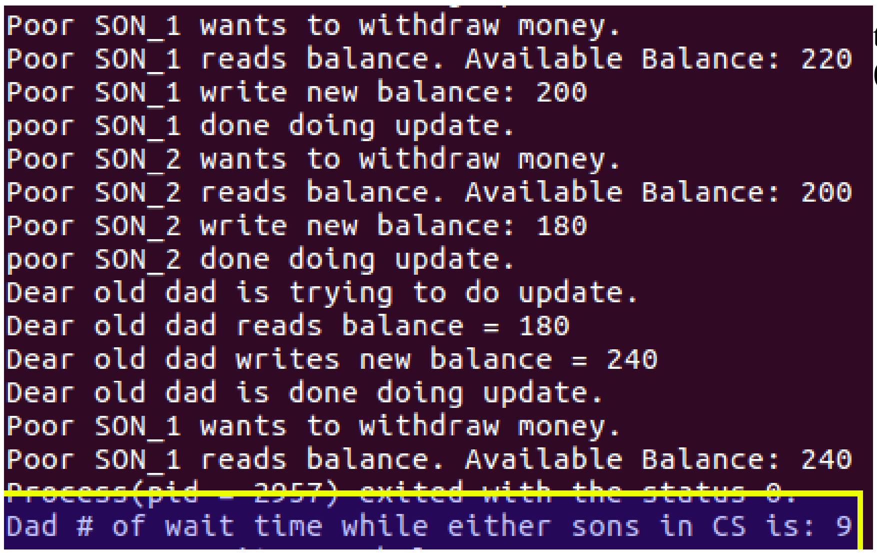
Demetrios Doumas 4/20th/17

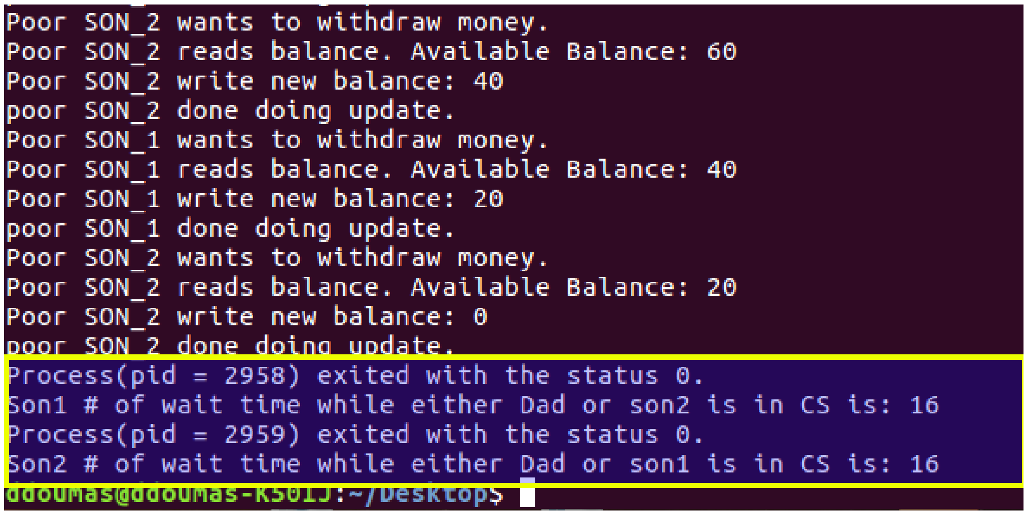
# Task 5 Process Synchronization Dad-Son problem

There is a way to prevent race condition in the Dad-son problem with the use of a single semaphore (sem) and several P(sem) and V(sem) statements. There are three critical sections (CS). The first critical section is the dad depositing money into a balance. The other two critical section are both sons withdrawing money from the same balance that the dad is depositing. Race conditioning was prevented by enclosing the critical sections with the statements P(sem) before CS and V(sem) after CS. This will prevent other processes entering the CS when a process is already inside CS. The figures below shows that each process isn't interrupted while it is executing.

To keep track of how many times each process has waited there needs to be three variables that act like counters. Each counter represents a process. The dad wait counter is waitTimeDad. The sons wait counter is waitTimeSon1 and waitTimeSon2. Once a process enters the cs, it increments the counters of the other two processes. An example would be the Dad is in CS, both sons will be waiting and their counters will be incremented. The data of the each counter is saved in a separate file since fork() will only create copies of the parent variables. A way for the processes to share resources was to save the variable into a text file. This was done to meet the semester deadline. Every time a process enters the CS it will access the counter files of the other two processes and increase their values by 1. The figure below shows how many times the Dad and sons have waited (the blue highlighted rectangles).



```
Poor SON_1 wants to withdraw money.
Poor SON_1 reads balance. Available Balance: 220
Poor SON_1 write new balance: 200
poor SON_1 done doing update.
Poor SON_2 wants to withdraw money.
Poor SON_2 reads balance. Available Balance: 200
Poor SON_2 write new balance: 180
poor SON_2 done doing update.
Dear old dad is trying to do update.
Dear old dad reads balance = 180
Dear old dad writes new balance = 240
Dear old dad is done doing update.
Poor SON_1 wants to withdraw money.
Poor SON_1 reads balance. Available Balance: 240
Process(pid = 2957) exited with the status 0.
Dad # of wait time while either sons in CS is: 9
```



```
Poor SON_2 wants to withdraw money.
Poor SON_2 reads balance. Available Balance: 60
Poor SON_2 write new balance: 40
poor SON_2 done doing update.
Poor SON_1 wants to withdraw money.
Poor SON_1 reads balance. Available Balance: 40
Poor SON_1 write new balance: 20
poor SON_1 done doing update.
Poor SON_2 wants to withdraw money.
Poor SON_2 reads balance. Available Balance: 20
Poor SON_2 write new balance: 0
poor SON_2 done doing update.
Process(pid = 2958) exited with the status 0.
Son1 # of wait time while either Dad or son2 is in CS is: 16
Process(pid = 2959) exited with the status 0.
Son2 # of wait time while either Dad or son1 is in CS is: 16
ddoumas@ddoumas-K501J:~/Desktop$
```