

Trabalho prático 01 - Algoritmos e Estrutura de Dados I

Arthur De Bellis Gomes - 03503

Pablo Ferreira - 03480

Saulo Miranda Silva - 03475

Universidade Federal de Viçosa - Campus Florestal

Objetivo do trabalho

O objetivo do trabalho foi colocar em prática o que aprendemos nas aulas ministradas pela professora Thais, de Algoritmos e Estrutura de Dados, aplicando nossos conhecimentos sobre Tipos Abstratos de Dados, Alocação Dinâmica e Listas Simplesmente Encadeadas.

Projeto do Sistema Implementado

Inicialmente, a divisão das tarefas foi definida de modo com que todos pudessem tomar decisões acerca da implementação de cada um dos TADs, sendo a equipe dividida geralmente em 2 pessoas para desenvolver o código de cada um deles, porém, de acordo com as necessidades e por questões de melhor aproveitamento de tempo, os três integrantes do grupo trabalharam juntos em alguns dos TADs em prol do projeto.

Principais decisões do grupo

Algumas das decisões tomadas em equipe foram, por exemplo, as nomeações das variáveis e funções, separação das horas e minutos em dois inteiros distintos para facilitar as comparações e execuções de alguns dos subprogramas. Outra importante decisão foi sobre a criação de um TAD contendo somente os menus, diminuindo assim o tamanho do main e padronizando a aparência do programa. Porém, a principal decisão foi a respeito das divisões das tarefas, a divisão foi feita de acordo com a afinidade de cada um dos membros com as atividades propostas por cada TAD.

Módulos desenvolvidos

Primeiramente fizemos as projeções de o que as funções fariam e como seria a integração entre elas, após este processo, começamos o planejamento das execuções, tal ação ajudou muito no procedimento do algoritmo implementado.

Funções implementadas:

Tad voo:

- ☐ IniciaVoo
- ☐ GetVid;
- ☐ GetHrDecolagem
- ☐ GetMinutosDecolagem
- ☐ GetHoraPouso
- ☐ GetMinutosPouso
- ☐ GetAeroportoDecolagem
- ☐ GetAeroportoPouso
- ☐ GetIdPista
- ☐ SetVid
- ☐ SetHrDecolagem
- ☐ SetMinutosDecolagem
- ☐ SetHrPouso
- ☐ SetMinutosPouso
- ☐ SetAeroportoPouso
- ☐ SetAeroportoDecolagem
- ☐ SetIdPista

Tad Lista de Voos:

- ☐ IniciaLista
- ☐ ListaVazia
- ☐ InserirNovo
- ☐ InserirUltimo
- ☐ RemoverVoo
- ☐ ProcurarVoo
- ☐ ImprimirLVoos

Tad Item Matriz:

- ☐ InicialItem
- ☐ ImprimirItem
- ☐ SetLista
- ☐ SetNumVoo
- ☐ SetHreMntsLast
- ☐ GetLista
- ☐ GetNumVoo
- ☐ GetHrLast
- ☐ GetMntsLast

Tad Matriz de Voss:

- ☐ IniciaMatriz
- ☐ InserirMVoo
- ☐ RemoverMVoo
- ☐ ProcurarMVoo
- ☐ ImprimirVooHrDecolagemHrPouso
- ☐ ImprimirVooHrDecolagem
- ☐ ImprimirVooHrPouso
- ☐ ImprimirMatriz
- ☐ EncontrarMaiorHorario
- ☐ EncontrarMenorHorario
- ☐ EncontrarListaMaisRecente
- ☐ EncontrarListaMenosRecente
- ☐ MatrizEsparca

Tad Menu:

- ☐ menu_de_entradas
- ☐ menu_de_opcoes

- ❑ menu_de_confirmacao
- ❑ menu_de_saida
- ❑ menu_arquivo

Métodos implementados

Para a edição do código os três integrantes do grupo usaram o editor de texto Atom. Para a integração do código feito em 3 computadores distintos, usamos a excelente ferramenta GitHub, já para realizar a compilação do programa usamos o GCC no terminal do linux.

Após certificarmos que estávamos na pasta do trabalho que contém os arquivos “.c”, a pasta *Sources*, o seguinte comando era executado no terminal:

gcc main.c -o EXEC menu.c voo.c ListadeVoos.c ItemMatriz.c MatrizVoos.c

E para executar:

./EXEC

Início da execução do programa:

Após entrar no programa, um menu mostra as opções de entrada possíveis para o usuário:

1. Modo interativo
2. Arquivo
0. Sair

Utilização do modo interativo:

No modo interativo, espera-se que o usuário entre com as opções desejadas por ele.

Após escolher a utilização deste modo, uma tela de opções é exibida na tela, permitindo assim que o utilizador saiba as funções possíveis de se empreender no programa, bem como iniciar matriz, entrar com novo item etc.

O modo é bem intuitivo e auto-explicativo, permitindo um fácil entendimento por parte da pessoa que opera o programa.

Utilização do modo por arquivo:

No modo por arquivo, o programa exigirá que o usuário digite o nome do arquivo que estará localizado na mesma pasta em que o programa está. Nesse arquivo estará digitado todas as funções e dados esperados para cada uma delas.

O arquivo deverá seguir esse padrão de digitação:

Exemplo de arquivo:

a
b 06:35 09:25 MBH MFL 7
c 41
d 41
e 06:35 09:25
f 06:35
g 09:25
h
i
j
k
l
m

Observação: O nome do arquivo precisa seguir o padrão de extensão “.txt” no final.

Exemplo: *Teste.txt*

Opção de saída:

Caso o consumidor do programa decida parar sua execução, ele deverá digitar “0” no menu de opções. Para se evitar paradas de execução indesejadas ou inesperadas, é exibido um campo com pedido de confirmação, este pede para usuário digitar ‘s’ ou ‘S’, para parar a execução, evitando assim, certos desconfortos .

Conclusão

Concluimos que esse trabalho nos ajudou muito em nossa evolução como programadores. Aprendemos técnicas novas, aperfeiçoamos nossos conhecimentos pré existentes e aprendemos principalmente a trabalhar em equipe, além da utilização do GitHub, e boas práticas de programação.

Dedicamos também um agradecimento especial para nossos monitores, Angelo e Kayque, além de nossa professora Thais.