

## Laboratório 1

Aluno: Lucas Favilla Ferreira Alves da Silva DRE:119156518

### Atividade 1 -

```
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ ./hello
--Cria a thread 0
--Cria a thread 1
Hello World
Hello World
--Cria a thread 2
--Cria a thread 3
Hello World
--Cria a thread 4
Hello World
--Cria a thread 5
Hello World
--Cria a thread 6
Hello World
--Cria a thread 7
Hello World
--Cria a thread 8
Hello World
--Cria a thread 9
Hello World
--Thread principal terminou
Hello World
```

Sim, isso ocorre porque nesse programa não há nenhum mecanismo de controle de execução dessas threads, assim essa ordem de execução está completamente a merce do Sistema Operacional.

### Atividade 2-

```
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ gcc hello_arg.c -o hello -Wall -lpthread
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ ./hello
--Cria a thread 0
--Cria a thread 1
Hello World da thread: 0
Hello World da thread: 1
--Cria a thread 2
--Cria a thread 3
Hello World da thread: 2
--Cria a thread 4
Hello World da thread: 3
--Cria a thread 5
--Cria a thread 6
Hello World da thread: 5
Hello World da thread: 4
--Cria a thread 7
Hello World da thread: 6
--Cria a thread 8
Hello World da thread: 7
--Cria a thread 9
Hello World da thread: 8
--Thread principal terminou
Hello World da thread: 9
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ ./hello
--Cria a thread 0
--Cria a thread 1
Hello World da thread: 0
Hello World da thread: 1
--Cria a thread 2
--Cria a thread 3
Hello World da thread: 2
--Cria a thread 4
Hello World da thread: 3
--Cria a thread 5
Hello World da thread: 4
--Cria a thread 6
Hello World da thread: 5
--Cria a thread 7
Hello World da thread: 6
--Cria a thread 8
Hello World da thread: 7
--Cria a thread 9
Hello World da thread: 8
--Thread principal terminou
Hello World da thread: 9
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$
```

A diferença desse para o outro programa é a passagem de argumento para a função `pthread_create`, com isso foi possível numerar as `printf` das threads e demonstrar melhor qual thread foi executada em qual momento no terminal. Como demonstrado acima.

### Atividade 3-

```
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ ./hello
--Aloca e preenche argumentos para thread 0
--Cria a thread 0
--Aloca e preenche argumentos para thread 1
--Cria a thread 1
Sou a thread 0 de 10 threads
--Aloca e preenche argumentos para thread 2
--Cria a thread 2
Sou a thread 1 de 10 threads
--Aloca e preenche argumentos para thread 3
--Cria a thread 3
Sou a thread 2 de 10 threads
--Aloca e preenche argumentos para thread 4
--Cria a thread 4
Sou a thread 3 de 10 threads
--Aloca e preenche argumentos para thread 5
--Cria a thread 5
Sou a thread 4 de 10 threads
--Aloca e preenche argumentos para thread 6
--Cria a thread 6
Sou a thread 5 de 10 threads
--Aloca e preenche argumentos para thread 7
--Cria a thread 7
Sou a thread 6 de 10 threads
Sou a thread 7 de 10 threads
--Aloca e preenche argumentos para thread 8
--Cria a thread 8
--Aloca e preenche argumentos para thread 9
--Cria a thread 9
--Thread principal terminou
Sou a thread 8 de 10 threads
Sou a thread 9 de 10 threads
```

Sim, o programa funcionou da maneira esperada, ele conseguiu pegar mais de um argumento dentro da função a ser executada pela thread através da utilização de uma struct passada por referencia no 4º argumento da função pthread\_create. Como demonstrado acima.

### Atividade 4 -

A diferença consiste que a thread principal esta sempre terminando por último, isto é a thread principal está esperando todas as outras terminarem seus processos antes de acabar com o dele próprio, diferentemente dos outros que poderiam ocorrer de a thread principal terminar antes de as outras threads acabarem de ser executadas. Como demonstrado ao lado, pela linha “-- Thread principal terminou” por último

```
lks@Favsnote:~/Área de Trabalho/Facul/2021.1/CompConc/cods-mod1-lab1$ ./hello
--Aloca e preenche argumentos para thread 0
--Cria a thread 0
--Aloca e preenche argumentos para thread 1
--Cria a thread 1
Sou a thread 0 de 10 threads
--Aloca e preenche argumentos para thread 2
--Cria a thread 2
Sou a thread 1 de 10 threads
--Aloca e preenche argumentos para thread 3
--Cria a thread 3
Sou a thread 2 de 10 threads
--Aloca e preenche argumentos para thread 4
--Cria a thread 4
Sou a thread 3 de 10 threads
--Aloca e preenche argumentos para thread 5
--Cria a thread 5
Sou a thread 4 de 10 threads
--Aloca e preenche argumentos para thread 6
--Cria a thread 6
Sou a thread 5 de 10 threads
--Aloca e preenche argumentos para thread 7
--Cria a thread 7
Sou a thread 6 de 10 threads
--Aloca e preenche argumentos para thread 8
--Cria a thread 8
Sou a thread 7 de 10 threads
--Aloca e preenche argumentos para thread 9
--Cria a thread 9
Sou a thread 8 de 10 threads
Sou a thread 9 de 10 threads
--Thread principal terminou
```