**Instituto Federal da Paraíba - Campus Campina Grande**

**Aluno**: Wesley Wilson Almeida Araújo

**Professor**: David Candeia

**Turma**: Engenharia de Computação

**Atividade Prática sobre Processos**

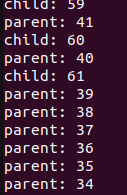
**9.** A saída consiste em dois processos que são criados que realizam uma contagem cada, um processo filho incrementa o valor 1 a cada instante e o outro processo pai decrementa 1 do valor 50 que é uma variável global. No código main ele vai criar uma variável c que vai dar um fork no processo pai e vai chamar a função adjustX(), que vai incrementar a variável chamada na função e vai formar um looping invertendo entre processo pai e processo filho.



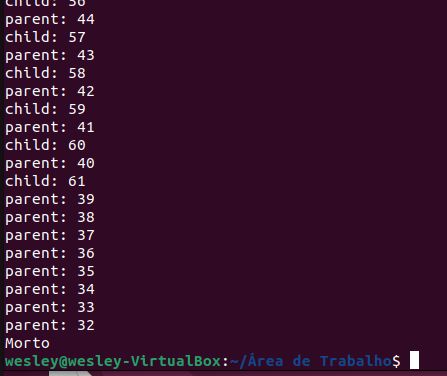
**11.** O processo pai é o 2106, pois o processo 2107 contém o 2106 no PPID indicando que ele é o processo pai, e consecutivamente o 2107 é o processo filho.



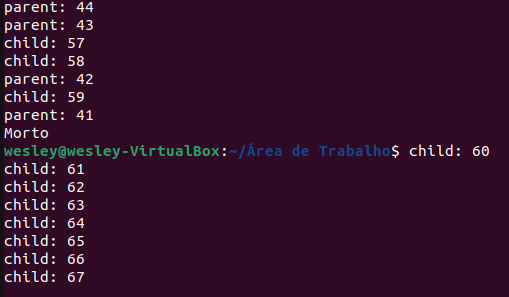
**12.** Ficou somente o processo pai rodando, e a contagem do processo filho parou de acontecer.



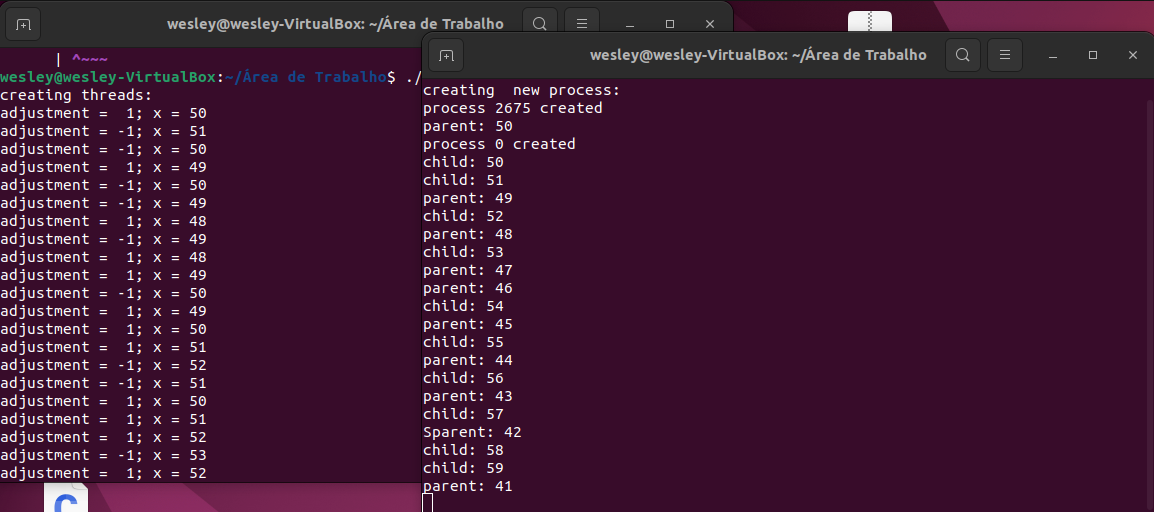
**13.** Os dois processos foram encerrados e a contagem de ambos pararam e apareceu uma mensagem “Morto”.



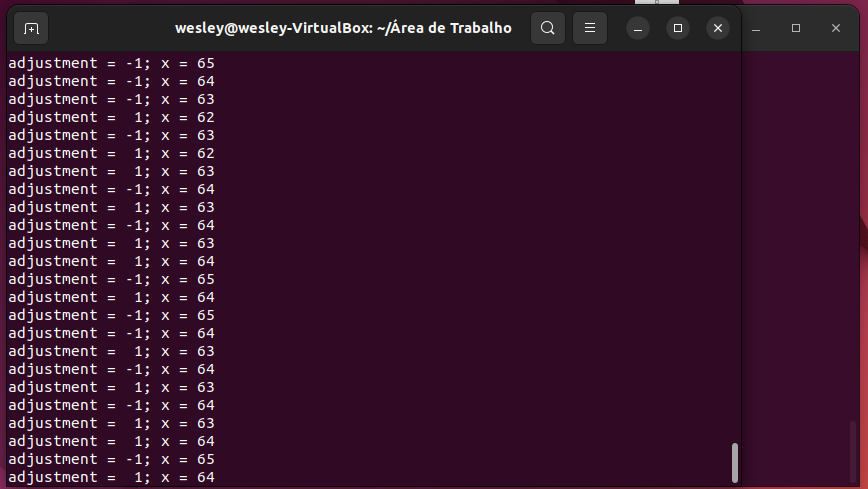
**14.** Apareceu uma mensagem de “Morto” e parou a contagem do processo pai, e em seguida quando eu encerrei o processo filho acabou a contagem de ambos.

****

**15.** Em ambos os casos a contagem foi parada enquanto o outro processo continuava correndo até que fosse encerrado.

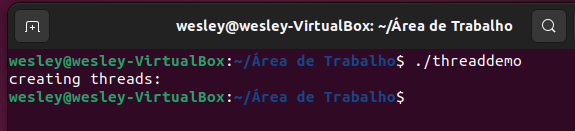
**18.** Na thread ocorre que as operações compartilham a mesma variável global, fazendo com que não siga uma sequência em si, como o programa processdemo.c faz, em que cada um tem a sua variável separada e não interfere na outra contagem.

**19.** O programa threaddemo é mais rápido em comparação com o processdemo em questão de linhas de saída, é possível ver isso pela barra de rolagem do programa threaddemo pois têm mais linhas. A thread é mais rápida pois ela compartilha o mesmo lugar na memória, não precisando gastar tempo tendo que ir lá em 2 processos diferentes como no processdemo.



**21.** O programa apenas cria a thread e não faz mais nada, é necessário que o programa fique em loop pois somente assim ele pode ficar chamando as funções

e realizando a contagem, se rodar apenas 1 vez não tem como alterar o valor na variável global.



**22.** Modificando o threaddemo.c, uma forma de conseguir fazer com que a thread funciona da mesma forma que o processdemo.c é na função adjustX(), em que no processo se tinha uma separação entre um processo e outro e no thread não tinha, então se separar uma variável x e y e usar em uma comparação que nem os processos, vai poder ser possível a manipulação.

