

Lecture 33 — Starling City Vigilantes

Jeff Zarnett

2018-11-13

In Class Exercise:

Background. Your name is Oliver Queen. After five years in hell, you have come home with only one goal: to save your city. Now others have joined your crusade. To them, you're Oliver Queen. To the rest of Starling City, you're someone else. You are something else. You are... the Green Arrow.

The League of Assassins has planted an explosive device in one of the buildings around town. There are too many buildings for you to search by yourself, but you have a team of your fellow vigilantes who will help you. It's your team, so you will need to coordinate the actions of your team members – telling them what areas they should search, and checking in with them as to whether they've found it. You are a hands-on individual, so you will also be searching in the meantime.

Either you or a member of your team will find the device eventually. Once it has been found, you can tell your team members they can stop searching. If you found the device, deal with it immediately. If someone else found it, they will tell you where it is so you can deal with it. Once it's disabled, the city is safe for another night and you can go back to the lair.

Primary Objective. The primary objective of this exercise is to practice using asynchronous I/O in a program.

Secondary Objective(s). This is an additional opportunity to practice with system calls, program design, . You may also improve your ability to work with version control (git) and gitlab.

Starter Code. The starter code can be found at <https://git.uwaterloo.ca/jzarnett/ece252/ece252-e4> – fork this repository to your own space. Set permissions for this repository to be private, but add the group for the course staff with read access so your code can be evaluated.

Submitting Your Code. When you're done, use the command `git commit -a` to add all files to your commit and enter a commit message. Then `git push` to upload your changes to your repository. You can commit and push as many times as you like; we'll look at whatever was last pushed. And check that you gave the course staff permissions!

Grading. Binary grading: 1 if you have made a meaningful attempt at implementing the code; 0 otherwise.

Description of Behaviour. The goal is to implement the program so that the following behaviour occurs:

- Your program should not leak memory; be sure to destroy/deallocate anything initialized/allocated.
- There should not be any race conditions in your program either.

Hints & Debugging Guidance. Some general guidance is below. If you're having trouble, try running through these steps and it may resolve your problem. If you're still stuck you can ask a neighbour or the course staff.

- Check the documentation for how functions work if you are unfamiliar with them (google is your friend!)
- Have you initialized all variables? It is easy to forget; `malloc` does not initialize the value...
- Is there a missing or extra `*` (dereference) on a pointer somewhere?

- Does every memory allocation have a matching deallocation?
- It may be helpful to put `printf()` statements to follow along what the program is doing and it may help you narrow down where the issue is.
- Don't be shy about asking for help; the TAs and instructor are here to help you get it done and will help you as much as is reasonable.

References