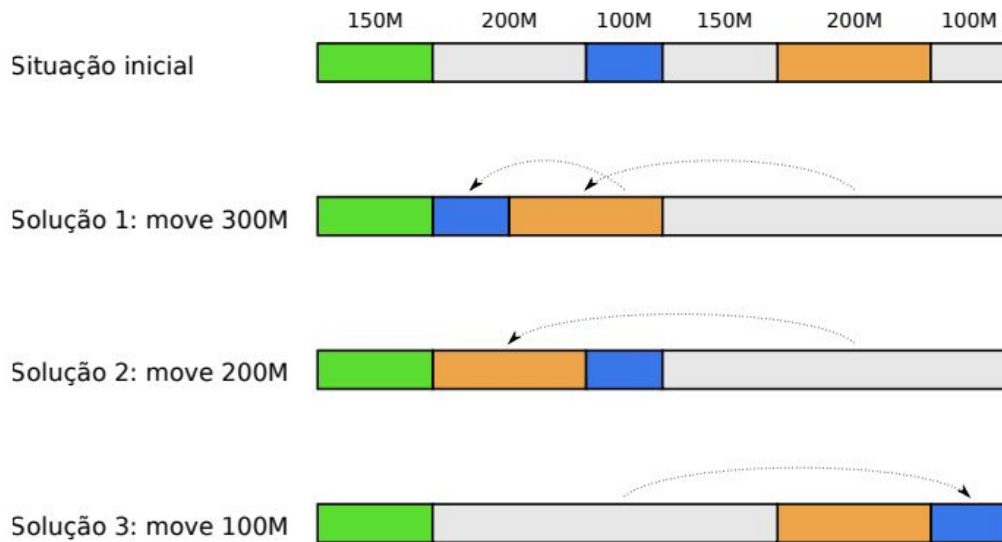




# Desfragmentação

Outra forma de tratar a **fragmentação externa** consiste em **desfragmentar** a memória periodicamente.

Como as áreas de memória **não podem ser acessadas** durante a **desfragmentação**, é importante que esse procedimento seja executado **rapidamente e com pouca frequência**, para não interferir nas atividades normais do sistema.



Possibilidades de desfragmentação

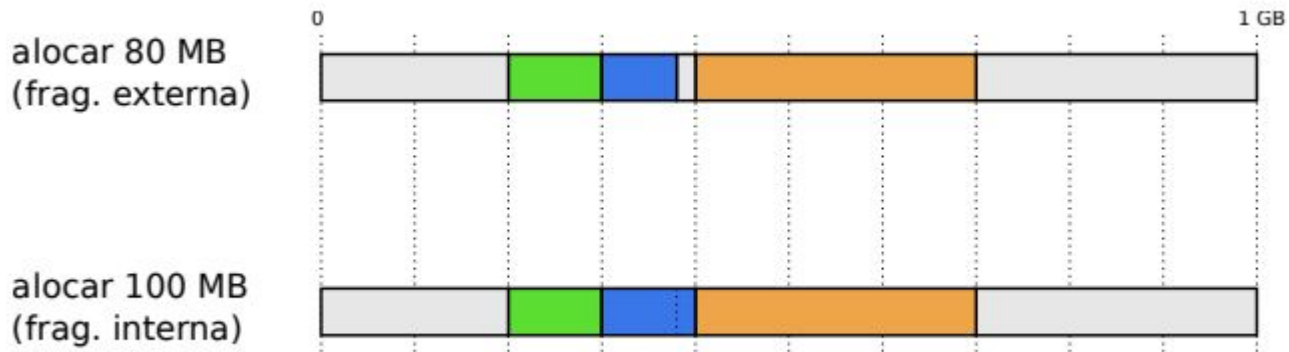


# Fragmentação interna

Uma alternativa para minimizar o impacto da **fragmentação externa** consiste em arredondar algumas requisições de alocação, para evitar sobras muito pequenas.

Por exemplo, na alocação com *best-fit*, a área alocada poderia ser arredondada de 80 MB para 100 MB, evitando a sobra da área de 20 MB.

Dessa forma, evita-se a geração de um fragmento de memória livre, mas a memória adicional alocada provavelmente não será usada por quem a requisitou. **Esse desperdício de memória dentro da área alocada é denominado fragmentação interna** (ao contrário da fragmentação externa, que ocorre nas áreas livres).



# Agenda

- Segmentação;
- Paginação;

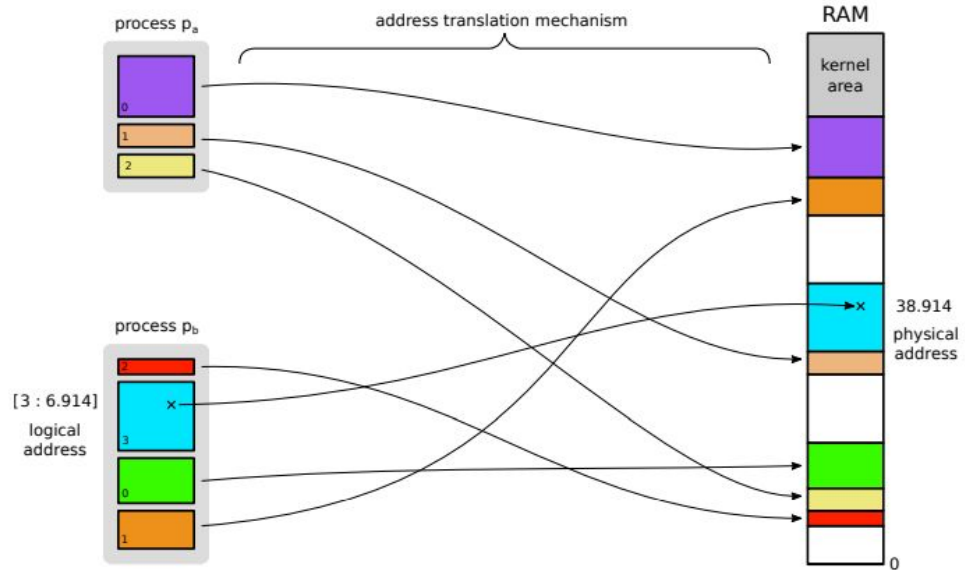


# Segmentação



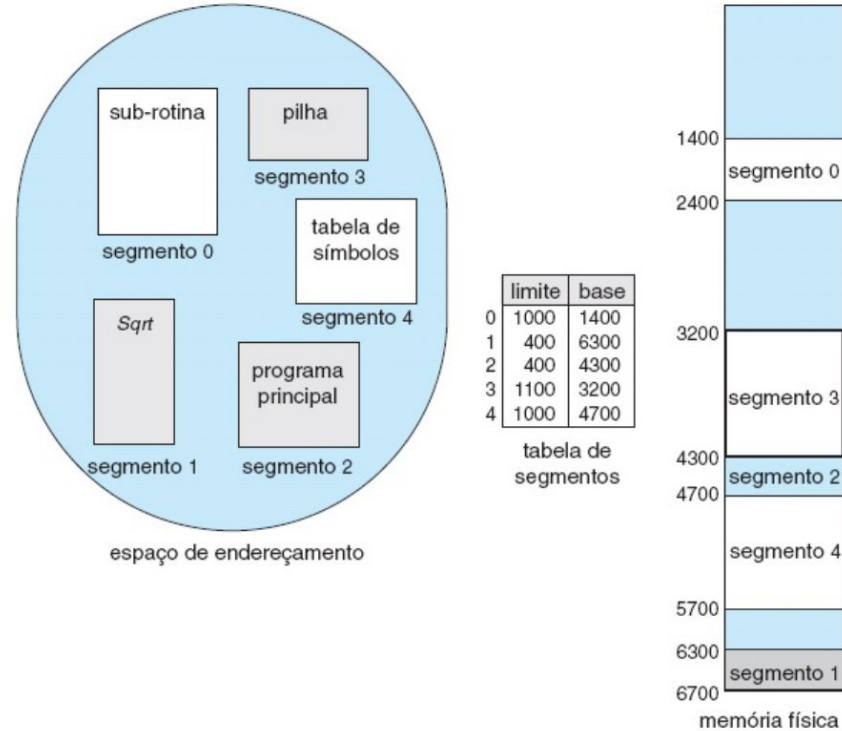
# Segmentação

- A segmentação permite que o espaço de endereçamento físico de um processo seja **não contíguo**.
- A memória como um **conjunto de segmentos** de tamanho variável, sem que haja necessariamente uma ordem entre eles.



Esquema de partições **por segmentos**.

# Segmentação



Exemplo de segmentação.

# Paginação





# Paginação

A segmentação permite que o espaço de **endereçamento físico** de um processo seja não contíguo. A **paginação** é outro esquema de gerenciamento da memória que oferece essa vantagem.

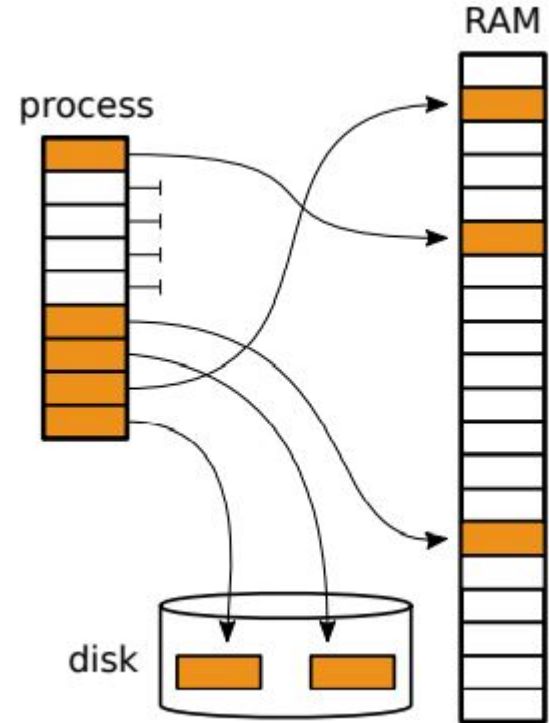
No entanto, a **paginação** evita a **fragmentação externa** e a necessidade de compactação, enquanto a segmentação não faz isso.

Ela também resolve o considerável problema de acomodar trechos de memória de vários tamanhos na memória de retaguarda.

# Paginação

- Usar dispositivos de **armazenamento secundário** como extensão da memória RAM;
- Partes **ociosas** da memória podem ser **transferidas** para um **disco**, liberando a memória RAM para outros usos;
- Caso algum **processo** tente **acessar** esse **conteúdo** posteriormente, ele deverá ser trazido de **volta à memória RAM**;
- A **transferência** dos **dados** entre **memória** e **disco** é feita pelo **sistema operacional**, de forma **transparente** para os **processos**;

A ideia central da **paginação** em disco consiste em transferir páginas ociosas da memória RAM para uma área em disco, liberando memória para outras páginas.





# Método Básico de Paginação

O **método básico** para a implementação da paginação envolve:

- A. A fragmentação da **memória física** em blocos de tamanho fixo chamados **quadros**
- B. E a fragmentação da **memória lógica** em blocos do mesmo tamanho chamados **páginas**.

Quando um **processo** está para ser **executado**, suas **páginas** são carregadas em quaisquer **quadros** de memória disponíveis a partir de sua origem (um sistema de arquivos ou a memória de retaguarda).

Essa ideia bastante simples tem grande funcionalidade e amplas ramificações.

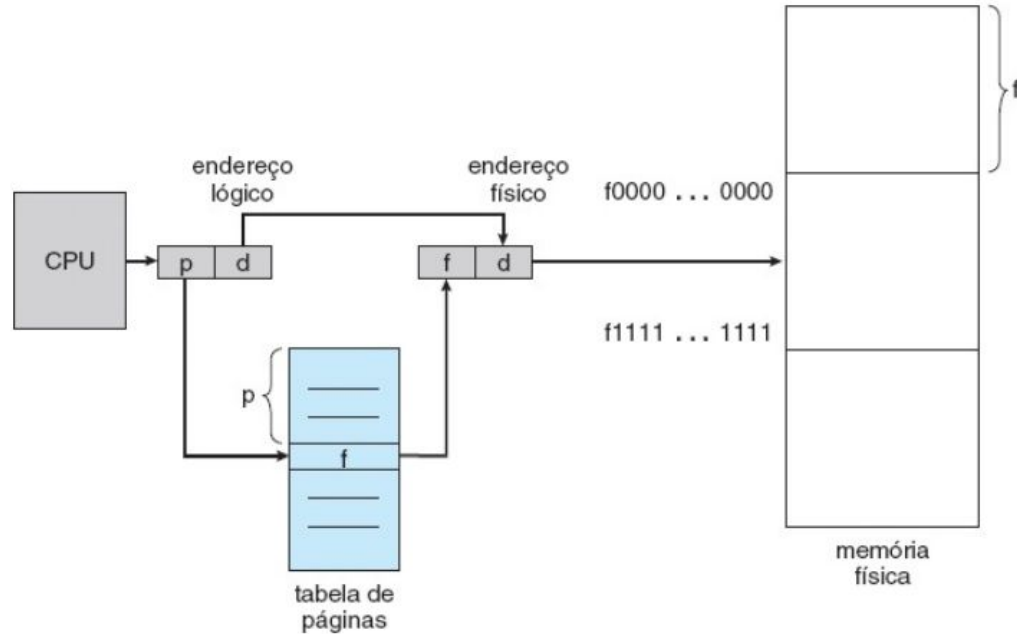
# Suporte de Hardware à Paginação

Cada endereço gerado pela CPU é dividido em duas partes: um **número de página** (p) e um **deslocamento de página** (d).

O **número de páginas** é usado como índice em uma **tabela de páginas**.

A **tabela de páginas** contém o endereço base de cada página na memória física.

Esse **endereço base** é combinado com o **deslocamento de página** para definir o **endereço de memória física** que é enviado à unidade de memória.





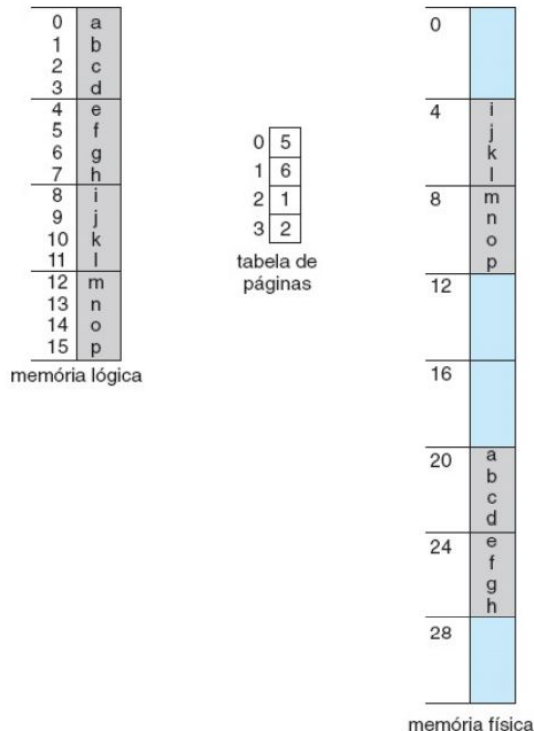


# Suporte de Hardware à Paginação

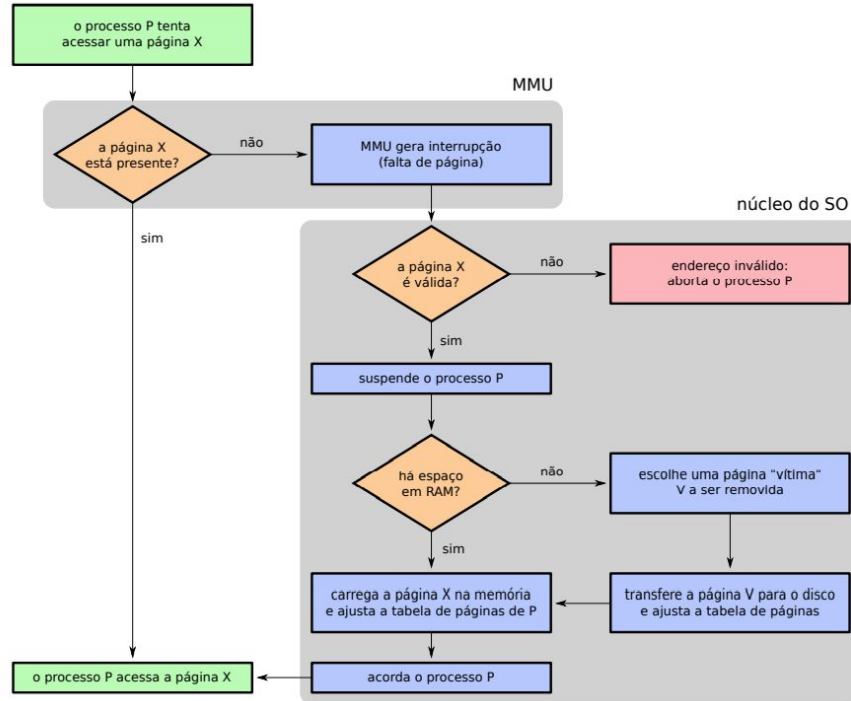
Por exemplo, considere a memória da figura ao lado.

Usando um tamanho de **página** de 4 bytes e uma memória física de 32 bytes (8 páginas), mostramos como a visão que o programador tem da memória pode ser mapeada para a memória física.

- A. O endereço lógico 0 é a página 0, deslocamento 0. Indexando na tabela de páginas, vemos que a página 0 está no quadro 5. Logo, o endereço lógico 0 é mapeado para o endereço físico 20 [=  $(5 \times 4) + 0$ ].
- B. O endereço lógico 3 (página 0, deslocamento 3) é mapeado para o endereço físico 23 [=  $(5 \times 4) + 3$ ].
- C. O endereço lógico 4 é a página 1, deslocamento 0; de acordo com a tabela de páginas, a página 1 é mapeada para o quadro 6. Consequentemente, o endereço lógico 4 é mapeado para o endereço físico 24 [=  $(6 \times 4) + 0$ ].
- D. O endereço lógico 13 é mapeado para o endereço físico 9.



# Suporte de Hardware à Paginação





# Critérios de Seleção

**Idade da página:** Há quanto tempo a página está na memória; páginas muito antigas talvez sejam pouco usadas.

**Frequência de acessos à página:** páginas mais acessadas em um passado recente possivelmente ainda o serão em um futuro próximo.

**Data do último acesso:** páginas há mais tempo sem acessar possivelmente serão pouco acessadas em um futuro próximo (sobretudo se os processos respeitarem o princípio da localidade de referências).

**Prioridade do processo proprietário:** processos de alta prioridade, ou de tempo real, podem precisar de suas páginas de memória rapidamente; se elas estiverem no disco, seu desempenho ou tempo de resposta poderão ser prejudicados.

**Conteúdo da página:** páginas cujo conteúdo seja código executável exigem menos esforço do mecanismo de paginação, porque seu conteúdo já está mapeado no disco (dentro do arquivo executável correspondente ao processo). Por outro lado, páginas de dados que tenham sido alteradas precisam ser salvas na área de troca.

**Páginas especiais:** páginas contendo buffers de operações de entrada/saída podem ocasionar dificuldades ao núcleo caso não estejam na memória no momento em que ocorrer a transferência de dados entre o processo e o dispositivo físico. O processo também pode solicitar que certas páginas contendo informações sensíveis (como senhas ou chaves criptográficas) não sejam copiadas na área de troca, por segurança.