

UNIVERSIDADE FEDERAL DO MATO GROSSO *CAMPUS* ARAGUAIA

ANTHONY MUNIZ PRADO DE OLIVEIRA, VINICIUS SPANHOL
FERRARI

TRABALHO 2 - TABELA HASH

BARRA DO GARÇAS - MT

2021

INTRODUÇÃO

O objetivo deste trabalho é exercitar a implementação e manipulação de uma tabela Hash com a linguagem de programação C. Também foi trabalhado a manipulação de arquivos para inserir dados na tabela hash em questão.

Foi utilizado o editor de textos Visual Studio Code para a programação.

DESENVOLVIMENTO

O desenvolvimento consiste de 5 etapas onde 3 delas serão tratadas neste trabalho:

Etapa 1:

Elabore um algoritmo em linguagem C/C++ que gere um arquivo com um conjunto de valores e suas respectivas chaves, para serem inseridos em uma tabela Hash. Este algoritmo recebe como parâmetro o número de elementos a ser gerado, o tipo das entradas e o nome do arquivo resultante. Exemplo:

```
> ./algoritmo 1000 1 saida.txt
```

Onde algoritmo é o nome do programa, 1000 é o número de elementos a ser gerado, 1 é o tipo de arquivo (explicado a seguir) e saida.txt o nome do arquivo resultante.

Como resultado, no arquivo resultado, cada linha deve conter uma chave inteira com valor entre 0 e 1023, e um valor composto por 3 letras. Estes valores serão definidos aleatoriamente, mas garanta que não existem chaves com valores repetidos.

São dois tipos de arquivos de entrada possíveis:

1. Todas as chaves geradas têm valor par;
2. Sem restrição.

Nesta etapa houveram problemas para se gerar os arquivos (foi resolvido), também em gerar a função rand de forma que os valores não se repetissem(não resolvido), e também não conseguimos fazer os parâmetros serem passados pelo próprio executar.

Etapa 2:

Implemente um algoritmo que receba um arquivo resultante da etapa 1 a aplique em uma tabela Hash de tamanho 100 e que trate as colisões com encadeamento externo.

Para o arquivo o resultante da etapa 1 teste duas 2 funções Hash distintas de tratamento de colisões a seu critério.

Desenvolva todas as funções necessárias para manipulação da estrutura (inserção, busca, etc).

Nesta etapa houveram problemas em se retirar os valores linha por linha, também de trabalhar com char pelas funções e também de se implementar o novo hash linear para tratar as colisões conforme foi pedido(todos foram resolvidos).

Etapa 3:

Avaliando as funções hash.

Com o algoritmo da etapa 01:

- Gere 5 arquivos do tipo 1 com 50 números gerados, com 100 números gerados e 150 números gerados. Num total de 15 arquivos do tipo 1;
- Gere 5 arquivos do tipo 2 com 50 números gerados, com 100 números gerados e 150 números gerados. Num total de 15 arquivos do tipo 2;

Desta forma, ao final desta etapa teremos 30 arquivos.

Execute os 30 arquivos no algoritmo da etapa 2 e para cada execução, contabilize o número de colisões que ocorreram em cada uma das funções hash.

RESULTADOS

O trabalho ficou incompleto devido a não termos conseguido fazer rand gerar números que não se repetissem, com isso o nosso resh linear não funciona para fazermos a comparação.

CONSIDERAÇÕES FINAIS

;~;