

MC202AB - Estrutura de Dados

Lab 10 - Hashing (Endereçamento aberto)

Data da Primeira Chance: 21 de novembro de 2022

Peso: 5

She-Hulk ainda está com problemas para lidar com os zilhões de processos que seu chefe deu para ela cuidar. Agora ela percebeu que os números dos processos são números extremamente altos e está cada vez mais difícil para ela ter que fazer as operações de inserir, buscar e remover processos do sistema.



Para ajudar a advogada de heróis, você precisa desenvolver um TAD da estrutura de dados **Hashing, implementado com endereçamento aberto**.

O dado que precisa ser armazenado é uma estrutura chamada **Processo**, em que se tem um valor inteiro **ID** ($10^9 \leq ID \leq \text{INT_MAX}$) que será o número identificador e a chave do processo, uma string **nome_proc** (string sem espaço) que armazenará o nome do arquivo do processo, e outro valor inteiro **tempo** ($10 \leq \text{tempo} \leq 1000$) representando quanto tempo, em dias, o processo está esperando para ser atendido.

Com esse TAD implementado, as seguintes operações poderão ser solicitadas:

- *I [id do processo] [nome do processo] [tempo do processo]*
 - Comando para inserir processo na tabela hash
 - Imprime a mensagem “*PROCESSO [id do processo] INSERIDO!*”
 - Obs.: Assuma que o processo não está armazenado, isto é, não acontecerá um comando para inserir um processo já inserido.
- *R [id do processo]*
 - Comando para remover processo da tabela hash

- Imprime a mensagem "*PROCESSO [id do processo] REMOVIDO!*"
 - Nessa operação, **é garantido** que o processo existe
- *C [id do processo]*
 - Comando para consultar o nome do processo na tabela hash
 - Imprime a mensagem "*PROCESSO [id do processo]: [nome do processo]*"
 - Nessa operação, **não é garantido** que o processo exista na tabela. Caso não exista, a seguinte mensagem deve ser impressa: "*PROCESSO [id do processo] NAO ENCONTRADO!*"
- *T [id do processo]*
 - Comando para consultar o tempo do processo na tabela hash
 - Imprime a mensagem "*TEMPO DO PROCESSO [id do processo]: [valor do tempo do processo] DIAS.*"
 - Nessa operação, **é garantido** que o processo existe

Entrada

A entrada do programa será um valor N ($5 \leq N \leq 2000$) representando a quantidade de operações a serem feitas, seguido das operações em si, com cada uma em uma única linha.

Exemplo:

```
5
I 1321468128 caso_wong 10
I 1846162235 abominavel_detalhes 125
I 1545135462 recorrencia_titania 342
C 1321468128
R 1545135462
```

Saída

A saída do programa deverá ser a impressão de cada operação realizada, finalizando com a mensagem "*FINALIZADO!*". Exemplo (seguindo o exemplo da Entrada):

```
PROCESSO 1321468128 INSERIDO!
PROCESSO 1846162235 INSERIDO!
PROCESSO 1545135462 INSERIDO!
PROCESSO 1321468128: caso_wong
PROCESSO 1545135462 REMOVIDO!
FINALIZADO!
```

Exemplos

Exemplo 1:

Entrada

```
11
I 1037515542 abominavel_v2 56
I 1125176107 caso_mallory 502
I 1321468128 caso_wong 10
I 1846162235 abominavel_detalhes 125
I 1545135462 ocorrencia_titania 342
I 1548865512 sokovia_revisao 687
I 1896572551 donny_blaze 20
R 1037515542
C 1125176107
C 1545135462
T 1548865512
```

Saída

```
PROCESSO 1037515542 INSERIDO!
PROCESSO 1125176107 INSERIDO!
PROCESSO 1321468128 INSERIDO!
PROCESSO 1846162235 INSERIDO!
PROCESSO 1545135462 INSERIDO!
PROCESSO 1548865512 INSERIDO!
PROCESSO 1896572551 INSERIDO!
PROCESSO 1037515542 REMOVIDO!
PROCESSO 1125176107: caso_mallory
PROCESSO 1545135462: ocorrencia_titania
TEMPO DO PROCESSO 1548865512: 687 DIAS.
FINALIZADO!
```

Exemplo 2:

Entrada

```
13
I 1037515542 abominavel_v2 10
I 1125176107 caso_mallory 15
I 1321468128 caso_wong 53
I 1846162235 abominavel_detalhes 587
I 1545135462 ocorrencia_titania 789
C 1545135462
C 1321468128
R 1037515542
R 1545135462
C 1125176107
```

```
C 1846162236
R 1846162235
R 1321468128
```

Saída

```
PROCESSO 1037515542 INSERIDO!
PROCESSO 1125176107 INSERIDO!
PROCESSO 1321468128 INSERIDO!
PROCESSO 1846162235 INSERIDO!
PROCESSO 1545135462 INSERIDO!
PROCESSO 1545135462: recorrencia_titania
PROCESSO 1321468128: caso_wong
PROCESSO 1037515542 REMOVIDO!
PROCESSO 1545135462 REMOVIDO!
PROCESSO 1125176107: caso_mallory
PROCESSO 1846162236 NAO ENCONTRADO!
PROCESSO 1846162235 REMOVIDO!
PROCESSO 1321468128 REMOVIDO!
FINALIZADO!
```

Regras e Avaliação

Importante: é obrigatório o uso dos conceitos de **HASHING (ENDEREÇAMENTO ABERTO)** para implementar a solução deste lab.

Seu código será avaliado não apenas pelos testes do CodePost, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código iremos analisar: o uso apropriado de funções, de comentários, e de structs; a escolha de bons nomes de funções, variáveis e de structs e seus campos; o correto uso de Tipos Abstratos de Dados e a correta separação em vários arquivos; a ausência de diversos trechos de código repetidos, e o tempo de execução e uso de memória dos algoritmos projetados. Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Submissão

Você deverá submeter no CodePost, na tarefa Lab 10 - Hashing, os arquivos *hash.h*, *hash.c* e *lab10.c*. No arquivo *hash.h* é esperado que você declare a(s) estrutura(s) com os devidos campos e funções necessárias para implementar as operações com o *Hashing* desta atividade. No arquivo *hash.c* é esperado que você implemente as funções declaradas no arquivo *hash.h*. Por fim, no arquivo *lab10.c* é esperado que você faça a leitura da entrada do programa e execute as chamadas das funções por meio do TAD implementado.

Após o prazo estabelecido para a atividade, será aberta uma tarefa Lab 10 - Segunda Chance, com prazo de entrega até o fim do semestre.