

Resolução Laboratório 01 - Fork

Alunas: Andreza Fernandes de Oliveira - 384341
Arina de Jesus Amador Monteiro Sanches - 392476

02 QUESTÃO.

ITEM a) Os parâmetros utilizados para execução foram:

- **início = 1**
- **fim = 20**

```
andreza@andrezavm:~/Área de Trabalho/Operational-systems/Lab1Fork$ ./1b  
  
Posição Inicial: 1  
  
Posição Final: 20  
  
Eu sou o processo 6115, filho do processo 6114. Fiquei encarregado de imprimir  
essa parte da Sequência de Fibonnaci  
0 1 1 2 3 5 8 13 21 34  
Olá, eu sou o pai 6114, o meu filho é 6115  
  
Eu sou o processo 6116, filho do processo 6084. Fiquei encarregado de imprimir  
essa parte da Sequência de Fibonnaci  
55 89 144 233 377 610 987 1597 2584 4181  
Olá, eu sou o pai 6084, o meu filho é 6116  
andreza@andrezavm:~/Área de Trabalho/Operational-systems/Lab1Fork$
```

ITEM b) Utilizamos aqui parâmetros de entrada que pudessem “sobrecarregar” o programa, priorizando assim o processo para que pudéssemos vê-lo utilizando o comando **top**. Também utilizamos o **sleep()** com parâmetro **50**.

- **início = 1**
- **fim = 99999999**

```
top - 00:24:23 up 4:30, 1 user, load average: 4,60, 2,92, 1,41
Tarefas: 173 total, 4 em exec., 133 dormindo, 1 parado, 2 zumbi
%CPU(s): 56,5 us, 43,5 sis, 0,0 ni, 0,0 oc, 0,0 ag, 0,0 ih, 0,0 is 0,0 tr
KB mem : 1949156 total, 133576 livre, 1109424 usados, 706156 buff/cache
KB swap: 483800 total, 445232 livre, 38568 usados, 651300 mem dispon.
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPO+	COMANDO
6359	andreza	20	0	14016	2040	1896	R	28,3	0,1	1:04.58	1b
18043	andreza	20	0	808180	40204	27620	R	27,4	2,1	3:59.04	gnome-terminal-
920	andreza	20	0	3069548	438148	61732	R	20,1	22,5	31:24.71	gnome-shell
6189	root	20	0	0	0	0	I	14,0	0,0	0:16.39	kworker/u2:0
711	andreza	20	0	463772	90032	24096	S	6,4	4,6	10:44.83	Xorg
1086	andreza	20	0	380120	11960	8044	S	2,2	0,6	0:58.65	ibus-daemon
6397	andreza	20	0	51332	4260	3584	R	1,0	0,2	0:00.21	top
1268	andreza	20	0	818444	22404	15288	S	0,3	1,1	0:06.59	gsd-color
1	root	20	0	225552	7108	4924	S	0,0	0,4	0:28.87	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:07.64	ksoftirqd/0
8	root	20	0	0	0	0	I	0,0	0,0	0:10.34	rcu_sched

Como podemos observar, o processo **6359** é um dos processos filhos calculando parte da série de fibonacci. Infelizmente com o comando **top** não foi possível visualizar o processo principal do executável.

Abaixo utilizamos o comando **ps -e** para listar os processos. Aqui vemos alguns dos processos (**6348, 6349, 6359**) correspondente ao nosso executável **1b**.

Usando este comando e pelo número mais “antigo” de processo relacionado ao executável **1b**, estipulamos que o processo principal seja o **6348**.

```
6189 ?      00:00:11 kworker/u2:0
6233 ?      00:00:07 kworker/u2:2
6347 ?      00:00:03 kworker/u2:1
6348 pts/1    00:00:00 1b
6349 pts/1    00:00:00 1b <defunct>
6359 pts/1    00:00:45 1b
6383 pts/2    00:00:00 ps
14989 ?      00:00:02 apache2
14991 ?      00:00:00 apache2
14992 ?      00:00:00 apache2
17942 ?      00:00:00 loop7
17978 tty1    00:00:22 gnome-software
18043 ?      00:03:41 gnome-terminal-
18932 ?      00:00:00 wpa_supplicant
22164 ?      00:00:00 xfsalloc
22167 ?      00:00:00 xfs_mru_cache
22176 ?      00:00:00 jfsIO
22177 ?      00:00:00 jfsCommit
22178 ?      00:00:00 jfsSync
22306 tty1    00:00:01 debconf-communi <defunct>
andreza@andrezavm: /proc$
```

O comando abaixo utilizado é o **top -p número_do_processo**, que no caso é o **6348**. Aqui podemos ter algumas informações mais detalhadas desse processo.

```
andreza@andrezavm: /proc
Arquivo Editar Ver Pesquisar Terminal Ajuda
top - 00:22:00 up 4:28, 1 user, load average: 4,17, 1,83, 0,86
Tarefas: 1 total, 0 em exec., 1 dormindo, 0 parado, 0 zumbi
%CPU(s): 53,6 us, 46,4 sis, 0,0 ni, 0,0 oc, 0,0 ag, 0,0 ih, 0,0 is 0,
KB mem : 1949156 total, 140684 livre, 1103072 usados, 705400 buff/cac
KB swap: 483800 total, 445232 livre, 38568 usados, 657560 mem disp

  PID USUARIO  PR  NI   VIRT   RES   SHR S %CPU %MEM  TEMPO+ COMANDO
  6348 andreza  20   0  14016   2040  1896 D  0,0  0,1   0:00.00 1b
```

Neste outro caso abaixo, utilizamos o comando acima para o processo **6359**, que é um processo filho mostrando para gerar sua sub sequência de Fibonacci.

```
andreza@andrezavm: /proc
Arquivo Editar Ver Pesquisar Terminal Ajuda
top - 00:25:32 up 4:32, 1 user, load average: 4,70, 3,31, 1,66
Tarefas: 1 total, 1 em exec., 0 dormindo, 0 parado, 0 zumbi
%CPU(s): 56,0 us, 44,0 sis, 0,0 ni, 0,0 oc, 0,0 ag, 0,0 ih, 0,0 is 0,0 tr
KB mem : 1949156 total, 133772 livre, 1109396 usados, 705988 buff/cache
KB swap: 483800 total, 445232 livre, 38568 usados, 651300 mem dispon.

  PID USUARIO  PR  NI   VIRT   RES   SHR S %CPU %MEM  TEMPO+ COMANDO
  6359 andreza  20   0  14016   2040  1896 R 29,7  0,1   1:25.48 1b
```

ITEM c)

Ainda utilizando os parâmetros iniciais do item passado e usando o comando **ps aux | grep 1b**, temos:

```
andreza@andrezavm: /proc$ ps aux | grep 1b
andreza  6348  0.0  0.1 14016  2040 pts/1    D+   00:19   0:00 ./1b
andreza  6349  0.0  0.0      0      0 pts/1    Z+   00:19   0:00 [1b] <defunct>
andreza  6359 29.3  0.1 14016  2040 pts/1    R+   00:20   1:45 ./1b
andreza  6407  0.0  0.0 21536  1048 pts/2    S+   00:26   0:00 grep --color=auto 1b
```

Porém, devido a sobrecarga não foi possível ver o outro processo filho pois o meu pc deu tela azul.

Mas, felizmente, no próprio programa é impresso o número do processo filho e o seu pai. Isso pode ser visto claramente no screenshot do **item a)**.

ITEM d)

Aqui utilizamos o comando **top -p 6359** para visualizar mais informações sobre esse processo.

```
top - 00:34:21 up 4:40, 1 user, load average: 4,61, 4,38, 2,95
Tarefas: 1 total, 1 em exec., 0 dormindo, 0 parado, 0 zumbi
%CPU(s): 56,5 us, 43,5 sis, 0,0 ni, 0,0 oc, 0,0 ag, 0,0 ih, 0,0 is 0,0 tr
KB mem : 1949156 total, 132780 livre, 1109316 usados, 707060 buff/cache
KB swap: 483800 total, 445232 livre, 38568 usados, 651404 mem dispon.

  PID USUARIO  PR  NI   VIRT   RES   SHR S  %CPU  %MEM   TEMPO+  COMANDO
  6359 andreza  20   0   14016   2040   1896 R  28,9   0,1   4:02.10 1b
```

Utilizando o comando **ps -eo pid, ppid, cmd, %mem, %cpu --sort=-%cpu | head** listamos os 9 primeiros processos ordenados de forma decrescente em relação ao CPU.

Podemos visualizar aqui que o processo filho, **6359**, o mesmo citado anteriormente, está gastando bastante CPU para calcular a parte de sua subsequência. Os parâmetros de entrada ainda são os mesmos citados no **item b)**.

```
andreza@andrezavm:/proc$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head
PID  PPID CMD                                %MEM %CPU
6359  6348 ./1b                                0.1 29.6
920   757 /usr/bin/gnome-shell                 22.4 11.7
711   709 /usr/lib/xorg/Xorg vt1 -dis        4.6  4.0
6347   2 [kworker/u2:1]                       0.0  2.6
18043 662 /usr/lib/gnome-terminal/gno       2.0  2.2
6189   2 [kworker/u2:0]                       0.0  1.6
6233   2 [kworker/u2:2]                       0.0  0.9
2091   1 /usr/bin/python3 /usr/bin/u        8.1  0.5
1086  920 ibus-daemon --xim --panel d     0.6  0.3
andreza@andrezavm:/proc$
```

Agora, fazendo uso do comando **sudo strace -p numero_processo** podemos visualizar as system calls feitas por tal processo.

Nos dois screenshots abaixo vemos as system calls do processo **1480**, pai do processo **1481**, e o processo **1633**, pai do processo **1647**.


```
es Terminal dom, 10:08
andreza@andrezavm: ~/Área de Trabalho/Operational-systems/Lab1Fork
Arquivo Editar Ver Pesquisar Terminal Ajuda
Eu sou o processo 1481, filho do processo 1480. Fiquei encarregado de imprimir
ssa parte da Sequência de Fibonnaci
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1771
28657 46368 75025 121393 196418
Olá, eu sou o pai 1480, o meu filho é 1481

Eu sou o processo 1625, filho do processo 1478. Fiquei encarregado de imprimir
ssa parte da Sequência de Fibonnaci
317811 514229 832040 1.34627e+06 2.17831e+06 3.52458e+06 5.70289e+06 9.22747e+0
1.49304e+07 2.41578e+07 3.90882e+07 6.3246e+07 1.02334e+08 1.6558e+08 2.67914e
08 4.33494e+08 7.01409e+08 1.1349e+09 1.83631e+09 2.97122e+09 4.80753e+09 7.778
4e+09 1.25863e+10 2.0365e+10 3.29513e+10 5.33163e+10 8.62676e+10
l-systems

andreza@andrezavm: /proc
Arquivo Editar Ver Pesquisar Terminal Ajuda
andreza@andrezavm:/proc$ sudo strace -p 1480
strace: Process 1480 attached
restart_syscall(<... resuming interrupted nanosleep ...>) = 0
write(1, "\n", 1) = 1
write(1, "Ol\303\241, eu sou o pai ", 19) = 19
getpid() = 1480
write(1, "1", 1) = 1
write(1, "480", 3) = 3
write(1, ",", 1) = 1
write(1, " o meu filho \303\251 ", 16) = 16
write(1, "1", 1) = 1
write(1, "481", 3) = 3
write(1, "\n", 1) = 1
exit_group(0) = ?
```

```
andreza@andrezavm: ~/Área de Trabalho/Operational-systems/Lab1Fork
Arquivo Editar Ver Pesquisar Terminal Ajuda
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711
28657 46368 75025 121393 196418
Olá, eu sou o pai 1636, o meu filho é 1637

Eu sou o processo 1647, filho do processo 1633. Fiquei encarregado de imprimir e
ssa parte da Sequência de Fibonnaci
317811 514229 832040 1.34627e+06 2.17831e+06 3.52458e+06 5.70289e+06 9.22747e+06
1.49304e+07 2.41578e+07 3.90882e+07 6.3246e+07 1.02334e+08 1.6558e+08 2.67914e+
08 4.33494e+08 7.01409e+08 1.1349e+09 1.83631e+09 2.97122e+09 4.80753e+09 7.7787
4e+09 1.25863e+10 2.0365e+10 3.29513e+10 5.33163e+10 8.62676e+10
Olá, eu sou o pai 1633, o meu filho é 1647
andreza@andrezavm:~/Área de Trabalho/Operational-systems/Lab1Fork$

andreza@andrezavm: /proc
Arquivo Editar Ver Pesquisar Terminal Ajuda
andreza@andrezavm:/proc$ sudo strace -p 1633
strace: Process 1633 attached
restart_syscall(<... resuming interrupted nanosleep ...>) = 0
write(1, "\n", 1) = 1
write(1, "Ol\303\241, eu sou o pai ", 19) = 19
getpid() = 1633
write(1, "1", 1) = 1
write(1, "633", 3) = 3
write(1, ",", 1) = 1
write(1, " o meu filho \303\251 ", 16) = 16
write(1, "1", 1) = 1
write(1, "647", 3) = 3
write(1, "\n", 1) = 1
exit_group(0) = ?
```