FAQ: Labs

Quick Links: [Home (https://q.utoronto.ca/courses/419441/assignments/syllabus)](https://q.utoronto.ca/courses/419441/assignments/syllabus) [Lecture material (https://q.utoronto.ca/courses/419441/pages/lecture-material)](https://q.utoronto.ca/courses/419441/pages/lecture-material) [Lab assignments (https://q.utoronto.ca/courses/419441/pages/lab-assignments)](https://q.utoronto.ca/courses/419441/pages/lab-assignments) [Piazza Discussion (https://piazza.com/utoronto.ca/winter2026/ece353/home)](https://piazza.com/utoronto.ca/winter2026/ece353/home)

See FAQs for: [Lab 1 (https://q.utoronto.ca/courses/419441/pages/faq-lab1)](https://q.utoronto.ca/courses/419441/pages/faq-lab1) Lab 2 Lab 3 Lab 4 Lab 5

Q: How do I setup VNC on Windows?

A: Start by configuring ssh, as described in ssh.html. Then use the putty program to ssh into a UG machine and start the VNC server:

```
vncserver -geometry 1920x1080 -depth 24 :1
```

Then setup putty to open a tunnel to the server. Under

```
Connection > SSH > Tunnels
```

setup source port to any number, e.g., 2000
setup destination to 127.0.0.1:5901

Now use putty to login to the UG machine. This will start the tunnel from port 2000 on your Windows machine to port 5901 on the UG machine.

Then you can tightvnc to access the vncserver. There, the remote host should be specified as

```
localhost:2000
```

Q: Are we allowed to use some open source code for data structure handling? (e.g. if I wanted to implement a hashtable can I use the code that's written by others for inserting/iterating the hashtable?)

A: In general, you are not supposed to use third-party code (i.e. code that does not belong to the code base that we have provided) for this course.

Q: Cscope segmentation fault. Hey, I'm having an error when I run cscope, I get "*Building cross-reference...Segmentation fault*". How do I fix the bug?

A: Try deleting all the previously generated cscope files and then regenerating them.

Q: Is there a way to change the ssh default shell? I use bash on the ug machine. When I login via ssh, the default shell is csh. Is there a way to change the default running shell to bash?

A: use

```
chsh username /bin/bash
```

Also, if you have trouble with backspace, look at:

http://unix.stackexchange.com/questions/43103/backspace-tab-not-working-in-terminal-using-ssh (http://unix.stackexchange.com/questions/43103/backspace-tab-not-working-in-terminal-using-ssh)

Q: How can I suppress make errors due to unused functions?

A: You can add this source code to suppress unused function/parameter warnings:

```
static int unused(int a)
{
    // this suppresses warning for unused parameter a
    (void)a;
}

// this suppresses unused static function warning
(void)unused;
```

Q: What does the following code do:

```
minfo = mallinfo();
assert(minfo.uordblks == 0);
assert(minfo.hblks == 0);
```

A: This code checks for heap memory leaks. uordblks are dynamically allocated bytes that are still in use, and hblks are the number of regions of memory that are currently allocated. See **http://man7.org/linux/man-pages/man3/mallinfo.3.html (http://man7.org/linux/man-pages/man3/mallinfo.3.html)**

Q: valgrind shows that my program does not have any memory leaks. When I run my program without valgrind, I do not get the uordblks/hblks assertion failure, but when I run my program with valgrind, I get the uordblks/hblks assertion failure. Why does that happen?

A: When you run without valgrind, you are not leaking memory. With valgrind, you leak memory, because valgrind is leaking memory (and we do not have a fix for this issue)! However, valgrind knows that your program is not leaking memory.

Q: What does this error mean?

```
Program received signal SIGSEGV, Segmentation fault.
__brk (addr=0x62c000, addr@entry=<error reading variable: Cannot access memory at address 0x2702db8
>)
    at ../sysdeps/unix/sysv/linux/x86_64/brk.c:40
40      ../sysdeps/unix/sysv/linux/x86_64/brk.c: No such file or directory.
```

A: Your code, or a function your code uses, tried to access memory it shouldn't.

You should use the gdb tips (**see gdb page (https://q.utoronto.ca/courses/419441/pages/debugging-with-gdb)**) to try and identify the line it's failing at, the backtrace (bt), and the variables it's trying to access.

It could be you're trying to free an invalid stack pointer or you've corrupted memory somehow. You can search piazza for others who have experienced similar problems and try their solutions.

Q: A process can run concurrently or in parallel. Is there a difference between these two concepts?

A: Concurrency refers to the ability to run multiple executions, e.g., start a second process before a first process is finished. In Labs 2 and 3, the user-level threads run concurrently.

Parallelism refers to the ability to run multiple threads of execution simultaneously (at the same physical time), for which you require multiple CPUs (or cores). Labs 2 and 3 do not have parallelism because the user-level threads that you need to implement are run in the context of a single kernel thread, which runs on one core. So two user-level threads cannot be running at the same time.

In Labs 4 and 5, we will explore parallelism, because the web server threads are kernel threads, and they run on different cores. Unlike concurrency on a single core, parallelism enables speeding up execution by utilizing the multiple cores simultaneously.

Q: when connecting to the ug machine using ssh, the vim editor does not recognize my backspace. Is there a way to fix that?

A: type in shell:

stty -a

```
The erase character should be "^?". If not, try
```

```
stty erase ^?
```

Q: My program runs correctly in gdb, but it crashes without gdb. How do I debug my program.

If you find that your program works in gdb but not without gdb, it is a good idea to turn the disable-randomization option "off" in gdb, and check whether your program still runs in gdb.

```
(gdb) set disable-randomization off
```

Q: Is the grade provided when I run the marker the final grade I will get?

A: The marker is used by the TAs to generate your final mark. There are no additional test cases or changes to any code when marked.

There are some tests where non-determinism that could affect your mark from run to run. For example, if you have synchronization bugs it is possible they are not exposed in each single run.

Cheating/plagiarism/etc (anything malicious that would break academic integrity or defeat the purpose of the lab) will also override the mark generated by the marker.