



INSTITUTO FEDERAL DA PARAÍBA - IFPB

CENTRO DE ENGENHARIA DE COMPUTAÇÃO - CEC

DISCIPLINA: SISTEMAS OPERACIONAIS

PROFESSOR: DAVID CANDEIA MEDEIROS MAIA

ESTUDANTE: JOSÉ ROBERTO PEIXOTO FERREIRA FILHO

Atividade Prática

Processos

9. A saída é nada mais que uma string relacionada ao tipo do processo que está em execução no momento, que de forma aleatória varia entre o processo pai e o processo filho.

11. O processo pai é aquele que o PID é igual a PPID do filho

```
~/processdemo$ ps xl
F  UID      PID      PPID  PRI  NI     VSZ   RSS  WCHAN  STAT TTY      TIME COMMAND
4  1000        1          0   20    0 2064460 36896 futex_ Ssl ?      0:56 /nix/store/zmwpz4bxbh25nkmp
4  1000       31          1   20    0 1079240 72536 futex_ SL ?      0:00 /nix/store/l2nfqy8hhq86lg8g
4  1000     11798          1   20    0   8032   4608 do_wai Ss    pts/1    0:00 /nix/store/8kgsjv57icc18qhp
4  1000     11894          1   20    0   3564   1792 -      Rs+    pts/0    0:02 ./main
1  1000     11895     11894   20    0   3564    896 -      R+    pts/0    0:01 ./main
4  1000     11928     11798   20    0   8832   2944 -      R+    pts/1    0:00 ps xl
```

12. O processo filho foi excluído e o pai continuou executando

```
~/processdemo$ ps xl
F  UID      PID      PPID  PRI  NI     VSZ   RSS  WCHAN  STAT TTY      TIME COMMAND
4  1000        1          0   20    0 2064460 36896 futex_ Ssl ?      0:56 /nix/store/zmwpz4bxbh25nkmp
4  1000       31          1   20    0 1079240 72536 futex_ SL ?      0:00 /nix/store/l2nfqy8hhq86lg8g
4  1000     11798          1   20    0   8032   4608 do_wai Ss    pts/1    0:00 /nix/store/8kgsjv57icc18qhp
4  1000     11894          1   20    0   3564   1792 -      Rs+    pts/0    0:02 ./main
1  1000     11895     11894   20    0   3564    896 -      R+    pts/0    0:01 ./main
4  1000     11928     11798   20    0   8832   2944 -      R+    pts/1    0:00 ps xl
~/processdemo$ kill -9 11895
~/processdemo$
```

13. O processo pai foi excluído.

```
~/processdemo$ ps xl
F  UID      PID      PPID  PRI  NI     VSZ   RSS  WCHAN  STAT TTY      TIME COMMAND
4  1000        1          0   20    0 2064460 37084 futex_ Ssl ?      1:26 /nix/store/zmwpz4bxbh25nkmp
4  1000       31          1   20    0 1079240 61784 futex_ SL ?      0:00 /nix/store/l2nfqy8hhq86lg8g
4  1000     11798          1   20    0   8048   3968 do_wai Ss    pts/1    0:00 /nix/store/8kgsjv57icc18qhp
4  1000     11894          1   20    0   3564   1792 -      Rs+    pts/0   11:54 ./main
1  1000     11895     11894   20    0      0      0 -      Z+    pts/0    0:06 [main] <defunct>
4  1000     16538     11798   20    0   8832   3072 -      R+    pts/1    0:00 ps xl
~/processdemo$ kill -9 11894
~/processdemo$
```

```
parent: -630
parent: -631
parent: -632
parent: -633
parent: -634
parent: -635
parent: -636
parent: -637
parent: -638
parent: -639
parent: -640
parent: -641
parent: -642
parent: -643
parent: -644
parent: -645
parent: -646
parent: -647
parent: -648
parent: -649
parent: -650
parent: -651
parent: -652
parent: -653
parent: -654
parent: -655
parent: -656
parent: -657
exit status -1
$
```

14. O processo pai foi morto e junto com ele o filho, já quando o comando de matar o filho foi enviado recebeu-se uma mensagem que não existia mais tal processo (já que ele foi morto junto com o pai na primeira execução do código)

```
~/processdemo$ ps xl
F  UID      PID    PPID  PRI   NI     VSZ   RSS WCHAN  STAT TTY      TIME COMMAND
4  1000        1        0   20    0 2064460 37140 futex_ Ssl ?      1:38 /nix/store/zmwpz4bxbh25nkmp
4  1000       31        1   20    0 1079240 61784 futex_ Sl ?      0:00 /nix/store/l2nfqy8hhq86lg8g
4  1000     11798        1   20    0   8048   3968 do_wai Ss pts/1    0:00 /nix/store/8kgsjv57icc18qhp
4  1000     18445        1   20    0   3564   1792 -      Rs+ pts/0    0:15 ./main
1  1000     18446     18445   20    0   3564    900 -      R+ pts/0    0:13 ./main
4  1000     18692     11798   20    0   8832   3072 -      R+ pts/1    0:00 ps xl
~/processdemo$ kill -9 18445
~/processdemo$ kill -9 18446
bash: kill: (18446) - No such process
```

```
parent: 35
parent: 34
child: 64
parent: 33
child: 65
child: 66
parent: 32
child: 67
parent: 31
parent: 30
parent: 29
child: 68
parent: 28
child: 69
child: 70
parent: 27
child: 71
parent: 26
parent: 25
child: 72
exit status -1
```

15. após a execução do processo pai não havia necessidade de matar o filho, já que matar o processo pai mata o filho também

18. O processo gera duas threads que operam sobre um valor comum. A semelhança entre este código e o código que envolve processos está na execução simultânea e não determinística, tanto para as threads quanto para os processos.

19. Acredito que o processo do arquivo 'threaddemo' seja mais rápido, possivelmente devido ao fato de que as threads consomem menos memória, uma vez que compartilham esse recurso.

20. O processo foi interrompido porque o código compilado não conclui sua execução, ao contrário do que ocorre quando há um loop infinito

21. criei uma variável Z para que cada thread tivesse sua variável
Criei a função adjustZ, na linha da função adjustX e chamei a função criada na thread de acréscimo

```
5 int x = 50; /* a global (shared) variable */
6 int z = 50;
7 const clock_t MAXDELAY = 2000000;
8
9 void delay(clock_t ticks) { /* a "busy" delay */
10     clock_t start = clock();
11     do
12         ; while (clock() < start + ticks);
13 }
14
15 void * adjustX(void *n)
16 { int i = (int)n;
17   while (1) /* loop forever */
18   { printf("adjustment = %2i; x = %i\n", i, x);
19     x += i;
20     delay(rand() % MAXDELAY);
21   }
22   return(n);
23 }
24
25 void * adjustZ(void *n)
26 { int i = (int)n;
27   while (1) /* loop forever */
28   { printf("adjustment = %2i; z = %i\n", i, z);
29     z += i;
30     delay(rand() % MAXDELAY);
31   }
32   return(n);
33 }
34
35 main()
36 { int a;
37   srand(time(NULL));
38   pthread_t up_thread, dn_thread;
39
40   pthread_attr_t *attr; /* thread attribute variable */
41   attr=0;
42
43   printf("creating threads:\n");
44   pthread_create(&up_thread,attr, adjustZ, (void *)1);
45   pthread_create(&dn_thread,attr, adjustX, (void *)-1);
46
47   while (1) /* loop forever */
48   { ;}
49
50 }
```

```
> make -s
> ./main
creating threads:
adjustment = 1; z = 50
adjustment = -1; x = 50
adjustment = 1; z = 51
adjustment = -1; x = 49
adjustment = 1; z = 52
adjustment = 1; z = 53
adjustment = -1; x = 48
adjustment = -1; x = 47
adjustment = -1; x = 46
adjustment = 1; z = 54
adjustment = -1; x = 45
adjustment = 1; z = 55
adjustment = -1; x = 44
adjustment = -1; x = 43
adjustment = 1; z = 56
adjustment = 1; z = 57
adjustment = -1; x = 42
adjustment = -1; x = 41
adjustment = -1; x = 40
adjustment = 1; z = 58
adjustment = -1; x = 39
adjustment = 1; z = 59
adjustment = -1; x = 38
adjustment = 1; z = 60
adjustment = -1; x = 37
```