Ahmad Malik

11/2/22

ECE-357

Problem Set 4

Question 1)

a) stty(1) is the terminal command that changes that changes the control keys and terminal line settings. You can use “intr” to send an interrupt signal SIGINT using the command “sty intr CHAR” where CHAR is a special character. In this case we want use Control-I for TAB key, so the CHAR will be ^I.

b) In the handler function, the static variable “cnt” is initialized to 0 and is incremented by 1. In the main function, a signal of SIGSEGV is passed when the handler is called.
In the next line, the expression (int*)0 yields a null pointer of type int* and it is being derefrenced to a value of 123. This will result in the signal SIGSEGV being sent. Thus, because of “*(int *)0 = 123” , the main function will keep sending the SIGSEGV signal and keep calling the handler function which will continuously increment “cnt ”.

c) SIGKILL and SIGSTOP can’t be blocked or ignored.

d) A background process that attempts to read or write from the terminal, the shell sends either the SIGTTIN or SIGTTOU signal to tell that process that it's not allowed to read or write, respectively. Upon running the fg command, It would then send SIGCONT to allow the process to run in the foreground if it has already been stopped.

e) Because the kill system call is used to send signal#1 twenty times, the signal won’t be blocked or ignored regardless of masking so it is delivered twenty times to pid#123.

Question 2)

a) It should not be able to print “ABA” because there are two child processes that are writing to the same pipe. Since the writes occur atomically, the first child process writes 4K bytes without obstruction from the other child process, so alternating between child processes is not possible and the sequence would not be possible.

b) Close(pp[1]) closes the pipe so that the pipe cannot be written to and an EOF is received during he read, resulting in 0’s in the pipe. When the line is removed, the parent process will continue to wait for a write since the pipe is still open and the pipe will expects a write to happen and won’t report an EOF to the parent’s read.