**CIS 452 02 – Lab 3 – Gabrielle Munson and Gabe Baksa**

**Sample Program 1**

1. **What does the program print, and in what order?**

First, the program prints “waiting…”

Then, it prints “received an interrupt”

Which is then followed one second later by “time to exit”

1. **Describe exactly what is happening to produce the answer observed for the above question.**

The signal call waits for there to be an interrupt to the program, hence it printing out “waiting…”. When an interrupt occurs that would terminate the program, the program then jumps to the signal handler code block for a graceful shutdown and then runs that, resulting in “received an interrupt” and “time to exit” to be printed.

**File I/O**

1. **Where does the standard output of the child process go? Explain your answer.**

The standard output of the child process goes to the *temp* file. This is because the location of the standard output is assigned to the parent process before it is forked, making the location of the standard output for the child process the same *temp* file.

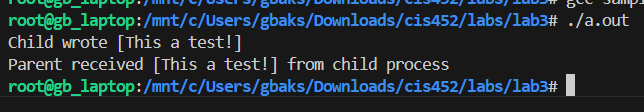
1. **Where does the standard output of the child process go? Explain your answer.**

The standard output of this child goes to the terminal because the fork was applied before the redirection of the standard output to *temp*.

**Sample Program 2**

1. **What exactly does the program do (i.e., describe its high-level functionality)?**

The program first creates a pipe and then it is forked, creating a child process. The child process writes a defined output number to a file and then the parent process reads the number that was written by the child process.



**Programming Assignment**

