ECE 362 – Embedded Operating Systems
Assignment 3: Due date: Feb 28th 2024
Total Points: 100
Process system calls and signals

Turn in: a single Makefile and your source code. The Makefile must include targets "timer" and "alarmSig" that will compile individually the programs and a target "all" that compiles both programs.

This assignment will provide the opportunity to develop programs that use the fork, exec, and alarm system calls. Please use the guidelines described in previous assignments. This is an individual assignment. Please submit one assignment per student.

Be careful! Often students unintentionally create extra processes that don't terminate. Please regularly run the following command, especially before logging off.
ps -ef | grep $USER | less

Reviewing the list of your processes will help you determine if you've created "runaway" processes. If you have, use the command *kill -9 pid*, where *pid* is the id of the process you want to terminate.

1. Executable name: timer
Timer should run a specified program with its arguments and then print the output of the program and the amount of time (in seconds) the program needed to execute (use the time() function). For example, "*timer ls -l /bin*" would run the command *ls* and pass the arguments *-l and /bin* . The output of this program and the time taken would be printed as a result of the execution. You can use the "*sleep*" bash command to test your program.

2. Executable name: alarmSig
Write a program to read a line of text from standard input and after the user

presses the enter key, output the same line to the standard output. In addition, your program must timeout after 10 seconds in case the user is not responding or keeps typing for that long.

You can accomplish this task by taking the following steps:

- Register a signal handler for alarm (use SIGALRM to catch an alarm signal)
- start the alarm using *alarm* system call
- start reading the input from stdin
- Disable the alarm with *alarm* system call
- Write the user string back to the stdout if the read was successful

You can learn about the usage of the alarm system call and SIGALRM from Unix *manpages* on online.

Grading Rubric

20 points each for:

- Right high-level program flow
- Makefile behavior (compile/build/clean)
- Correct execution of timer including corner cases
- Correct execution of piper including corner cases
- Right handling of system call errors