

Segundo Trabalho Prático de Estrutura de Dados I

**** Aline Pontes Dório**

DI/UFES**

Introdução

> O projeto visa simular uma implementação de um programa para avaliar expressões aritméticas por meio de uma Árvore Binária de Expressões Aritméticas, com a utilização de sensores, implementando os conceitos estudados durante o semestre. O código foi elaborado utilizando-se diversas TAD 's, atribuindo cada funcionalidade do sistema a uma biblioteca específica. A intenção final do projeto era criar um programa que transforma expressões aritméticas em árvores binárias e, por meio dela, calcula o resultado das expressões e os escreve em um arquivo de saída.

> Nos tópicos a seguir, um pouco de cada uma dessas funcionalidades implementadas, seguindo a linha de raciocínio utilizada.

1. info.h e info.c

Nessas TAD's foram implementados os trechos dos códigos relacionados à manipulação das árvores e à utilização dos arquivos disponibilizados e criados durante o processo do trabalho, como os de entrada e saída. Foram feitas funções que trabalhassem com a criação da árvore binária, com os cálculos das expressões numéricas e, também, uma função inicial chamada "inicializaTrab" que junta todas as informações e as organiza de forma intuitiva para facilitar na organização do código

2. arv.h e arv.c

Aqui, foram trabalhadas as funções diretamente relacionadas às árvores, juntamente a uma struct "Arv", que foi usada por todo o código para facilitar sua manipulação. Na TAD arv.h as funções foram criadas e nomeadas e na arv.c elas foram implementadas para serem utilizadas na info.h.

Arquivos

É importante ressaltar aqui da importância da localização do arquivo de entrada, que deve estar localizado dentro da pasta "data" para que o código funcione de forma apropriada.

Como arquivo de entrada, foi utilizado o arquivo de teste disponibilizado pela professora, a saber: entrada.txt. Como arquivos de saída, o programa deve gerar saídas para incluir os resultados das expressões aritméticas calculados por meio das árvores, com o sensor devidamente lido (saida.txt).

Conclusão

O trabalho permitiu trabalhar bem o conceito de árvores, uma vez que todo o projeto girava em torno dessa ideia. Além disso, exigiu que se trabalhasse no quesito organização, visto que foi utilizada uma grande quantidade de funções e algumas TAD's para este projeto. Foi um trabalho desafiador e houveram muitas dificuldades. Dessa forma, frente ao desafio, foi possível aprimorar e fixar bem os conceitos vistos, de tal maneira a contribuir para a minha

evolução enquanto cientista da computação. Como maiores dificuldades vistas, apresentam-se como principais:

- Parsing, uma vez que demorei duas semanas para conseguir encontrar um raciocínio correto para o que foi pedido, até que percebi que era necessária a utilização da recursão, como ensinado em aula, para que o programa funcionasse.
- A liberação de memória também foi uma parte na qual tive dificuldade, uma vez que optei por implementar as funções de liberação de memória ao final e não durante a implementação do código. Com isso, foi necessário analisar com cuidado a forma como o código estava sendo executado, para que as liberações de memória fossem encaixadas nos locais corretos.
- Dificuldade com relação ao graphviz, que infelizmente não consegui fazer o preenchimento do arquivo corretamente, por falta de conhecimento sobre o assunto e de tempo para pesquisá-lo e entendê-lo melhor. Percebo agora que a decisão de deixá-lo para o final não foi muito interessante e eu deveria tê-lo feito simultaneamente à criação do código.

Bibliografia

<https://pt.stackoverflow.com/>

<https://cboard.cprogramming.com/>

<https://www.geeksforgeeks.org/>

https://www.tutorialspoint.com/c_standard_library/index.htm