

Departamento de Engenharia Eletrônica e de Computação
Linguagens de Programação - EEL 670 – Prof. Flávio Mello
Trabalho ref. Módulo 14: Gerência de Memória

Entrega do trabalho: 04Out24, na chegada em sala de aula

Exercício 1) Faça download do arquivo `Exercise14.java`.

Neste arquivo encontra-se a `HeapManager` que implementa o mecanismo `first-fit`, tal como apresentado nos slides de aula. Ele utiliza o primeiro bloco da lista de segmentos livres em que seja possível alocar o tamanho de memória desejado. Outro mecanismo relativamente simples é o `best-fit`. A idéia é percorrer a lista de segmentos livres por um bloco suficientemente largo que seja o mais próximo possível do tamanho que se deseja alocar. Se existe um bloco na lista de segmentos livres que seja do tamanho exato desejado, o mecanismo de `best-fit` pode interromper a busca sem precisar completar todo seu percurso. Caso contrário, ele precisa percorrer toda a lista de segmentos livres. A vantagem deste procedimento é que ele não quebra blocos grandes desnecessariamente. Por exemplo, se existe um bloco de tamanho exato daquele desejado, então o mecanismo de `best-fit` irá encontrá-lo, e assim não será preciso quebrar nenhum bloco.

a) Implemente uma versão do `HeapManager` com o mecanismo `best-fit`. Comece com uma nova cópia da classe `HeapManager` com o nome alterado, e modifique o método `allocate` para implementar o `best-fit`.

b) Depois disto estar testado e funcionando, determine uma seqüência simples de operações para a qual o método `best-fit` obtem sucesso enquanto o `first-fit` falha. Dica: Existe uma seqüência que é iniciada por:

```
mm = new HeapManager2(new int[7]);  
int a = mm.allocate(2);  
int b = mm.allocate(1);  
int c = mm.allocate(1);  
mm.deallocate(a);  
mm.deallocate(c);
```

Extendendo a seqüência com apenas mais duas chamadas `mm.allocate`, você consegue uma configuração de sucesso para o `best-fit` e insucesso para o `first-fit`.

c) Apesar do `best-fit` em geral ser uma estratégia de locação melhor que o `first-fit`, existem configurações nas quais ele é pior. Determine uma seqüência simples de operações para a qual o método `first-fit` obtem sucesso enquanto o `best-fit` falha. Dica: Existe uma seqüência que é iniciada por:

```
mm = new HeapManager2(new int[11]);  
int a = mm.allocate(4);  
int b = mm.allocate(1);  
int c = mm.allocate(3);  
mm.deallocate(a);  
mm.deallocate(c);
```

Extendendo a seqüência com apenas mais três chamadas `mm.allocate`, você consegue uma configuração de sucesso para o `first-fit` e insucesso para o `best-fit`.