SY110
OS - Remote Access and Unix Shell

Major Brian Hawkins, USMC

U.S. Naval Academy

Fall AY 2018

Outline

1. Review

2. Remote Computing

3. Some Unix & Windows Differences
   - Unix File System

Review & Way Ahead

Last time we talked about the Windows, the OS most of you are familiar with – today we'll be talking about Unix, which is a very different Operating System

Local vs. Remote Computing

Up to this point, everything we've done has been on our own machine. But many times, we need to use computers we don't have *physical* access to.

Remote Computing

rona

The SY110 course server is a Unix machine called `rona`, which we all have accounts on. We can use `rona` by running a special program called `ssh` (secure shell), which allows us to connect to a remote machine, and get a shell over a network.

Connecting to `rona`

To connect to `rona`, type:

```
ssh m21xxxx@rona.academy.usna.edu
```

- What is the command name? What arguments are present? Any options?
- `ssh` into `rona` yourself!

Remote Computing

When connected to `rona` by `ssh`, the commands we enter are executed on `rona`, not our laptop

It isn't just the machine that we execute commands on, but also the user that executes them – remember permissions, usernames, and ownership of files/processes?

Some Unix and Windows Differences

As we've discussed, Windows and Unix-based OSes are different. For example, Unix-based OSes use forward slashes between directory names, *e.g.*

/home/rye/

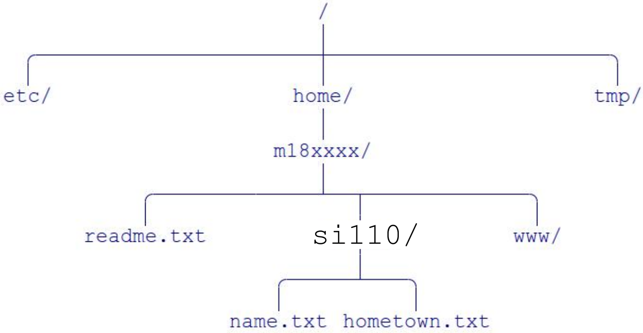
instead of backslashes, like in Windows, *e.g.*

C:\Users\rye

No drive letters in Unix

Whereas Windows may have many file hierarchies (`C:`, `D:`, `E:`), Unix-based OSes have only one. It starts with "/" (one root to rule them all)

The Unix File System



More Unix and Windows Differences

Some commands are the same

- `cd arg1` – still changes directories to arg1
- `mkdir arg1` – still makes directory arg1
- `rmdir arg1` – still removes directory arg1

But some are different

Windows command → Unix command

- `cd`(with no arguments) → `pwd`
- `dir` → `ls`
- `del arg1` → `rm arg1`
- `copy arg1 arg2` → `cp arg1 arg2`
- `move arg1 arg2` → `mv arg1 arg2`

Last words on Unix

The same basic permissions and ownership concepts that applied in Windows apply to Unix. Each process, file, and directory is owned by a user, and the OS enforces what these processes are allowed to do.

Administrator/root

The one exception is the Administrator account on Windows, or the root account on Unix. Also called *super-users*, processes run by these users can access any file/directory.

Questions

Questions?