CS 1550

Lab 1 – Linux/Shell/C Pointers

Basic commands Introduction

Teaching Assistant

Xiaoyu (Veronica) Liang

Recitation TA – Office Hours

- Office Hours (tentative)
  - Tuesday:
    - 5:00pm – 6:00pm
  - Thursday:
    - 3:00pm – 4:00pm
    - 5:00pm – 6:00pm
- Office
  - SENSQ 6410
- Email
  - xil160@pitt.edu
- Slides Website
  - http://people.cs.pitt.edu/~xil160/CS1550_Fall2019/

CS 1550 – Introduction to Operating Systems

- Common operating systems abstractions and mechanisms

iOS

ANDROID

Windows 10

ubuntu

CS 1550 – Introduction to Operating Systems

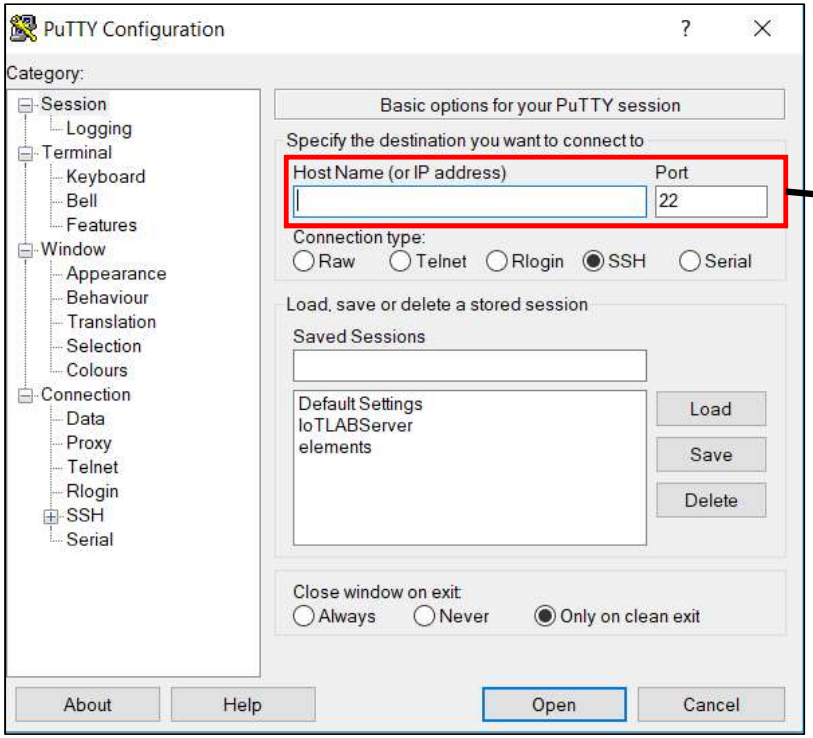
- Common operating systems abstractions and mechanisms
- Will provide basic knowledge common to many modern Operating Systems

SSH Clients

- Windows
  - **Putty**
- MacOS/Ubuntu
  - **Terminal**

Windows - Putty

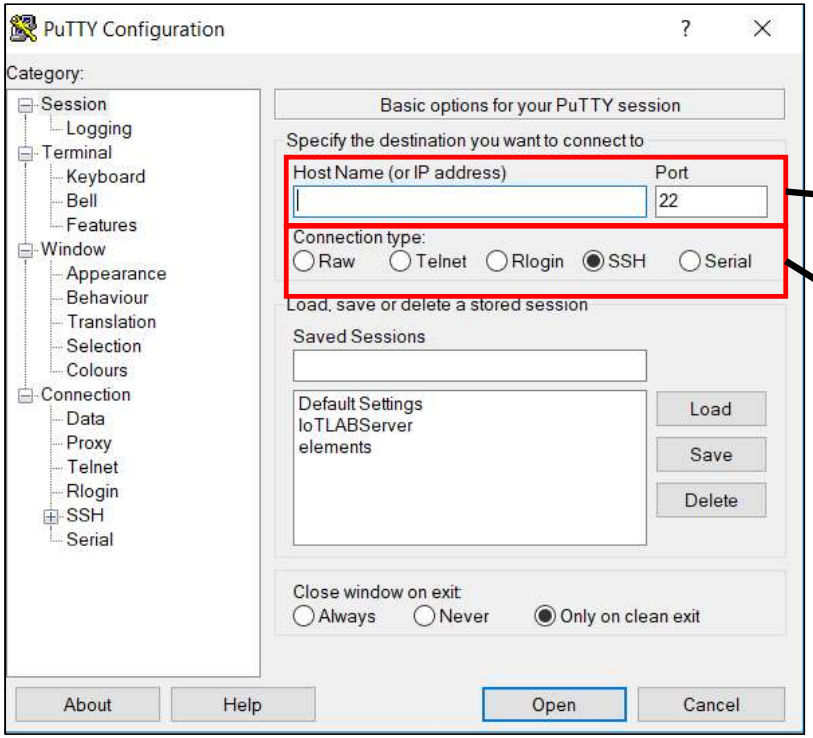
- Download from **www.putty.org**



server address:
thoth.cs.pitt.edu at port 22

Windows - Putty

- Download from **www.putty.org**

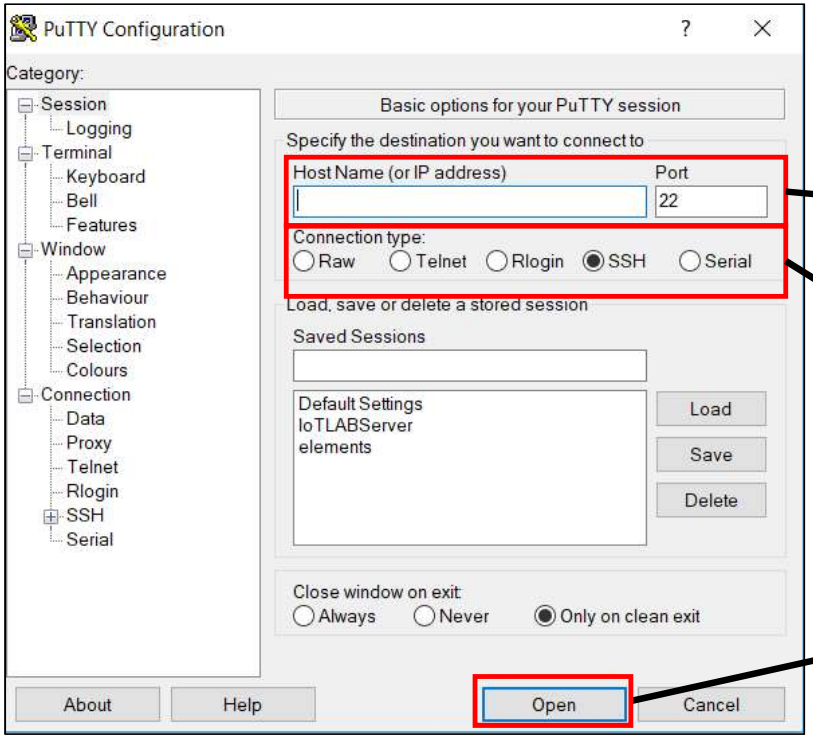


server address:
thoth.cs.pitt.edu at port 22

Use SSH

Windows - Putty

- Download from **www.putty.org**



server address:
thoth.cs.pitt.edu at port 22

Use SSH

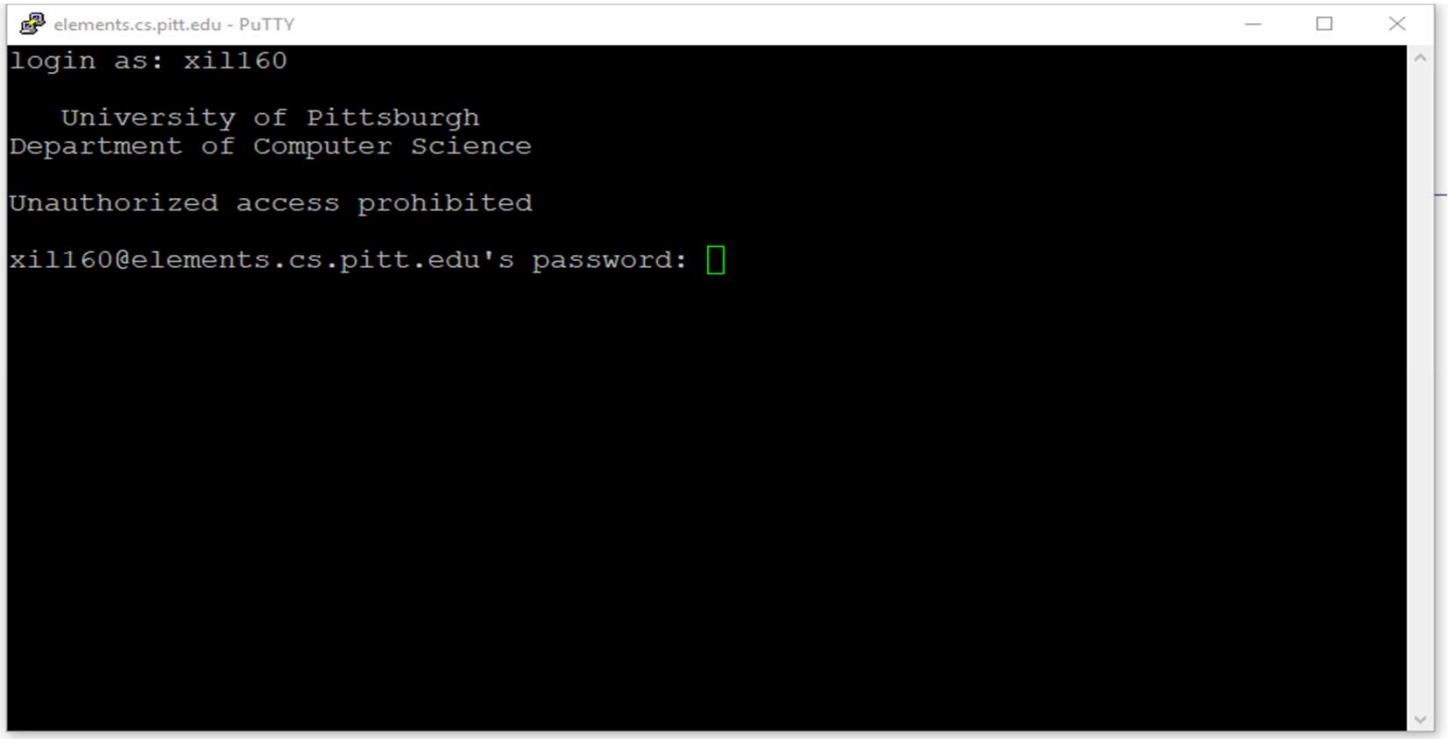
Click Open

Windows - Putty

- Download from **www.putty.org**

Windows - Putty

- Download from **www.putty.org**



MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

```
xiaoyu in ~home-laptop->$
```

MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

```
xiaoyu in ~home-laptop->$ ssh user@IPaddress:port
```

MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

```
xiaoyu in ~home-laptop->$ ssh user@IPaddress:port
```

call ssh command on this IP:Port

MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

```
xiaoyu in ~home-laptop->$ ssh xil160@thoth.cs.pitt.edu:22
```

call ssh command

on this IP:Port

MacOS/Ubuntu - Terminal

- Go to **Applications > Utilities**, and open **Terminal**.

```
xiaoyu in ~home-laptop->$ ssh xil160@thoth.cs.pitt.edu:22

   University of Pittsburgh
Department of Computer Science

Unauthorized access prohibited

xil160@elements.cs.pitt.edu's password:
```

Just type and **press enter,** no cursor will show

Basic Linux Shell commands

- Once into a elements machine
  - Read, create directories and files
  - Compile C/C++ code
  - Whatever program/service you install or the OS already offers

Basic Linux Shell commands

- Check Current Directory – **pwd**
- List directories - **ls**
- Create/Remove directory – **mkdir/rmdir**
- Remove files – **rm**
- Copy files from anywhere to anywhere – **cp**
  - cp **<current path> <new path>**
  - cp **some_text.txt Desktop/**
- Move files from anywhere to anywhere – **mv**
  - mv **<current path> <new path>**
  - mv **some_text.txt Desktop/**

Environment variables

- Environment variables can hold textual information stored within the system that can be used by OS programs
  - ***env*** – Lists all of the environment variables in the shell
  - ***printenv*** – Prints all (if no environment variable is specified) of environment variables and definitions of the current environment
  - ***export*** – Assigns or defines an environment variable
    - export PATH=$PATH:/usr/local/bin
  - ***unset*** – Deletes the environment variable

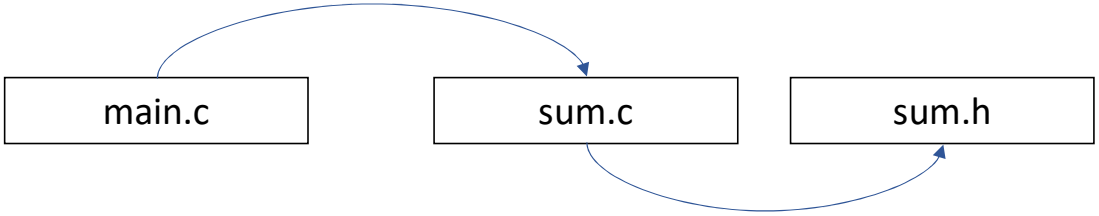
The Makefile

- Small programs (easy to compile)
  - single file

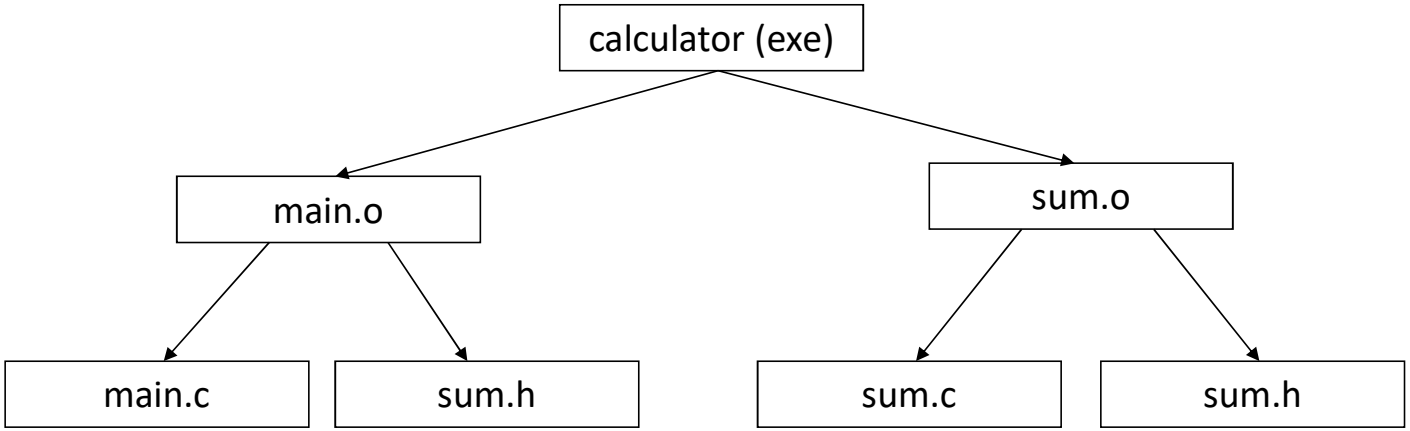
```
~home-laptop->$ gcc main.c -o calculator
```

The Makefile

- Small programs (easy to compile)
  - single file
- Bigger programs
  - multiple files



The Makefile



The Makefile

```
target: dependencies
 action
```

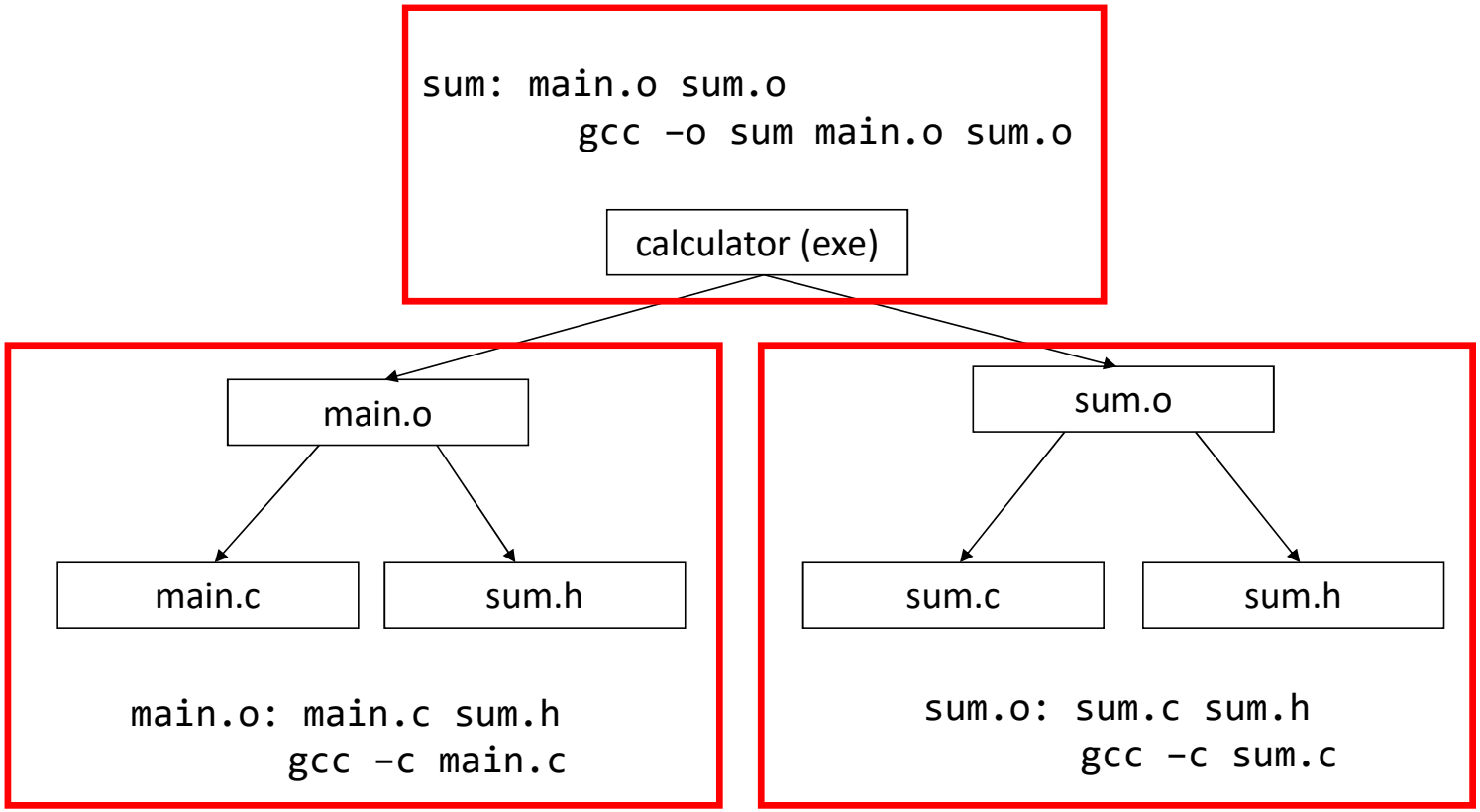
The Makefile

```
calculator: main.o sum.o
 gcc -o calculator main.o sum.o

main.o: main.c sum.h
 gcc -c main.c

sum.o: sum.c sum.h
 gcc -c sum.c
```

The Makefile



The Makefile

- Running “make”

```
~home-laptop->$ make calculator
```

The Makefile

```
calculator: main.o sum.o
 gcc –o calculator main.o sum.o
main.o: main.c sum.h
 gcc –c main.c
sum.o: sum.c sum.h
 gcc –c sum.c
clean:
    -rm -f *.o
```

The Makefile

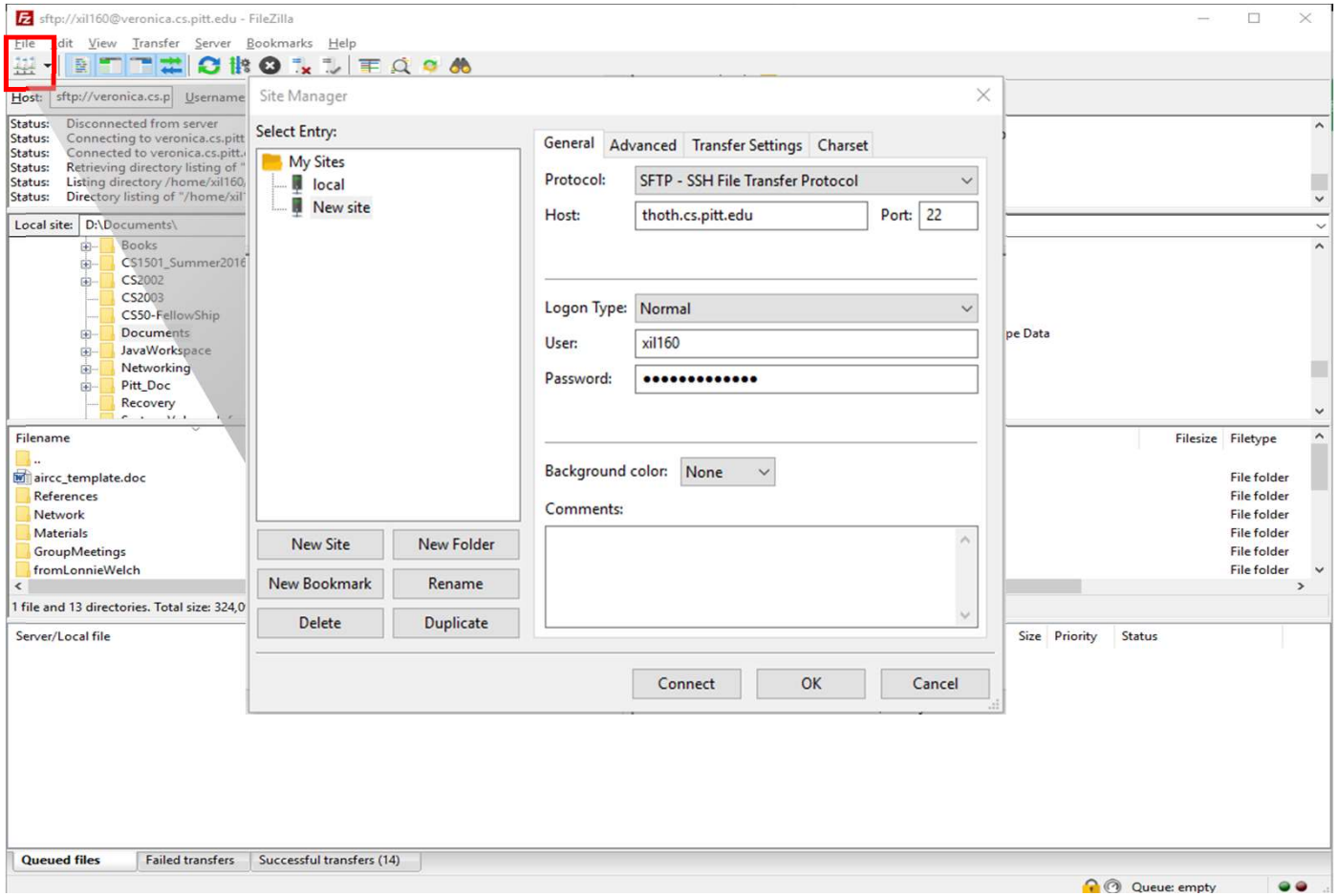
- Running “make”

```
~home-laptop->$ make clean
```

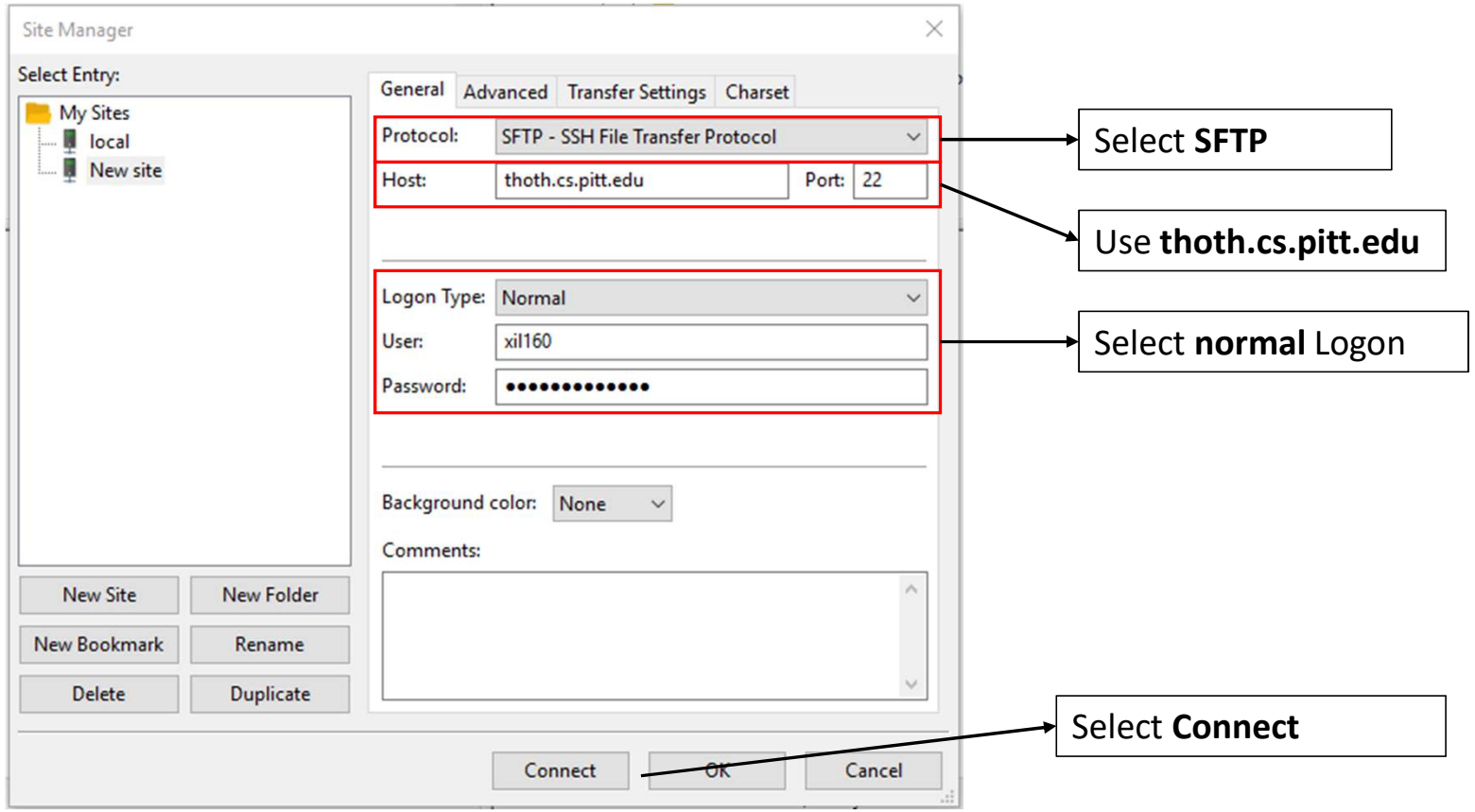
GUI based FTP Clients

- FileZilla – Windows/MacOS
  - Copy files with drag and drop
  - Create directories
  - Delete files

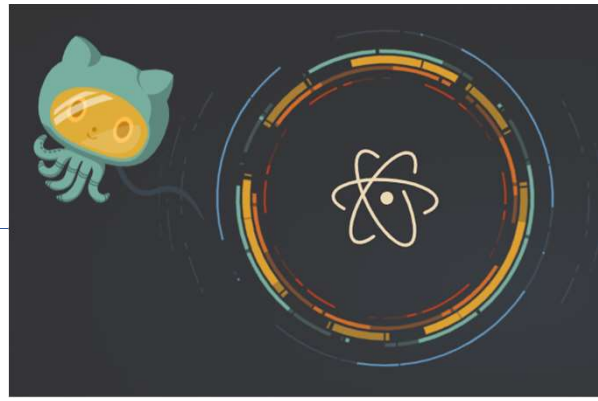
GUI based FTP Clients



GUI based FTP Clients



Using Atom



- Good coding tool
  - Easily search for variables names in any file inside a folder
  - Lots of plugins
- Synchronize Files
  - install ***remote-sync*** (recommended)
  - Right click main project folder
  - Navigate to Remote Sync > Configure
  - Fill in the details / select options
  - Hit save
  - https://atom.io/packages/remote-sync

C Pointers

- Pointers are variables that contain *memory addresses* as their values.
- A variable name *directly* reference a value.
- A pointer *indirectly* reference a value.
- A pointer variable must be declared before it can be used.

C Pointers

- Examples of pointer declarations:
  - int *a;
  - float *b;

  ** tells the compiler that the variable is to be a pointer, and the type of data that the pointer points to*
- & and *:
  - & -- “address operator” gives or produces the memory address of a data variable.
  - * -- “dereferencing operator” provides the contents in the memory location specified by a pointer.

C Pointers

- A simple Example

```
int number;
int *ptr1;
int *ptr2;

ptr1 = &number;
number = 2;
ptr2 = &number;
printf("\n*ptr1 = %d\t*ptr2 = %d\n", *ptr1, *ptr2);
```

C Pointers

- Arrays

```
int demoArray[5] = {8, 19, 34, 0, 3};
printf("Element \t Address \t Value \n");

for (int i = 0; i < 5; i++) {
    printf("demoArray[%d]\t%p\t%d\n", i, &demoArray[i], demoArray[i]);
}

//array names are just pointers to the first Element
printf("\ndemoArray\t\t%p\n", demoArray);

//dereference
printf("\n*demoArray\t\t%d\n", *demoArray);
printf("\n*(demoArray+2)\t\t%d\n", *(demoArray+2));
```

CS 1550

Lab 1 – Linux/Shell/C Pointers

Basic commands Introduction

Teaching Assistant

Xiaoyu(Veronica) Liang