

2º Trabalho Prático

O programa referente ao 2º trabalho prático da disciplina Organização e Recuperação de Dados deverá ser desenvolvido na linguagem C por grupos de até três pessoas. O código-fonte deverá estar devidamente identificado com os nomes das pessoas do grupo e ser entregue exclusivamente via *Classroom*. Posteriormente, o grupo deverá fazer a apresentação do trabalho em horário previamente agendado.

Especificação

O objetivo deste trabalho é criar um programa que, a partir de um arquivo contendo uma lista de chaves, construa um **hashing extensível**. Além disso, o programa também deverá fornecer funcionalidades para imprimir informações sobre o *hashing*.

O arquivo de chaves será um arquivo texto contendo apenas chaves numéricas, uma por linha. Cada chave lida (um número inteiro) deverá ser inserida em um *hashing* extensível com *buckets* de tamanho fixo. Para facilitar a experimentação com *buckets* de diferentes tamanhos, faça o tamanho máximo do *bucket* ser uma constante TAM_MAX_BUCKET e use-a ao longo do código sempre que precisar referenciar o tamanho do *bucket*.

A estrutura interna dos *buckets* e do diretório devem ser similares às vistas em aula. **Os *buckets* deverão ser criados e mantidos no arquivo de *buckets***. O diretório será armazenado no arquivo de diretório, mas será construído e manipulado em memória.

O programa deverá oferecer as seguintes funcionalidades:

- Importação do arquivo de chaves para o *hashing* extensível;
- Impressão das informações do *hashing* extensível.

Assim como no 1º trabalho, **as operações serão realizadas sem interação direta com o usuário, utilizando para isso a linha de comando com passagem de parâmetros**.

Importação das chaves

A funcionalidade de importação do arquivo de chaves para o *hashing* extensível (`-i`) deve ser acessada pela linha de comando, da seguinte forma:

```
$ programa -i nome_arquivo_chaves
```

sendo `programa` o nome do arquivo executável do seu programa e `nome_arquivo_chaves` o nome do arquivo que contém as chaves. Sempre que ativada, essa funcionalidade fará a leitura do arquivo de chaves e a inserção das chaves no *hashing* extensível. Os *buckets* serão armazenados no arquivo *buckets.dat* e o diretório será armazenado no arquivo *dir.dat*. Note que o formato do arquivo de chaves será sempre igual, porém **o número de chaves no arquivo pode variar**. Para simplificar o processamento do arquivo de chaves, considere que ele sempre será fornecido corretamente – uma chave por linha e sem repetições (i.e., o seu programa não precisa verificar a integridade desse arquivo).

Ao final da importação, o programa deverá apresentar uma mensagem na tela indicando se essa operação foi concluída com sucesso ou não.

Impressão das informações do diretório

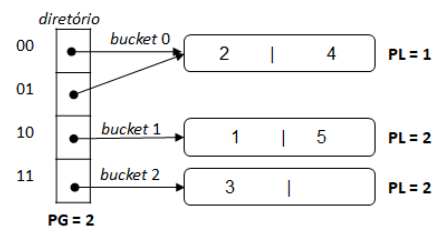
A funcionalidade de impressão das informações do diretório (`-pd`) também será acessada via linha de comando, no seguinte formato:

```
$ programa -pd
```

sendo `programa` o nome do arquivo executável do seu programa. Note que, para essa execução, o *hashing* extensível precisa ser inicializado/carregado para a memória e os arquivos *buckets.dat* e *dir.dat* devem existir.

Sempre que ativada, essa funcionalidade apresentará na tela o conteúdo de todas as células do diretório, além das seguintes informações: (a) profundidade; (b) tamanho atual; e (c) número total de *buckets* referenciados.

Como exemplo, considere o *hashing* extensível com *buckets* de tamanho 2 mostrado abaixo.



Supondo que o *hashing* esteja nessa configuração, seu programa deverá apresentar as seguintes informações sobre o diretório:

```
----- Diretorio -----
dir[0] = bucket(0)
dir[1] = bucket(0)
dir[2] = bucket(1)
dir[3] = bucket(2)

Profundidade = 2
Tamanho atual = 4
Total de buckets = 3
```

Impressão das informações dos *buckets*

A funcionalidade de impressão das informações dos *buckets* (`-pb`) também será acessada via linha de comando, no seguinte formato:

```
$ programa -pb
```

sendo `programa` o nome do arquivo executável do seu programa. Note que, para essa execução, o *hashing* extensível precisa ser inicializado/carregado para a memória e os arquivos *buckets.dat* e *dir.dat* devem existir.

Sempre que ativada, essa funcionalidade apresentará na tela o conteúdo de todos os *buckets* do arquivo *buckets.dat*.

Como exemplo, considere o *hashing* extensível mostrado na figura acima. Supondo que o *hashing* esteja nessa configuração, seu programa deverá apresentar as seguintes informações sobre os *buckets*:

```
----- Buckets -----
Bucket 0 (Prof = 1):
Chave[0] = 2
Chave[1] = 4

Bucket 1 (Prof = 2):
Chave[0] = 1
Chave[1] = 5

Bucket 2 (Prof = 2):
Chave[0] = 3
Chave[1] = -1
```

Qualquer dúvida referente a esta especificação deverá ser sanada com os respectivos professores.

BOM TRABALHO!