

Introdução à Programação de Computadores

Segundo Trabalho

8 de julho de 2018

1 Objetivos e organização

Esta tarefa possui como objetivos:

- Aumentar a experiência de uso do compilador GCC.
- Possibilitar uma compreensão mais aprofundada da última versão padrão da linguagem C (C11).
- Desenvolver as habilidades no manuseio correto de arquivos, arranjos, estruturas, apontadores e funções.

Os programas implementados serão apresentados ao *monitor* da disciplina que os avaliará. Trabalhos não apresentados não serão avaliados. Os códigos fontes deverão ser encaminhados para o professor via e-mail. **Não** coloque arquivo executável pois o Gmail recusará os e-mails.

Todos os códigos devem ser documentados em conformidade com os exemplos já disponibilizados. Assim, o código deve conter, entre outras informações, o nome e o propósito de cada algoritmo implementado e o código desse algoritmo. Além disso ele deve conter instruções específicas sobre como compilar e executar os programas no Ubuntu.

O trabalho é em **dupla** e qualquer desonestidade acadêmica será punida com nota zero para todos os envolvidos.

1.1 Informações importantes

1.1.1 Sobre o código

- **Não** use variáveis globais. Além de ser um péssimo estilo de programação, nesse trabalho eu quero ver as passagens de parâmetros por valor ou por endereço, conforme for o caso.

- Sempre verifique as saídas das funções de biblioteca que lidem com arquivos, conforme exemplos vistos em aula.
- Modularize o seu programa, ou seja, crie várias funções.
- Evite criar funções que ocupem mais que 50 linhas de texto.
- Mantenha a sua função `main` com um mínimo de chamadas de funções e de código.
- São esperadas pelo menos 5 funções em seu programa.
- Documente seu código usando o padrão visto nos exemplos dados.
- O programa deve ser compilado com GCC padrão C11 no Ubuntu e também deve ser executado no Ubuntu.

1.1.2 Detalhes da entrega

entrega 16 de julho de 2018 até as 13h, segunda-feira, para o monitor. O horário limite de entrega para o monitor, em nenhuma situação poderá ser ultrapassado. Por favor combine um horário para a apresentação. Você pode também antecipar a apresentação se terminar antes desse dia.

e-mail prof.asoares@gmail.com Esse é o endereço para onde deve-se enviar somente o código fonte.

assunto IPC - Trabalho 2

2 Enunciado

Este trabalho consiste em escrever um programa em C11, usando o GCC no Ubuntu, que mantenha registros e realize uma análise estatística de uma turma de estudantes.

Para cada estudante, precisaremos de um nome com até 50 caracteres, um matrícula com 11 caracteres, três notas de provas, duas notas de trabalhos, um inteiro para o número de faltas, uma nota final, um caractere para a situação final: 'A' - aprovado, 'R' - reprovado, 'F' - reprovado por faltas.

Os dados dos estudantes serão armazenados em um arranjo de **estruturas** de estudantes. Esse arranjo deve ser capaz de manter até 80 estudantes.

O Programa conterá o menu principal mostrado na Figura 1.

Cada uma das opções acima deve se comportar como especificado nas próximas subseções.

```
=====
                        Menu Principal
=====
Opções:

1 - Ler turma de estudantes do arquivo
2 - Imprimir turma de estudantes
3 - Imprimir estatísticas da turma
4 - Cadastrar novo estudante
5 - Editar informações de um estudante
6 - Reler turma de estudantes do arquivo
7 - Salvar e Sair

Digite uma opção:
```

Figura 1: Menu principal do programa

2.1 Ler turma de estudantes do arquivo

Ao selecionar esta opção, surgirá um novo menu como o mostrado na Figura 2.

```
=====
                        Ler Turma
=====
Opções:

1 - Ler turma existente
2 - Criar nova turma
3 - Voltar ao menu principal

Digite uma opção:
```

Figura 2: Menu de leitura de turma

Caso a opção selecionada for:

- 1 - O programa deverá solicitar o nome do arquivo texto contendo os dados da turma:

Digite o nome do arquivo com a turma: turma1.txt

Aqui, `turma1.txt` é um exemplo de nome de arquivo texto. Se a leitura for bem sucedida, a mensagem a seguir será exibida

Arquivo turma1.txt lido com sucesso!

Os dados desta turma serão armazenados no arranjo de estruturas, o arquivo será *fechado* e o menu da Figura 2 será mostrado novamente. Salve o nome do arquivo digitado para uso posterior.

Caso o arquivo **não** exista, a seguinte mensagem será impressa

Arquivo turma1.txt não existe!

e o menu mostrado na Figura 2 será mostrado novamente.

- 2 - O programa deverá solicitar o nome do novo arquivo texto que conterá os dados da turma:

Digite o nome do arquivo para a nova turma: turma2.txt

Caso o arquivo **não** exista, um novo será criado e a mensagem a seguir será mostrada

Arquivo turma2.txt criado com sucesso.

e o menu da Figura 2 será mostrado novamente. Salve o nome do arquivo digitado para uso posterior.

Se o arquivo já existir, informe ao usuário e pergunte o que fazer

Arquivo turma2.txt já existe. Deseja sobrescrevê-lo? (S/N)

Se a resposta for S, abra o arquivo em modo de escrita de texto com apagamento de conteúdo. Se a resposta for N, não faça nada. Em qualquer caso o menu da Figura 2 será mostrado novamente.

- 3 - Retorne ao menu principal mostrado na Figura 1.

2.1.1 Fomato do arquivo com os dados

A entrada será lida de um arquivo **texto**.

Cada linha do arquivo contém o nome do estudante, seu número de matrícula, suas três notas de provas, suas duas notas de trabalhos, seu número de faltas, sua nota final e sua situação final, nessa ordem. Se uma prova ou trabalho ainda não foi realizado, a nota é zero. A informação de

Turma de Estudantes									
Nome	Matrícula	Nota 1	Nota 2	Nota 3	Trab 1	Trab 2	Faltas	Nota final	Situação
Jackson do Pandeiro	10011BSI497	18	27	15	10	12	2	82	A
Luiz Gonzaga	10011BSI498	19	28	16	10	13	2	86	A
Alberto Roberto	10011BSI499	5	10	10	5	3	17	33	R
Chico Bento	10011BSI500	5	10	15	5	11	40	46	F

Tabela 1: Exemplo de estudantes

cada estudante será separada por uma nova linha no arquivo. As informações dos demais estudantes seguirão na próxima linha do arquivo.

Como exemplo, considere que tenhamos a turma de estudantes mostrada na Tabela 1.

O arquivo texto para essa turma de estudantes é aquele mostrado na Figura 3.

2.2 Imprimir turma de estudantes

Ao selecionar essa opção, o programa verificará se há uma turma já carregada – olhe se existe um nome de arquivo salvo – e, se existir, listará as informações dos estudantes na ordem em que elas forem lidas do arquivo. Não é necessário ordenar.

Imprima cada estudante em uma linha separada com a informação adequada em cada coluna. Para a turma mostrada na Tabela 1 a saída será aquela mostrada na Figura 4.

Caso não haja um turma carregada, imprima mensagem a seguir e retorne para o menu da Figura 1:

Não há turma carregada!

2.3 Imprimir estatísticas da turma

Se for esta a opção selecionada, o programa verificará se há uma turma já carregada - e, se existir, imprimirá a maior, a menor e a média da turma nas provas, trabalhos e na nota final. Além disso, também serão impressos os números de estudantes aprovados, reprovados e reprovados por faltas, juntamente com o seu percentual em relação à turma toda.

Na Figura 5 é ilustrada a saída das estatísticas da turma.

As estatísticas deve sempre ser tomadas em relação aos alunos cadastrados (no exemplo é 4) e não em relação à capacidade total do arranjo (80).

```
Jackson do Pandeiro
10011BSI497
18
27
15
10
12
2
82
A
Luiz Gonzaga
10011BSI498
19
28
16
10
13
2
86
A
Alberto Roberto
10011BSI499
5
10
10
5
3
17
33
R
Chico Bento
10011BSI500
5
10
15
5
11
40
46
F
```

Figura 3: Arquivo texto contendo a turma de estudantes do exemplo

Caso não haja um turma carregada, imprima mensagem a seguir

Não há turma carregada!

e retorne para o menu principal mostrado na Figura 1.

Estudantes										
Nome	Matrícula	N1	N2	N3	T1	T2	Fal	Final	Sit	
Jackson do Pandeiro	10011BSI497	18	27	15	10	12	2	82	A	
Luiz Gonzaga	10011BSI498	19	28	16	10	13	2	86	A	
Alberto Roberto	10011BSI499	5	10	10	5	3	17	33	R	
Chico Bento	10011BSI500	5	10	15	5	11	40	46	F	

Figura 4: Listagem das informações dos estudantes do exemplo

Estatísticas						
	N1	N2	N3	T1	T2	Final
Maiores notas da turma	19	28	16	10	13	86
Menores notas da turma	5	10	10	5	3	33
Notas médias da turma	13.0	20.4	14.2	8.0	10.2	65.8
Número de estudantes aprovados: 2 (50.0 %)						
Número de estudantes reprovados: 1 (25.0 %)						
Número de estudantes reprovados por falta: 1 (25.0 %)						

Figura 5: Saída das estatísticas da turma do exemplo

2.4 Cadastrar novo estudante

Se for esta a opção selecionada, o programa verificará se há uma turma já carregada - e, se existir, o texto da Figura 6 será mostrado.

Note que a nota e a situação final são calculadas pelo programa da seguinte forma:

- Se a nota final for maior ou igual a 60 e o número de faltas for menor ou igual a 18, a situação do estudante é **Aprovado**.
- Se o número de faltas for maior que 18, a situação do estudante é **Reprovado por faltas**. Isso é independente da nota final.
- Se a nota final for menor que 60, a situação do estudante é **Reprovado**.

```
=====
Cadastrar Novo Estudante
=====

Digite o nome: Aldovandro Cantagalo
Digite a matrícula: 10011BSI0501
Digite as três notas das provas: 15 5 10
Digite as duas notas dos trabalhos: 5 5
Digite o número de faltas: 5

Nota final calculada: 40
Situação final: Reprovado
```

Figura 6: Cadastro de novo estudante

Caso não haja um turma carregada, imprima mensagem a seguir

Não há turma carregada!

e retorne para o menu principal mostrado na Figura 1.

2.5 Editar informações de um estudante

Se for esta a opção selecionada, o programa verificará se há uma turma já carregada - e, se existir, surgirá um novo menu como o mostrado na Figura 7.

As opções 1 e 2 permitem que o usuário acesse o registro do estudante por matrícula ou por nome. Em qualquer um dos casos o programa deve procurar no arranjo de estruturas o registro correto.

```
=====
      Editar Informações do Estudante
=====
Opções:

1 - Selecionar por número de matrícula
2 - Selecionar por nome
3 - Voltar ao menu principal

Digite uma opção:
```

Figura 7: Menu de edição de um estudante

A comportamento a seguir é válido para ambas as opções, fazendo-se as devidas alterações de **matrícula** para **nome**.

O programa exibirá a mensagem a seguir

Digite o número de matrícula: 10011BSI497

Depois, ele verificará se esta matrícula existe no arranjo de estruturas. Se existir, o menu da Figura 8 será mostrado.

Se o número de matrícula for inválido, escreve a mensagem

Número de matrícula inválido!

e exibe novamente o menu da Figura 7.

```
=====
      Editar Informações do Estudante
=====
Opções:

1 - Alterar nome
2 - Alterar matrícula
3 - Alterar nota da prova 1
4 - Alterar nota da prova 2
5 - Alterar nota da prova 3
6 - Alterar nota do trabalho 1
7 - Alterar nota do trabalho 2
8 - Alterar número de faltas
9 - Voltar ao menu anterior

Digite uma opção:
```

Figura 8: Menu com as informações a serem editadas para um estudante

Caso seja selecionada a opção:

1 - Faz a solicitação

Digite o nome:

2 - Faz a solicitação

Digite a matrícula:

3 - Faz a solicitação

Digite a nota da prova 1:

4 - Faz a solicitação

Digite a nota da prova 2:

5 - Faz a solicitação

Digite a nota da prova 3:

6 - Faz a solicitação

Digite a nota do trabalho 1:

7 - Faz a solicitação

Digite a nota do trabalho 2:

8 - Faz a solicitação

Digite o número de faltas:

9 - retorna ao menu da Figura 7.

Após atualizar a informação do estudante no arranjo de estruturas, re-apresenta o menu da Figura 8.

Note que a nota e a situação final precisarão ser recalculadas.

Caso não haja um turma carregada, imprima mensagem a seguir

Não há turma carregada!

e retorne para o menu principal mostrado na Figura 1.

2.6 Reler turma de estudantes do arquivo

Ao selecionar essa opção, o programa verificará se há uma turma já carregada e, se existir, ele lerá as informações desse arquivo no arranjo de estruturas.

Se a leitura for bem sucedida, a mensagem a seguir será exibida

Arquivo turma1.txt lido com sucesso!

Os dados desta turma serão armazenados no arranjo de estruturas, o arquivo será *fechado* e o menu da Figura 3 será mostrado novamente. Salve o nome do arquivo digitado para uso posterior.

Caso o arquivo **não** exista, a seguinte mensagem será impressa

Arquivo turma1.txt não existe!

e o menu mostrado na Figura 3 será mostrado novamente.

2.7 Salvar e Sair

Ao selecionar essa opção, o programa verificará se há uma turma já carregada e, se existir, ele salvará o conteúdo do arranjo de estruturas para o arquivo existente.

Se o arquivo **não** existir exiba a seguinte mensagem

O arquivo turma1.txt não existe. Deseja criar um novo? (S/N)

Se a resposta for **S** crie o arquivo em modo de escrita de texto com apagamento de conteúdo, escreva todo o conteúdo do arranjo de estruturas para ele e feche o arquivo aberto. Ao terminar a escrita ou se a resposta for **N**, imprima

Até a próxima.

e termine o programa.

Se o arquivo **já** existir, informe ao usuário este fato e pergunte o que fazer

Arquivo turma1.txt já existe. Deseja sobrescrevê-lo? (S/N)

Se a resposta for **S** abra o arquivo em modo de escrita de texto e escreva todo o conteúdo do arranjo de estruturas para ele. Ao terminar a escrita ou se a resposta for **N**, imprima

Até a próxima.

e termine o programa.