

## Relatório EP1

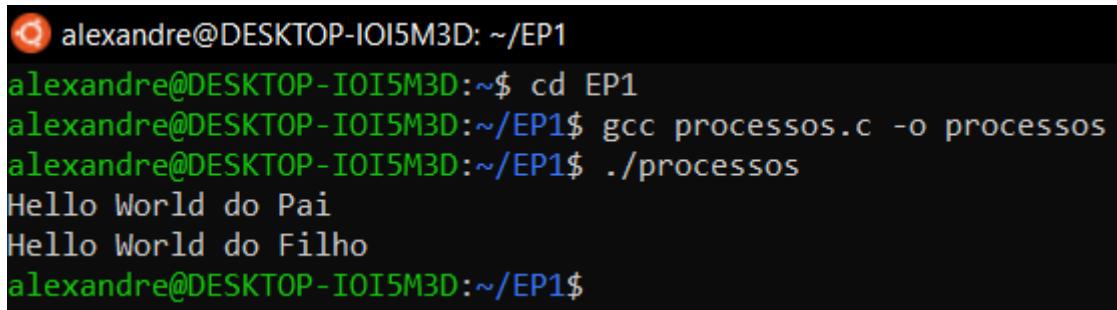
ACH2044 - Sistemas Operacionais  
Profª. Gisele S. Craveiro - Turma 94/2020

Alexandre Kenji Okamoto - 11208371  
Mirela Mei - 11208392

Para compilar e executar os programas, utilizamos um notebook com Windows 10 (versão 10.0.19041 - compilação 19041) com Subsistema do Windows para Linux 2 (WSL 2) com a distribuição Ubuntu 20.04.1 LTS.

### 1 – Processos

Para compilar foi utilizado o comando “gcc processos.c -o processos” e para executar o comando “./processos”.



```
alexandre@DESKTOP-IOI5M3D: ~/EP1
alexandre@DESKTOP-IOI5M3D:~$ cd EP1
alexandre@DESKTOP-IOI5M3D:~/EP1$ gcc processos.c -o processos
alexandre@DESKTOP-IOI5M3D:~/EP1$ ./processos
Hello World do Pai
Hello World do Filho
alexandre@DESKTOP-IOI5M3D:~/EP1$
```

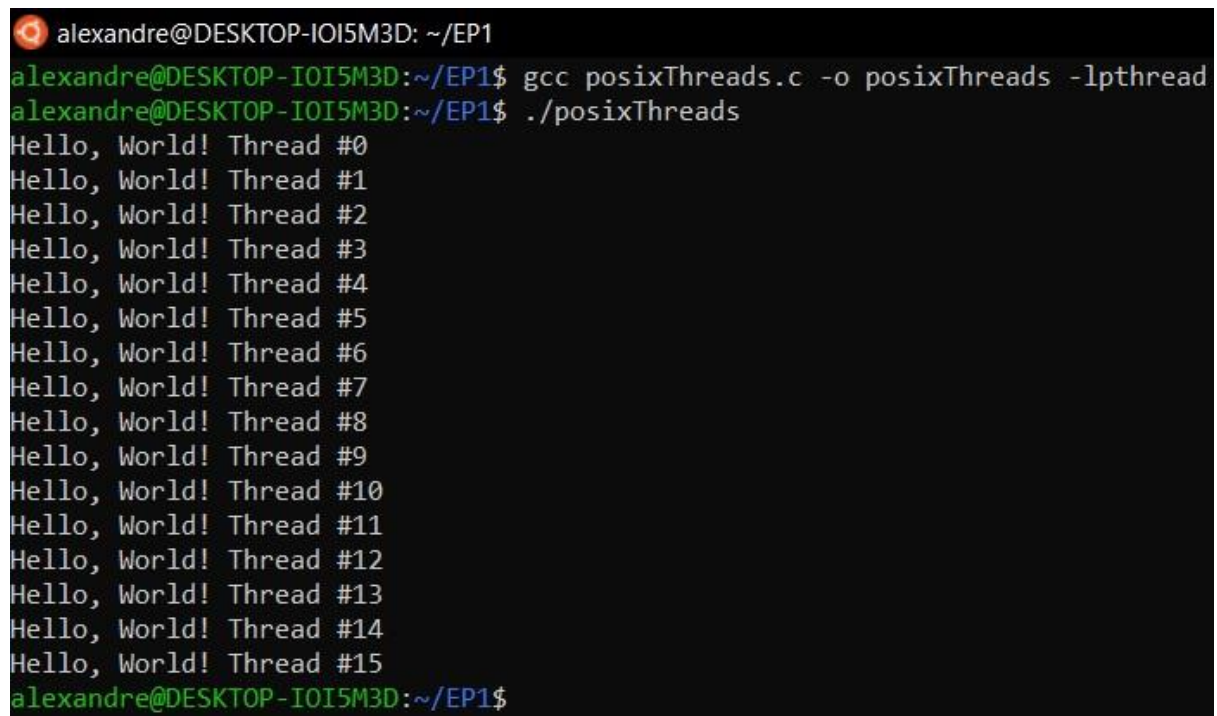
## 2 – Java Threads

Para compilar foi utilizado o comando “\*.java” e para executar o comando “java testaThread”.

```
alexandre@DESKTOP-IOI5M3D: ~/EP1
alexandre@DESKTOP-IOI5M3D:~/EP1$ javac *.java
alexandre@DESKTOP-IOI5M3D:~/EP1$ java testaThread
Thread 2 Hello World! 0
Thread 1 Hello World! 0
Thread 3 Hello World! 0
Thread 1 Hello World! 1
Thread 1 Hello World! 2
Thread 2 Hello World! 1
Thread 1 Hello World! 3
Thread 1 Hello World! 4
Thread 3 Hello World! 1
Thread 1 Hello World! 5
Thread 2 Hello World! 2
Thread 2 Hello World! 3
Thread 3 Hello World! 2
Thread 2 Hello World! 4
Thread 3 Hello World! 3
Thread 2 Hello World! 5
Thread 3 Hello World! 4
Thread 3 Hello World! 5
alexandre@DESKTOP-IOI5M3D:~/EP1$
```

### 3 – POSIX Threads

Para compilar foi utilizado o comando “gcc posixThreads.c -o posixThreads -lpthread” e para executar o comando “./posixThreads”.

A terminal window with a black background and green text. The prompt is 'alexandre@DESKTOP-IOI5M3D: ~/EP1'. The first command is 'gcc posixThreads.c -o posixThreads -lpthread'. The second command is './posixThreads'. The output consists of 16 lines, each saying 'Hello, World! Thread #' followed by a number from 0 to 15. The prompt returns to 'alexandre@DESKTOP-IOI5M3D: ~/EP1\$'.

```
alexandre@DESKTOP-IOI5M3D: ~/EP1$ gcc posixThreads.c -o posixThreads -lpthread
alexandre@DESKTOP-IOI5M3D: ~/EP1$ ./posixThreads
Hello, World! Thread #0
Hello, World! Thread #1
Hello, World! Thread #2
Hello, World! Thread #3
Hello, World! Thread #4
Hello, World! Thread #5
Hello, World! Thread #6
Hello, World! Thread #7
Hello, World! Thread #8
Hello, World! Thread #9
Hello, World! Thread #10
Hello, World! Thread #11
Hello, World! Thread #12
Hello, World! Thread #13
Hello, World! Thread #14
Hello, World! Thread #15
alexandre@DESKTOP-IOI5M3D: ~/EP1$
```

Nossas referências foram:

A chamada de sistema fork() - Como criar e Gerenciar Processos do site Programação Progressiva (<https://www.programacaoprogessiva.net/2014/09/A-Chamada-de-Sistema-fork-Como-Criar-e-Gerenciar-Processos.html>)

Tutorial processos no Linux - Comando fork() - pt-BR de Bruno Sampaio Pinho da Silva (<https://www.youtube.com/watch?v=3VgLkCqqKWo>)

Curso de Java 68: Threads: Interface Runnable de Loiane Groner (<https://www.youtube.com/watch?v=oWoU0uTEaA0>)

POSIX Threads Programming de Blaise Barney (<https://computing.llnl.gov/tutorials/pthreads/#Pthread>)

How to create a simple thread in C do site Educative (<https://www.educative.io/edpresso/how-to-create-a-simple-thread-in-c>)

Threads - Linguagem C de Duca Siqueira (<https://www.youtube.com/watch?v=cwT3EJJQhlo>)

Tutorial Thread em C pt-BR de Bruno Sampaio Pinho da Silva  
(<https://www.youtube.com/watch?v=CyIpD8zXHZA>)

Livro Fundamentos de Sistemas Operacionais de Abraham Silberschatz, Peter B. Galvin e Greg Gagne, 9ª edição