

Diogo Rodrigues - 2380242
Marcos Vinicius de Quadros - 2380560
Carolina Yumi Fujii - 2335468

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
Departamento Acadêmico de Computação – DACOM
Bacharelado em Ciência da Computação – BCC

Campo Mourão
Março / 2023

Sumário

1	Introdução	3
2	Objetivos	3
3	Materiais	3
4	Procedimentos e Resultados	3
4.1	Identifique no seu sistema Linux quantas threads estão em execução? Qual o processo com o maior número de threads?	3
4.2	Qual o número máximo de threads que o seu sistema suporta?	4
5	Conclusões	5

1 Introdução

Este relatório tem como objetivo, abordar a respeito de threads no linux, utilizando programas em C/C++ ou pelo terminal Linux.

2 Objetivos

- Compreender as principais operação usadas em threads.
- Desenvolver aplicações usando threads
- Explorar programação com threads POSIX (pthreads)

3 Materiais

- Distribuição Linux.
- Ambiente de desenvolvimento para C/C++.
- Comandos do sistema e bibliotecas de programação.

4 Procedimentos e Resultados

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

Utilizando o comando "top -H", é possível exibir o total de threads no sistema. No exemplo, foi verificado que havia um total de 1161 threads em execução. Além disso, o comando também apresenta a quantidade de threads por processo, representada pela coluna "nTH". Dessa forma, foi possível observar que mais de um processo possuía 141 threads, sendo esse o maior número de threads encontrado, conforme pode ser visto na Figura 1.

```
top - 11:59:43 up 2:49, 1 user, load average: 0,30, 1,27, 1,38
Threads: 1161 total, 1 em exec., 1159 dormindo, 1 parado, 0 zumbi
%Cpu(s): 7,6 us, 6,5 sy, 0,0 ni, 85,4 id, 0,0 wa, 0,0 hi, 0,5 si, 0,0 st
MB mem : 3923,3 total, 226,0 livre, 2597,8 usados, 1099,4 buff/cache
MB swap: 3220,0 total, 2153,6 livre, 1066,4 usados, 946,4 mem dispon.
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPO+	COMANDO	nTH
1552	marcos	9	-11	1956124	13544	10308	S	8,9	0,3	0:02.99	pulseaudio	4
1705	marcos	20	0	5345176	417480	91100	S	6,6	10,4	10:03.33	gnome-shell	18
1740	marcos	20	0	5345176	417480	91100	S	6,6	10,4	6:24.32	llvmpipe-2	18
1738	marcos	20	0	5345176	417480	91100	S	6,3	10,4	6:46.86	llvmpipe-0	18
1739	marcos	20	0	5345176	417480	91100	S	6,3	10,4	6:05.77	llvmpipe-1	18
10706	marcos	20	0	38,8g	311556	73004	S	5,0	7,8	2:28.23	Isolated Web Co	23
2271	marcos	20	0	12,3g	487120	143664	S	1,7	12,1	3:23.82	firefox	141
24980	root	20	0	0	0	0	I	1,7	0,0	0:18.85	kworker/u6:3-even+	1
1650	marcos	-6	0	1956124	13544	10308	S	1,3	0,3	0:12.52	alsa-sink-Intel	4
1749	marcos	20	0	5345176	417480	91100	S	1,0	10,4	0:25.71	gnome-shell	18
5580	marcos	20	0	2699020	209056	85320	S	1,0	5,2	1:01.01	Isolated Web Co	21
7645	marcos	20	0	578300	38992	27176	S	1,0	1,0	0:19.91	gnome-terminal-	4
8128	marcos	20	0	2661192	120340	49672	S	1,0	3,0	0:52.94	Isolated Web Co	20
10719	marcos	20	0	38,8g	311556	73004	S	1,0	7,8	0:10.33	TaskCon~ller #1	23
2048	marcos	20	0	213332	35432	31196	S	0,7	0,9	3:27.71	Xwayland	1
2378	marcos	20	0	12,3g	487120	143664	S	0,7	12,1	2:08.61	IPC I/O Parent	141
10720	marcos	20	0	38,8g	311556	73004	S	0,7	7,8	0:07.62	TaskCon~ller #2	23
25830	marcos	20	0	23112	5204	3444	R	0,7	0,1	0:00.30	top	1
210	root	20	0	0	0	0	S	0,3	0,0	0:03.60	jbd2/sda3-8	1
1106	root	20	0	304380	1584	1312	S	0,3	0,0	0:00.33	VBoxService	9
3747	marcos	20	0	5345176	417480	91100	S	0,3	10,4	0:00.20	threaded-ml	18
1914	marcos	20	0	873944	12364	8640	S	0,3	0,3	0:00.67	gsd-media-keys	4
2259	marcos	20	0	227512	1000	756	S	0,3	0,0	0:57.04	VBoxClient	5
2432	marcos	20	0	12,3g	487120	143664	S	0,3	12,1	1:56.19	Softwar~cThread	141
2490	marcos	20	0	12,3g	487120	143664	S	0,3	12,1	2:08.16	WRRende~ckend#1	141
8143	marcos	20	0	2661192	120340	49672	S	0,3	3,0	0:01.22	TaskCon~ller #2	20
8556	marcos	20	0	2658348	59384	46944	S	0,3	1,5	0:00.52	JS Watchdog	18
9914	marcos	20	0	2802544	38684	29648	S	0,3	1,0	0:03.55	gjs	7
10715	marcos	20	0	38,8g	311556	73004	S	0,3	7,8	0:07.04	Timer	23

Figura 1 – Threads em execução

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

Para identificar o número máximo de threads suportado pelo sistema, foi utilizado o comando "ulimit -a", que exibi várias limitações do sistema incluindo o número máximo de threads permitidos que está nomeado como "max user processes". O resultado pode ser observado na FIGURA 2.

```
diogo@diogo-pc:~$ ulimit -a
real-time non-blocking time (microseconds, -R) unlimited
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
scheduling priority (-e) 0
file size (blocks, -f) unlimited
pending signals (-i) 14520
max locked memory (kbytes, -l) 474596
max memory size (kbytes, -m) unlimited
open files (-n) 1024
pipe size (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
real-time priority (-r) 0
stack size (kbytes, -s) 8192
cpu time (seconds, -t) unlimited
max user processes (-u) 14520
virtual memory (kbytes, -v) unlimited
file locks (-x) unlimited
```

Figura 2 – Numero maximo de threads permitidos

5 Conclusões

Com esse laboratório, foi possível observar que, as threads podem auxiliar atividades que possuem diversas tarefas, podendo separalas para que cada thread execute uma tarefa específica.