Lab: Signals IC221, Spring AY22

100 points total

**Learning Goals**

1. Using signal handlers, signal() and sigaction()
2. Scheduling alarm signals with alarm()

**Test script**: ./test.sh

**Submit**: shredder.c

**Shredder: Alarm Management**

You will develop a program to time the execution of a child process using alarms and signal handling. Your program will execute another program specified on the command line and schedule SIGALRM to occur ever 1 second. Depending on how many alarms are set off, different information will print to the screen. If too many alarms are set off, the program will terminate.

The source file to complete is shredder.c. Requirements:

* Compiles with no warnings or errors
* Executes child process command with exec()
* Prints correct number of ticks, one per second
* Proper output for (ticks < 3)
* Proper output for (3 <= ticks <=5)
* Kills child process, proper output for (ticks > 5)

If the child ran for < 3 seconds, print:

Blast that grotesque ganglion! You let them get away!

If the child ran for >= 3 seconds, but < 5 seconds, print:

Sayonara you shell-backed simpletons. I'll get you next time!

If the child was terminated because it ran >= 5 seconds, print:

Tonight I dine on turtle soup! Muhaha!

Sample Output:

$ ./shredder ls

Makefile shredder shredder.c

Blast that grotesque ganglion! You let them get away!

$ ./shredder sleep 2

1: tick-tock

2: tick-tock

Blast that grotesque ganglion! You let them get away!

$ ./shredder sleep 4

1: tick-tock

2: tick-tock

3: tick-tock

4: tick-tock

Sayonara you shell-backed simpletons. I'll get you next time!

$ ./shredder sleep 5

1: tick-tock

2: tick-tock

3: tick-tock

4: tick-tock

5: tick-tock

Tonight I dine on turtle soup! Muhaha!

$ ./shredder cat

1: tick-tock

2: tick-tock

3: tick-tock

4: tick-tock

5: tick-tock

Tonight I dine on turtle soup! Muhaha!

$ ./shredder BAD\_FILE

execvp: No such file or directory

Blast that grotesque ganglion! You let them get away!