

CS 1550 – thoth.cs.pitt.edu

We're going to use `thoth.cs.pitt.edu` to do our work for this course. This is a quad-core x86 machine with many GBs of RAM and over a terabyte of disk space. In other words, a modern, powerful machine.

To use it, we first need to modify the path to point to some helper programs, and since the man pages also might be useful, it's worth a digression to fix this on `thoth.cs.pitt.edu` permanently.

```
cd ~
vim .bash_profile
```

Type a capital **G** to go to the end of the file. Scroll up a bit until your cursor is under the line:

```
# Define your own private shell functions and other commands here
```

Type the letter **i** to enter insert mode, and add the lines (**spacing around the [] is critical!**):

```
if [ "$HOSTNAME" = "thoth.cs.pitt.edu" ]; then
    source /opt/set_specific_profile.sh;
fi
```

Hit the **ESCAPE** key, then type **:w!** to save, followed by **:q** to quit.

We did it this way because the file was read only, and we can just let `vim` take care of the permissions via the forced write. This won't take effect until the next time you log in, so log out and reconnect to `thoth`.

To check that it worked type at the prompt:

```
man open
```

If you see:

```
No manual entry for open.
```

that you made a mistake in typing the code. Go back and check your spaces and the characters and try again.

Note

We have also set up an alias for the machine as: `cs1550.cs.pitt.edu`

You can connect with either name, but the script above has to be in terms of the canonical hostname, `thoth`.

CS 1550 – Hello, World!

Part I:

- 1) To login to the computers, you will need to use an SSH client. The SSH client that we will be using is **Putty** (at home, download from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)

We will connect to the machine (host) by its name: `thoth.cs.pitt.edu` or by its alias: `cs1550.cs.pitt.edu`

- 1) When you login first, you are placed in your home directory. The command

```
ls
```

will **LiSt** all of the files and directories there. The one we are most concerned with is the *private* directory. It is special in that only you can access files inside this directory. It will keep your work safe from other people.

- 3) Let's move into the *private* directory so we can work there:

```
cd private
```

Changes **D**irectory to the *private* directory

- 4) For this class, we'll keep all of our files organized into a *cs1550* directory. Make it by typing:

```
mkdir cs1550
```

If you want to double check that it worked, type `ls` to list.

- 5) We now want to move into the *cs1550* directory to do our actual work.

```
cd cs1550
```

Part II:

1) While still in the `cs1550` directory type:

```
mkdir lab1
cd lab1
```

to make a file for today's lab. Now type:

```
nano lab1.c
```

`nano` is a very simple text editor, a lot like Notepad on windows. It is one option for creating and editing code under UNIX/Linux.

2) Type the following text in exactly as it is shown:

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

3) Save the file by hitting **Ctrl + O** and then enter. Exit `nano` by typing **Ctrl + X**. At the bottom of the `nano` window, it shows what keys do special things. The **^** means to hold **Ctrl** while pressing the key

4) Back at the prompt type:

```
gcc -o lab1 lab1.c
```

which will make our program. A file named `lab1` will be in the directory if we type `ls`

5) Run it by typing:

```
./lab1
```

Part III:

1) Type:

```
nano lab1.sh
```

2) Type the following text in exactly as it is shown:

```
#!/bin/bash  
echo "Hello, World"
```

3) Save the file by hitting **Ctrl + O** and then enter. Exit `nano` by typing **Ctrl + X**. At the bottom of the `nano` window, it shows what keys do special things. The **^** means to hold **Ctrl** while pressing the key

4) Back at the prompt type:

```
chmod +x lab1.sh
```

which will give execute permission to our shell script.

5) Run it by typing:

```
./lab1.sh
```

Part IV:

Thoth has ~5mb disk quota which means you can store up to 5mb in your private directory. When you exhaust this space, you would not be able to compile your c files. There is a temporary directory ending with your Pitt username under the following path

```
/u/OSLab/YOUR_PITT_ID
```

You can store up to ~500mb in this folder. So, the best way is to start developing your projects under this directory.