



Assignment 4

Preface

Operating systems provide various tools for investigating the run time behavior of programs, including programs whose source code is unavailable. In the chapters we have studied so far you learned about multi-threaded programs, dynamic libraries, and signals. This assignment gives you an opportunity to learn about tools related to these topics. Because we have Ubuntu Linux installed on our systems, this assignment is designed around Linux.

Overview

In this assignment, you will use a handful of tools to answer questions about a multithreaded program named `hwk4prog`, and write them in a file that you will then submit using the `submithwk_cs340` command. In this respect it is like Assignment 3. To facilitate the grading of this assignment, I have created a file, in essence a template, that you must copy into your own home directory, and then edit and submit. ***You must follow the instructions below exactly in order to receive full credit. If you do not do exactly what the instructions state, the scripts that process your submission will not find your complete answers and you will lose points.***

The program `hwk4prog` is a multi-threaded program. Whenever you run it, it will create many threads. The exact number of threads will be the same each time you run it. The program calls various functions and responds to signals sent to it in various ways. Your job is to learn the behavior of the program when you run it.

Guidance

There are various commands that you will need to complete this assignment. ***Following is important advice to read before you start this assignment.***

1. You need to make sure that, whenever you finish running the program, **there are no instances or threads running after you terminate it**. Recall that the `ps` command displays information about processes and has many options. You will need to use the `ps` command to do this assignment. Chapter 4 of the lecture notes describes some options for viewing threads and process ids and more. It is strongly recommended that you read the man page for the `ps` command.
2. **You need to open two ssh connections to the same cslab machine to do the first part of this assignment**. In one terminal, you should run the program. In the other terminal, you will need to execute commands to examine or control the program's behavior.
3. Before you run this program you must know how to terminate all threads that it creates in case things go wrong. The command

```
$ killall -9 hwk4prog
```

will send signal SIGKILL (number 9) to all processes and threads whose name is `hwk4prog`. This will terminate the program and all of its threads.

4. Read the man page for the `kill` command:

```
$ man kill
```

This will give you information that you need to answer some questions.



5. Every signal in UNIX systems has both a numeric value and a mnemonic name that starts with the letters “SIG”. For example, SIGINT has number 2. If a question asks for the number of a signal, for SIGINT it is 2. If it asks for its name, it is SIGINT.
6. Because you will need to trace the execution of this program, you will need to use the `ltrace` command. Read the man page for `ltrace` and review Chapter 2’s slides.
7. The `ltrace` command uses breakpoints and various internal signals to trace a program’s execution. The program `hwk4prog` has some code that may sometimes cause your terminal window to become garbled when it is run under `ltrace`. If this happens, you must use the `reset` command to restore your terminal’s settings. To do this, you type

```
reset
```

followed by <ENTER> or <LF>, which you enter by typing Control-J, and your terminal will be fixed. **You will not be able to see what you type.** Do not worry; this is because character output to the terminal was turned off. Your keystrokes are read by the terminal driver and the command will be executed. You may have to type `reset` more than once.

Deadline

You must complete this assignment before its *deadline*, which is **Wednesday, April 13, at 7:00 PM.**

Detailed Instructions

1. Using any *ssh client* on your computing device, remotely login to `eniac.cs.hunter.cuny.edu` using your CSDN username and password.
2. When you login successfully, ssh to **any cs lab host**. For example, to ssh to `cslab12` you would type:

```
$ ssh cslab12
```

3. The remaining instructions assume that you have logged into some `cs lab` host.
4. Copy the file `/data/biocs/b/student.accounts/cs340_sw/hwks/hwk4_questions` to your home directory using the command

```
$ cp /data/biocs/b/student.accounts/cs340_sw/hwks/hwk4_questions ~
```

5. Answer the questions in the file `hwk4_questions` that you just copied into your home directory. You will need to use a text editor to do this. **Do not use a word processor to do this.** If you do, the file will not be readable and you will get a zero on the assignment. Instead, if you edit it on Linux, use an editor such as `vim`, `emacs`, `pico`, or `nano`. If you choose not to do this and instead use *Windows*, use `WordPad` or `NotePad`, but be warned - these programs may create a file with the wrong end-of-line characters and may make your assignment unreadable. You must convert your file to a UNIX format before submitting it. Learn how to do this by searching on the web. On a Mac, use an editor such as `NotePad++`, `atom`, or any commandline editor like the ones in the Linux list above. Make sure that you follow the instructions for how to write the answers in order to receive full credit.
6. Save your file and **make sure that you do not add an extension to its name**. It can be named anything as long as it does not have periods in the name and has no extensions such as “.txt”.
7. Assuming the file is named `myhwk4_questions`, you would submit it by running the command

```
submithwk_cs340 -t 4 myhwk4_questions
```

The program will copy your file into the directory

```
/data/biocs/b/student.accounts/cs340_sw/hwks/hwk4/
```



and if it is successful, it will display the message, “File `hwk4_username` successfully submitted.” where `username` is your username. If you do not receive this message, it was not submitted successfully and you should try again. You will not be able to read this file, nor will anyone else except for me.

8. *You can do step 7 as many times as you want. Newer versions of the file will overwrite older ones.*

Grading Rubric

This assignment is 2.5% of your final grade. The relative value of each question is listed in the template file for the assignment. Partial credit may be given for partially correct answers. The deadline again is **Wednesday, April 13 at 7:00 PM**.