**Fragmentação da memória**

Externa: pequenos pedaços de memória livre que sobram entre os blocos de memória alocadas. Mesmo havendo memória livre suficiente, a quantidade de memória requisitada não pode ser alocada pois não há quantidade contígua de memória.

Interna: desperdício de memória ao alocar a memória com espaços maiores do que o necessário.

**Métodos de ajuste**

Ajuste sequencial: lista de blocos disponíveis, busca sequencial na lista de blocos pelo mais "adequado" para alocação.

Ajuste não sequencial: área disponível de forma estruturada, decisão sobre o bloco de alocação com critérios mais eficientes.

**Ajuste Sequencial**:

Quando a alocação é realizada, é feito uma busca para verificar qual o melhor bloco para fazer a alocação.

Critérios de ajuste para alocação de blocos:

- Primeiro ajuste (first fit): escolha da primeira área livre ordenada por endereço que atende á solicitação.

- Melhor ajuste (best fit): escolha da área livre com tamanho mais próximo ao da solicitação. (Fragmentação externa grande) (ordenação por tamanho)

- Pior ajuste (worst fit): escolha do bloco de maior tamanho para atender á solicitação. (ordenação por tamanho)

- Próximo ajuste (next fit): escolha da próxima área livre (pode-se montar uma lista circular)

Todas com o tempo irão gerar uma fragmentação externa, mas não vão gerar fragmentação interna pois elas alocam a porção exata da requisição. Nenhum deles é "O Melhor", suas filosofias são diferentes. É tudo uma questão de perspectiva.

(exemplo dos ajustes sequenciais no slide do prof.)

**Ajuste não sequencial**

Organização geral: uma estrutura maior do que uma lista de encadeamento de blocos livres.

Uso de várias listas, cada lista com um tamanho predefinido de bloco. Pega a memória, separa ela em tamanhos padrões que são julgados como usuais para a alocação.

Uso de várias listas, cada uma com um tamanho específico, ou seja, uma lista com tamanho 10, outra com tamanho 20, por assim vai. Porém o tamanho dos blocos vão alterando de acordo com os tamanhos que são mais requisitados, com o tempo ele vai aprendendo quais os melhores tamanhos.

Divisão de blocos de memória em pares associados.

**Lista de tamanho predefinidos**:

Divisão de memória em um números arbitrário de listas.

Cada lista tem blocos de um único tamanho.

Blocos maiores podem ser divididos em blocos menores, gerado novas listas.

Quando há duas desalocações sequenciais, os blocos não se juntam, pois cada bloco pertence à sua lista. Problema é que as listas vão mudando e alguns blocos podem ir se "perdendo" e nunca mais serem utilizados.

**Listas de ajuste exato adaptativo:**

Manter uma lista de valores de blocos liberados de tamanhos conhecidos. Porém ela é criada dinamicamente de acordo com os tamanhos de blocos que vão sendo requisitados, vai criando várias listas, cada uma com um valor de tamanho distinto, quando um tamanho requisitado não pertence a nenhuma das listas existentes, há uma alocação no final da memória com aquele tamanho e uma nova lista é criada. Quando a lista é parada de ser usada depois de um determinado tempo, há um processamento para avaliar se aquela lista não deve mais ser mantida, então o espaço da memória torna-se livre novamente.

**Sistemas amigos:**

Eu tenho o espaço total de memória que é considerado todo livre, para que o gerenciamento possa ser mais eficiente, é definido o termo bloco amigo. É preciso fazer uma alocação, divide-se o espaço da memória em 2 espaços amigos. É possível saber quem é amigo de quem apenas pelo endereço do bloco pois serão sempre potência de 2. Quando há uma liberação libera o bloco inteiro, se dois blocos amigos estiverem livres, junta-se os dois. Gera fragmentação interna, principalmente em situações críticas! Mesmo havendo espaço livre em blocos consecutivos, não é permitido a alocação (se houver 10GB, quebra-se em 2x5GB, não é mais possível alocar 6GB).

Uma parte do bloco é utilizada para manter controle: alocado/livre, tamanho, ponteiro da lista.

\*Existe um sistema chamado de amigos de Fibonacci, utilizando a sequencia de Fibonacci na alocação de memória, ao invés de ir quebrando a memória em blocos de potência de dois, utiliza-se Fibonacci\*

**Gerenciador de memória**

Função: alocar, desalocar blocos de memória.

Gerenciamento da área livre.

Gerenciamento de acesso a dados compartilhados.

Transferência de dados entre memória principal e secundária.

**Leitura básica:**

Drozdek (2002). Capítulo 12. Seções 12.1 e 12.2

**Leitura para próxima aula:**

Drozdek (2002). Capítulo 12: Seção 12.3