

João Vitor Moraski Lunkes

## **Laboratório 03**

### **Manipulação de Threads**

Relatório técnico de atividade prática solicitado pelo professor Rodrigo Campiolo na disciplina de Sistemas Operacionais do Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Universidade Tecnológica Federal do Paraná – UTFPR

Bacharelado em Ciência da Computação – BCC

Campo Mourão

Abril / 2025

# Sumário

1	Introdução . . . . .	3
2	Configurações de Hardware . . . . .	4
2.1	Hardware . . . . .	4
3	Parte 1 - Manipulação de Threads . . . . .	4
3.1	Questão 01 . . . . .	4
3.2	Questão 02 . . . . .	4
3.3	Questão 03 . . . . .	4
4	Parte 2 - Programação . . . . .	5
4.1	Questão 01 . . . . .	5
4.2	Questão 02 . . . . .	6
	<b>Referências . . . . .</b>	<b>6</b>

## 1 Introdução

A evolução dos computadores tem elevado cada dia mais o nível de processamento de nossas máquinas. Um dos principais fatores, que deixa isso evidente, é a capacidade de executar diversos processos de forma paralela. O que permitiu esse multiprocessamento foram as Threads do processador, que são resumidamente, a divisão do processo principal de um programa dentro do sistema operacional.

Buscamos através de testes práticos e implementações, observar o comportamento das Threads no sistema operacional Linux, fazendo com que o entendimento acerca do tema fosse aprofundado ao observar processos reais em execução.

## 2 Configurações de Hardware

### 2.1 Hardware

Processador: Inter I7-1165G7 (8)<sup>TM</sup>4.700GHz 11th Gen

Memória RAM: 16 GB

Armazenamento em SSD

Sistema 64 bits

## 3 Parte 1 - Manipulação de Threads

### 3.1 Questão 01

Identifique no seu sistema Linux quantas threads estão em execução? Qual o processo com o maior número de threads?.

**Resposta:** No momento em que foi feita a verificação utilizando o comando *ps -eLo pid,tid | wc -l*, foi obtido um resultado de 1483, como mostrado na Figura 1

Como indicado na figura, o processo com maior numero de threads era o processo de PID 90085, que com o comando *ps -p 90085 -o comm=* foi possível descobrir que era a IDE de da JetBrains, conhecida como CLion.

### 3.2 Questão 02

Qual o número máximo de threads que o seu sistema suporta?.

**Resposta:** A verificação foi feita utilizando o comando *ulimit -u*, foi obtido um resultado de 62638, como mostrado na Figura 1

### 3.3 Questão 03

Verifique o tempo de execução do programa da questão 2, parte 2, considerando: 1, 2, 4, 8 e 16 threads. Descreva o hardware (processador, memória e número de núcleos, tamanho da matriz usada nos testes e o tempo de execução para cada teste).

Resposta: a execução de cada teste foi feita na Figura 3

```
magalu@wlx1099sup:~  
└─magalu@wlx1099sup ~  
$ ps -elo pid,tid | wc -l  
1483  
└─magalu@wlx1099sup ~  
$ ps -elf | awk '{print $2}' | sort | uniq -c | sort -nr | head  
128 90085  
57 90175  
51 90563  
37 7121  
34 4757  
30 4426  
27 4786  
24 7162  
22 4607  
19 91995  
└─magalu@wlx1099sup ~  
$ ulimit -u  
62638  
└─magalu@wlx1099sup ~  
$ ps -p 90085 -o comm=  
clion  
└─magalu@wlx1099sup ~  
$
```

Figura 1 – Execução dos comandos da parte 01

## 4 Parte 2 - Programação

### 4.1 Questão 01

```
└─magalu@wlx1099sup ~/pessoal/faculdade/sistemas-operacionais/lab03/codigos <master>  
└─$ make run01  
gcc ativ01.c -o ativ  
./ativ 4 4  
0 elemento 4 foi encontrado pela thread de número: 1 na posição 29  
0 elemento 4 foi encontrado pela thread de número: 2 na posição 64  
└─magalu@wlx1099sup ~/pessoal/faculdade/sistemas-operacionais/lab03/codigos <master>  
└─$
```

Figura 2 – Execução da atividade 01

## 4.2 Questão 02

```
magalu@wlx1099sup ~/pessoal/faculdade/sistemas-operacionais/lab03/codigos <master>  
$ make run02  
gcc ativ02.c ../matriz/matriz.c -o ativ -lpthread -lm  
./ativ 1 10000 10000  
Numero de threads: 1  
Tempo percorrido nas operações de calculo com 1 threads: 7s  
./ativ 2 10000 10000  
Numero de threads: 2  
Tempo percorrido nas operações de calculo com 2 threads: 7s  
./ativ 4 10000 10000  
Numero de threads: 4  
Tempo percorrido nas operações de calculo com 4 threads: 7s  
./ativ 8 10000 10000  
Numero de threads: 8  
Tempo percorrido nas operações de calculo com 8 threads: 8s  
./ativ 16 10000 10000  
Numero de threads: 16  
Tempo percorrido nas operações de calculo com 16 threads: 8s  
magalu@wlx1099sup ~/pessoal/faculdade/sistemas-operacionais/lab03/codigos <master>  
$
```

Figura 3 – Execução dos comandos da parte 01