

## Griffin Hurt

Undergraduate Teaching Fellow

/ Teaching Assistant

[griffhurt@pitt.edu](mailto:griffhurt@pitt.edu)

<https://griffinhurt.com>

Spring 2024, Term 2244

Friday 2 PM Recitation

Jan 19<sup>th</sup>, 2024

Slides adapted from  
Shinwoo Kim, Martha Dixon, and  
Vinicius Petrucci

Department of Computer Science  
School of Computing & Information  
University of Pittsburgh

## REC-0: Welcome to 449!

---

- Introduction
- Understanding the command-line
- Accessing the Thoth machine
- Your first C program!
- Debugging with GDB

# TA Introduction

---

- **Griffin Hurt**

- Senior in CS
- Research Medical Applications of Mixed Reality
- TA for
  - CS 0449: Introduction to System Software (this course)
  - CMPINF 0010: Big Ideas in Computing and Information
- [griffhurt@pitt.edu](mailto:griffhurt@pitt.edu)
  - preface subject line with '[CS 0449]'
- **Recitation Materials will be posted on my website (once I update it)**



# TA Office Hours

---

Day	Time/Location
Tuesday	11:00 AM - 2:00 PM @ 130 N Bellefield Ave., 5th Floor
Another Time	TBA
By appointment	Message me to schedule a meeting (in-person/virtual)

- **Changes to office hours will be announced on Discord/Canvas**

# Lab 0 – Hello Lab: Setting up to environment

---

## Goals

### 1. Set up Computing Environment

- Accessing Thoth
- Getting familiar with the Shell

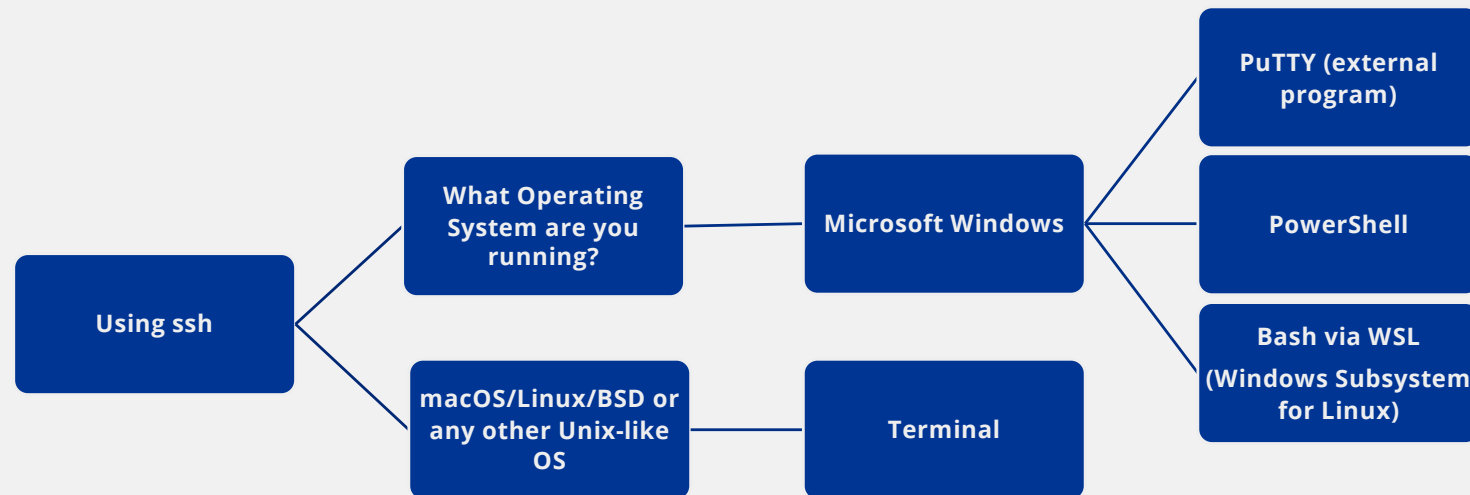
### 2. Learn to compile and run C code on Linux

### 3. Learn to debug C programs using GDB

# Accessing Thoth

---

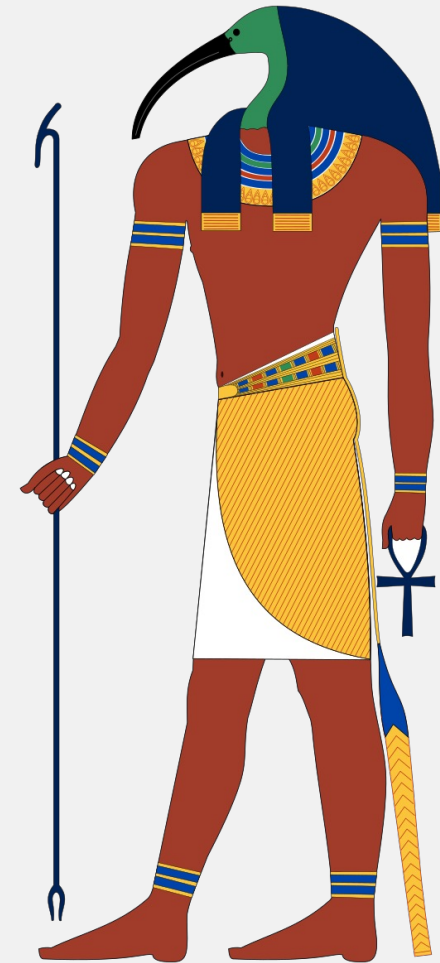
- **You will be using the CS department's Thoth machine for your labs/projects/assignments**
  - This ensures that everyone's code is compiled and run in the same way
    - Your code must work on Thoth to receive full marks!
  - To access the Thoth server, we need to use ssh (**secure shell**) to connect.



# Who was Thoth?

---

- **Thoth was the Egyptian God of writing, wisdom, and magic**
  - Associated with order and justice
  - Advisor and mediator to the Gods
- **In many ways, our Thoth is like this**
  - You'll be writing and gaining wisdom from it
  - C often feels like magic (just wait until 1550)



## Accessing Thoth (cont.)

---

- Open Terminal/Command-line

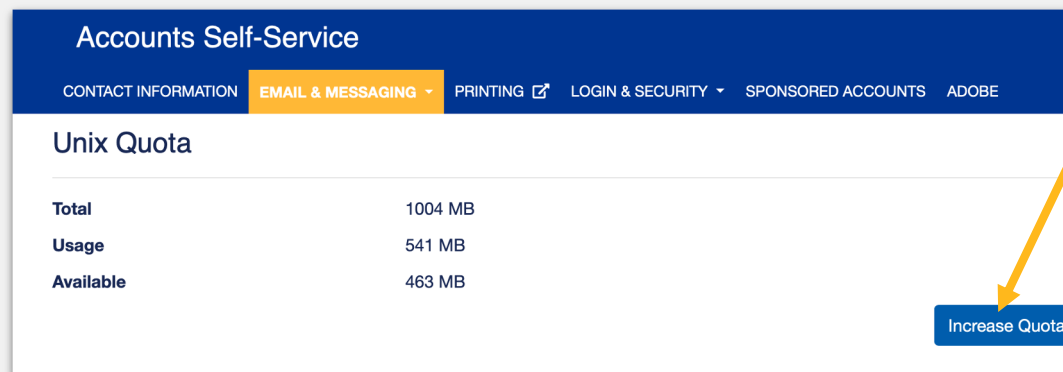
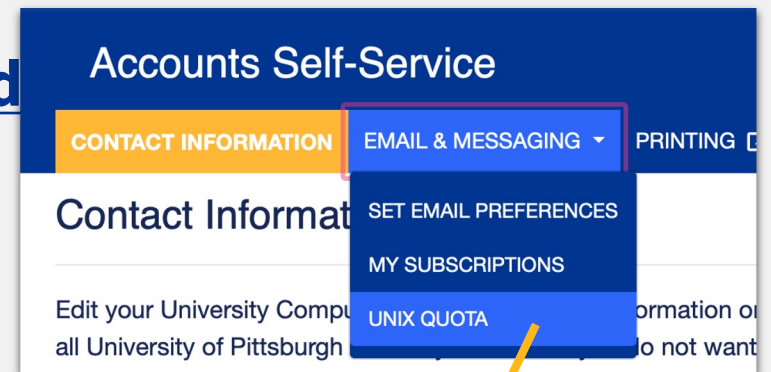
**ssh <PittID>@thoth.cs.pitt.edu**

- Then use your Pitt university computing account password (my.pitt.edu)
- If you are on windows, use PowerShell, or download PuTTY
  - <https://www.chiark.greenend.org.uk/~sgtatham/putty/>
  - Alternatively, you can enable WSL (Windows Subsystem for Linux), then use Bash
    - wsl --install
    - Install Ubuntu from Windows Store

# Increasing AFS Space

Once you can connect, you should increase your AFS space to ensure you have enough storage space to run the assignments

1. Login to <https://accounts.pitt.edu>
2. “Email & Messaging”
3. “Unix Quota”
4. “Increase Quota”





# Linux Shell: Running Commands

---

- **The shell is just a different interface to your computer!**
  - Old, but useful
  - Just like using a keyboard + mouse
  - Useful for automating tasks (especially if we don't need a graphical environment)
  - Useful for interacting with remote computers
    - Like Thoth!

# Basic Shell Commands

---

- **/ (forward slash) is used as path separator**
- **Directory Shortcuts**

/	Root directory
.	Current directory
..	Parent directory (one above)
~	Home directory

- **pwd prints the working directory**
- **man <command> displays the manual for a command**

# Editing Text

---

- **Thoth has several text editors installed**
  - Nano (basic, easiest to use)
  - Vim (popular, steep learning curve)
  - Emacs (popular, steep learning curve, heavily customizable)
- **We can also use GUI text editors such as**
  - Atom, VS Code, Notepad ++, others
- **But we will use the native text editors for today's lab**

# Creating a C Program

---

```
#include <stdio.h>

int main (int argc, char* argv[]) {

    //Declare a variable

    int x;

    //Assign a variable

    x = 2;

    //Print a string and a variable

    printf("Hello world! x is currently %d \n", x);

    return 0;

}
```

# Creating a C Program

```
#include <stdio.h>
int main (int argc, char* argv[]) {
    //Declare a variable
    int x;
    //Assign a variable
    x = 2;
    //Print a string and a variable
    printf("Hello world! x is currently %d \n", x);
    return 0;
}
```

**Preprocessor command:**  
tells compiler to include  
contents of the standard  
input and output file

# Creating a C Program

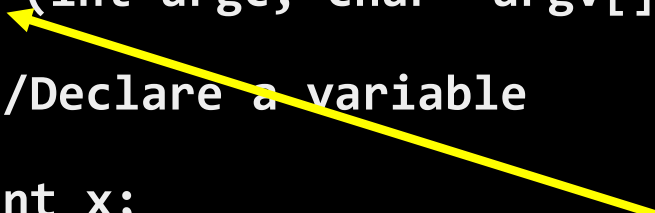
```
#include <stdio.h>
int main (int argc, char* argv[]) {
    //Declare a variable
    int x;
    //Assign a variable
    x = 2;
    //Print a string and a variable
    printf("Hello world! x is currently %d \n", x);
    return 0;
}
```

**Standard input/output file:** Contains functions like scanf() (take input) and printf() (display output)

# Creating a C Program

```
#include <stdio.h>

int main (int argc, char* argv[]) {
    //Declare a variable
    int x;
    //Assign a variable
    x = 2;
    //Print a string and a variable
    printf("Hello world! x is currently %d \n", x);
    return 0;
}
```



Execution of C files starts from main()

# Creating a C Program

```
#include <stdio.h>
```

```
int main (int argc, char* argv[]) {
```

```
    //Declare a
```

```
    int x;
```

```
    //Assign a variable
```

```
    x = 2;
```

```
    //Print a string and a variable
```

```
    printf("Hello world! x is currently %d \n", x);
```

```
    return 0;
```

```
}  
  
printf(): library function to send  
formatted output to screen
```

Using printf()  
without using #include  
<stdio.h> results in  
compilation error

Placeholders:

- %d int
- %u unsigned int
- %f float
- %s "string"
- %x hexadecimal
- %p pointer



# Creating a C Program

```
#include <stdio.h>

int main (int argc, char* argv[]) {

    //Declare a variable

    int x;

    //Assign a variable

    x = 2;

    //Print a string and a variable

    printf("Hello world! x is currently %d \n", x);

    return 0;

}
```

**Exit status:** Returning 0 basically means we exit without error

# Moving files to and from Thoth

---

- **To move files in-and-out of Thoth, we can use Secure Copy (SCP)**
- **To copy local.txt to Thoth (saved as remote.txt in home directory of Thoth)**
  - `scp local.txt user@thoth.cs.pitt.edu:remote.txt`
- **To copy remote.txt from Thoth (saved as local.txt on your device)**
  - `scp user@thoth.cs.pitt.edu:remote.txt local.txt`
- **Check the contents of main.c**
  - `cat main.c`
- **We can make adjustments using nano**
  - `nano main.c`

**-W:** sets warnings  
**-Wall:** enable all warnings

**-g:** turns on debugging symbols

**-o:** used for renaming executable

What you want the executable to be named

Source filename

## • Compile using GCC (Gnu Compiler Collection)

○ `gcc -Wall -g -std=c99 -o hello main.c`

## • At last, we can run our Hello World program

**-std:** sets what version of C is being used

`-std=c99: use C99`

Run the executable

○ `./hello`

Source File

hello  
.c

Compiler

Executable File

a.out

Program Running

`$ gcc hello.c`

To compile

`$ ./a.out`

To run

# Lab 0A Submission

---

- **Rename `main.c` to `username_lab0.c`**
  - `mv main.c username_lab0.c`
    - `mv` command is also used for moving files
- **Copy `username_lab0.c` to Dr. Oliveira's folder**
  - `cp username_lab0.c /afs/pitt.edu/home/l/u/lun8/public/lab0/submissions`
- **To ensure your lab was submitted, run:**
  - `/afs/pitt.edu/home/l/u/lun8/public/lab0/materials /check_submission.sh username`

## Part B – Debugging using GDB

---

- **In later labs, we will use GDB (GNU Project debugger), to debug programs**
- **Today, we will get use to using GDB by running it on a demo program**
  - `wget https://cs0449.gitlab.io/sp2024/labs/00/calculator.c -O calculator.c`
  - `gcc -Wall -g -std=c99 -o calculator calculator.c`
  - `./calculator 4 5 +`
- **Run GDB**
  - `Gdb calculator`
    - Notice we are running GDB with our executable (not the source .c file)
  - `run 4 5 +`

## Part B - GDB Tutorial

---

- **Run through the GDB tutorial (part B)**
  - <https://cs0449.gitlab.io/sp2024/labs/00/>
  - Start from B.5
- **Once you've finished Part B, you are done for today!**