

INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO I

TRABALHO 2

20 de abril de 2018

Gabriel de Andrade Dezan
Nº USP: 10525706

Universidade de São Paulo

1 Introdução

1.1 Grupo

O trabalho foi feito individualmente pelo aluno Gabriel Dezan (P = 0).

1.2 O Programa

O trabalho é uma implementação de um sistema de controle de alunos de Introdução à Ciência da Computação em C.

As informações dos alunos são armazenadas em arquivo de nome "students.dat". É utilizado um outro arquivo de nome "temp.dat" para facilitar a manipulação de dados no arquivo principal.

No programa, o usuário pode incluir/consultar/excluir alunos, provas e trabalhos.

2 Descrição do Projeto

2.1 Ambiente de Desenvolvimento

O jogo foi desenvolvido no sistema operacional Windows 64-bit com o editor de textos Visual Studio Code. Foi utilizada a plataforma GitHub para controle de versão.

2.2 Compilador Usado

Foi utilizado o compilador MinGW (GCC 6.3.0) no Windows sem parâmetros de compilação.

2.3 Códigos Fonte

Os arquivos do projeto são os abaixo:

- main.c
- students.c
- students.h
- grades.c
- grades.h

2.4 Versão Compilada

Há uma versão compilada do projeto, de nome "files.exe", onde o projeto pode ser testado sem a necessidade de compilação.

3 Tutorial

3.1 Compilação e Execução

Para compilar o projeto, o compilador GCC deve estar instalado. Basta acessar o terminal de comando na pasta do projeto e executar o seguinte comando:

```
"gcc -o alunos main.c students.c grades.c"
```

Para executar o projeto, execute em seguida o comando:

```
"./files"
```

É possível também compilar e executar o projeto com o uso da IDE Code::Blocks. Basta iniciar um novo projeto, abrir os arquivos do projeto recursivamente e usar a opção "Build and Run".

3.2 Como utilizar

Ao iniciar o programa, caso o usuário não tenha o arquivo "students.dat", o programa irá criar esse arquivo e, então, o usuário deverá inserir os alunos iniciais que ele deseja. Em seguida, é apresentado a um menu com opções para manipular os dados dos alunos.

```
=====
Informacao de Alunos
=====

-----ALUNOS-----
[1] Novo Arquivo de Alunos
[2] Adicionar Alunos
[3] Consultar Aluno
[4] Excluir Aluno
-----

-----PROVAS-----
[5] Adicionar/Atualizar Prova
[6] Consultar Prova
[7] Excluir Prova
-----

-----TRABALHOS-----
[8] Adicionar/Atualizar Trabalho
[9] Consultar Trabalho
[10] Excluir Trabalho
-----

[11] Mostrar Todos Alunos

[0] Sair
=====
```

Exemplos das opções sendo executadas:

- Consultar aluno

```
=====
3
=====

Entre o nome completo do aluno ou seu NUSP: 123456789

=====
Nome: João da Silva
NUSP: 123456789
=====
PROVAS

Prova 1: 7.80
=====
TRABALHOS

Trabalho 1: 9.76
=====

Pressione ENTER para continuar
```

A consulta só exibe as provas e trabalhos que foram adicionados.

- Adicionar/Atualizar Prova

```
=====
Escolha a prova
=====

[1] Prova 1
[2] Prova 2 (Prova nao adicionada ainda)

[0] Voltar

=====

2

=====
Prova 2
=====

Entre com a nota para João da Silva: 9.7
Entre com a nota para Maria de Souza: 6.8798
```

Caso o usuário escolha uma prova já adicionada (como a Prova 1 no exemplo), ele será perguntado se ele deseja substituir as notas antigas da prova por novas.

- Consultar Trabalho

```
=====
Escolha o Trabalho
=====

[1] Trabalho 1
[2] Trabalho 2 (Trabalho nao adicionado ainda)
[3] Trabalho 3 (Trabalho nao adicionado ainda)
[4] Trabalho 4 (Trabalho nao adicionado ainda)

[0] Voltar

=====

1

=====
Trabalho 1
=====

João da Silva: 9.76
Maria de Souza: 7.89

Pressione ENTER para continuar
```

- Adicionar Aluno

```
=====

2

=====

Entre o nome do aluno: Carlos Santos
Entre o NUSP do aluno: 1111111111
Entre a nota do aluno na Prova 1: 6.789
Entre a nota do aluno na Prova 2: 9.12
Entre a nota do aluno no Trabalho 1: 5.55
```

Caso provas/trabalhos já tenham sido inseridos anteriormente, o usuário terá que por as notas dessas provas/trabalhos.

4 Limitações

O programa está constantemente escrevendo e lendo arquivos, algumas funções abrem e fecham dois arquivos duas vezes. O programa talvez seria mais eficiente com o uso de listas encadeadas, fazendo-se necessária a abertura do arquivo apenas uma vez no programa. Seria possível, também, a organização alfabética dos alunos com mais facilidade.

5 Observações

Caso o usuário deseje ler o arquivo "students.dat", foi incluído um programa "reader.c" que pode ser compilado e executado para a leitura do arquivo.

Para compilar:

```
"gcc -o leitor reader.c"
```

Para executar:

```
"./leitor"
```