Introduction to Linux

Jerry Ebalunode

Center for Advanced Computation and Data Systems
(CACDS)
http://support.cacds.uh.edu

University of Houston
Houston, TX

@UHCACDS twitter

facebook.com/CACDSATUH

Overview

- Operating Systems and Linux
- Linux File System Hierarchy
- Basic Linux Commands
- Working with Files & Folders
- Text Editors
- I/O Redirection & Pipes
- Archiving
- File Permission Management
- Network File & Folder Transfer
- Open Lab and Homework

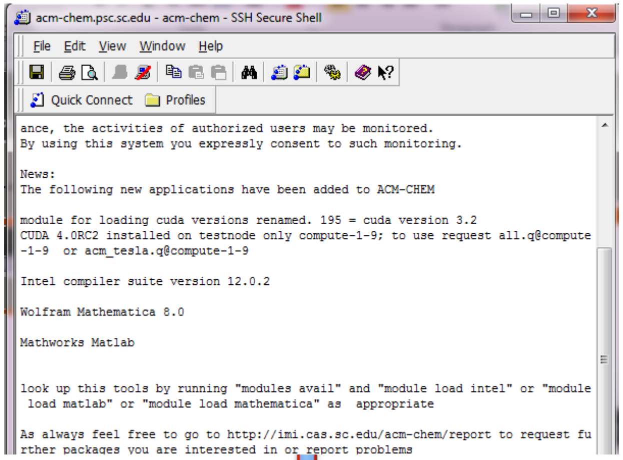
First Access Your Account

- Log into your accounts
  - Username or login = hpc_user**X**
  - Where x = sign in serial number 1 – 47
  - Password = **cacds2014**

  - Use your web browser
    - Firefox, Chromium or Google chrome
- Slides could be downloaded from URL below
  - http://129.7.249.171/workshops/intro2linux.pdf

The Role of an Operating System (OS)

- OS=Software & data that manages computer hardware resources (e.g. processor, memory, storage)


acm-chem.psc.sc.edu - acm-chem - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
ance, the activities of authorized users may be monitored.
By using this system you expressly consent to such monitoring.

News:
The following new applications have been added to ACM-CHEM

module for loading cuda versions renamed. 195 = cuda version 3.2
CUDA 4.0RC2 installed on testnode only compute-1-9; to use request all.q@compute
-1-9  or acm_tesla.q@compute-1-9

Intel compiler suite version 12.0.2

Wolfram Mathematica 8.0

Mathworks Matlab


look up this tools by running "modules avail" and "module load intel" or "module
 load matlab" or "module load mathematica" as  appropriate

As always feel free to go to http://imi.cas.sc.edu/acm-chem/report to request fu
rther packages you are interested in or report problems
```

- Provides a platform for running applications on phones, tablets, desktops, servers, clusters.

Android Phone/Tablet/Laptops

HPC Cluster

What is Linux?

- Linux is an OS just like Windows or Mac OS X
  - Technically speaking, Linux is the kernel: the program in a system that allocates the computer/server hardware resources to the other programs. Linux is normally used in combination with the GNU operating system utilities: the whole system is basically GNU with Linux added, or GNU/Linux
- Under development since 1991, started by Linus Torvalds

- Lightweight operating system

Why Create Linux

- Personal computers were becoming popular
- Microsoft's DOS was too limiting
- Commercial UNIX was expensive
- Needed compatibility with UNIX (IEEE POSIX)

Why Use Linux?

- General features of Linux:
  - Most distributions are free
  - Open-source (completely customizable)
  - Portable to nearly any hardware platform
    - cell phones, roku, steam0S devices, PS3, tablets, TVs, routers
  - Highly scalable to lots of cores, and or lots of memory
    - for instance: Blacklight supercomputer system at PSC
      - **4096 CPU cores**
      - **32 TB** main system memory
  - Highly efficient, therefore useful for computation
  - Robust and proven security model
  - Includes a complete development environment

Linux Distributions

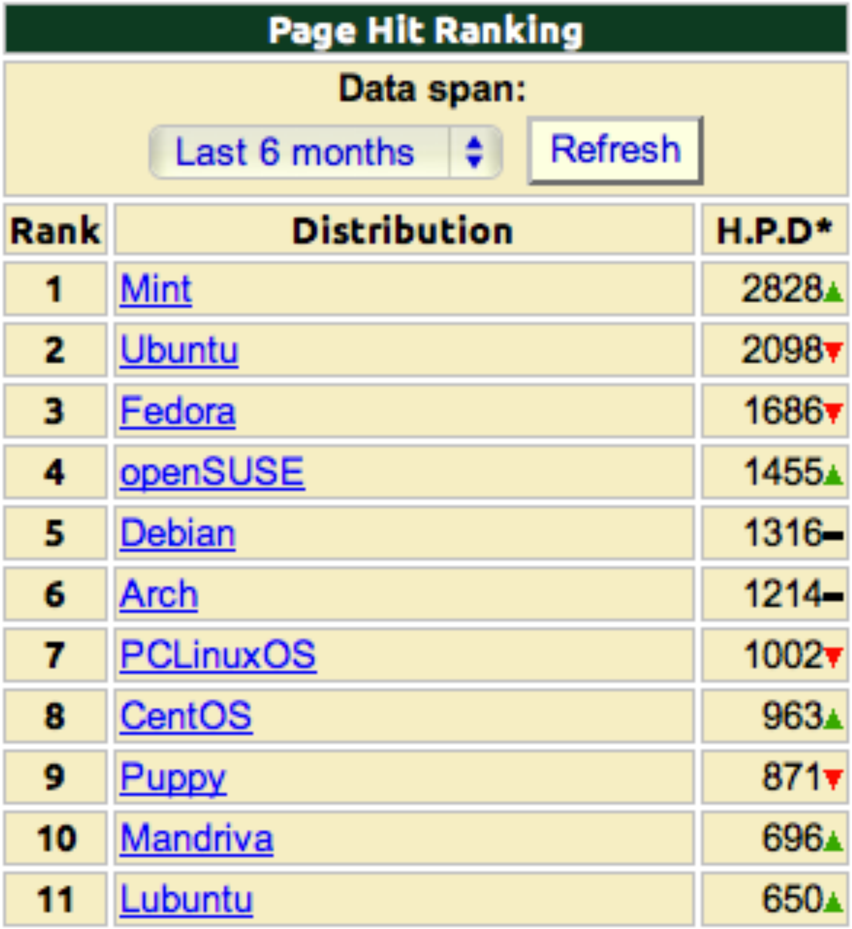
- Today there over 100 different versions of the Linux OS
  - also called ***distributions***

- Each "*distribution*" offers a <u>unique combination of features</u> and applications to suit needs of different users.

Tracking Linux Distributions

Distrowatch

- *Distrowatch.com* provides news, comparisons, popularity ranking ….. of various Linux distributions
  - *Moto: put the fun back in computing…*



Rank	Distribution	H.P.D*
1	Mint	2828▲
2	Ubuntu	2098▼
3	Fedora	1686▼
4	openSUSE	1455▲
5	Debian	1316–
6	Arch	1214–
7	PCLinuxOS	1002▼
8	CentOS	963▲
9	Puppy	871▼
10	Mandriva	696▲
11	Lubuntu	650▲

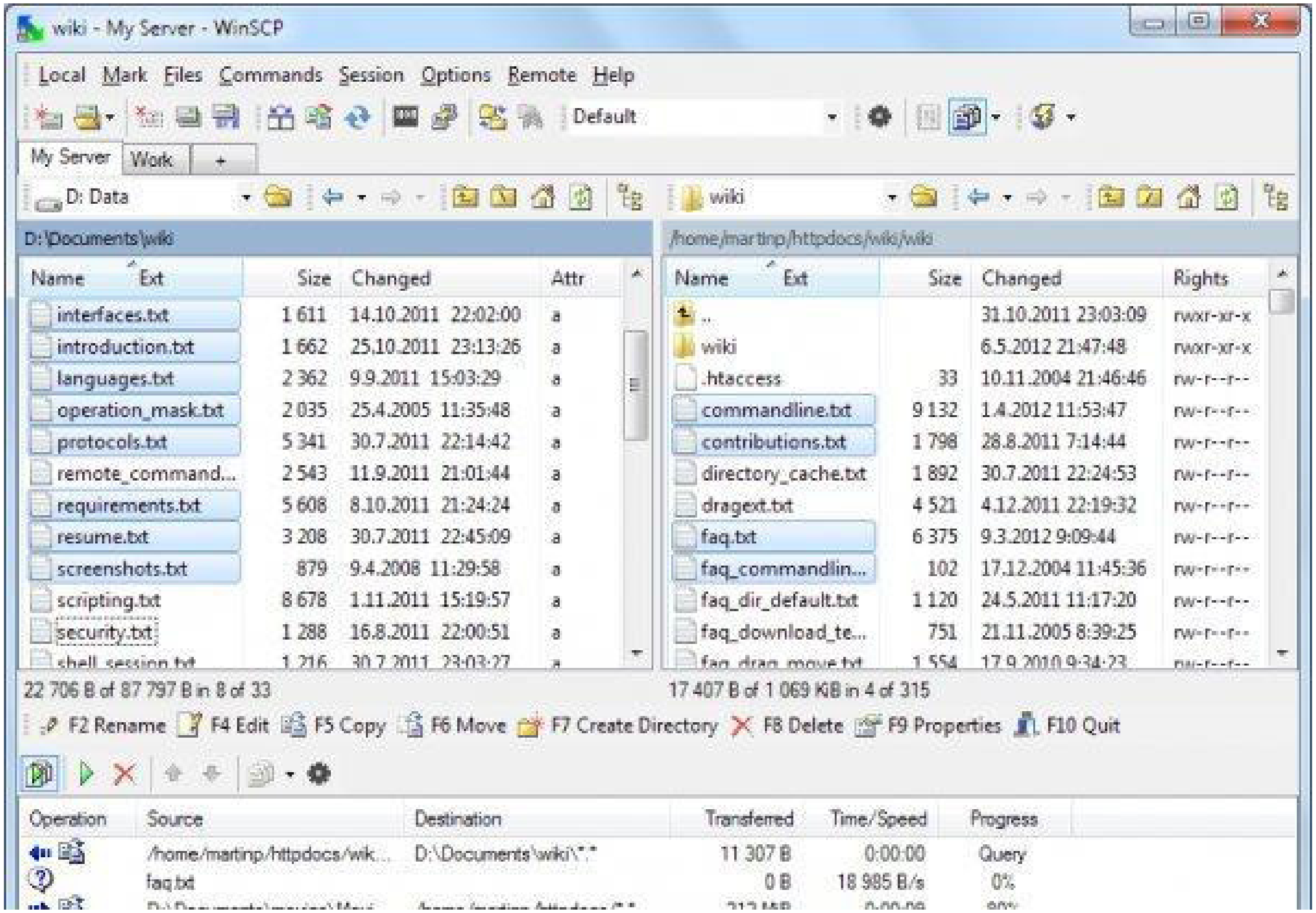
*H.P.D = hits per day

Connecting to Linux Systems

- Most popular way: Secure Shell (SSH) clients
  - Assumption: SSH server application Installed and Running on server
- SSH client other functions
  - Provide security, encryption, performance.
- Popular clients
  - OpenSSH (Linux & Mac OSX)
  - Putty for windows

Transferring Data Between Linux and Windows

- USE winSCP
- http://winscp.net/eng/index.php



D:\Documents\wiki

Name Ext	Size	Changed	Attr
interfaces.txt	1 611	14.10.2011 22:02:00	a
introduction.txt	1 662	25.10.2011 23:13:26	a
languages.txt	2 362	9.9.2011 15:03:29	a
operation_mask.txt	2 035	25.4.2005 11:35:48	a
protocols.txt	5 341	30.7.2011 22:14:42	a
remote_command...	2 543	11.9.2011 21:01:44	a
requirements.txt	5 608	8.10.2011 21:24:24	a
resume.txt	3 208	30.7.2011 22:45:09	a
screenshots.txt	879	9.4.2008 11:29:58	a
scripting.txt	8 678	1.11.2011 15:19:57	a
security.txt	1 288	16.8.2011 22:00:51	a
shell_session.txt	1 216	30.7.2011 23:03:27	a

22 706 B of 87 797 B in 8 of 33

/home/martinp/httpdocs/wiki/wiki

Name Ext	Size	Changed	Rights
..		31.10.2011 23:03:09	rwxr-xr-x
wiki		6.5.2012 21:47:48	rwxr-xr-x
.htaccess	33	10.11.2004 21:46:46	rw-r--r--
commandline.txt	9 132	1.4.2012 11:53:47	rw-r--r--
contributions.txt	1 798	28.8.2011 7:14:44	rw-r--r--
directory_cache.txt	1 892	30.7.2011 22:24:53	rw-r--r--
dragext.txt	4 521	4.12.2011 22:19:32	rw-r--r--
faq.txt	6 375	9.3.2012 9:09:44	rw-r--r--
faq_commandlin...	102	17.12.2004 11:45:36	rw-r--r--
faq_dir_default.txt	1 120	24.5.2011 11:17:20	rw-r--r--
faq_download_te...	751	21.11.2005 8:39:25	rw-r--r--
faq_drag_move.txt	1 554	17.9.2010 9:34:23	rw-r--r--

17 407 B of 1 069 KiB in 4 of 315

Operation	Source	Destination	Transferred	Time/Speed	Progress
	/home/martinp/httpdocs/wik...	D:\Documents\wiki\*.*	11 307 B	0:00:00	Query
	faq.txt		0 B	18 985 B/s	0%
	D:\Documents\movies\Movi...	/home/martinp/httpdocs/*.*	[illegible]	[illegible]	[illegible]

Command Line Interface (CLI)

- Most Linux desktop systems can be full-featured, user-friendly graphics
  - i.e. graphical user interfaces (GUIs) to access most utilities

- But in High Performance Computing (HPC) environment, the CLI is the most common way to access & use the OS.
  - Reason: performance is more important than eye candy
    - prefer to dedicate all CPU cycles to solving the computational problem
  - CLI is light weight
    - not CPU intensive

- Therefore, knowing how to complete tasks from the **command line is very critical**.

Getting Started

Use the terminal to download intro2linux.zip file to your home directory

- Run the following commands

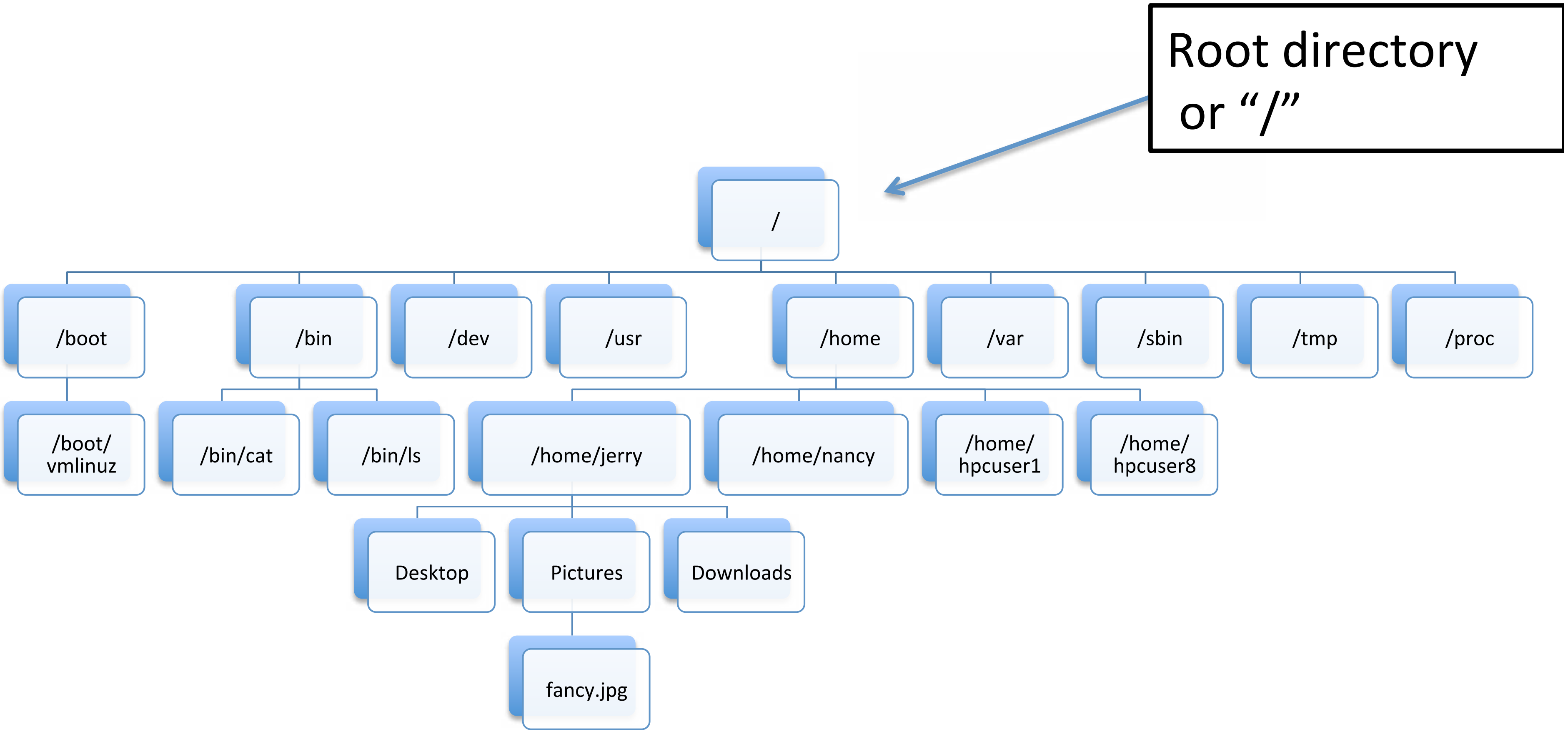
```
cd

cp   /share/apps/tutorials/intro2linux.zip    ~

unzip    intro2linux.zip


cd  intro2linux
```

Now, you can begin working with tutorial files on your terminal

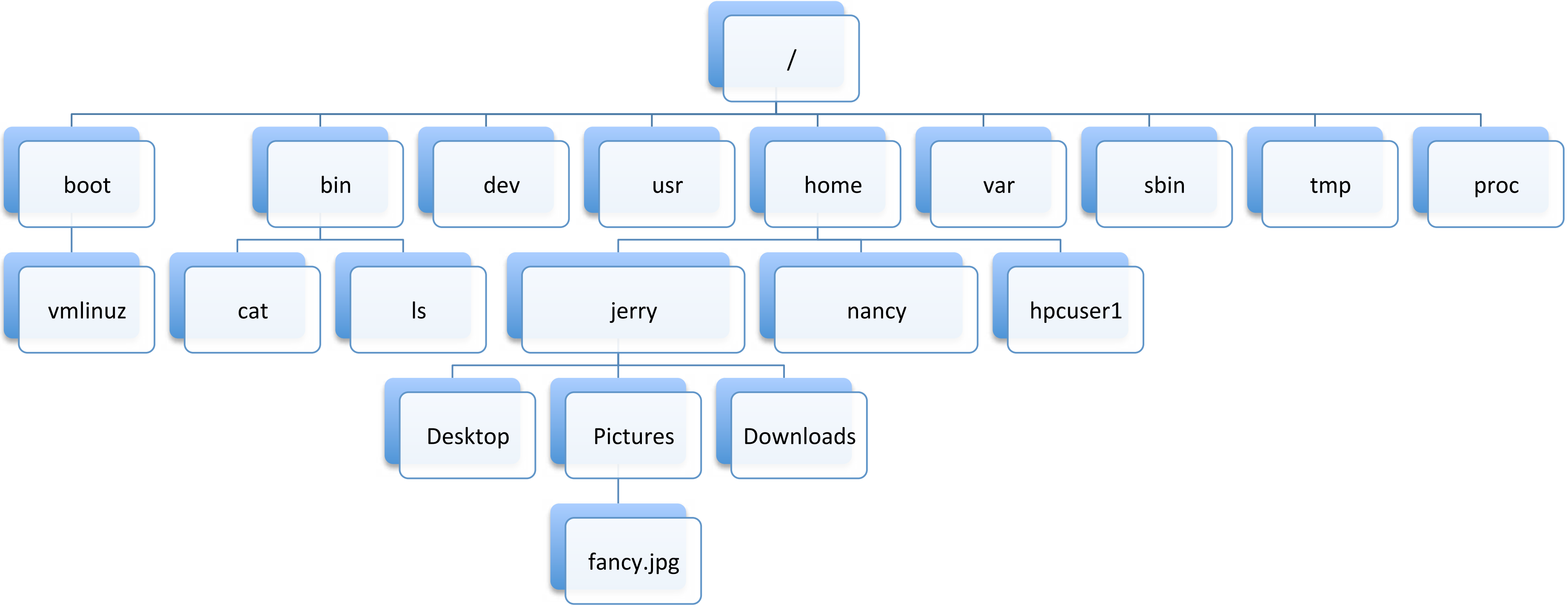
Linux File System

- A file system is the way files are organized on the disk
  - the methods and data structures that an operating system uses to keep track of files on a disk or partition
  - Linux uses several types of file systems
    - Extended file systems : Ext2, Ext3, Ext4, fat, ntfs**
    - Read, write and execute operations possible on Ext2-4, fat
    - Read and Execute operations only for ntfs
- The operating system stores data (i.e, files and directories) in the file system in a hierarchal order

File System Hierarchy



File System Hierarchy



Full PATH to "Desktop" folder in Jerry's Account

/home/jerry/Desktop

Full PATH to fancy.jpg file
/home/jerry/Pictures/fancy.jpg

Some Basic Linux Commands

pwd – prints your current working directory

ls – list the contents of the directory

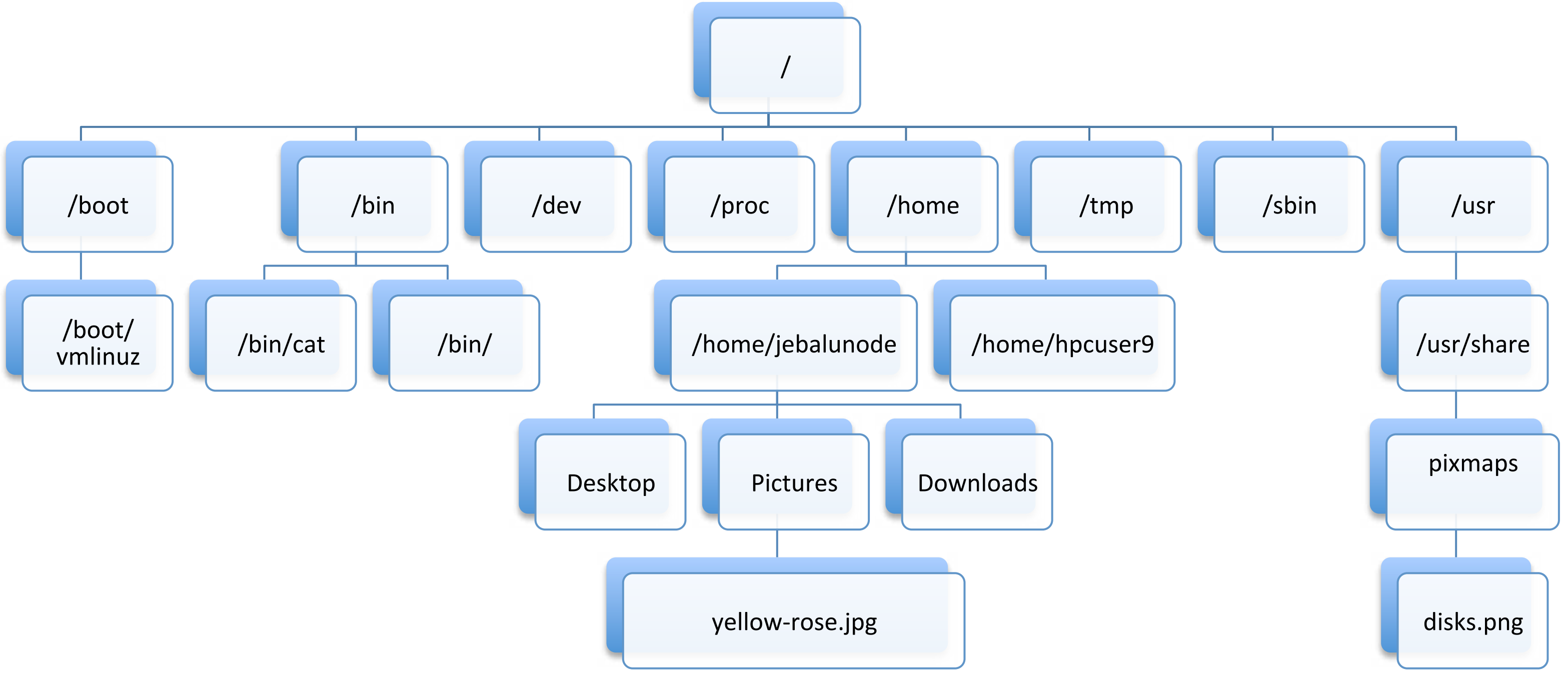
cd – change directory (defaults to home directory)

cp -- copy file(s)

example: cp file1.jpg file2.jpg

File System Hierarchy

Navigating around



Exercise 1: change directory to pixmaps folder

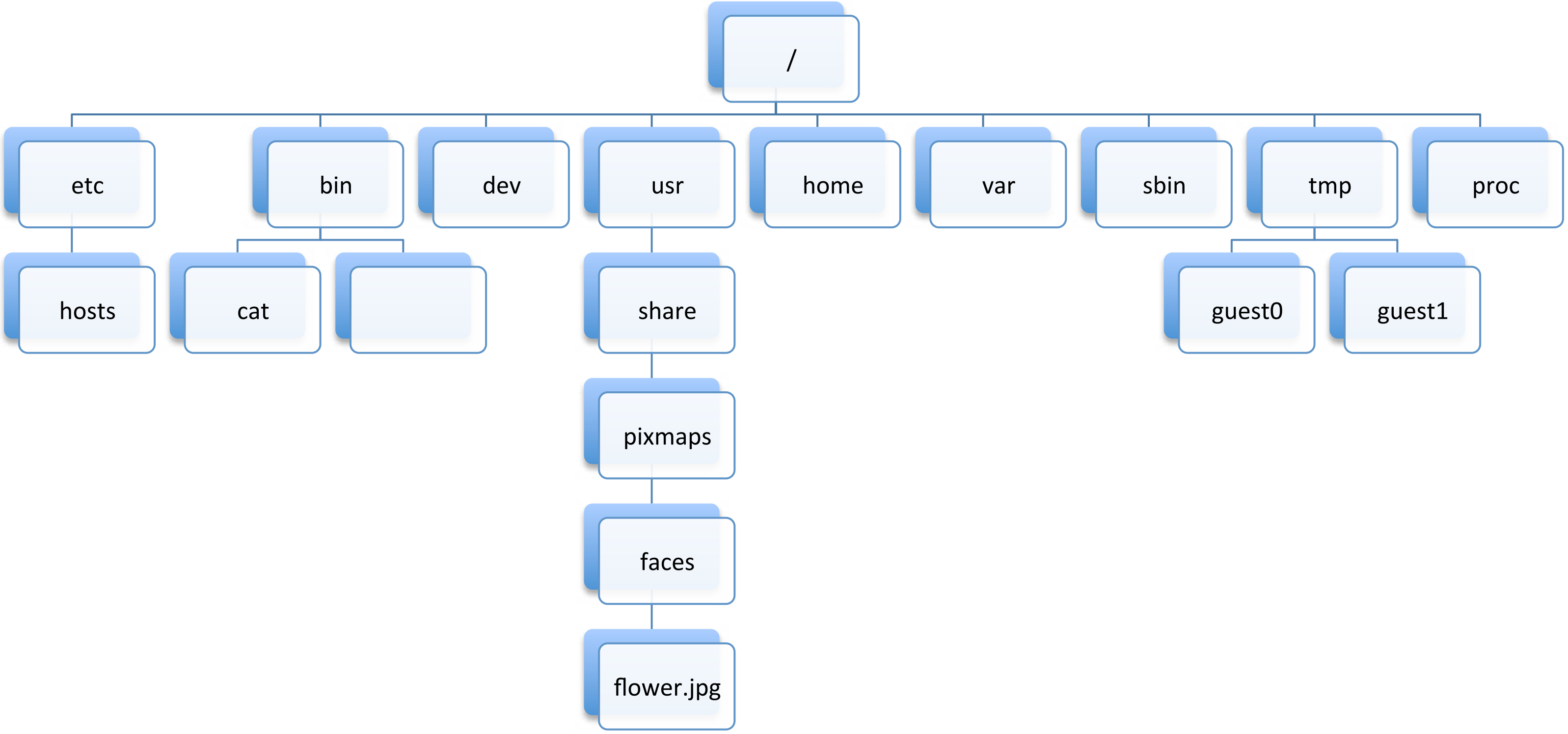
```
cd   /usr/share/pixmaps
pwd
```

Exercise2: copy "disks.png" file to /tmp directory

```
cp    /usr/share/pixmaps/disks.png    /tmp
cd  /tmp
 ls    disks.png
```

Pop Quiz (1)

File System Hierarchy



write full PATH to flower.jpg

write a change directory command to go to the faces directory starting from root (/)

write a copy command to copy flower.jpg to your home directory

write a copy command to copy flower.jpg to intro2linux folder

Basic Linux Commands

pwd	– prints your current working directory
whoami	– prints the name of the current user
who	– prints a list of all users who are logged-in
ls	– list the contents of the directory
cd	– change directory (defaults to home directory)
clear	-- clears printed content on terminal window/console
date	– prints the current date and time
ps	– prints snapshot of current shell processes
env	– list all environment variables/settings
df	– prints summary of disk usage
time	-- print the execution time of an application

Linux Commands Accept Arguments

- Some commands accept “arguments” that change the behavior of the command, or tell the command exactly what to do.

df -h – prints “human readable” disk usage

echo Hello – prints adjacent string to screen

mkdir new_project – creates a new directory called “new_project“

cd new_project – change directory (move into “new_project” directory)

cd .. – back up/out of the directory you’re in

cd ../.. – back up 2 levels/directories
cd or cd ~ - change directory to home directory

which mkdir – shows any command’s full path

Pop Quizz(2)

- Give a command to identify logged in users

- How can you change your working directory to 4 levels/directories higher than your current working directory?

Pop Quizz(2)

- Give a command to identify logged in users

  ```
  who
  ```

- How can you change your working directory to 4 levels/directories higher than your current working directory?

  ```
  cd  ../../../..
  ```

Working with Files

Here are some commands that are useful for working with files and folders:

cp file1 file5	– create a copy of a file
mv file3 new_name	– move (or rename) a file
rm file4	– delete a file (rm -r [dir] for a folder)
file file1	– print the type of file
more dictionary.txt	– read a text file, one “page” at a time
head -n file1	– print the first n lines of a file
tail -n file1	– print the last n lines of a file
grep ing file1	– print lines that match pattern “ing”
cat file5	– print the contents of a file to the screen

Sorting Data

- We use the “sort” command to display the contents of a file or data stream in **order** by lines. Note it does not change the contents of the file

- Examples

  quickly check file content

  ```
  head z-a.txt
  ```

  ```
  sort  z-a.txt
  ```

- Reverse sort

  ```
  sort  -r    a-z.txt
  ```

- Many more options available

Pasting Files

“paste” command lets you merge two or more input streams side by side

```
cat  serial.txt
output:          1
                 2
                 3
cat    data.txt
 output:        0.2
               -0.3
                1.2

paste     serial.txt     data.txt
   output:    1   0.2
              2  -0.3
              3   1.5
```

Cutting files

- Use the “cut” command to print out selected sections from each line of an input stream or file
  - Assumes contents of file or input stream are “tab” delimited

```
cut  -f 1,2  all.lanes.txt
```

  - Prints field(s) 1 and 2

```
cut  -d  ‘ ‘   -f 2-     all.lanes2.txt
```

print all fields starting from field 2

Note the –d ‘ ‘ informs program that data is delimited with spaces rather than default tab.

Man Pages & History

- Nearly all commands available for use on a particular system have an accompanying “manual page”:

```
man  cp
man  ls
```

- Note: To exit the manual page (man page) viewer
  - simply type the letter **Q**
  - ↑ or “up” arrow to scroll through commands you’ve used.
- You can view the entire history of commands you have used by executing

```
history
```

Text Editors

- Nearly all Linux distributions come with a variety of text editors for writing and editing files or scripts.
- Some of the most common are nano, **gedit**, vi, vim, and emacs.

- We will be using nano for this session
  - Example:
    - **nano hello.txt** - opens a file called hello.txt for editing
    - [write something]
    - **CTRL+o or (^o)** to save
      - note you might be prompted to rename the file, but you don't have to. Just hit enter key when prompted to save with same name
    - **CTRL+x or (^x)** to exit nano

I/O Redirection

- By default, command line programs print to “stdout” (standard out = the computer monitor).
- I/O redirection is a way of manipulating the input/output of Linux programs, allowing you to capture the output in a file, or send it to another program.
- Example: Get the first 9 lines from the dictionary:

```
head  -n  9  dictionary.txt

head  -n  9  dictionary.txt  >  temp.txt

more  temp.txt

wc  -l  temp.txt
```

-counts the number of lines in a file

- The **“>”** character performs a “redirect,” taking the output of the head command and putting it into the file temp.txt.

I/O Redirection: Append

- Use **“>>”** to append to a file without overwriting:

```
echo “Right now it’s Friday”   >>   temp.txt
cat   temp.txt
```

I/O Redirection: Pipes

- Another useful technique is to redirect one program's output (stdout) into another program's input (stdin). This is done using a "pipe" character.

```
 cat   z-a.txt   |  sort

cat dictionary.txt
cat dictionary.txt  |  grep   ing
cat dictionary.txt  |  grep   ing   |  grep  un
```

Pattern Matching with grep

```
grep  ing   dictionary.txt
```

searches the file for lines containing “ing” and prints them to stdout

```
grep  -v   ing   dictionary.txt
```

searches the file for lines that do NOT contain pattern “ing” and prints them to stdout

```
grep  -f  items2searchFor.txt   dictionary.txt
```

Reads a database of patterns from file “**item2searchFor.txt”**
searches file “***dictionary.txt”*** for lines that matches any of the patterns and prints them to stdout

```
grep  -f   items2searchFor.txt   theraven.txt
```

Pop Quizz (3)

- Give an examples of using the “grep” command for the following
  - With only output redirected
  - Within a pipeline

Also Good to Know

top -will list processes/tasks running on your system … similar to task manager on windows

q or CTRL-c can help you get “unstuck”

tr -translate or delete characters

```
echo   linux   |  tr   ‘a-z’  ‘A-Z’

echo  'world}}}' | tr  '}'   '!’
```

TAB - used for command completion

Find a file:

```
find  ./   -name    "name.of.my.file.txt”
locate   name.of.my.file.txt
```

To close your shell.

```
exit
```

File Permissions by User Types

```
cd intro.linux
ls -l
```

```
• -rwxr-xr-x 1 jebalunode public 622783 2010-12-03 09:15 dictionary.txt
• -rwxr-xr-x 1 jebalunode public   8262 2010-12-03 09:15 icb.txt
• -rwxr-xr-x 1 jebalunode public 891777 2010-12-03 09:15 personnel.txt
• -rwxr-xr-x 1 jebalunode public   6599 2010-12-03 09:15 theraven.txt
```

- Three user types associated with Linux files

Owner(u)	Group(g)	Other/ world (o)	
rwx	r-x	r-x	jebalunode public theraven.txt

File & Directory Permissions

- Control access to files & directories by setting permissions

- Setting permissions using read /write or executable :
  - **chmod ug+r file0** --makes a file readable by owner (**u**) and group (**g**)
  - **chmod ug+w file0** –writes to the file are permitted
  - **chmod ug+x file0** --makes a file executable
  - **chmod ug+rwx file0** --makes a file executable, writable and readable

- **chmod ugo+r file0** --makes a file readable by owner (**u**) and group (**g**) and world(**o**)

- For directories you apply the recursive "R"
  - **chmod -R u+rx directory** --makes a directory readable

```
cd   intro.linux
ls  -l
        •   -rwxr-xr-x 1 jebalunode public 622783 2010-12-03 09:15 dictionary.txt
        •   -rwxr-xr-x 1 jebalunode public   8262 2010-12-03 09:15 icb.txt
        •   -rwxr-xr-x 1 jebalunode public 891777 2010-12-03 09:15 personnel.txt
        •   -rwxr-xr-x 1 jebalunode public   6599 2010-12-03 09:15 theraven.txt
```

File Permissions cont.
Using Octal Notation

0	–	no permission	
1		--x	execute
2		-w-	write
3		-wx	write and execute
4		r--	read
5		r-x	read and execute
6		rw-	read and write
7		rwx	read, write, execute

“-rwxr-xr-x”	=	755
“-rw-rw-r--”	=	664
“-r-x--------”	=	500

- you can change permission with octal notation

```
chmod  755  dictionary.txt
chmod  -R   755   ../intro2linux/
```

Pop Quizz (4)

- write a command to make the file called dictionary.txt to be only readable by you

Accessing Remote Linux Servers

- use ssh to login to remote system
  - syntax:

```
ssh   username@server_hostname_or_ip_address
```

- ssh jebalunode@compute-0-0
- ssh compute-0-0
- ssh   jebalunode@10.1.1.1

Archiving your Work
Pack and Unpack

- creating an archive
  - option (-cvzf)== create a compressed file archive in verbose mode

```
tar   -cvzf     my_compressed_archive.tar.gz     my_directory/
```

  - “**tars**” (like “zipping”) a directory into a single compressed file

- unpacking/extracting an archive
  - option (-xvzf) ==extract a compressed file archive in verbose mode

```
tar   -xvzf     my_compressed_archive.tar.gz
```

“**tar -xvzf** ” (like “unzipping”) a compressed file, which may contain a folder

Data Transfer in Linux systems

Sending Data to a Remote Location

- use scp for file and folder transfers
  - syntax:

  scp *filename* username@server:path_to_destination

  - scp dictionary.txt jerry@cusco.hpcc.uh.edu:/home/jerry/

- useful for directory or folder transfers. note -r option

  scp -r *my_directory* username@server:path_to_destination

- scp => Secure Copy. Used to copy a file or folder or directory to another computer where you have a user account.

More on SCP
Copying Data from a Remote Location

- syntax:

  ```
  scp  username@server:path_to_remote_file    path_to_destination_file
  ```

- Example

  ```
  scp  jerry@cusco.hpcc.uh.edu:/home/jerry/dictionary.txt   mycopy.txt
  ```

- For directories include “-r” or recursive option

-
  ```
  scp  -r  username@server:path_to_remote_dir   path_to_destination_dir
  ```

Open Lab

- Take a few minutes to try some of the commands you've learned. Perhaps try combining commands to give you very specific results.
- If you have not done so already, use your web browser to copy intro2linux.zip where you can begin working with it: then

```
cd
mv   intro2linux   intro2linux.old   ###Cleanup

cp   /share/apps/tutorials/intro2linux.zip   ~
## to get tutorial package

unzip   intro2linux.zip

cd    intro2linux
```

- execute the commands you learnt

Open Lab

Exercise

- Give a command to
  - Redirect standard output from a “sort” command to a file named “phone_list”, using the file named “numbers” as input.
  - Translate all occurrences of the characters [and { to the character (, and all occurrences of the characters] and } to the character), in the file add.c . (Hint: Refer to tr)
  - Create a file named book.txt that contains the contents of two other files: part1.txt and part2.txt

Recommended Literature

- The Linux Command Line: A Complete Introduction Paperback by Shotts

- Practical Guide to Linux Commands, Editors, and Shell Programming by Sobell

- Learning the bash Shell: Unix Shell Programming (In a Nutshell (O'Reilly))
- Free Ebooks
  - Advanced Bash-Scripting Guide
    - http://tldp.org/LDP/abs/html/
  - Bash Guide for Beginners
    - http://tldp.org/LDP/Bash-Beginners-Guide/html/

Acknowledgements

- Center for Advanced Computation and Data Services
  - Barbara Chapman
  - Tony Curtis