Introduction

Rob Lane

HPC Support
Research Computing Services
CUIT
Introduction

Linux Basics for Beginners
Introduction

Linux Basics for Beginners

(No HPC Yet)
Introduction

1st Hour: Slides + Hands-on

2nd Hour: Hands-on
Introduction

First Workshop
History

This will be quick
History

Linux is old

1991
History

For our purposes

Linux = Unix
History

Unix is very old

1969
History

Unix created by engineers for engineers
Unix was designed to be simple and powerful
Access

Did everyone get a user ID?
Access

Windows Instructions
1. Search for putty on Columbia home page
2. Select first result
3. Follow link to Putty download page
4. Download putty.exe
5. Run putty.exe
Access

Mac Instructions

1. Run terminal
Access

Mac (Terminal)
$ ssh userNN@didius.cc.columbia.edu

Windows (Putty)
Host Name: didius.cc.columbia.edu
Access

Aside

System: cunix.columbia.edu
User: Your UNI
[user1@didius ~]$ 

- User name
- System name
- Name of current directory
  ~ is special
pwd

$ pwd

“Print working directory”
Directory Path

$ pwd
/workshop/home/user1
$ ls

“List directory”

Not very interesting.
cd

$ cd /

“Change directory”

$ pwd
`ls`

```
$ ls

$ ls -l

Long listing.
```
cd

$ cd

$ pwd

cd with no arguments takes you back home
$ pwd
$ cd ..
$ pwd

“..” means “the directory above this one”
$ pwd
$ cd .
$ pwd

“.” means “this directory”
ls -a

$ cd

$ ls -a

Can combine options

$ ls -la
$ pwd
$ cd ~
$ pwd

“~” means “home directory”
Paths

$ cd tmp
$ cd /tmp

Absolute: starts with “/”

Relative: doesn’t
cp

$ cd
$ cp /tmp/keets .
$ ls
```bash
$ cp keets junk
$ ls
$ rm junk
$ ls
```
cat

$ cat keets
mv

$ mv keets keats

$ ls
mkdir

$ mkdir tmp
$ mv keats tmp
$ cd tmp
$ ls
rmkdir

$ pwd
$ mv keats ..
$ cd ..
$ rmdir tmp
$ ls
who am i

$ whoami
$ who am i
$ id
$ groups
id, groups

$ id rl2226
$ groups rl2226
who

$ who
W
bash

- bash is a “shell”
- It prints the prompt and interprets what you enter
- It has many keyboard shortcuts that can really speed up your work
bash

$ ls jeats
ls: jeats: No such file or directory

- Up arrow to retrieve the command
- Left and right arrows to navigate on line
- Change the “j” to a “k” and rerun
"^" means "hold down control"

^a : go to beginning of line
^e : go to end of line
^k: delete to end of line

Many more useful bash commands
bash

$ ls k[tab]
$ ls keats

• Tab completion
• Works for commands as well
$ man ls

Display manual for “ls” command
ls -l

$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

- File type and permissions
- Link count
- User
- Group
- Date last changed
- File name
File Type

$ ls -l

total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

- Normal File
  d Directory
  l Link

Others Various Special Files
Permissions

$ ls -l

```bash
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats
```

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>read</td>
</tr>
<tr>
<td>w</td>
<td>write</td>
</tr>
<tr>
<td>x</td>
<td>execute</td>
</tr>
<tr>
<td>others</td>
<td>various special settings</td>
</tr>
</tbody>
</table>
$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

We’ll ignore links for now.
User

$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

The user that owns this file.
Group

$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

The group that owns this file.
Size

$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

The size of this file.

Here listed in bytes.
$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

The last time the file was changed.
$ ls -l
total 4
-rw------- 1 user1 workshop 573 Sep 29 22:00 keats

The file name.
Permissions

$ ls -l /bin/bash
-rwrxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash
Permissions

$ ls -l /bin/bash
-rwxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

r    read
w    write
x    execute
others various special settings
Permissions

$ ls -l /bin/bash
-rwrxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

Nine permission settings
Permissions

$ ls -l /bin/bash
-rwxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

Three groups of three
Permissions

$ ls -l /bin/bash
-rwxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

First group: owner

read    yes
write   yes
execute yes
Permissions

$ ls -l /bin/bash
-rwxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

Second group: group

read yes
write no
execute yes
Permissions

$ ls -l /bin/bash
-rwxr-xr-x 1 root wheel 768952 Sep 25 15:31 /bin/bash

Third group: everyone else

<table>
<thead>
<tr>
<th>Access</th>
<th>Third group: everyone else</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>yes</td>
</tr>
<tr>
<td>write</td>
<td>no</td>
</tr>
<tr>
<td>execute</td>
<td>yes</td>
</tr>
</tbody>
</table>
$ date
Tue Sep 30 13:27:29 EDT 2014
Output Redirection

$ cd
$ date > thedate
$ cat thedate
Output Redirection

$ cd
$ ls -l > myhome
$ cat myhome
sort

$ sort keats
Input Redirection

$ sort < keats
Input/Output Redirection

$ sort < keats > sorted
$ cat sorted
Appending

$ date >> thedate
$ cat thedate

Useful for log files
$ grep planet keats
When a new planet swims into his ken;

Find all lines containing “planet” in “keats”
Pipes

$ cat keats
$ cat keats | grep planet
When a new planet swims into his ken;

Pipes connect output from one command to
the input of another command
Editing

No single obvious choice for editor

- vi – simple but difficult at first
- emacs – powerful but complex
- nano – simple but not really standard
$ nano keats

“^” means “hold down control”
\(^a\) : go to beginning of line
\(^e\) : go to end of line
\(^k\) : delete line
\(^o\) : save file
\(^x\) : exit
less

$ less /var/log/messages

• Used to read (not edit) files
• Very useful command
less

(space] : next page
b : previous page
g : go to end of file
^g : display location in file

/ : search
n: repeat search
sleep

$ sleep 5

• More useful than it seems
echo

$ echo hi
hi

• More useful than it seems
Multiple Commands

$ sleep 5; echo hi
hi

• Use ; to separate commands
Job Control

$ sleep 60
^c
$
$

• Use control-c to stop a running command
Job Control

$ sleep 60; echo hi
^c

What happened? Why?
exit

$ exit

• Logs out
• There are other ways of logging out

$ logout
$ ^d
$ ps

- Lists the processes you have running in this session
$ ps -e

• List every process
$ ps -aux

- List every process with a lot more information
- Flags for ps are numerous and inconsistent
$ top

- Lists running processes
- Updates every 3 seconds
- Many options to change display
- Many other top-like commands exist.
Questions?

Any questions?
Other Topics

• This is the end of the slides proper

• Many other commands and concepts that could be covered

• Following slides just list possibilities
Other Topics

- Environment variables
- $PATH
- which
- type
Other Topics

• More process control
• ^Z
• bg, fg
• jobs (command)
Other Topics

• Standard directories
• bin
• dev
• etc
• tmp
• usr
• var
Other Topics

• File systems
• df
• Space usage
• du
Other Topics

• Processes
• fork()
• exec()
• kill
Other Topics

• Shell scripts

• Programming