Introduction to Linux

Jerry Ebalunode
jebalunode@uh.edu
Center for Advanced Computing and Data Systems
(CACDS)
http://cacds.uh.edu

http://support.cacds.uh.edu

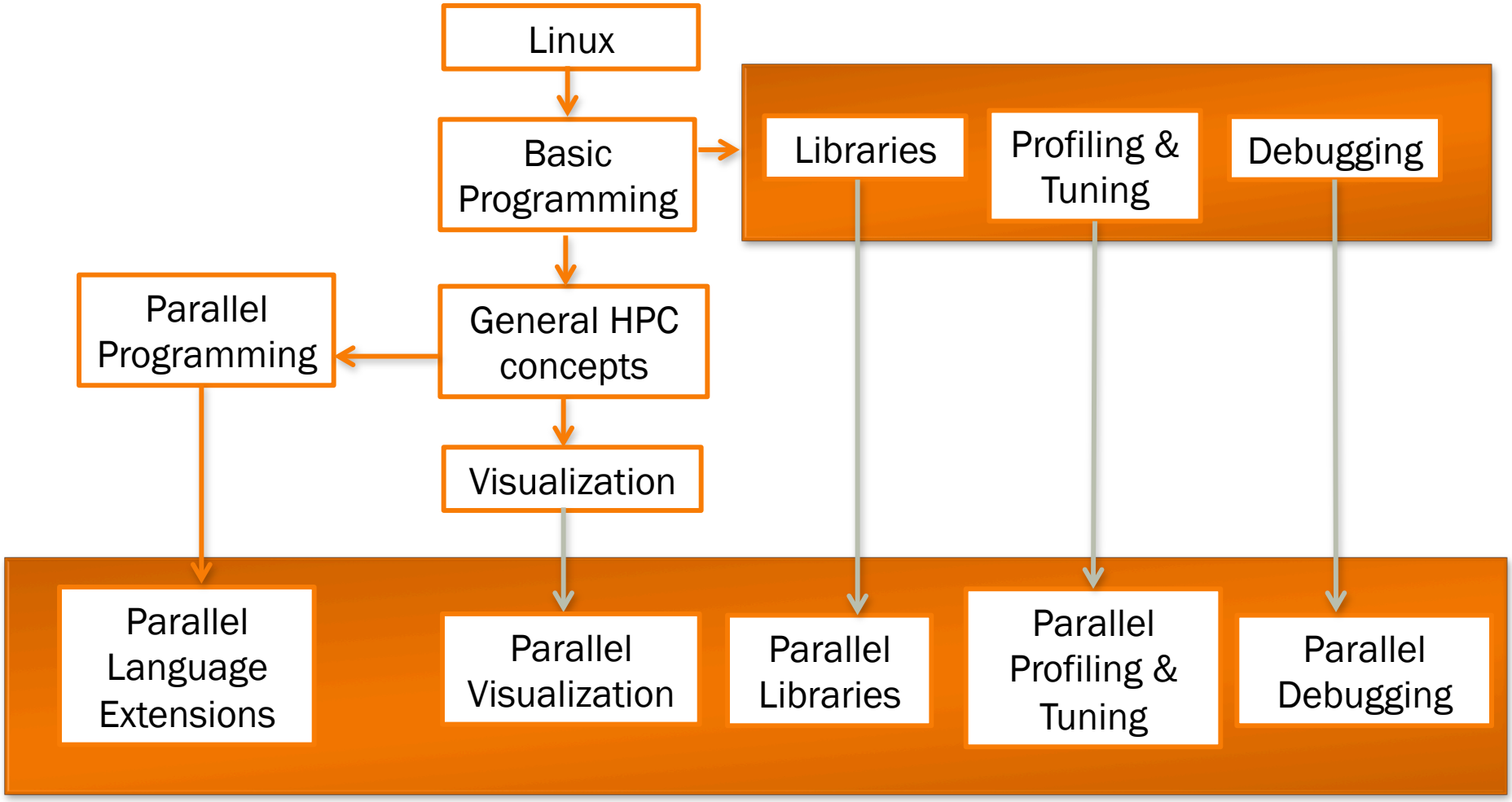
University of Houston
Houston, TX

CACDS

Mission Statement

- The Center for Advanced Computing and Data Systems (CACDS) provides high performance computing resources to advance Tier One research and education goals at the University of Houston (UH).

HPC User Training Road Map



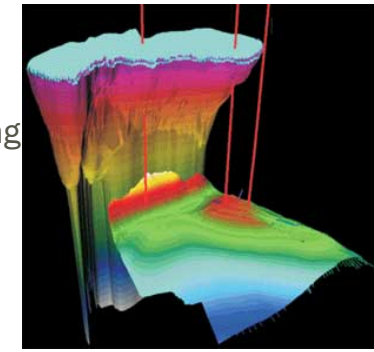
HPC Training

Location: PGH 200
48 seat class room

- Linux
- Shell programming
- Cluster computing on Maxwell
- XSEDE Cyberinfrastructure and allocation process
- Python
- MATLAB
- C++
- Fortran
- Parallel Programming using OpenMP
- Parallel Programming using MPI
- Parallel numerical libraries

Location: PGH 235
12 seat class room

- Nvidia CUDA
  - introductory, intermediate and advanced
  - profiling, debugging & tuning
- OPENACC
  - intermediate, advanced
  - profiling, debugging & tuning
- Intermediate and Advanced MPI
  - profiling, debugging & tuning
- Intel Xeon Phi Programming
  - Native computing
    - OpenMP, Cilk++
  - Offload Execution
  - Symmetric Execution
  - profiling, debugging & tuning
- Visualization at scale
  - Paraview
  - VisIt



CACDS Training
www.cacds.uh.edu/education/courses

UNIVERSITY of HOUSTON
CENTER FOR ADVANCED COMPUTING & DATA SYSTEMS

ABOUT | RESEARCH | EDUCATION | RESOURCES | NEWS

Training Courses

EDUCATION

Training Courses>>
- Events
- Lectures
- Workshops
- Technical Seminars

CACDS has launched several HPC training courses for members of the UH research community. Registration is required to take the courses. Please click the date you would like to attend under each course to register.

CURRENT TRAINING

CLASS NAME
Introduction to Linux
Introduction to Shell Programming
Introduction to Cluster Computing and Maxwell Cluster
Introduction to C++ Programming
Introduction to Python Programming
Introduction to Parallel Programming with OpenMP
GPGPU Parallel Programming with OpenACC
GPGPU Parallel Programming with CUDA
Introduction to Parallel Programming with MPI
Introduction to Matlab
Introduction to Intel Xeon Phi Programming
Introduction to High Performance Numerical Libraries
Visualization with VISIT I
Visualization with VISIT II
Visualization with Paraview I

Upcoming Training Opportunities
Workshops

- XSEDE HPC Monthly Workshop - Big Data
  - August 4, 2015
  - PGH 200
- Supercomputing for Everyone Series: Performance Tuning Summer School
  - August 17 - 21, 2015
  - PGH 200
- Science Visualization
  - August 24 - 25, 2015
  - PGH 200
- Visit http://www.cacds.uh.edu/education/workshops/ to learn more

Course 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

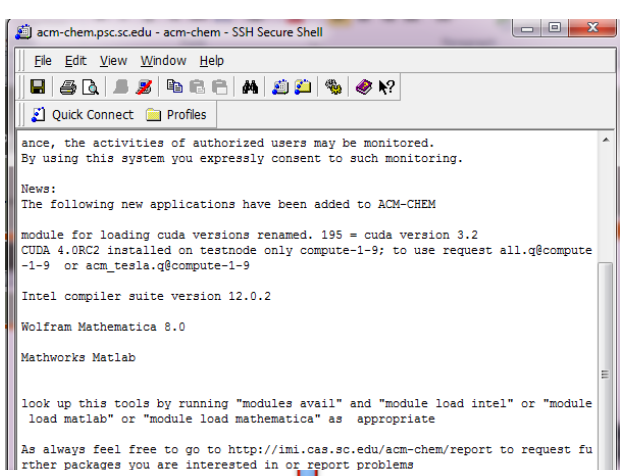
- Log into your accounts
  - Username or login = hpc_user**X**
  - Where x = sign in serial number 1 – 47
  - Password = **cacds2014**

  - Use your web browser
    - Firefox, Chromium or Google chrome

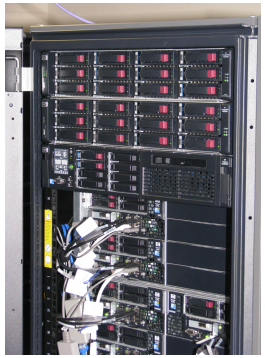
- Slides are available on your desktop
  - http://129.7.249.171/workshops/intro2linux_summer2015.pdf

The Role of an Operating System (OS)

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



- 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:
      - U.S. Postal Service
        - » **1024 CPU cores** and 16 TB main memory
      - 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 unique combination of features 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…*

Page Hit Ranking

Data span: Last 6 months Refresh

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▲

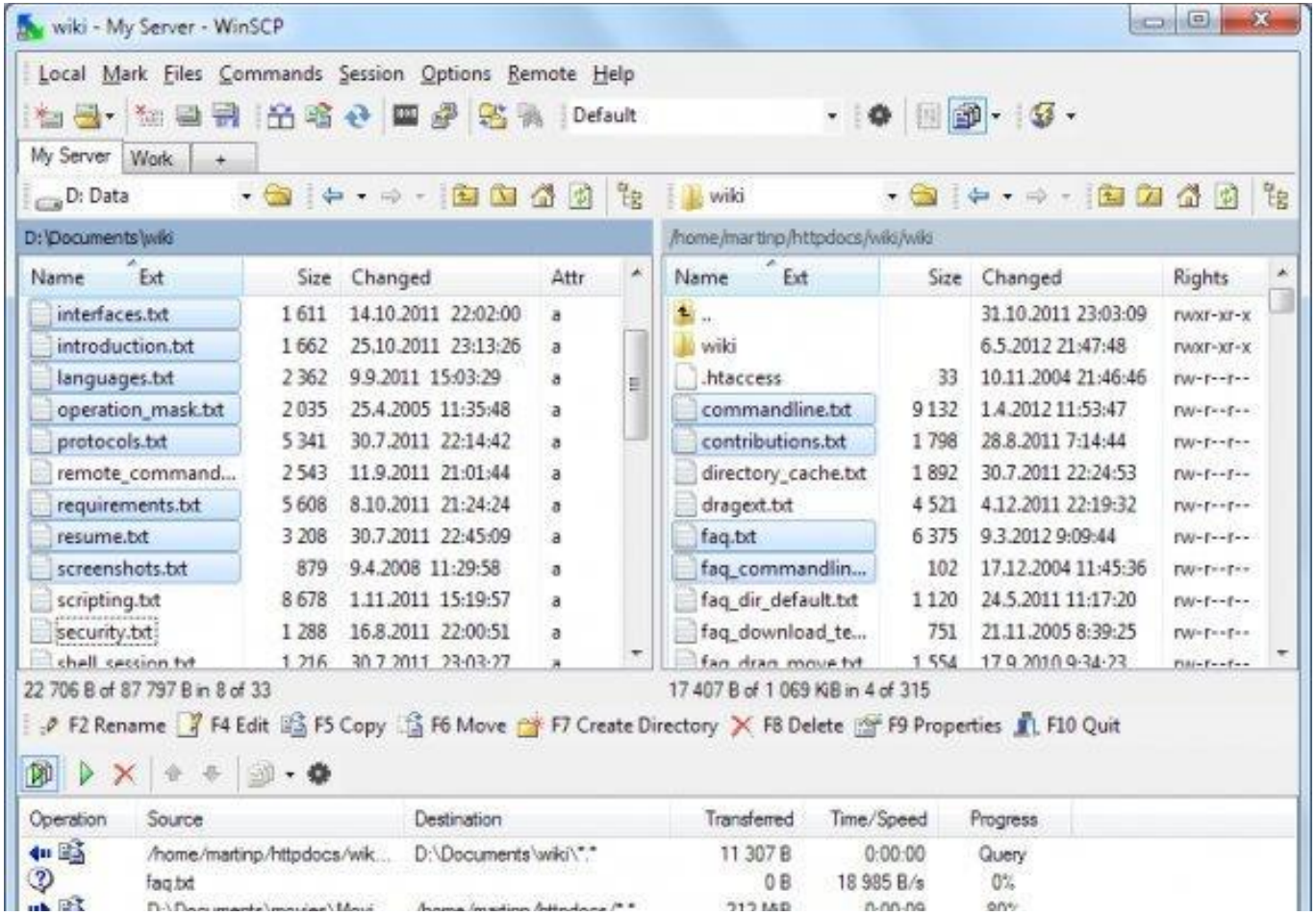
*H.P.D = hits per day

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



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

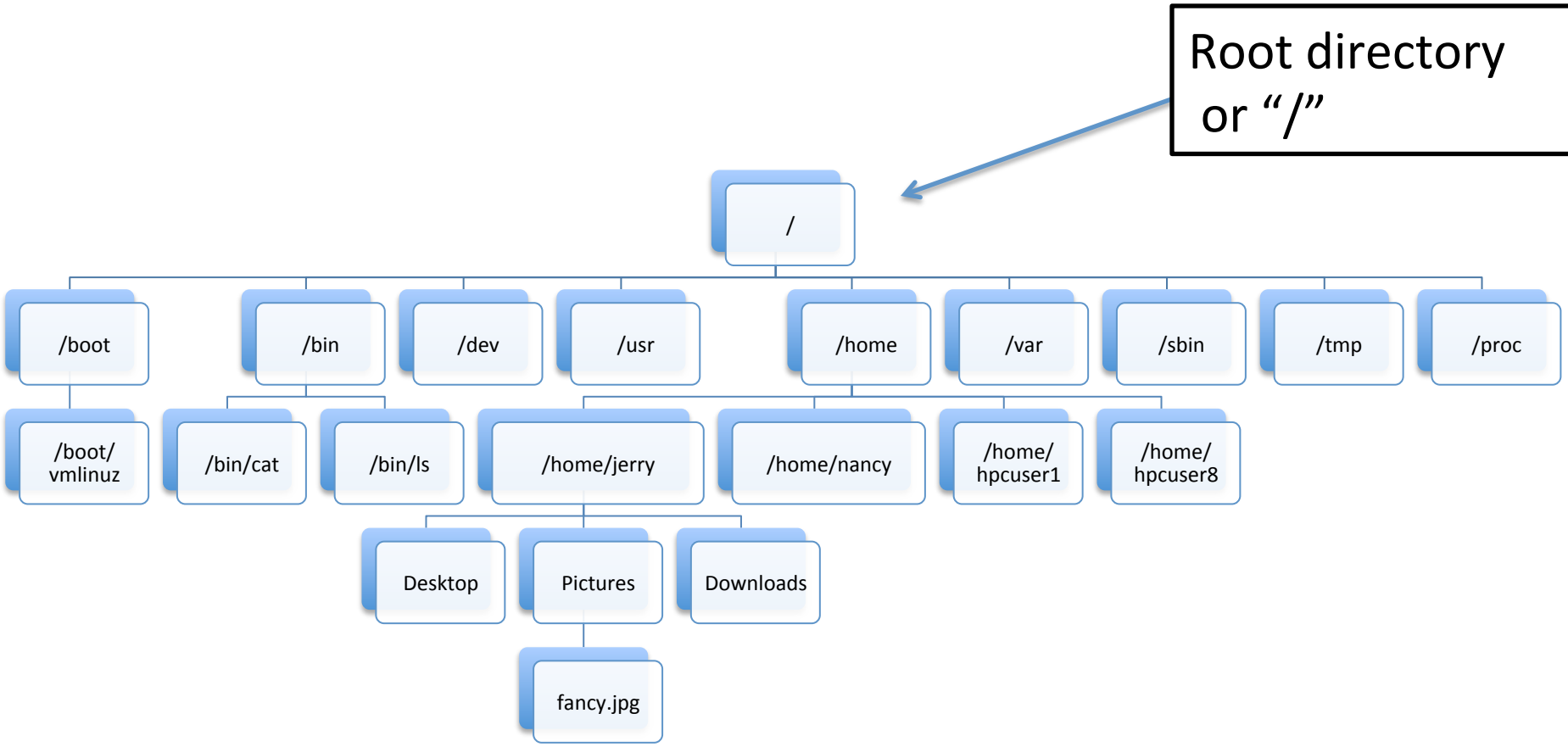
```
cd

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

unzip   intro2linux.zip


cd  intro2linux
```

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

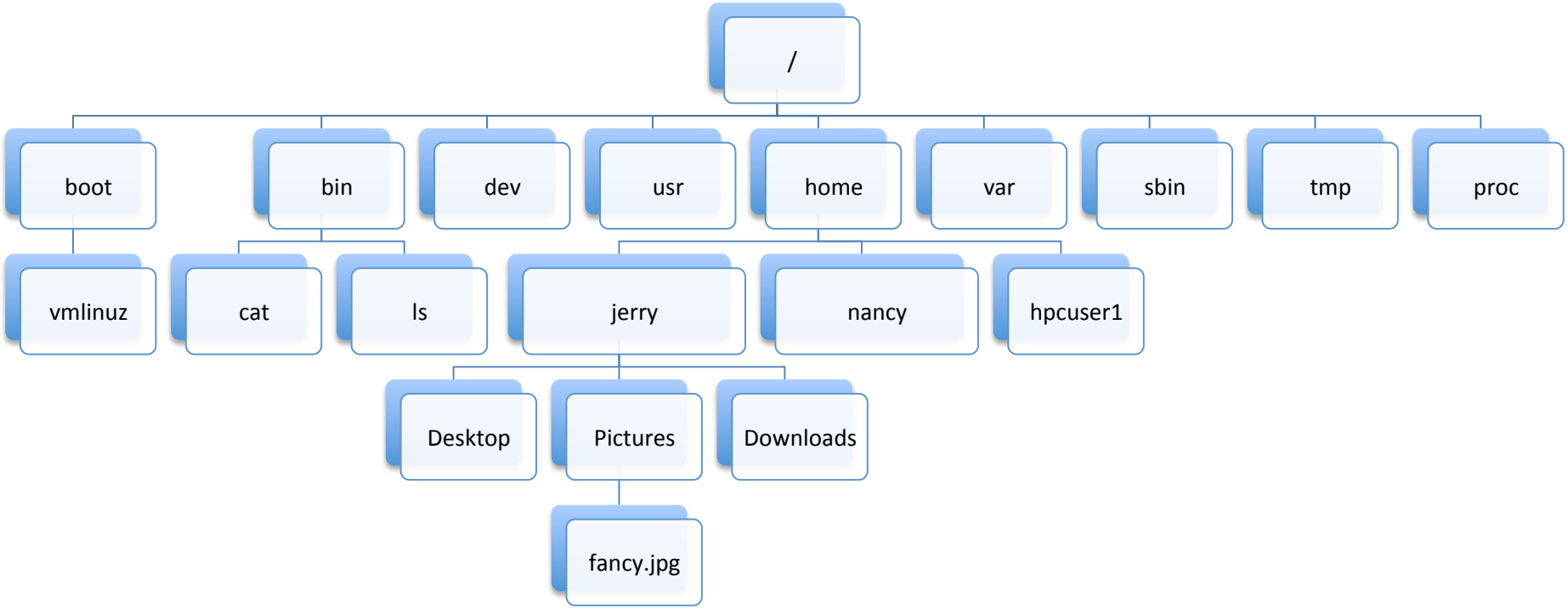
Linux File System

- A file system is the way files are organized on the disk
  - 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
    - Typically Read and Execute operations only for ntfs on most distros like Redhat
- 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)

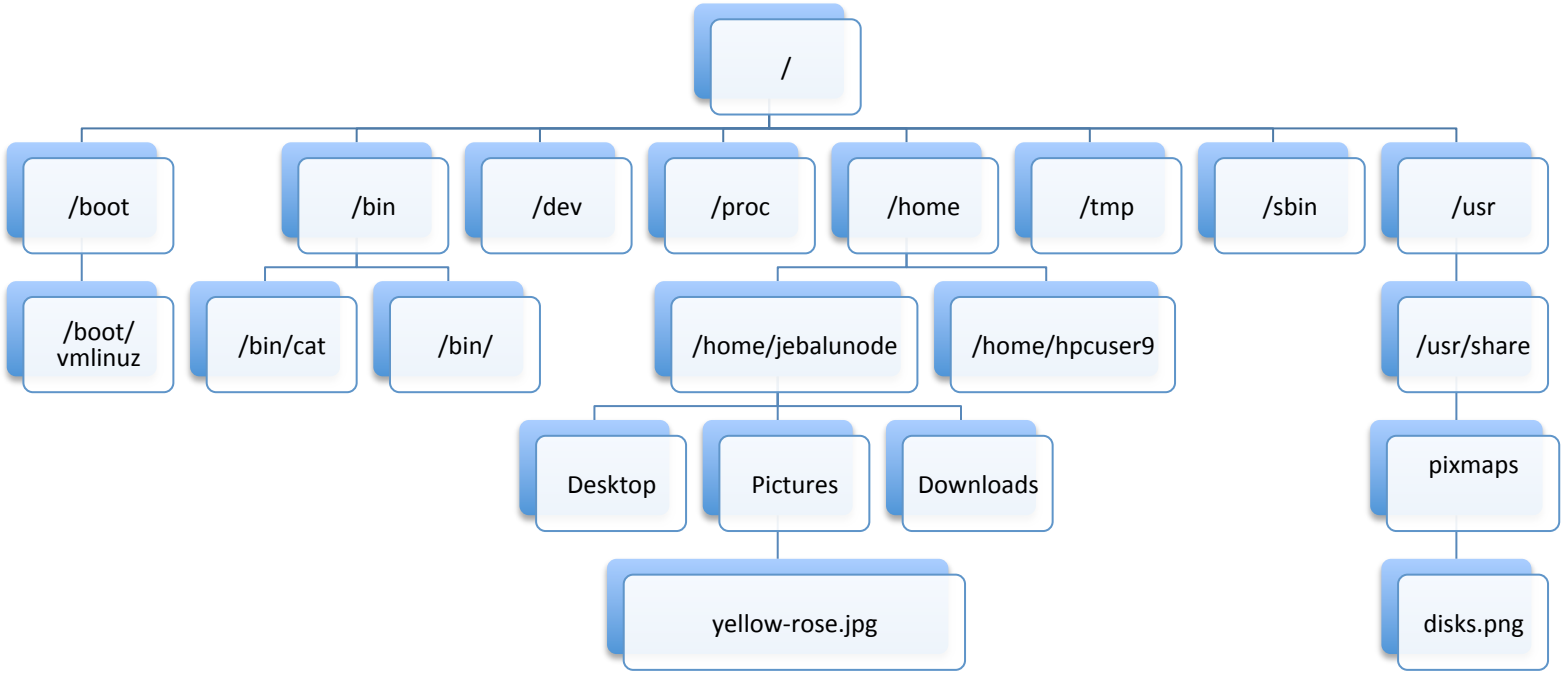
cp -- copy file(s)

example:

cp file1.jpg file2.jpg

File System Hierarchy

Navigating around



Exercise 1: change directory to pixmaps folder

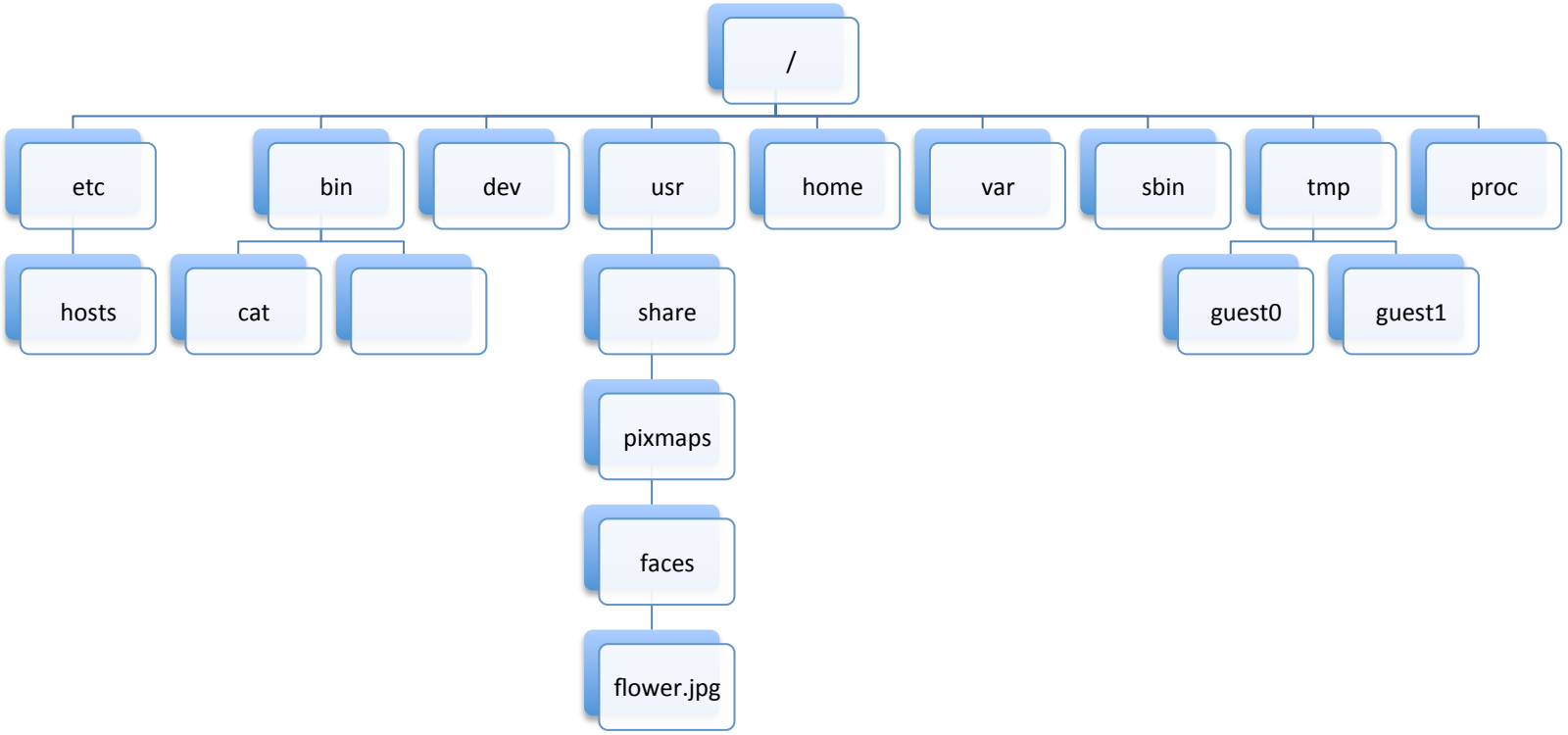
```
cd   /usr/share/pixmaps
pwd
```

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

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

Pop Quiz

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 Quiz

- 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 Quiz

- 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

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

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

  sort z-a.txt > result.txt

- Reverse sort

  sort -r a-z.txt

  sort -r a-z.txt > result.txt

- Many more options available

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 Quiz

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

- write a command to make the file called theraven.txt to be writable & readable by you and your group

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

Pop Quiz

- write a command to change to your home directory
- write a command to make a compressed archive "tar.gz" of your work using the intro2linux folder as input
  - call the archive "Serial#_my_archive.tar.gz"
- write a command to copy the archive to /tmp
- write a command to unpack the archive

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.

Pop Quiz

- write a command to change to your home directory

- write an scp command to copy the compressed archive “Serial#_my_archive.tar.gz” to a remote location “/tmp” on server with ip address 10.1.1.1
  - Note remote path is /tmp

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

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/

Support

- CACDS Support
- http://support.cacds.uh.edu

- Jerry Ebalunode
  - jebalunode@uh.edu
  - PGH 231
  - Office hours: 10-12pm Wednesday
  - 713-743-1755