

Getting started: Basics

Outline:

- I. Connecting to cluster: ssh
- II. Connecting outside UCF firewall: VPN client
- III. UCF resources
- IV. Introduction to Linux
- V. Introduction to vi text editor-- other options

II. Connecting outside UCF firewall: VPN client

1. First go to the NOC page

<http://www.noc.ucf.edu/VPN/default.htm>

2. Then log on to the NOC intranet with your PID and myucf password
3. Click on VPN link to download and install VPN client

Then....

1. First open VPN client, and then log on to VPN server
2. Finally you can ssh to cluster as in preceding slide

III. UCF resources

<http://www.computerlabs.ucf.edu/>

BA 148 8:00am-10:00pm

CC II-104 (West Computer Lab) 8:00am-12:00am

CCII-113 (East Computer Lab) 8:00am-8:00pm

CL-101 (Classroom Computer Lab) 8:00am-7:00pm

Putty, SSH secure shell, ssh secure ftp, microsoft
Office 2007

IV. Introduction to Linux

1. Linux is an operating system similar to UNIX
2. Graphical interfaces exist, but we will just use text-based
3. After ssh to cluster...

4. The `ls` command shows the contents of the directory

```
[student@odin student]$ ls  
prog1.f  
[student@odin student]$
```

Only one file there, a starting code for the first project

Creating a new directory...

1. Let's create a new directory for project #1
2. From your bash shell

```
[student@odin student]$mkdir proj1  
[student@odin proj1]$ ls  
proj1 prog1.f  
[student@odin ]$ cd proj1  
[student@odin proj1]$ ls  
[student@odin proj1]$
```

New directory is empty! We will place source code for project #1 here.

Moving files around... copying

1. Rather than clicking and dragging, we need the `cp` command to create duplicate files
2. In the student top directory, you will need `prog1.f` for the first project
3. Copy this file to `student/group1/proj1` in the following way:

```
[student@odin proj1]$ cd ../
[student@odin student]$ ls
prog1.f
[student@odin student]$ cp prog1.f proj1
[student@odin proj1]$ cd proj1
[student@odin proj1]$ ls
prog1.f
[student@odin proj1]$
```

The program `prog1.f` is where you need it for your first project!

Moving files around... moving

1. Like *cp*, the *mv* command is for manipulating files
2. The *mv* command eliminates the original file however

Uses...moving to a new directory

```
[student@odin proj1]$ mv prog1.f ../  
[patrick@odin proj1]$ ls  
[patrick@odin proj1]$ cd ../  
[patrick@odin student]$ ls  
proj1 prog1.f  
[patrick@odin student]$ mv prog1.f proj1  
[patrick@odin student]$ cd proj1  
[patrick@odin student]$ ls  
prog1.f
```

Moving files around... moving

Uses...renaming a file

```
[student@odin proj1]$ mv prog1.f prog.f
[student@odin proj1]$ ls
prog.f
[student@odin proj1]$ mv prog.f prog1.f
[student@odin proj1]$ ls
prog1.f
```

In contrast, if we had used cp

```
[student@odin proj1]$ cp prog1.f prog.f
[student@odin proj1]$ ls
prog1.f prog.f
[student@odin proj1]$
```

Removing files... and being careful!

```
[student@odin proj1]$ rm prog.f
rm: remove `prog.f'?y
[student@odin proj1]$ ls
prog1.f
[student@odin proj1]$
```

If we didn't want to be prompted...

```
[student@odin proj1]$ rm -f prog.f
[student@odin proj1]$ ls
prog1.f
[student@odin proj1]$
```

For removing directories, they must first be empty of any other files, then use the rmdir command

IV. Introduction to vi text editor-- other options

1. The simple vi is easy to use and surprisingly powerful
2. Some people hate it so we have other options

Let's start our new computer project with rs.f as the starting point...

```
[student@odin proj1]$ ls  
prog1.f
```

```
[student@odin proj1]$ vi prog1.f
```

program proj1

C code to model radioactive decay

Etc.....

Command mode and insert mode

1. The vi editor starts in the command mode
2. Arrow keys move cursor around the file
3. Insert mode accessed by typing “i”
4. In the insert mode, text can be inserted into file
5. Return to command mode with Esc

Let's insert a couple of lines into prog.f...

1. Go to top of file prog.f, type “i” for insert
2. Use “return” to get a blank line at top of file
3. Enter program name and end statement:

```
PROGRAM prog.f
```

```
END PROGRAM prog.f
```

4. Return to command mode with Esc

Saving the file you edit, prog1.f...

1. From the command mode, we need to write the changes to file...

`:w`

2. Then to quit and go back to the bash shell

`:q`

3. We could've written and quit in one command

`:wq`

4. To quit without saving changes

`:q!`

Compiling an F90 source code...

1. Assuming we had a code to compile... which we don't yet, we can compile and create an executable
2. The F90 compiling is invoked using...

```
[student@odin proj1]$ ifort -o prog1 prog1.f  
[student@odin proj1]$ ls  
prog1 prog1.f
```

3. If we didn't name the executable with the -o option,
[student@odin proj1]\$ ifort prog1.f
[student@odin proj1]\$ ls
a.out prog1.f

To run the executable.... and redirecting output

1. Running the executable prog1 is done with

```
[student@odin proj1]$ ./prog1
```

2. If code has output to the screen, you will see the output as it comes
3. Often it is good to redirect output to a file that can be viewed or analyzed later on. To do this,

```
[student@odin proj1]$ ./proj1 > output
```

4. To view the output, we can type

```
[student@odin proj1]$ vi output
```