



Universidade de Brasília

Faculdade UnB Gama (FGA)

Engenharia de Software

Fundamentos de Sistemas Operacionais

Professor: Thiago Alves

Trabalho 1

Eduardo Q. Gomes 14/0137068

Miguel Pimentel 14/0156143

Brasília, 1 de Setembro de 2016

Trabalho 1 Fundamentos de Sistemas Operacionais

Material Usado

- Sistema Operacional Linux, distribuição Ubuntu 16.04 com a interface Gnome;
- Compilador gcc (5.4.0);
- Editor de texto Sublime.

Conteúdo Utilizado

- Conteúdo do Moodle disponível em <http://aprender.ead.unb.br/course/view.php?id=3806>
- Livro Autor: Autor: SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P.B. Obra: Operating System Concepts

Questões

1) Escreva um programa em C que, em tempo de execução, receba do usuário 3 coordenadas cartesianas dos vértices de um triângulo e, como saída, calcule e imprima na tela o tamanho dos lados, o perímetro e a área do triângulo. Sua implementação deverá atender os seguintes requisitos:

- Na sua implementação deverá constar, no mínimo, três arquivos .h e três arquivos .c.
- Utilize um arquivo .h para definir os seguintes tipos compostos: ponto no plano cartesiano e triângulo, com três pontos no plano cartesiano.
- Utilize um arquivo .c/h para realizar as operações geométricas: cálculo do comprimento do lado do triângulo, cálculo da condição de existência do triângulo, cálculo do perímetro do triângulo e cálculo da área do triângulo.
- Utilize um arquivo .c/h com as funções de entrada.
- Utilize um arquivo .c com a função main().
- Sua implementação deverá ser construída a partir de um Makefile.

O programa contém uma estrutura que pode ser compilada utilizando um Makefile, além disso conta com um README.md para o usuário poder rodar o software. O programa trabalha com um conjunto de arquivos .h e .c, são eles:

- main.c e .h: Responsável pela execução inicial do programa
- geometric_calculus.c e .h: Responsável pelos cálculos geométricos;
- input.c e .h: responsável pelas entradas e pelo switch de opções;
- menu.c e .h: responsável por mostrar na tela o menu principal.

Para utilizar o software, inicialmente o usuário deve digitar coordenadas para três pontos, vale ressaltar que o usuário deve digitar números inteiros:

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T1/1 - Triangulo$ make run
./prog
Insira as Coordenadas x e y dos pontos
Ponto 1: 0 0
Ponto 2: 0 1
Ponto 3: 1 0
```

Após isso um menu irá aparecer, o usuário deve digitar a opção desejada. O usuário pode apenas digitar números das opções.

```
Escolha uma opção:
-----
-      0 - Imprime todos os resultados!      -
-      1 - Calcula comprimento dos lados     -
-      2 - Calcula condicao de existencia     -
-      3 - Calcula perimetro triangulo       -
-      4 - Calcula area do triangulo         -
-      5 - Sair                               -
-----
```

A opção 0 de imprimir todos os resultados apresenta uma opção mais rápida e simples de executar e apresentar os comandos de uma forma mais direta. Apertando esse opção a seguinte tela irá ser apresentada com informações pertinentes a esses pontos:

```
0
Esse triangulo existe!

Comprimento lado 1: 1.00
Comprimento lado 2: 1.41
Comprimento lado 3: 1.00

0 perimetro do triangulo é 3.41

Distancia 1 em calculo da área = 1.00
Distancia 2 em calculo da área = 1.41
Distancia 3 em calculo da área = 1.00
Semi-perimetro em calculo da área = 1.71
A área do triangulo é 0.50
```

Alguns casos de teste podem ser utilizados:

- 0 0 0 1 1 0 0 para um triangulo existente;
- 2 3 3 4 0 0 0 para outro triângulo existente;
- 1 1 2 0 1 1 0 para um triângulo inexistente.
- 1 1 2 2 3 3 0 para outro triângulo inexistente.

2) Escreva um programa que ordene uma lista de inteiros. A ordem de ordenação, bem como os inteiros a serem ordenados, deverão ser informados como parâmetros da linha de comando. Na ausência de uma switch (opção), a ordem deverá ser

crescente; caso o usuário indique a opção -d, a ordem deverá ser crescente e, se indicar -r, a ordem deverá ser decrescente.

- Na sua implementação deverá constar, no mínimo, dois arquivos .c e dois arquivos .h.
- Sua implementação deverá ser construída a partir de um Makefile.
- O vetor usado para armazenar temporariamente os inteiros recebidos pela linha de

comando deverá ser alocado dinamicamente e com quantidade de células adequada para a execução do programa.

O programa foi desenvolvido com o objetivo de ordenar de forma crescente e decrescente um conjunto de valores inteiros dinamicamente. As instruções de uso do programa estão presentes no README.md, O arquivo está estruturado da seguinte forma:

sort.c e sort.h - Responsáveis por funções de ordenação do vetor;
array.c e array.h - Responsáveis pela realização de funções de manipulação do array;
main.c - Função principal e impressão de algumas funções na tela;

O programa desenvolvido apresenta algumas limitações referentes a entradas, tais como:

- Não aceita o uso de caracteres;
- Aceita apenas as entradas de inteiros;

Para utilizar o programa, abra um terminal e pasta a qual se encontra os arquivos .c / .h, e realize o seguinte procedimento:

- **Para compilar :**

make

- **Para executar :** execute o programa com um dos seguintes comandos. As opções "./a" e "./a -d" são para utilizadas para a ordenação crescente de número inteiros. A opção "./a -r" é utilizada para ordenamento decrescente de números inteiros. Dessa forma, existem 3 possíveis formas de execução, as quais, realizam operações diferentes.

./a

./a -d

./a -r

Caso, seja utilizada uma das formas de execução já executadas o programa irá pedir para inserir a quantidade de números que se deseja inserir. Deve-se digitar esta quantidade.



Em seguida a operação irá ocorrer e será impresso os valores em ordem crescente ou decrescente como já foi discutido antes.

- **Em seguida, para excluir o executável, digite:**

`make clean`

Caso de teste:

Para opção `./a` ou `./a -d`:

Input: 5 1 4 45 6 56

Output: 1 4 6 45 56

Input : 3 12 3 4

Output: 3 4 12

Para opção `./a -r`:

Input: 5 1 2 3 4 5

Output: 5 4 3 2 1

Input: 3 -1 2 3

Output: 3 2 -1

Para opções inválidas (diferentes das opções anteriores):

Output: Somente é aceito os parâmetros '-d' e '-r'. Tente novamente !

Telas de exemplo:

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T1/2 - Lista em Ordem$ ./a -d
Insira a quantidade de valores que deseja ordenar: 3
-1 2 3

***** Valores em Ordem Crescente *****

-1
2
3
```

```
eduardo@Nautilus2:~/Documentos/GitE/SO_Trabalhos/T1/2 - Lista em Ordem$ ./a -r
Insira a quantidade de valores que deseja ordenar: 4
2 0 -20 100

***** Valores em Ordem Decrescente *****

100
2
0
-20
```

3) Escreva um programa em C que possua, como variáveis, os seguintes itens:

```
double number1 = 7.3, number2;
char s1[100], s2[100];
```

- Declare a variável dPtr como ponteiro para a variável do tipo double.
- Carregue o endereço da variável number1 no ponteiro dPtr .
- Imprima em tela o valor da variável apontada por dPtr partindo do ponteiro.
- Carregue o valor da variável apontada por dPtr na variável number2 .
- Imprima o valor armazenado em number2.
- Imprima o endereço de number1 na tela.
- Imprima o endereço armazenado em dPtr na tela. O valor impresso decorrente do enunciado que contempla o item anterior é igual ao valor do endereço gravado em dPtr?
R: Sim, o valor é igual.
- Leia uma string e a armazene no array s1. Copie a string armazenada em s1 para s2. Compare a string armazenada em s1 com a string armazenada em s2 e imprima o resultado na tela.
- Apense a string s2 à string s1. Imprima o resultado na tela.
- A execução do item anterior pode provocar algum erro em tempo de execução?
R: Pode, se a junção das duas strings superar o valor alocado de s2.
- Determine o comprimento da string armazenada em s1 e imprima o resultado na tela.

O programa contém uma estrutura que pode ser compilada utilizando um Makefile, além disso conta com um README.md para o usuário poder rodar o software. O programa trabalha com um conjunto de arquivos .h e .c, são eles:

- main.c e .h: Responsável pela execução inicial do programa

Para utilizar o software, o usuário deve executa-lo com os comandos encontrados no README.md:

```
eduardo@Nautilus2:~/Documentos/GitE/S0_Trabalhos/T1/3 - Variaveis$ make run
./prog
Valor apontado por dPtr: 7.30
Valor em number2 atualizado: 7.30
Endereco de number 1: 0x7ffd40d6b7d8
Endereco de dPtr: 0x7ffd40d6b7d8
Digite uma string de no máximo 100 caracteres: 
```

Uma opção de digitar uma string irá aparecer na tela para a resolução de um ponto do enunciado e o usuário deverá digitar uma string qualquer de tamanho máximo de 100 caracteres:

```
Digite uma string de no máximo 100 caracteres: abc
s1: abc
s2: abc
As strings são iguais
s2: abcabc
O tamanho da string 1 é: 3
```