login in to the server

First log in to bolt.cs.ucr.edu using your CS account:

```
ssh your_student_id@bolt.cs.ucr.edu
```

From here, you can log in to the lab machines in WCH133. They are named wch133-01, wch133-02 ... wch133-47. Reserve your machine in this google sheet. For example, if you reserve wch133-01, log in to it from bolt:

```
ssh wch133-01
```

First, enable the compiler, binaries, library paths, etc. for devtoolset-7. This needs to be done **every time**.

```
scl enable devtoolset-7 bash
```

Start programming and debugging with the provided sample code.

Once you finish the programming part and need to measure the running time, **go back to bolt**, and log in to ti-05

```
ssh ti-05
```

The reason why we do not recommend you to debug on ti-05 is because all students need to share this machine.

Same as the previous instructions, you need to enable the compiler, etc.

```
scl enable devtoolset-7 bash
```

Before running your code, open a new terminal and use **htop** to monitor CPU usage. If someone is using it, you should wait a while to avoid conflict and get the accurate running time. We would make ti-05 reservation-based if its workload is too heavy, so you'd better start early.

To open a new terminal, you're recommended to use tmux to split the panes. Here is a simple introduction page.

Using cilk

Include the header files:

```
#include <cilk/cilk.h>
#include <cilk/cilk_api.h>
```

In C++, replace the for loop to **cilk_for** to parallelize it. For example, output the numbers from 0 to 9 in parallel.

```
cilk_for(int i = 0; i < 10; i++) {
  cout << i << endl;
}
```

Creating and joining a thread by using **cilk_spawn** and **cilk_sync**:

```
L = cilk_spawn reduce(A, n/2);
R = reduce(A+n/2, n-n/2);
cilk_sync;
```

Compile with cilk

To compile with cilk, you need to add two compilation flags: -fcilkplus -DCILK.

A makefile is also provided to you, so you can easily compile the sample code with:

```
make
```

Timer

A timer file is provided to you to measure the running time.

The timer automatically starts when it's constructed:

```
timer t;
```

To start/stop/reset the timer, use:

```
t.start();
t.stop();
t.reset();
```

To get the total running time, use:

```
t.get_total();
```

Install cilk(Optional)

It's not recommended to install cilk on your machines, but if you're using Ubuntu 18.04, you can use the following command to install it.

```
sudo apt install libcilkrts5
```

On other OSes, you will probably need to compile from the source code.