

Exercício

O objetivo do exercício é gerenciar uma agenda de amigos. Essa agenda deve armazenar um código (chave primária), o nome da pessoa e o telefone da mesma.

O arquivo a ser criado deve ser de registros e campos de tamanho variável, com um inteiro (4 bytes) no início do registro indicando o tamanho do registro, e com campos separados pelo caractere '|'.

Código	Nome	Telefone
int	char *	char *

Ex.: **211**|Veronica|5555-5555|

As seguintes operações deverão estar disponíveis:

1. Inserção de amigos
2. Remoção de amigos
3. Compactação do arquivo

Inserção de amigos (1)

Ao adicionar um amigo vocês terão que percorrer a lista de espaços disponíveis verificando se o novo registro se encaixa em algum dos espaços (vide Opção 2). Para tanto, usem a estratégia first-fit e, para facilitar, podem considerar fragmentação interna. Caso nenhum elemento da lista supra o espaço necessário para o novo registro, acrescente-o no final do arquivo.

Remoção de amigos (2)

Dado o “Código” realize a remoção do respectivo registro. A remoção deve ser feita diretamente no arquivo de dados. Para reaproveitar o espaço removido vocês terão que acrescentar no arquivo uma lista ligada entre os espaços disponíveis. Assim, vocês terão que acrescentar as seguintes informações no arquivo:

- (1) criem um registro cabeçalho e nele um campo que indica o offset para o primeiro elemento da lista.
- (2) ao remover um registro, substitua-o no arquivo por: <tamanho em bytes do registro removido>*<offset para o próximo elemento da lista>), onde * é um marcador indicando que este espaço está disponível.
- (3) um novo espaço disponível deve ser acrescentado sempre no início da lista. Logo, vocês devem atualizar o offset do cabeçalho e guardar o seu antigo offset no novo elemento da lista.
- (4) o final da lista é indicado por -1 no campo offset para o próximo elemento.

Compactação do arquivo

A estratégia de remoção vai criar fragmentos (internos e externos). Reconstruam o arquivo, quando solicitado pelo usuário, compactando todos os registros e limpando esses fragmentos (internos e externos).

Observações:

- (1) Não criar os arquivos toda vez que o programa for aberto (fazer verificação).
- (2) Os arquivos principais devem ser manipulados totalmente em memória secundária!