Ch.5 Coding Homework

Tyler Holmquist

gcc (Ubuntu 13.2.0-4ubuntu3) 13.2.0

1. The variable in the child process is not changed, it is the same as was created before forking. When both child and parent change the value of x it has no effect on each other as they are different processes.
   (gcc q1.c -o q1)
2. Yes both the child and parent can access the file pointer. When both the child and parent are writing to the file it will pick up what is coming from the first process and then the second process will write to the file once the first is complete from what I have found in my code.
   (gcc q2.c -o q2)
3. You can do this without calling wait in the parent by forcing the parent to take more time than the child allowing it to finish first. I did this by calling usleep() in the parent to add time for the child to print hello before the parent prints goodbye.
   (gcc q3.c -o q3)
4. NA
5. Wait returns the pid of the process that you are waiting for. If you call wait in the child function it will return -1 or an error because it has no children to wait on.
   (gcc q5_q6.c -o q5_q6)
6. waitpid() would be useful when trying to manage multiple different child processes and you need to specify which child you are wanting to wait for.
   (gcc q5_q6.c -o q5_q6)
7. After closing the standard out in the child anything printed using printf() will not be displayed because the computer doesn't know where to send the output to.
   (gcc q7.c -o q7)