

# Trabalho 01 - Comunicação Entre Processos

Adailson Pinho dos Santos - 13/0140724

Vitor Nere Araújo Ribeiro - 13/0137413

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Ambiente de desenvolvimento</b>	<b>3</b>
<b>3</b>	<b>Questão 01</b>	<b>3</b>
3.1	Raciocínio lógico . . . . .	3
3.2	Instrução de uso . . . . .	3
<b>4</b>	<b>Questão 02</b>	<b>5</b>
4.1	Raciocínio lógico . . . . .	5
4.2	Instrução de uso . . . . .	5

# 1 Introdução

O presente documento visa descrever as ferramentas utilizadas no desenvolvimento do trabalho, quais foram os sistemas operacionais utilizados, qual ambiente de desenvolvimento, quais foram as entradas e saídas dos softwares desenvolvidos e quais são as limitações dos software, este trabalho foi contruído por uma dupla de estudantes da disciplina de Fundamentos de Arquitetura de Computadores da Universidade de Brasília.

## 2 Ambiente de desenvolvimento

Para a execução do desenvolvimento das aplicações foram utilizados os Sistemas Operacionais Linux Mint 18.2 Sonya e Debian 9, sendo que ambos são distribuições populares do Linux.

Foi utilizada a ferramenta Make que funciona como um sistema de alvos e dependências. Dessa forma, estará definido dentro do arquivo Makefile quais arquivos ele irá processar para realizar determinada tarefa. Sendo assim, utilizou-se a versão GNU Make 4.1 e GCC versão 5.4 nos sistemas GNU/Linux.

Para o desenvolvimento da especificação do presente trabalho, utilizou-se o Texmaker 4.4.1, sendo este um programa que serviu para compilação e exportação do trabalho em PDF por meio da linguagem LaTeX.

## 3 Questão 01

### 3.1 Raciocínio lógico

A questão de número 1 do trabalho solicitava que fosse criado um programa na linguagem C, que em tempo de execução, receba do usuário 3 coordenadas cartesianas de pontos pertencentes à circunferência de um círculo e retorne o raio do círculo, coordenadas do seu centro e a área da circunferência caso seja realmente convexo. No enunciado da questão foi sugerido como instrução a criação de três arquivos .h e três arquivos .c, além do arquivo Makefile.

Após um estudo matemático do problema, foi chegado a conclusão que era necessário utilizar a fórmula da equação reduzida da circunferência. Esta equação é dada por:

$$r^2 = (x + a)^2 + (y - b)^2$$

Sendo que  $r$  é o raio da circunferência,  $x$  e  $y$  são as coordenadas do centro da circunferência nos valores  $a$  e  $b$  de cada ponto do plano cartesiano.

### 3.2 Instrução de uso

Para explicar como é o funcionamento do programa, será feita uma simulação realizada no terminal do Linux, com as inserções de entrada e a saída correspondente.

Primeiramente, o usuário deve entrar na pasta que contém todos os arquivos do projeto, após isso, digitar o comando "make run", para que, dessa forma, o Makefile possa instruir a criar os arquivos .o, realizar a comunicação entre os arquivos .c e .h e habilitar que o compilador GCC compile o programa final produzido por esses arquivos.

Abaixo, temos a imagem que mostra a inserção de 3 pontos de um plano cartesiano, são eles: A(1,0), B(0,1) e C(-1,0).

Após inserir os valores e prosseguir, a leitura dos valores será computada, calculada e impressa no terminal por arquivos e funções específicas. São eles: operacoes.c, operacoes.h, the\_io.c, the\_io.h, tipos\_compostos.c, tipos\_compostos.h.

Sendo assim, o resultado do cálculo será ilustrado como na imagem abaixo:

```
adailson@localhost:~/Documentos/FAC/FAC.2017.2/trabalho01 96x27
[adailson@localhost trabalho01]$ make run
gcc -c tipos_compostos.c tipos_compostos.h -Wall -W -pedantic -ansi -I. -lm
gcc -c operacoes.c operacoes.h -Wall -W -pedantic -ansi -I. -lm
gcc -c the_io.c the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -c main.c tipos_compostos.h operacoes.h the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -o main.out main.o tipos_compostos.o operacoes.o the_io.o -Wall -W -pedantic -ansi -I. -lm
./main.out
0 0
0 1
1 1

Raio: 0.707107
Centro: (0.500000, 0.500000).
[adailson@localhost trabalho01]$ make run
gcc -c tipos_compostos.c tipos_compostos.h -Wall -W -pedantic -ansi -I. -lm
gcc -c operacoes.c operacoes.h -Wall -W -pedantic -ansi -I. -lm
gcc -c the_io.c the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -c main.c tipos_compostos.h operacoes.h the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -o main.out main.o tipos_compostos.o operacoes.o the_io.o -Wall -W -pedantic -ansi -I. -lm
./main.out
0 0
0 1
-1 0

Raio: 0.707107
Centro: (-0.500000, 0.500000).
[adailson@localhost trabalho01]$
```

Caso os pontos sejam colineares, o sistema retorna a seguinte resposta:

```
adailson@localhost:~/Documentos/FAC/FAC.2017.2/trabalho02 96x27
[adailson@localhost trabalho02]$ gcc give_me_parameters.c
[adailson@localhost trabalho02]$ ./a.out -a bababa -30 33 21

# de parametros: 5
Nome do executavel: a.out
Parametro #1: -a
Parametro #2: bababa
Parametro #3: -30
Parametro #4: 33
Parametro #5: 21

[adailson@localhost trabalho02]$ █
```

## 4 Questão 02

### 4.1 Raciocínio lógico

A questão de número 2 solicita que fosse criado um programa na linguagem C, que ao ser executado informasse a quantidade de parâmetros enviados no terminal. A exibição dos parâmetros deve ser numerada pela posição da variável. Esses parâmetros são obtidos pelos argumentos `argc` e `argv` injetadas na `main`. `Argc` informa a quantidade de parâmetros inseridos e `argv` contém um vetor de string com os parâmetros.

### 4.2 Instrução de uso

Primeiramente, o usuário deve entrar na pasta que contém todos os arquivos do projeto, após isso, digitar o comando `"gcc -o gmp.out give_me_parameters.c"` e executar o programa com os parâmetros que desejar na frente. Ex: `"./gmp param1 param2 param3"`

```
adailson@localhost:~/Documentos/FAC/FAC.2017.2/trabalho01 96x27
[adailson@localhost trabalho01]$ make run
gcc -c tipos_compostos.c tipos_compostos.h -Wall -W -pedantic -ansi -I. -lm
gcc -c operacoes.c operacoes.h -Wall -W -pedantic -ansi -I. -lm
gcc -c the_io.c the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -c main.c tipos_compostos.h operacoes.h the_io.h -Wall -W -pedantic -ansi -I. -lm
gcc -o main.out main.o tipos_compostos.o operacoes.o the_io.o -Wall -W -pedantic -ansi -I. -lm
./main.out
0 0
0 1
0 -1

Circulo nao viavel.
[adailson@localhost trabalho01]$
```