



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Fundamentos de sistemas operacionais

Relatório do Trabalho 1



Brasília, DF
28 de março de 2017

LUCAS VILELA TAVEIRA BRILHANTE

FUNDAMENTOS DE SISTEMAS OPERACIONAIS | TRABALHO 01

Trabalho elaborado para disciplina de
Fundamentos de sistemas operacionais do curso de
Engenharia de Software da
Universidade de Brasília campus Gama.
Orientador: Prof. Thiago

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA

1. Ambiente

1.1. Sistema operacional

O sistema operacional utilizado foi a distribuição Linux Mint 11, derivado do ubuntu.

1.2. Desenvolvimento

Para o desenvolvimento foi utilizado a versão X do C, em conjunto com o compilador GCC versão X. Para escrever as linhas de código foi usado o sublime editor, makefile para juntar toda a compilação e a IDE Eclipse para obter linha de comando.

Foi usado também o git para controle de versão, para poder continuar o desenvolvimento de qualquer computador.

2. Os programas

Estrutura de arquivos:

```
3 directories
lucas@brilhante: ~/git/FS02017 $ tree
.
├── ex1
│   ├── estruturas.h
│   ├── exercicio1
│   ├── exercicio1.c
│   ├── exercicio1.o
│   ├── func.c
│   ├── func.h
│   ├── io.c
│   ├── io.h
│   └── makefile
├── ex2
│   ├── exercicio2
│   └── exercicio2.c
├── ex3
│   ├── exercicio3
│   ├── exercicio3.c
│   ├── exercicio3.o
│   ├── lista_encadeada.c
│   ├── lista_encadeada.h
│   └── makefile
└── README.md
3 directories, 18 files
```

Para utilização do código dentro dos diretórios ex1, ex2, ex3, deve-se utilizar o comando “make” para compilar e “make run” para executar.

2.1. Exercício 1

2.1.1. Telas

Para criação do programa foi usada a teoria matemática em que se a determinante de 3 pontos com 3 conjuntos de pontos diferentes derem com o mesmo sinal, significa que o polígono é convexo.

Assim que executado deve-se digitar as coordenadas do quadrilátero em sentido horário, começando de qualquer dos pontos. Terminando a computação, o resultado será “Não é convexo” caso não seja convexo, ou “É Convexo” seguido da Área calculada “Área = 2”, por exemplo.

Exemplo:

```
lucas@brilhantepc ~/git/FS02017/ex1 $ make run
./exerciciol
1 0
0 0
0 3
5 5
É convexo
Area:10.00
```

2.1.2. Limitações

Usuário deve ter makefile instalado para que a compilação seja possível.

Caso 3 pontos forem colineares, não existe quadrilátero.

So recebe numeros de até 64bits.

Não pode ser digitado no sentido anti-horário.

2.1.3. Testes

```
lucas@brilhantepc ~/git/FS02017/ex1 $ ./exerciciol
0 0
0 1
0.3 0.3
1 0
Não é convexo
```

2.2. Exercício 2

2.2.1. Telas

Para compilar o exercício 2 deve-se utilizar o comando “gcc exercicio2.c -o exercicio”. Para utilizar o executavel deve-se utilizar o comando “./exercicio2 par1 par2”, onde par1 e par2 representam onde os parametros são colocados.

```
lucas@brilhantepc ~/git/FS02017/ex2 $ ./exercicio2 2 34 -1 -a
# de parametros #5
Parametro #1 : 2
Parametro #2 : 34
Parametro #3 : -1
Parametro #4 : -a
```

2.2.2. Limitações

Se der Ctrl C Ctrl V os parametros serão contados como um só. Eles devem ser digitados.

2.2.3. Testes

```
lucas@brilhantepc ~/git/FS02017/ex2 $ ./exercicio2 -a bababa -30 33 21
# de parametros #6
Parametro #1 : -a
Parametro #2 : bababa
Parametro #3 : -30
Parametro #4 : 33
Parametro #5 : 21
```

2.3. Exercício 3

2.3.1. Telas

Para implementar o exercício 3 foi utilizado uma estrutura de dados de lista encadeada, dessa forma os objetos sempre poderiam ser alocados de forma dinâmica. Como não foi especificado performance, não fiz questão que ordenar enquanto incluía os elementos, ao invés, decidi adicionar um bubble sort que é de simples implementação.

Para compilar o programa deve-se executar o “make” e para rodar o “make run ARGS="-d” ” ou -r para ordenar decrescente.

Para utilizar deve-se inserir os números a serem ordenados.

```
lucas@brilhantepc ~/git/FS02017/ex3 $ make run ARGS="-d"
./exercicio3 -d
9
1
4
3
-1
1 3 4 9
```

2.3.2. Limitações

Apenas inteiros.

2.3.3. Testes

2.3.3.1. Crescente

```
lucas@brilhantepc ~/git/FS02017/ex3 $ make run ARGS="-d"
./exercicio3 -d
9
1
4
3
-1
1 3 4 9
```

2.3.3.2. Decrescente

```
lucas@brilhantepc ~/git/FS02017/ex3 $ make run ARGS="-r"
./exercicio3 -r
4
2
6
1
-1
6 4 2 1
```

2.3.3.3. Invalido

```
lucas@brilhantepc ~/git/FS02017/ex3 $ make run ARGS="-g"
./exercicio3 -g
1
6
5
9
-1
./exercicio3: invalid option -- 'g'
1 5 6 9
```

2.3.3.4. Números grande e vários.

```
lucas@brilhantepc ~/git/FS02017/ex3 $ make run ARGS="-d"
./exercicio3 -d
5
4
8
34
23
12
3
5
6
8
3
9
2
233333
122
15551
2323
123512
123
-1
2 3 3 4 5 5 6 8 8 9 12 23 34 122 123 2323 15551 123512 233333
```