



UNIVERSIDADE  
FEDERAL DO CEARÁ

# Trabalho de Sistemas Operacionais I

Aluno : Pedro Anderson Costa Martins

Matrícula : 402454

Curso : EC

## - Descrição do ambiente de desenvolvimento

Na imagem a seguir consta dados do notebook utilizado e da versão do compilador GCC.

```
Atividades Terminal ter, 21:56
pedro@pedro-S14BW01: ~/Área de Trabalho/UFC_Pedro/SO/Trabalho 1
Arquivo Editar Ver Pesquisar Terminal Ajuda
pedro@pedro-S14BW01:~/Área de Trabalho/UFC_Pedro/SO/Trabalho 1$ neofetch
      .-/+oossssoo+/-.
      `:+ssssssssssss++:`
      -+ssssssssssssssyyss+-
      .ossssssssssssssdMMNNysssso.
      /ssssssssssshdmmNNmyNNMMNhs/
      +ssssssssshmydMMNNMNdddyssssss+
      /ssssssshNNMMYhhyyyhNNMMNhs/
      .sssssssdMMNNhsSSSSSSshNNMMdSSSSSS.
      +ssshhhyNNMMySSSSSSSSSYNNMMySSSSSS+
      ossyNNMMNymMhSSSSSSSSshmmhSSSSSSso
      ossyNNMMNymMhSSSSSSSSshmmhSSSSSSso
      +ssshhhyNNMMySSSSSSSSSYNNMMySSSSSS+
      .sssssssdMMNNhsSSSSSSshNNMMdSSSSSS.
      /ssssssshNNMMYhhyyyhdNNMMNhs/
      +sssssssdnydMMNNMNdddyssssss+
      /ssssssssshdmmNNmyNNMMNhs/
      .ossssssssssssssdMMNNysssso.
      -+ssssssssssssssyyss+-
      `:+ssssssssssss++:`
      .-/+oossssoo+/-.

OS: Ubuntu 17.10 x86_64
Host: S14BW01 1.20.X
Kernel: 4.13.0-46-generic
Uptime: 4 hours, 58 mins
Packages: 1711
Shell: bash 4.4.12
Resolution: 1366x768
DE: ubuntu:GNOME
WM: GNOME Shell
WM Theme: Adwaita
Theme: Ambiance [GTK3]
Icons: Ubuntu-mono-dark [GTK3]
Terminal: gnome-terminal
CPU: Intel Celeron N3060 (2) @ 2.480GHz
GPU: Intel Integrated Graphics
Memory: 2723MiB / 3800MiB

pedro@pedro-S14BW01:~/Área de Trabalho/UFC_Pedro/SO/Trabalho 1$ gcc --version
gcc (Ubuntu 7.2.0-8ubuntu3.2) 7.2.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

pedro@pedro-S14BW01:~/Área de Trabalho/UFC_Pedro/SO/Trabalho 1$
```

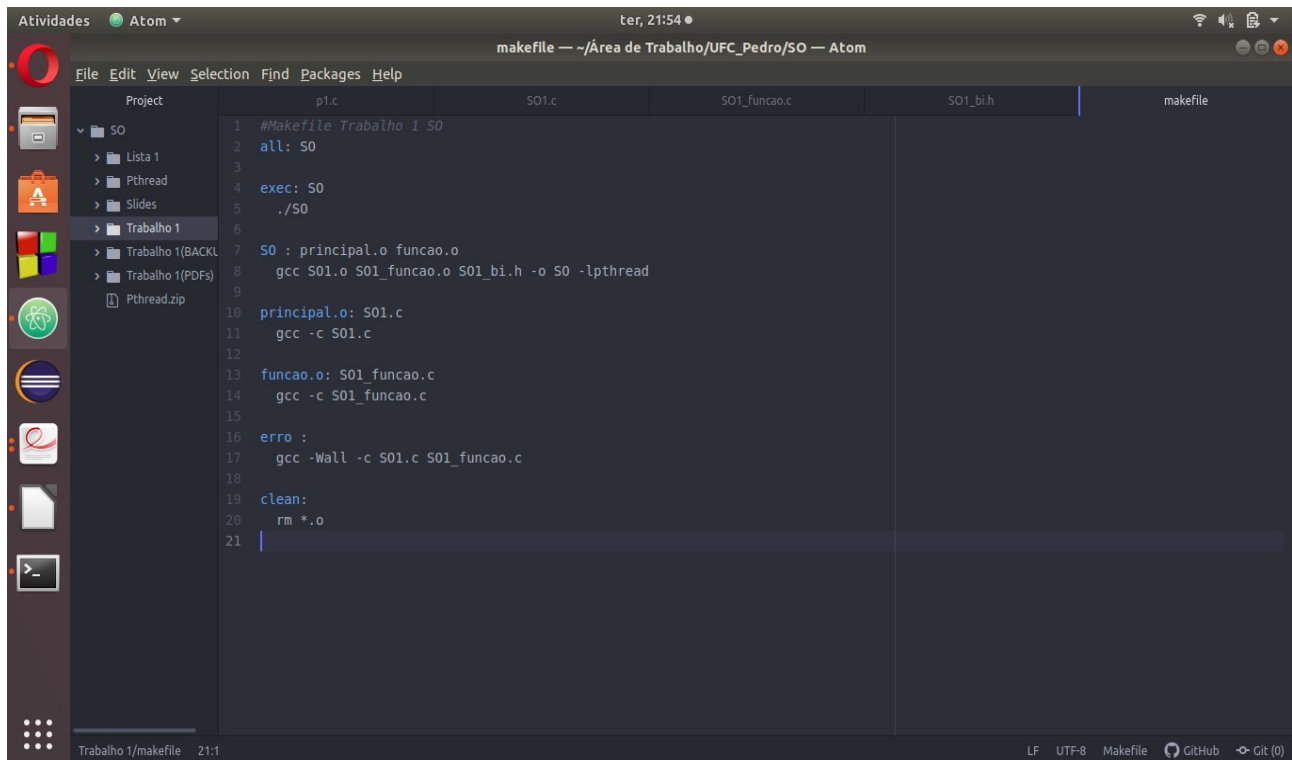
## - Instruções para compilação/execução :

O programa principal cria uma fila F e 10 threads para receber e ler os 10 arquivos gerados pelo programa p1.c , a cada processo de leitura ele inclui os valores do arquivo na fila. Após o preenchimento total da fila F ele imprime a fila desordenada mostrando o primeiro e último valor da fila e o programa pergunta ao usuário a quantidade de threads que o mesmo deseja ordenar a fila F. após o usuário decidir, o programa imprime toda a fila ordenada, o tempo de execução da ordenação, o primeiro e último valor da fila.

O trabalho foi dividido em 2 partes. A primeira é o programa “**p1.c**” que cria 10 arquivos binários com valores aleatórios, cada arquivo possui o tamanho de 4 kb. Para compilá-lo é necessário encontrar o diretório e usar o comando **gcc p1.c -o “nome\_do\_executável”** e para executá-lo **./”nome\_do\_executável”**. Exemplo :

- Compilação : **gcc p1.c -o gerar\_arquivos**
- Execução : **./gerar\_arquivos**

A segunda parte consiste no programa principal que dividi em 3 parte : **SO1.c** (Onde fica a função main) , **SO1\_funcao.c** (Onde ficam as funções do programa) , **SO1\_bi.h** (Onde ficam os protótipos de todas as funções). Para melhor organização o makefile foi utilizado, logo, para utilizar comandos de compilação e execução do programa é necessário usar os comandos a seguir :



```
1 #Makefile Trabalho 1 SO
2 all: SO
3
4 exec: SO
5     ./SO
6
7 SO : principal.o funcao.o
8     gcc SO1.o SO1_funcao.o SO1_bi.h -o SO -lpthread
9
10 principal.o: SO1.c
11     gcc -c SO1.c
12
13 funcao.o: SO1_funcao.c
14     gcc -c SO1_funcao.c
15
16 erro :
17     gcc -Wall -c SO1.c SO1_funcao.c
18
19 clean:
20     rm *.o
21
```

- Opção exec : Executa o programa executável criado pela opção SO.
- Opção SO : Monta os programas-objeto para formar o executável.
- Opção Principal : Cria o objeto do programa SO1.c
- Opção função : Cria o objeto do programa SO1\_funcao.c.
- Opção erro : Verifica se existe erro nos programas.
- Opção clean : Limpa todos os programas-objeto do diretório.

Para utilizar o makefile é necessário encontrar o diretório e utilizar o comando “**make**” para compilar e “**make exec**” para executar;

Exemplo :

- Compilação : **~terminal\$ make**  
**...compila...**
- Execução : **~terminal\$ make exec**  
**...executa...**