

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CIÊNCIA DA COMPUTAÇÃO

Estrutura de Dados

Relatório do Trabalho

Alunos:

Matheus de Oliveira Lima.

Yan Oliveira Maia.

Vitória – 26 Outubro 2022

MANEIRA DE TESTAR O TRABALHO

Desenvolvemos uma maneira mais fácil de testar o trabalho. Dentro da região onde está o trabalho coloque o a pasta com input e output dentro. Logo depois, vá até o “package.h” e no “#define TESTE x” troque o “x” pela pasta nome da pasta que inseriu como teste.

INTRODUÇÃO

Primeiramente foi feita uma análise referente a estrutura de dado que seria tomada para o desenvolvimento do trabalho. Neste início, notamos uma necessidade de adotar o método de lista encadeada com sentinela para um desempenho de uso de memória e organização.

DESENVOLVIMENTO

A primeira compreensão acerca da “main.c” do trabalho, foi a necessidade de uma lista de usuários criada e lida por funções de uma biblioteca “listaUsuario.h” que permitiria a utilização das funções presentes na “listaUsuario.c”.

Visualizamos que era necessário ir liberando e conferindo a liberação de memória do programa. Então, a medida que estávamos desenvolvendo fizemos testes parciais utilizando “valgrind”.

O tipo “tUsuario” começou de maneira simples, com nome, idade e localização. O segundo tipo a ser criado foi o “tPackage” necessário para a leitura dos arquivos de entrada que direcionavam o desenvolvimento do trabalho.

Como nós precisaríamos de abrir os arquivos “.package.txt” dos usuários para ler suas ações do ciclo, optamos por ir adicionando ao “tPackage” listas e structs que guardassem todas as informações utilizadas pelos seus respectivos usuários. OBS: Em cada usuário há um “tPackage” respectivo.

Após fazer as leituras e guardar as informações, foi feito a execução dos ciclos com a função “ExecutaEdMatch”. Ela basicamente foi pensada para desenvolver os ciclos de acordo com a entrada “num” dada no “argv”. Por exemplo: se você digita “./main 3”, esse valor 3 introduzido como argumento é usado como um valor fixo de ciclos que acontecerão no decorrer do “ExecutaEdMatch”.

Também foram criadas algumas listas e structs para o armazenamento de informações geradas pelos ciclos dentro do tUsuario para serem modificadas com o passar dos ciclos dentro do “ExecutaEdMatch”, como: registrar like, registrar unlike, registrar as amizades feitas e desfeitas, registrar hobby (caso tenha sido alterado), registrar post no feed e registrar

sugestões de amizades. Os primeiros registros desenvolvidos foram os likes e unlikes. Os demais, foram sendo criados e feitos de acordo a necessidade do problema proposto pelo trabalho.

Dentro do “ExecutaEdMatch” é criado o arquivo “logs.txt”, que com o passar dos ciclos é alimentado pelo o que acontece dentro do aplicativo “EdMatch”

Para a criação do arquivo reports.txt foi desenvolvida uma função na “main.c” que se chama “ImprimeListaUsuario”. Inicialmente, essa imprime lista usuário era utilizado para printar o usuário para saber se estávamos guardando de maneira correta suas informações. Com o passar do trabalho, essa função está sendo usada para que as informações geradas a partir do “ExecutaEdMatch” sejam “impressas” em um arquivo específico que mostra as informações gerais exigidas pelo trabalho.

A função “LiberaListaUsuario” também foi sendo desenvolvida e modificada com o desenvolvimento do trabalho. A cada lista e struct adicionada aos usuários, era necessário a adição desses novas estruturas para liberação da memória.

CONCLUSÃO

O início do desenvolvimento parece não sair do lugar, ao mexer com listas. Porém com o crescimento da complexidade do trabalho, estar com as listas e structs datadas e separadas da forma correta, cooperou para um melhor desempenho.