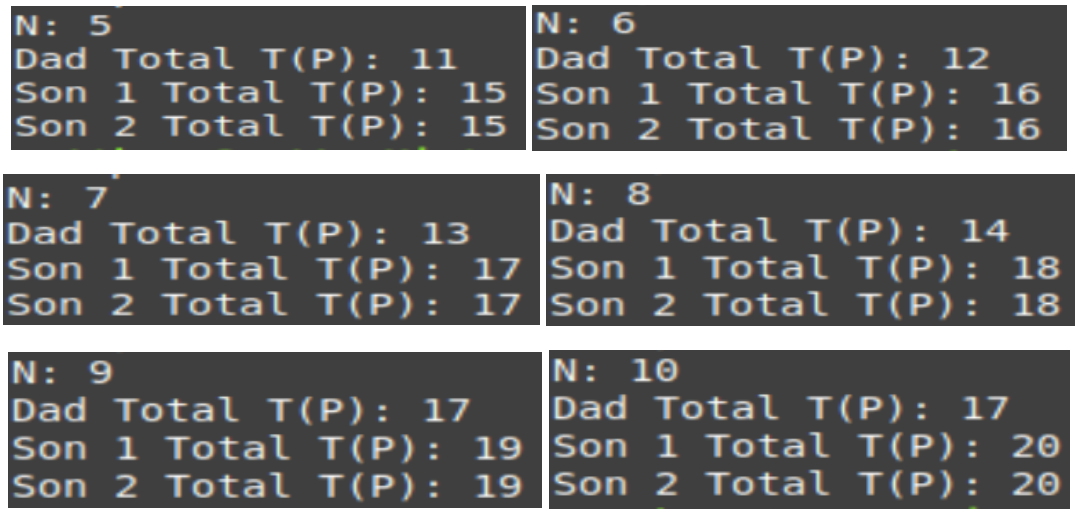
Matthews Cardenas



For this assignment we used semaphores to apply synchronization to the processes. One semaphore called sem was created and initialized with the value of one. The wait operation was used on all processes before they enter their critical section, and the signal operation was used after they are done with their critical section. This prevents race condition to the file balance because only one process at a time is able to open, read, and write into the balance file without the other process either reading or modifying the files content before the initial process had a chance to finish their work. This also holds true for the attempt file as only one of the son's processes can read and modify the file before the other is able to. To measure the wait time 3 new files were created called dad, son_1, and son_2. Each process increments these files whenever they enter the critical section, so when the value of the wait time is printed, the amount of times other processes entered their critical section after the initial process tries to enter it's own, is properly displayed. For the dad process, an extra file was created called dad_ask because the dad process does not ask for the semaphore immediately. This file saves a Boolean value, so the sons only increment the dad file when the dad process asks for the semaphores.