

Sistemas Operacionais

- Alocação de Memória
- Fragmentação



Alocação de Memória

Como ocorre com os demais recursos de hardware, a memória disponível no sistema deve ser gerenciada pelo SO, **para evitar conflitos entre aplicações e garantir justiça no seu uso.**

O sistema operacional aloca e libera áreas de memória para os processos (ou para o próprio núcleo), conforme a necessidade.

Alocação de Memória

Alocar memória significa **reservar áreas de memória RAM** que podem ser usadas por um processo, por um descritor de *socket*, arquivo no núcleo, por um cache de blocos de disco, etc.

No final de seu uso, cada área de memória alocada é liberada pela entidade que a solicitou e colocada à disposição do sistema para novas alocações.

Alocação de Memória

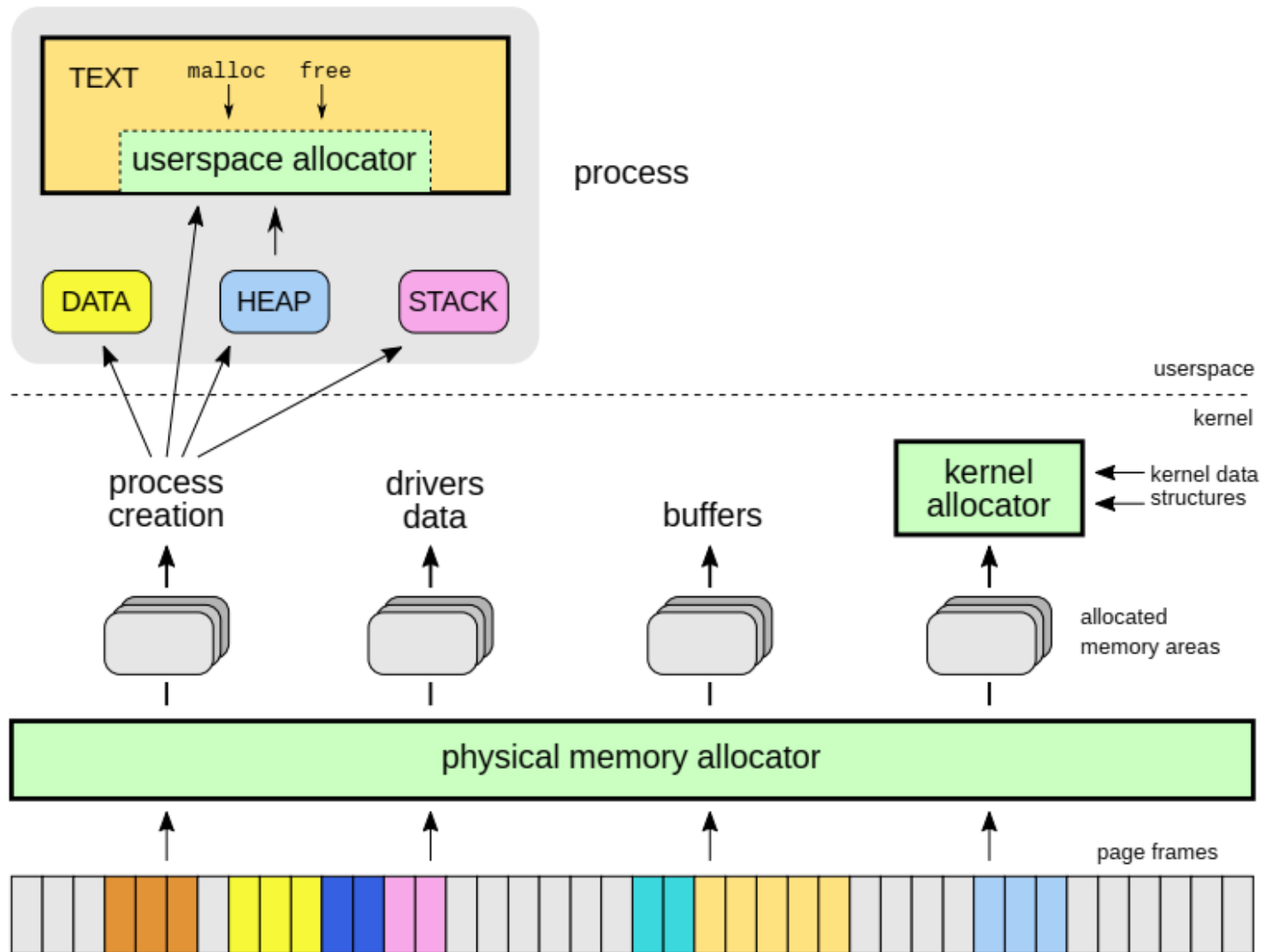
O mecanismo responsável pela alocação e liberação de áreas de memória é chamado um ***alocador de memória***.

Alocador de memória física : organiza a memória física do computador (DMA, BIOS, etc)

Alocador de espaço de núcleo: obtém áreas de memória do alocador físico e as utiliza para alocar essas estruturas para o núcleo (processos, *sockets* de rede, *pipes*).

Alocador de espaço de usuário: um processo pode solicitar blocos de memória para armazenar estruturas de dados dinâmicas (bibliotecas providas pelo sistema operacional)

Alocação de Memória



Alocação de Memória

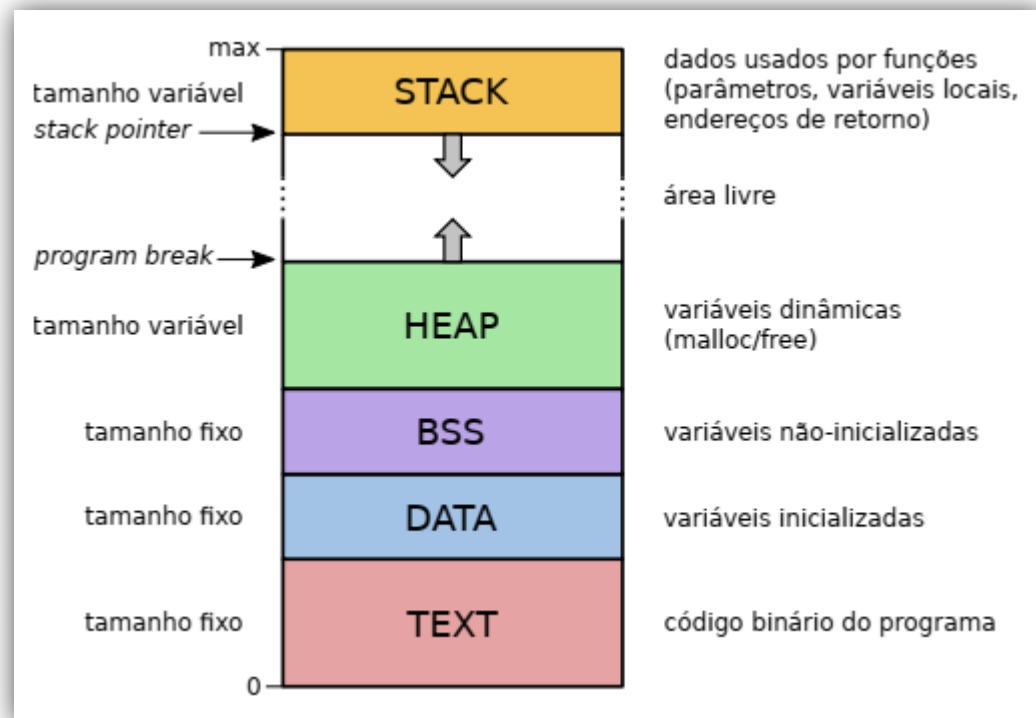
TEXT: contém o código binário a ser executado pelo processo.

DATA: esta seção contém as variáveis estáticas inicializadas.

BSS: *Block Started by Symbol*, esta seção contém as variáveis estáticas não-inicializadas.

HEAP: esta seção é usada para armazenar variáveis alocadas dinamicamente.

STACK: esta seção é usada para manter a pilha de execução do processo.



No linux pode ser usado o `pmap <pid>`

Alocação de Memória

Problema:

Como gerenciar uma grande área de memória RAM e atender um fluxo de requisições de **alocação e liberação** de partes dessa área para o sistema operacional ou para as aplicações de forma eficiente?

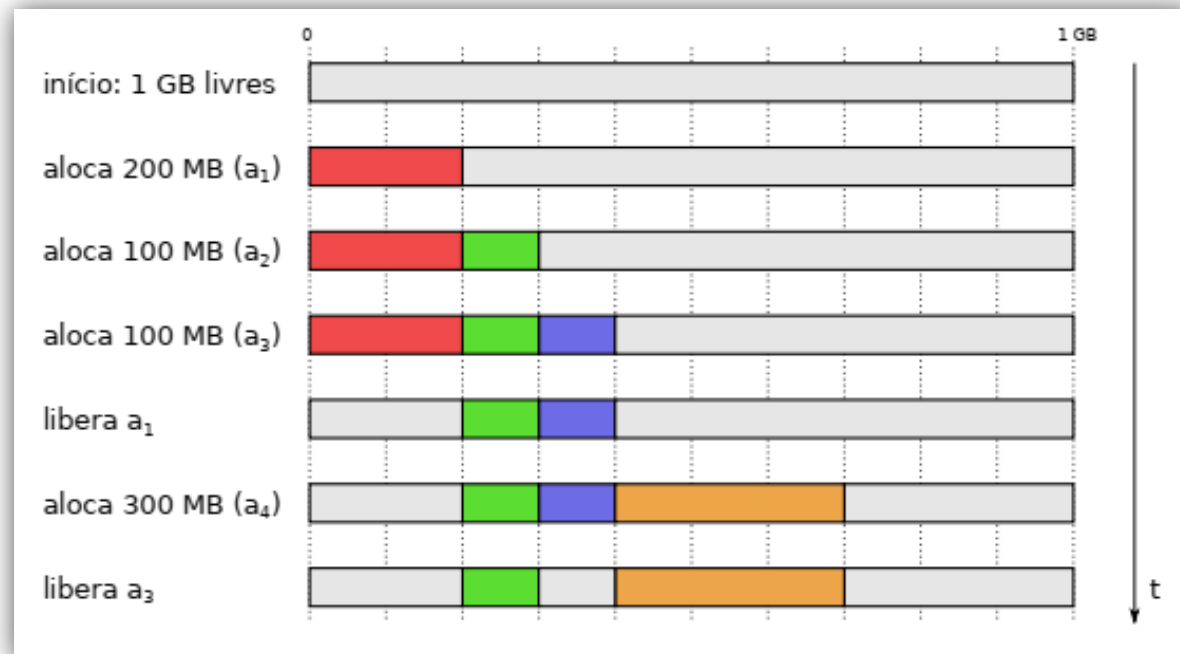
Essas requisições **ocorrem o tempo todo**, em função das tarefas em execução no sistema, e devem ser atendidas rapidamente.



Alocação de Memória

Considere um sistema hipotético com 1 GB de memória RAM livre em uma área única:

1. Aloca 200 MB (a_1)
2. Aloca 100 MB (a_2)
3. Aloca 100 MB (a_3)
4. Libera a_1
5. Aloca 300 MB (a_4)
6. Libera a_3



Fragmentação

Fragmentação são áreas livres de memória disponíveis entre as áreas alocadas (buracos) de difícil utilização posterior.

Isso ocorre devido as áreas de memória serem alocadas e liberadas continuamente.

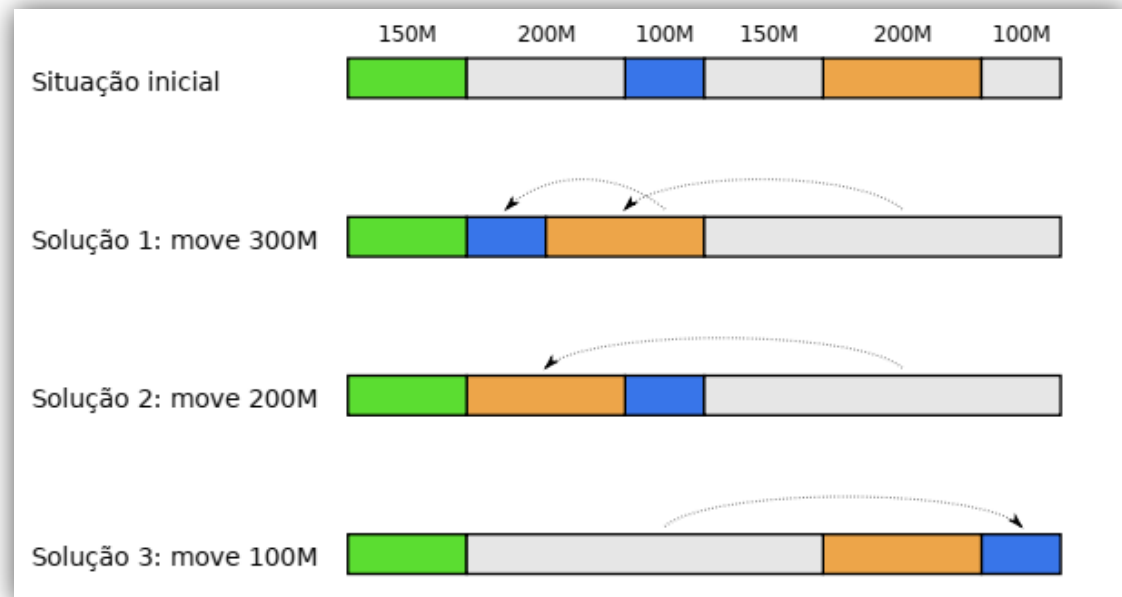
A fragmentação pode ser classificada em;

- **Fragmentação Externa:** fragmenta a memória livre
- **Fragmentação Interna:** sobra de alocação de reserva.
(Devido a área alocada pode ser arredondada - ***best-fit***)

DES Fragmentação

As áreas de memória usadas pelos processos devem ser movidas na memória de forma a **concatenar as áreas livres**

Ao mover um processo na memória, suas informações de endereçamento virtual (registrador base/limite, tabela de segmentos ou de páginas) devem ser devidamente ajustadas para refletir a nova posição do processo na memória RAM.



Estratégias de Alocação

First-fit (primeiro encaixe):

- Consiste em escolher a primeira área livre que satisfaça o pedido de alocação. (Rapidez)

Best-fit (melhor encaixe):

- Consiste em escolher a menor área que possa receber a alocação (minimiza o desperdício. Algumas áreas livres podem ficar inúteis. *Pode ocorrer a Fragmentação Interna*)

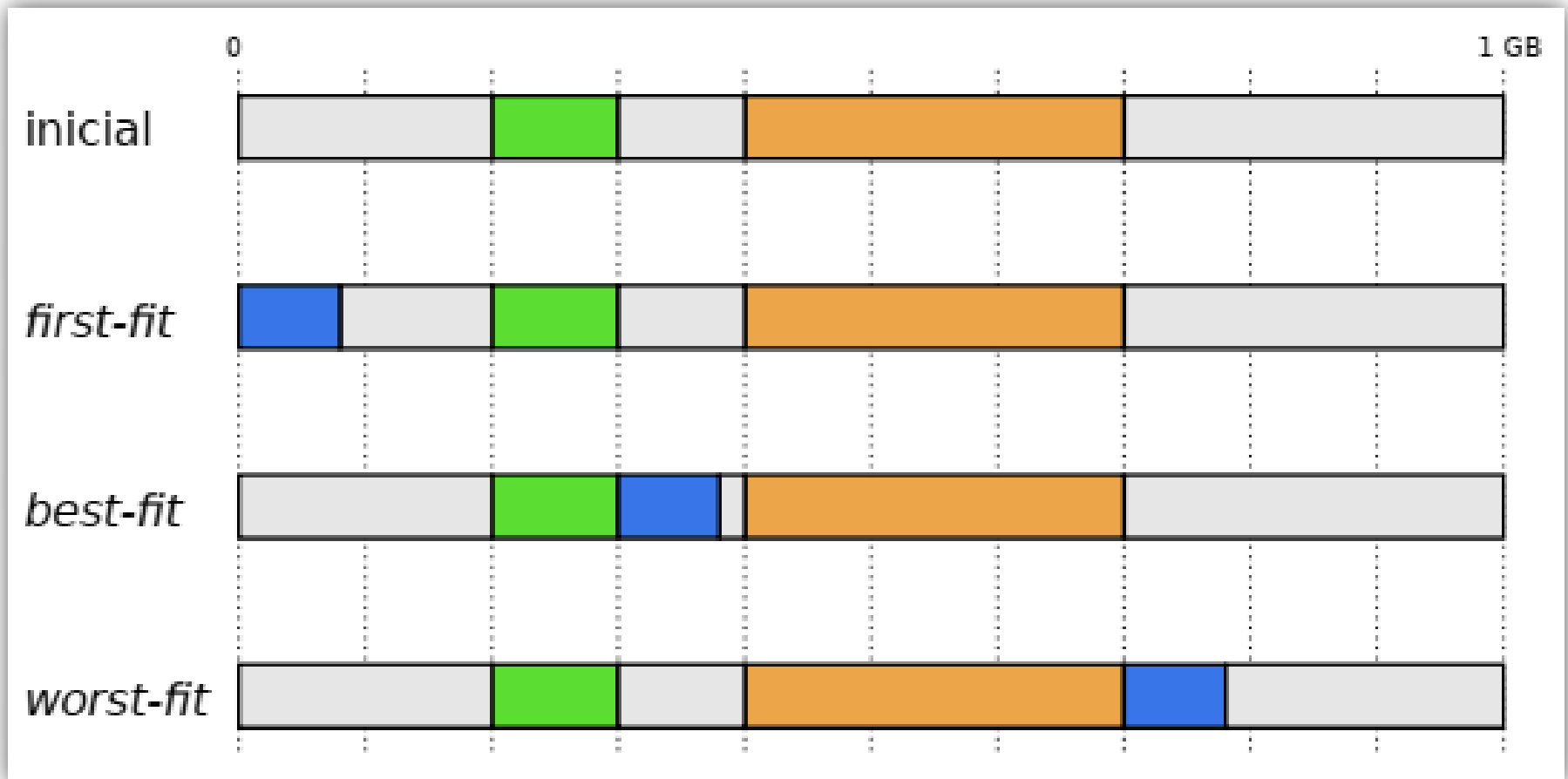
Worst-fit (pior encaixe):

- Consiste em escolher sempre a maior área livre possível, de forma que a “sobra” seja grande o suficiente para ser usada em outras alocações.

Next-fit (próximo encaixe):

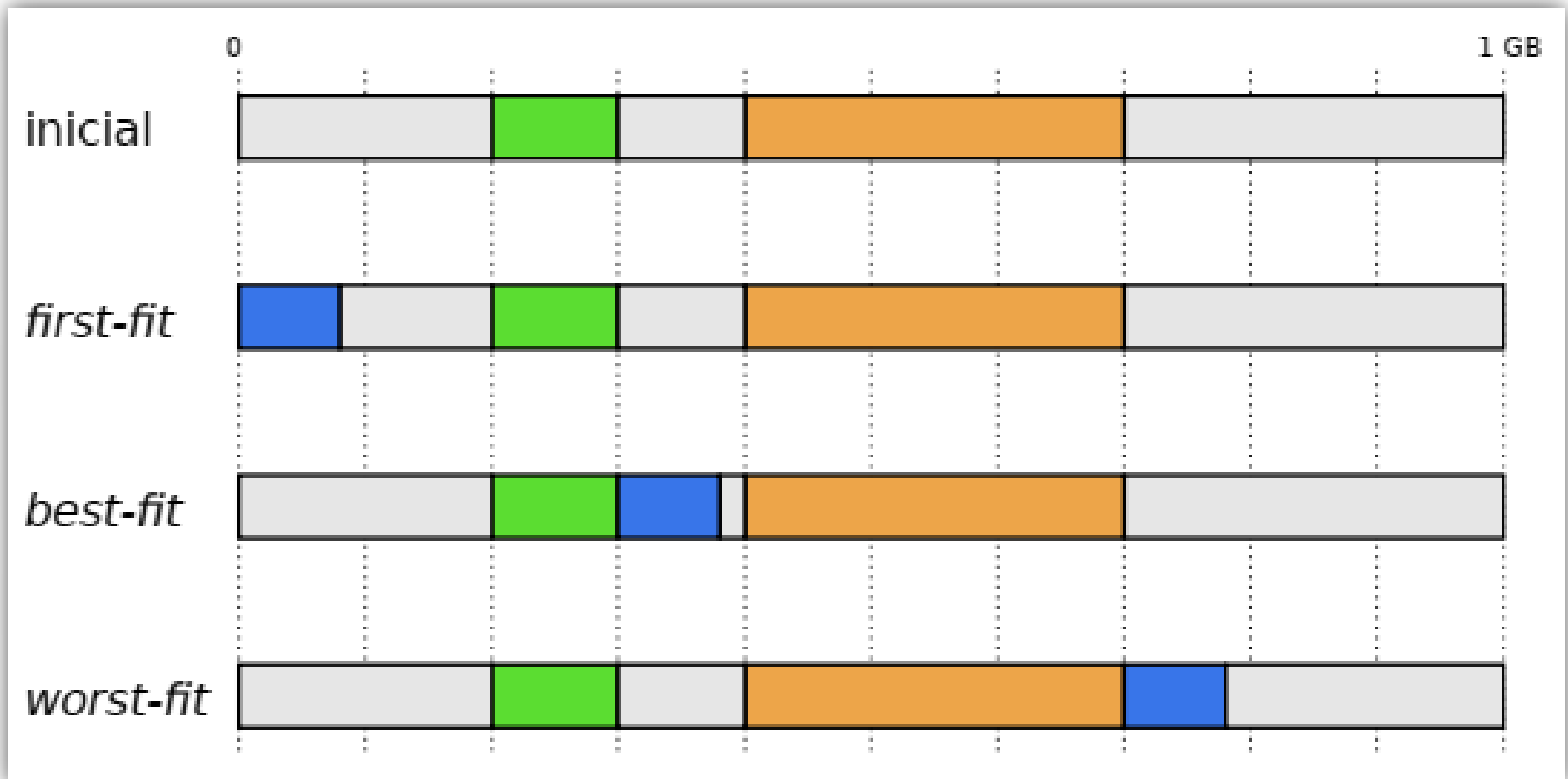
- Percorre a lista de áreas a partir da última área alocada ou liberada (distribui de forma mais homogênea).

Estratégias de Alocação



Estratégias na alocação de um bloco de 80 MB em uma memória de 1 GB parcialmente alocada.

Estratégias de Alocação



Estratégias na alocação de um bloco de 80 MB em uma memória de 1 GB parcialmente alocada.

Outras Estratégias de Alocação

- **O alocador Buddy**

- Em sua versão mais simples, a estratégia *Buddy* sempre aloca blocos de memória de tamanho 2^n
 - **Exemplo:** Para uma requisição de 85 KBytes será alocado um bloco com 128 KBytes (2^7 Kbytes e assim por diante).
 - Caso não encontre um bloco com 128 KBytes, procura um bloco livre com 256 KBytes, o divide em dois blocos de 128 KBytes (***os buddies***) e usa um deles para a alocação.
 - Quando liberar, verifica se o par (***buddy***) do bloco também está livre; se estiver, funde os dois em um bloco maior, sucessivamente .

Outras Estratégias de Alocação

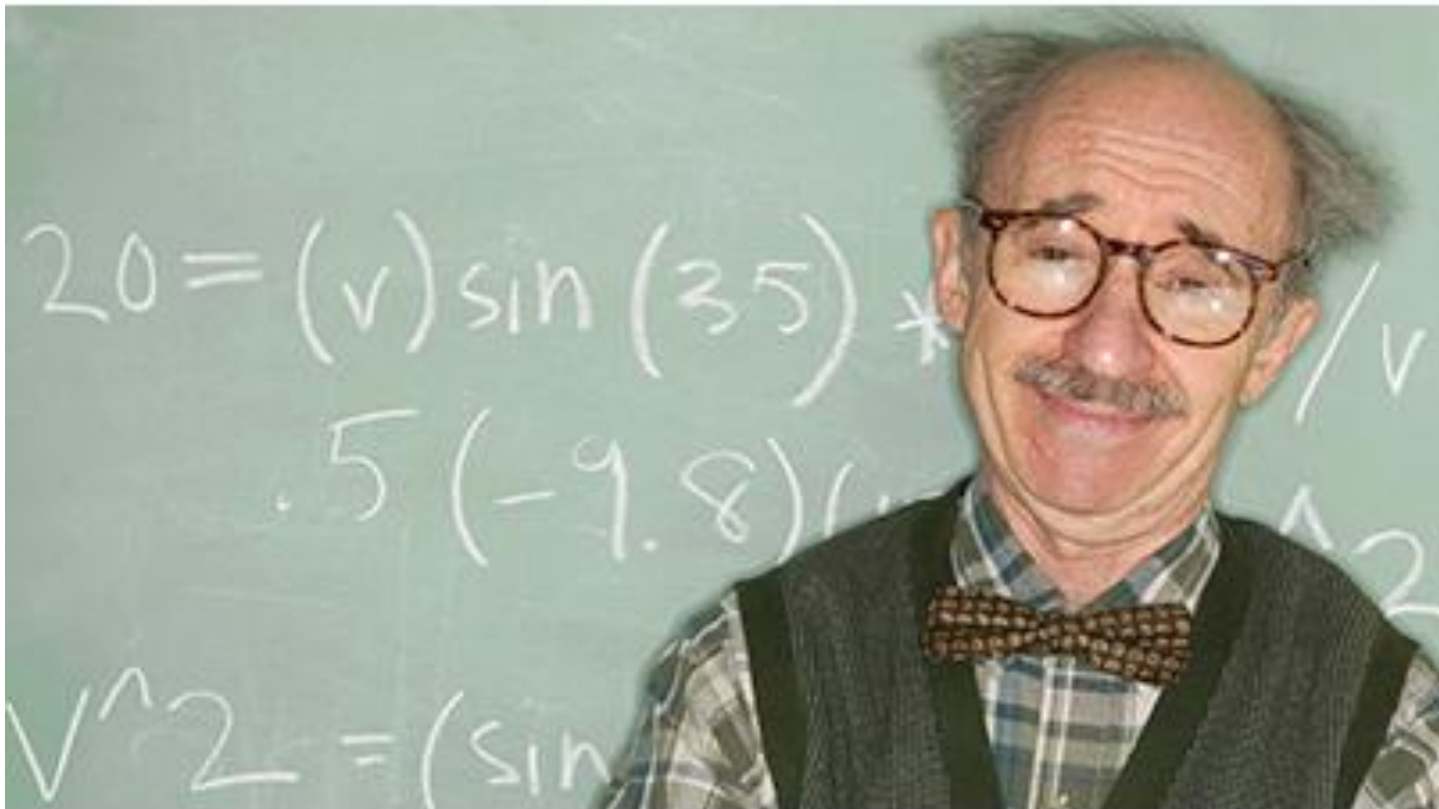
- **O alocador Slab**

- Foco na alocação de “**objetos de núcleo**”: estruturas de dados que são usadas para representar descritores de processos, de arquivos abertos, *sockets* de rede, *pipes*, etc (poucos dados).
- Utiliza uma estratégia baseada no **caching** de objetos.
 - Divide em subconjuntos (por tipo) – **slabs**
 - Quando um novo objeto de núcleo é requisitado, localiza um slab parcial; caso não haja slabs parciais, entrega um objeto livre de um slab vazio.
 - Caso não existam **slabs** vazios, o alocador pede mais páginas de RAM e marca como livres.
 - Quando um objeto é liberado, ele é marcado como livre;
 - Caso o sistema precise liberar memória para outros usos, o alocador pode descartar os slabs vazios (todos livres),

Outras Estratégias de Alocação

- **Alocação no espaço de usuário**
 - Normalmente fica a cargo de bibliotecas de sistema (*como a biblioteca C padrão (LibC), que oferecem funções básicas de alocação de memória como malloc, realloc e free*).
 - Existem várias implementações de alocadores de uso geral para o espaço de usuário. As implementações mais simples seguem o esquema *best-fit*.

Exercícios



Atividade 01

- Explique o que é *fragmentação externa* e o que é *fragmentação interna*.

Atividade 02

Considere um alocador de memória do tipo *first-fit*.

Considere as estratégias de alocação dada uma área contínua de memória RAM com 1 GByte (1.024 MBytes).

Apresente a evolução da situação da memória para a sequência de alocações e liberações de memória indicadas a seguir.

- (a) Aloca A1 200 MB
- (b) Aloca A2 100 MB
- (c) Aloca A3 150 MB
- (d) Libera A2
- (e) Libera A1
- (f) Aloca A4 100 MB
- (g) Aloca A5 40 MB
- (h) Aloca A6 300 MB

Atividade 03

Considere um alocador de memória do tipo *Best-fit*.

Considere as estratégias de alocação dada uma área contínua de memória RAM com 1 GByte (1.024 MBytes).

Apresente a evolução da situação da memória para a sequência de alocações e liberações de memória indicadas a seguir.

- (a) Aloca A1 200 MB
- (b) Aloca A2 100 MB
- (c) Aloca A3 150 MB
- (d) Libera A2
- (e) Libera A1
- (f) Aloca A4 100 MB
- (g) Aloca A5 40 MB
- (h) Aloca A6 300 MB

Atividade 04

Considere um alocador de memória do tipo *Worst-fit*.

Considere as estratégias de alocação dada uma área contínua de memória RAM com 1 GByte (1.024 MBytes).

Apresente a evolução da situação da memória para a sequência de alocações e liberações de memória indicadas a seguir.

- (a) Aloca A1 200 MB
- (b) Aloca A2 100 MB
- (c) Aloca A3 150 MB
- (d) Libera A2
- (e) Libera A1
- (f) Aloca A4 100 MB
- (g) Aloca A5 40 MB
- (h) Aloca A6 300 MB

Atividade 05

Considere um alocador de memória do tipo *Next-fit*.

Considere as estratégias de alocação dada uma área contínua de memória RAM com 1 GByte (1.024 MBytes).

Apresente a evolução da situação da memória para a sequência de alocações e liberações de memória indicadas a seguir.

- (a) Aloca A1 200 MB
- (b) Aloca A2 100 MB
- (c) Aloca A3 150 MB
- (d) Libera A2
- (e) Libera A1
- (f) Aloca A4 100 MB
- (g) Aloca A5 40 MB
- (h) Aloca A6 300 MB