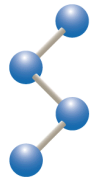


Introduction to Linux

Phil Mercurio

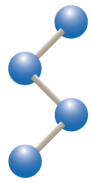
The Scripps Research Institute

`mercurio@scripps.edu`



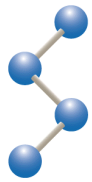
Session Overview

- What is Linux
- Shells & Windows
- The Linux File System
- Assorted Commands



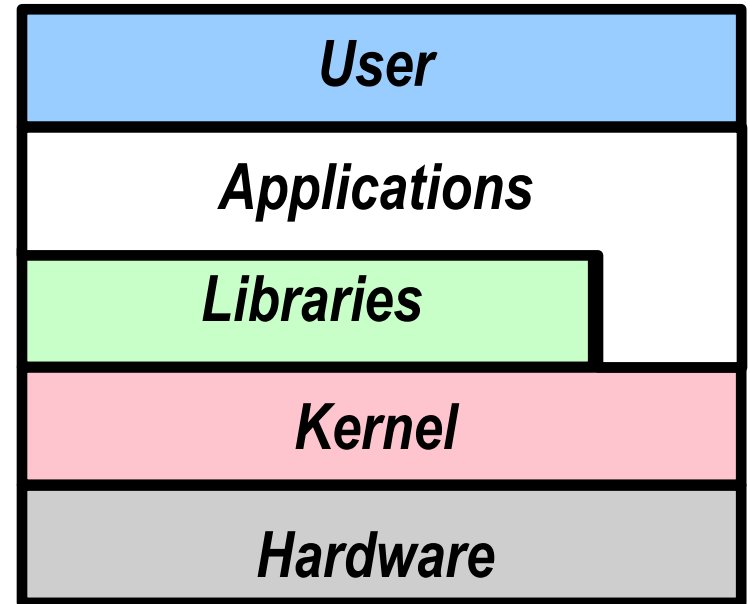
What Is Linux?

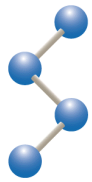
- Linux is a free open-source operating system based on Unix.
- Linux was originally created by Linus Torvalds with the assistance of developers around the world.
- Developed under the GNU General Public License, the source code for Linux is freely available to everyone.



An Operating System

- The Kernel: a program that's always running, providing an interface to the hardware
- The Filesystem: schemes for storing data on disk drives, networks, and other media
- Libraries & Applications: a collection of software that provide services or are used directly by the user





Source Code

- A program is a sequence of instructions in the binary language of a processor
- Programs are written in a higher-level language
- The human-readable text of a program is called its **source code**
- A **compiler** converts source code to binary code
- This process is irreversible

Foo.cpp

```
#include <iostream.h>

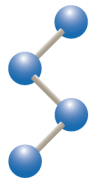
void main()
{
    double f = 42.0;
    double x;

    x = sqrt(f);
}
```

COMPILER

Foo.o

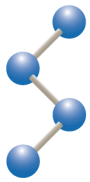
```
7F45 4C46 0102 0100
0004 0008 0015 0016
0023 0042 0000 0001
0001 0BE4 0F42 0034
0000 2030 0000 0000
0014 1D11 2F75 7372
2F6C 6992 2F54 642E
4E41 4D49 6D6F 6465
5F63 7869 7400 7F66
6D61 714D 714D 714D
```



The GNU Public License

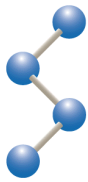
- Software can be delivered to the end user in binary form only (**closed source**)
- Without access to the source code, the user cannot modify the software
- The GNU Public License is an **open source** license agreement
 - Binaries can be given freely or sold, but the source code must be made freely available
 - Modifications to the the source code must also be released under the GPL





From Unix to Linux

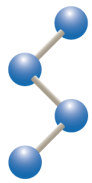
- The Unix operating system was developed by Bell Labs in the '70s
- The GNU (Gnu is Not Unix) project began in 1984 to implement a free Unix-like operating system
- In 1991, Linus Torvalds began to write a free Unix-like kernel called Linux, and released it under the GPL
- Combined with the GNU Project software, a complete operating system was built: **GNU/Linux**
- Thousands of developers throughout the world have contributed to GNU/Linux, enhancing the kernel and writing applications



Linux Distributions

- ▶ Although the source code is free, building a complete system from the source is difficult
- ▶ Linux vendors may add value by adding installers and commercial (closed-source) applications
- ▶ High compatibility across distributions that use the same kernel (current version: 2.6)



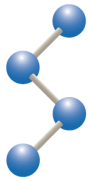


Shells & Windows

- Two most common human/computer interfaces:
 - GUI: Graphical User Interface
 - CLI: Command Line Interface

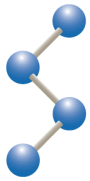


```
Terminal <@cb-demo01>
File Edit View Terminal Tabs Help
cryo 5: d
admin/      drivers/   image2000/ practicals/ tests/
bin/        em2em/    _imagic.dff spider/     tmp/
ccp4-5.0.2/ _em2em.dff jweb/      spire-1.4.0/ xtal/
chimera/    EMAN/     laser/     tcltk/     xtal_info/
Desktop/    EMAN2/    lib/       testimage2000/ XtalView/
cryo 6: d
```



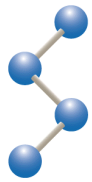
X Window System

- Most users are familiar with the Windows or Mac GUI desktops
- The GUI desktop used by most Unix systems is the X Window System:
 - X server: application that manages the display and input devices (keyboard, mouse, etc.)
 - X libraries: program pieces that talk to the X server
 - X applications: complete programs built using the X libraries
 - X window manager: an application that manages multiple X windows
 - Desktop suite: a collection of libraries, applications, and a window manager, creating a complete desktop environment

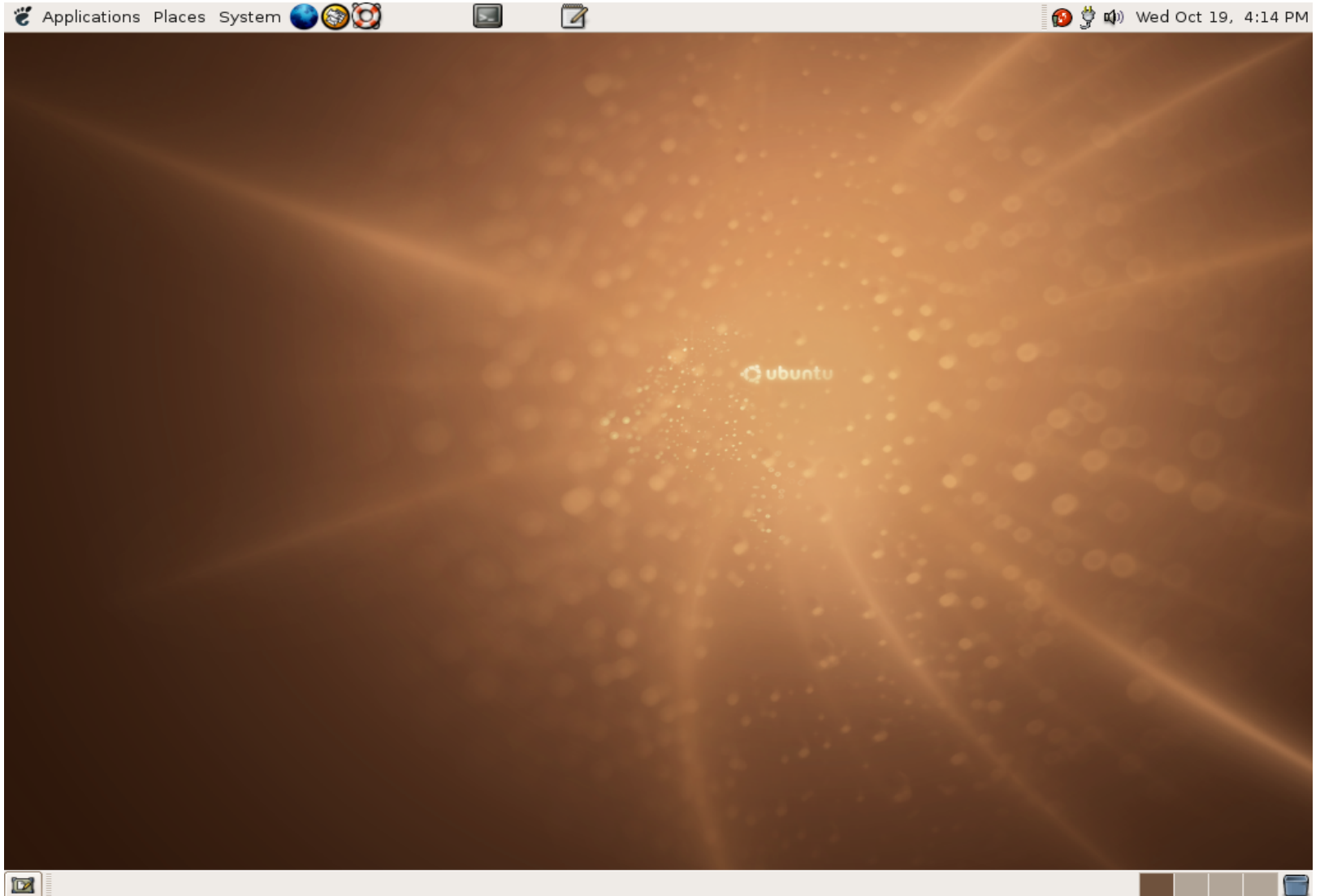


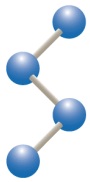
Logging In and Out

- Linux is a multi-user operating system
 - Your username is **student**
 - Your password is **cryo**
- If the X server is running, you'll get a GUI screen where you can log in
- If not, you'll get a textual `login:` prompt
 - After logging in, type **startx** to start X
- Use the “Log Out” entry on the “System” menu to log out or shutdown the machine
- CTRL-ALT-BACKSPACE will restart the X server if it gets stuck
 - This will kill all X Windows apps



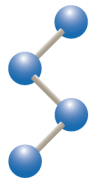
Gnome Desktop





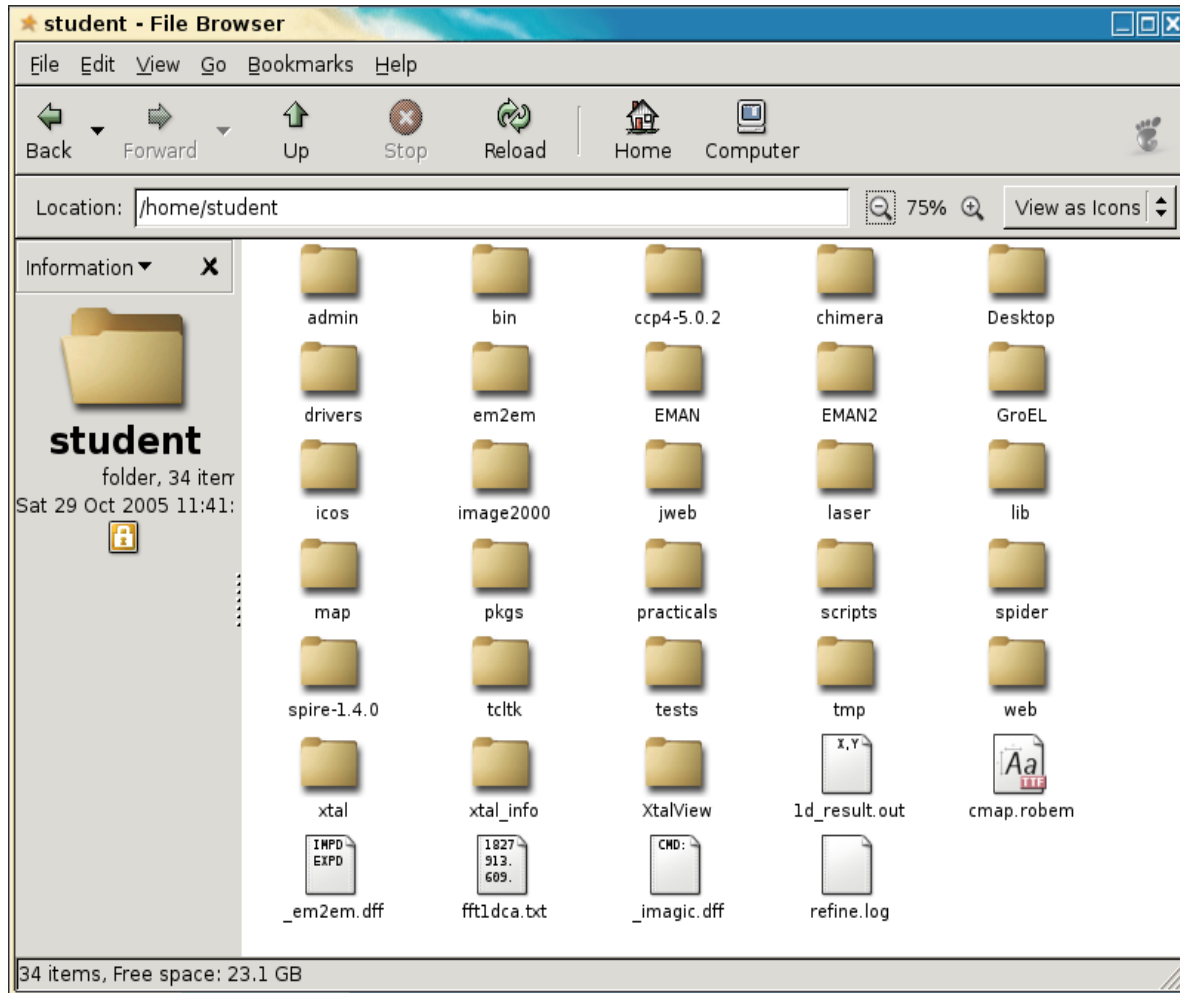
Using the Gnome Desktop

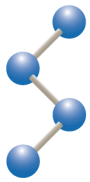
- Mac users will need to acquaint themselves with the right mouse button, sometimes used to bring up a menu
- Most operations performed with left mouse button
- Menubar at top
 - Program menus at left
 - Icons for frequently used programs in middle
 - System information at right
- Task bar at bottom
 - Button at left hides all open windows
 - Middle populated with a button for each open application
 - Squares at right switch between 4 desktops



File Browser

➤ The **Places** menu provides access to the File Browser

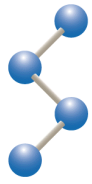




Shells

- The Command-Line Interface (CLI) predated the GUI
- Under Linux, the CLI is accessed via a shell program
 - Bourne Shell (`sh` or `bash`)
 - The C Shell (`csh` or `tcsh`)
- From the Gnome desktop, the **Terminal** program creates a new GUI window containing a CLI shell
 - You may have multiple terminals open
 - We'll use the C Shell for this class



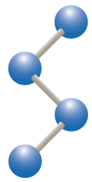


The Prompt

➤ The shell begins with a prompt consisting of:

- The name of the machine
- History number (1, 2, 3 ...)
- A colon

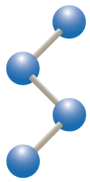
```
Terminal
File Edit View Terminal Tabs Help
cryo 1: █
```

A Command

- ▶ The user types a command, followed by **ENTER**
- ▶ The output appears below the command, followed by a new prompt

```
Terminal
File Edit View Terminal Tabs Help
cryo 1: d
admin/      em2em/      _imagic.dff  refine.log   web/
bin/        _em2em.dff  jweb/        scripts/     xtal/
ccp4-5.0.2/ EMAN/        laser/        spider/      xtal_info/
chimera/    EMAN2/       lib/          spire-1.4.0/ XtalView/
cmap.robem  GroEL/       map/          tcltk/
Desktop/    icos/        pkgs/         tests/
drivers/    image2000/   practicals/   tmp/
cryo 2:
```



Structure of a Command

```
cryo 1 : man -k color
```

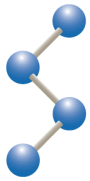
prompt

program

arguments

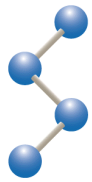
options

- Arguments are often file names
 - Each program handles arguments in its own way
- Options control the execution of the program and often begin with **-** or **--**
- Many commands accept **-h** or **--help** and display usage info



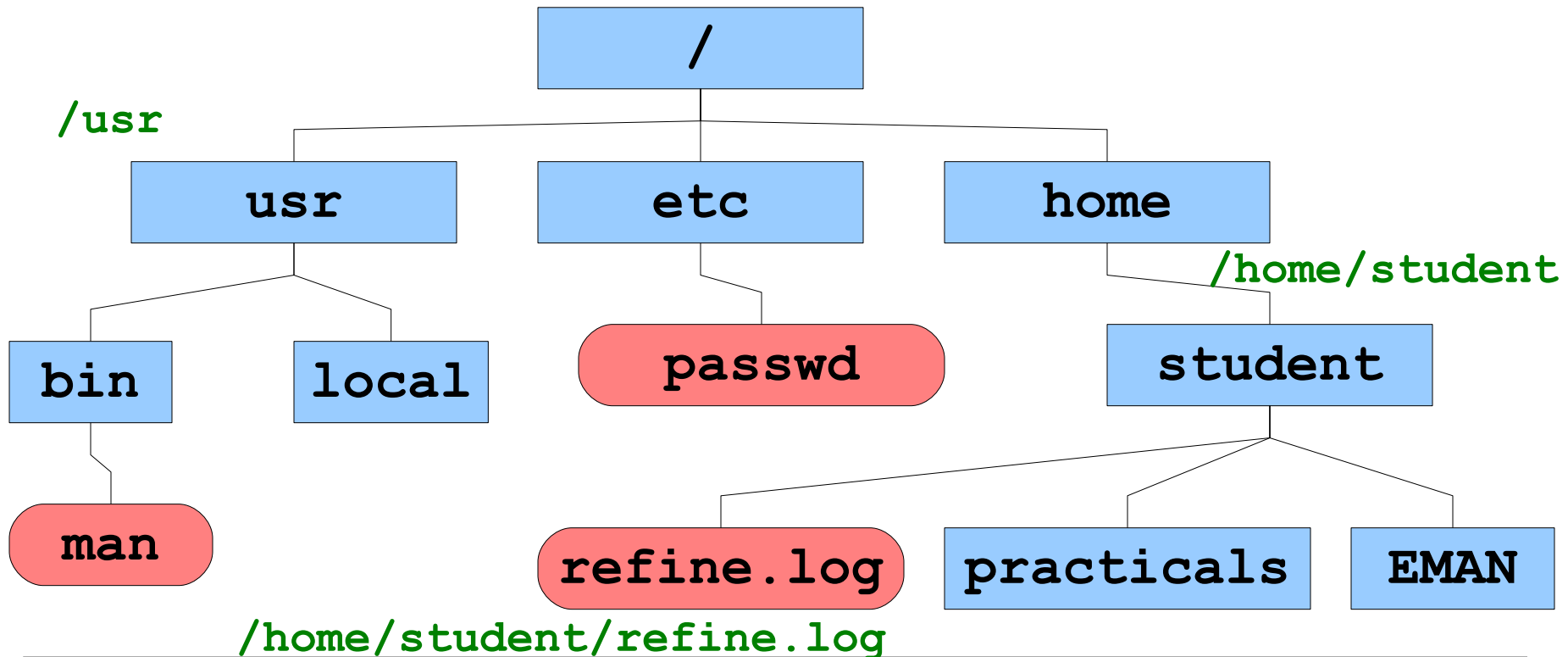
Command Editing & History

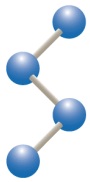
- ↑ and ↓ scroll through the history of commands
- ← and →, **HOME** and **END** for in-line editing
- **Control-U** erases entire line
 - The command is not issued until you hit **ENTER**
- History substitutions:
 - **!!** previous command
 - **!*** arguments from previous command
 - **!\$** last argument from previous command
 - **!4** command #4 from history
 - **!m** last command that began with 'm'
 - **^gif^png** replace “gif” with “png” in previous command
 - **history** prints list of commands executed



The Linux File System

- ▶ Linux has a hierarchical file system
- ▶ Folders are called **directories**
- ▶ The top of the hierarchy is **/**

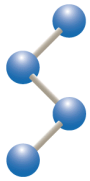




Current Working Directory

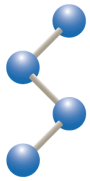
- The shell keeps track of your current location in the file system hierarchy
- Print your current working directory with **pwd**
- Change your current working directory with **cd**

```
cryo 1: pwd
/home/student
cryo 2: cd /usr
cryo 3: pwd
/usr
cryo 4: cd
cryo 5: pwd
/home/student
```



Pathnames

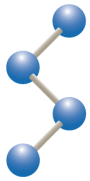
- A **pathname** is text specifying the location of a file or directory
- An **absolute** pathname begins with a `/` and uniquely locates a file or directory
 - `/home/student/refine.log`
- A **relative** pathname specifies a location relative to the current working directory
 - `refine.log`
 - refers to the same file if `cwd` is `/home/student`
- Spaces aren't allowed, use `\` or `"`
 - `/home/student/My\ Files/Paper\ Abstract.doc`
 - `/home/student/"My Files"/"Paper Abstract.doc"`



.. . and ~

- .. refers to the directory above, . refers to the *cwd*
- ~ refers to your home directory
- ~**user** refers to *user's* home directory

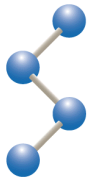
```
cryo 1: pwd
/home/student
cryo 2: cd ..
cryo 3: pwd
/home
cryo 4: cd ~
cryo 5: pwd
/home/student
```



Environment Variables

- ▶ The shell keeps track of environment variables available to all programs
- ▶ Environment variables are set with **setenv** and displayed with **printenv**

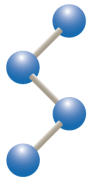
```
cryo 1: setenv LOGFILE ~/refine.log  
cryo 2: printenv LOGFILE  
/home/student/refine.log  
cryo 3:
```

Execution PATH

- The **PATH** is a list of directories to search for programs
- The program name is the name of a file somewhere on the path
 - Use an absolute pathname to specify a program exactly

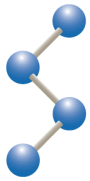
```
cryo 1: printenv PATH
/home/student/XtalView/bin/ibmpcLinux2:/home
/student/EMAN/bin:/home/student/EMAN2/bin:/u
sr/bin:/home/student/chimera/bin:/home/stude
nt/icos/allprograms:/home/student/em2em/bin:
/home/student/web/bin:/home/student/spider/b
in:./:/home/student/bin:/usr/local/bin:/usr/l
ocal/sbin:/bin:/sbin:/usr/sbin:/usr/X11R6/bi
n:/home/student/image2000/bin2000/local/stud
```



Built-Ins, Aliases, and **which**

- Some commands are built into the C shell
- Command shortcuts can be made with **alias**
- Identify a command with **which**

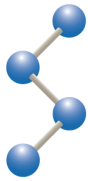
```
cryo 1: which cd
cd: shell built-in command.
cryo 2: which d
d:          aliased to ls --color -FC !*
cryo 3: which spider
/home/student/spider/bin/spider
cryo 4:
```



ls

ls lists the files in the current directory

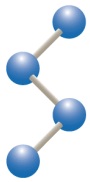
```
cryo 1: ls
1d_result.out  fft1dca.txt  scripts
admin          GroEL        spider
bin           icos         spire-1.4.0
ccp4-5.0.2    image2000    tcltk
chimera       _imagic.dff  tests
cmap.robem   jweb         tmp
Desktop      laser        web
drivers      lib          xtal
em2em        map          xtal_info
_em2em.dff   pkgs        XtalView
EMAN         practicals
EMAN2       refine.log
```



ls -F and ls -a

- ▶ **ls -F** adds **/** to directories and ***** to programs
- ▶ Filenames beginning with **.** are hidden, use **ls -a**

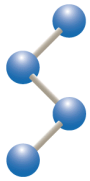
```
cryo 1: cd ~/tests/filesys
cryo 2: ls
directory1  file1  program
directory2  file2
cryo 3: ls -F
directory1/  file1  program*
directory2/  file2
cryo 4: ls -aF
./  directory1/  file1  .hidden
../  directory2/  file2  program*
```



d aliased to **ls --color -FC**

- **d** is a handy alias for listing directories
- Created in the hidden file `~/ .cshrc`, read by the C shell when it starts
 - Your PATH is also set in `~/ .cshrc`

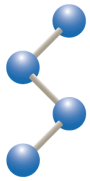
```
cryo 1: cd ~/tests/filesys
cryo 2: d
directory1/  file1  program*
directory2/  file2
cryo 3: d -a
./          directory1/  file1  .hidden
../         directory2/  file2  program*
```



ls -l

▶ **ls -l** shows all the details of file ownership

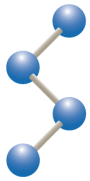
```
cryo 1: cd ~/tests/filesys
cryo 2: ls -l
total 24
drwxr-xr-x  2 student student 4096 2005-10-26
15:03 directory1
drwxr-xr-x  2 student student 4096 2005-10-26
15:03 directory2
-rw-r--r--  1 student student 3068 2005-10-26
15:32 file1
-rw-----  1 student student 6136 2005-10-26
15:33 file2
-rwxr-xr-x  1 student student   15 2005-10-26
15:33 program
```



ls -l Output

permissions	links	owner	group	size	
<code>-rw-r--r--</code>	<code>1</code>	<code>student</code>	<code>student</code>	<code>3068</code>	<code>2005-10-26</code>
<code>15:32</code>					
					<code>file1</code>

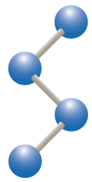
time modified file name date modified



Groups

- Each user belongs to one or more groups
- List the groups you belong to with **groups**
- Change the group a file belongs to with **chgrp**

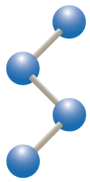
```
cryo 1: groups
student adm dialout cdrom floppy audio dip
video plugdev lpadmin scanner admin
cryo 2: chgrp admin file1
cryo 3: ls -l file1
-rw-r--r--  1 student admin 3068 2005-10-26
 15:32 file1
cryo 4:
```

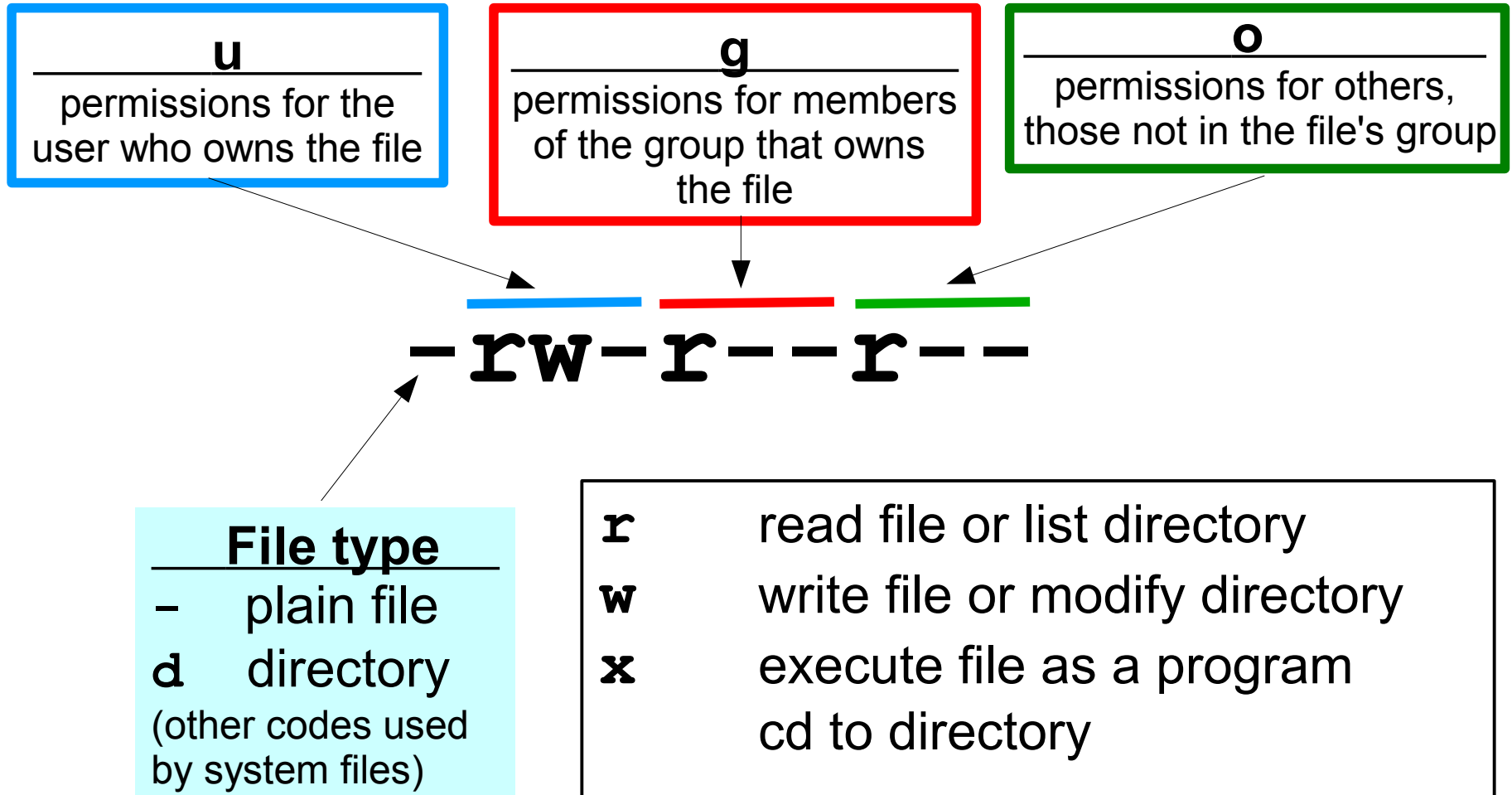
Users and the Superuser

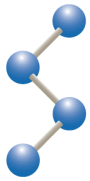
- One user is the **superuser**, or administrator
- The superuser logs in with the name **root**
- root can read or write any file, and can change the ownership of a file with **chown**

```
cryo 1: su root
Password: cryo2005
cryo 1# chown root file1
cryo 2# exit
cryo 2: ls -l file1
-rw-r--r--  1 root admin 3068 2005-10-26
 15:32 file1
cryo 3:
```



Permissions





chmod

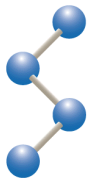
➤ **chmod** changes the permissions (mode) of a file

chmod *<mode>* **files...**

➤ *<mode>* is

ugoa	+	rwx
	-	

```
cryo 1: ls -l file1
-rw----- ... file1
cryo 2: chmod a+rwx file1
cryo 3: ls -l !$
-rwxrwxrwx ... file1
cryo 4: chmod o-w file1 ; !ls
-rwxrwxr-x ... file1
cryo 5:
```



Editing Files

- There are two popular CLI text editors: **vi** and **emacs**
 - Both are difficult to learn
- Two X Windows editors are available: **nedit** and **gedit**

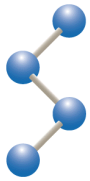
```
*.cshrc (/home/student) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
*.cshrc x
# tcsh options

set prompt = "%Ucryo%u %h: "

set autolist
set color
set colorcat
set correct = all

alias a alias
a - "cd ..;"
a = "cd \!:1"
a cdd "cd \!*; d"
a d "ls --color -FC \!*"
a dir d
a g grep
a m more
a mkx "chmod a+x \!*"
a mkw "chmod a+w \!*"
a pd pushd
a pop popd
a psg "ps auxww | grep \!*"
a rc "vi ~/.cshrc; source ~/.cshrc"
a recurse "find . -exec \!* {} \;"

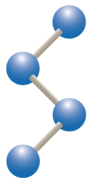
Ln 8, Col 18 INS
```



Copying, Moving, Deleting Files

- ▶ To copy a file: **cp** *<source>* *<destination>*
- ▶ To move a file: **mv** *<oldname>* *<newname>*
- ▶ To delete a file: **rm** *<files...>*

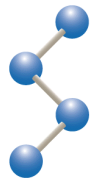
```
cryo 1: ls
file1  file2
cryo 2: cp file2 file3 ; ls
file1  file2  file3
cryo 3: mv file1 file99 ; ls
file2  file3  file99
cryo 4: rm file3 ; ls
file2  file99
cryo 5:
```



Directories

- To make a new directory: **mkdir** *<dir>*
- To delete an empty directory: **rmdir** *<dir>*
- To delete a non-empty directory: **rm -rf** *<dir>*
- **cp** and **mv** accept a directory as last argument

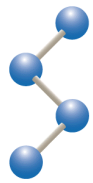
```
cryo 1: mkdir dir1
cryo 2: ls -F
dir1/  file1  file2
cryo 3: mv file1 dir1 ; ls -F
dir1/  file2
cryo 4: rmdir dir1
rmdir: `dir1': Directory not empty
cryo 5: rm -rf dir1 ; ls
file2
```



Specifying Filenames via Wildcards

- ▶ **?** matches a single character, ***** matches zero or more
- ▶ **[]** specifies a range of values for a single character

```
cryo 1: ls
a1  a2  b1  b2  bb1 bb2  c1  c2  d1  d2
cryo 2: ls *1
a1  b1  bb1  c1  d1
cryo 3: ls ?1
a1  b1  c1  d1
cryo 4: ls [a-c]*
a1  a2  b1  b2  bb1 bb2  c1  c2
cryo 5: rm *
cryo 6: ls
cryo 7:
```



Command Completion and Correction

- **TAB** causes the shell to expand filenames, presenting all the files that match what you've typed so far
 - Works for program names too
- The shell will also prompt you if it thinks a command is misspelled

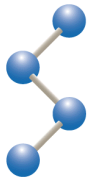
```
cryo 1: ls -l bbTAB
```

```
bb1  bb2
```

```
cryo 1: ls bb|
```

```
cryo 2: eman
```

```
CORRECT>eman (y|n|e|a)?
```

cat and more

- **cat** takes input from the keyboard and outputs it to the screen, or outputs a file to the screen
- **more** outputs to the screen one page at a time

```
cryo 1: cat
```

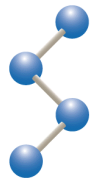
```
Hi There!
```

```
Hi There!
```

```
^D
```

```
cryo 2: more longfile.txt
```

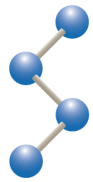
```
This is a long file that would just scroll  
off the screen if I used "cat", but because I  
used "more" it pauses at the end of each  
screenful and prompts me to press space to  
--More-- (42%)
```



Redirecting Input and Output

- ▶ Normal input is the keyboard, output is the screen
- ▶ Use **<** to read input from a file, **>** to send output to a file, **|** to send the output to another program

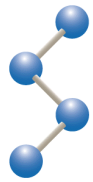
```
cryo 1: cat < onelinefile
This is the contents of onelinefile.
cryo 2: ls > outputfile
cryo 3: more outputfile
onelinefile
outputfile
cryo 4: ls -FC ~/dirWithLotsOfFiles | more
dir1/    dir2/    dir3/    file1    file2
file3    file4    file5    prog1*   prog2*
--More--
```



Job Control

- ▶ Use **&** to run a program in the background
- ▶ **jobs** lists the background tasks
- ▶ **tail -f** outputs the last lines of a file as its written

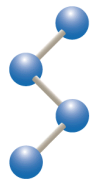
```
cryo 1: computeMeaningOfLife > answer &  
[1] 4815  
cryo 2: jobs  
[1] + Running computeMeaningOfLife  
cryo 3: tail -f answer  
00:00 Starting computation of Meaning of Life  
00:10 Hmm... this may take a while  
00:20 My, this is a tricky problem, isn't it?  
00:30 Still thinking ...  
00:40 Still thinking ...
```



More Job Control

- ▶ Use **^C** to cancel your foreground job, **^Z** to suspend it
- ▶ **bg** runs the suspended job in the background
- ▶ **kill** cancels the job, **fg** makes it the foreground job

```
cryo 1: computeMeaningOfLife
00:00 Starting computation of Meaning of Life
00:10 Hmm... this may take a while
^Z
[1] + Suspended          computeMeaningOfLife
cryo 2: bg %1
[1] computeMeaningOfLife &
cryo 3: kill %1
[1] Killed              computeMeaningOfLife
cryo 4:
```

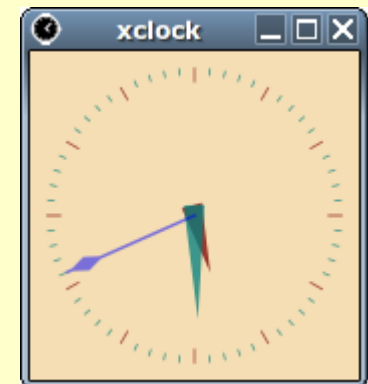


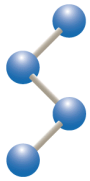
Remote Access

- Use **ssh** to open a shell on another computer
 - Communication is encrypted, your password is safe
- **-X** option causes X Windows apps run on remote computer to appear on your local desktop

```
cryo 1: ssh -X -l mercurio tippy.scripps.edu  
Password:  
Welcome to tippy.scripps.edu
```

```
tippy 1: xclock  
^C  
tippy 2: exit  
cryo 2:
```

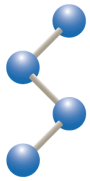




du and df

- **du** reports the amount of disk space used by a directory
 - Normally shows all subdirectories, use **-s** for summary
- **df** reports the amount of free space on a disk
- Use **-h** for human-readable output (**ls** too)

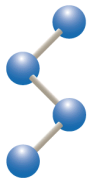
```
cryo 1: du -s ~
4815162 /home/student
cryo 2: du -sh ~
4.8G    /home/student
cryo 3: df -h ~
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda2       27G   13G   15G   47% /
```



file and Scripts

- **file** <files...> looks at the contents of a file and tries to identify it
- A script is a text file containing a list of commands
 - Giving a text file **x** (execute) permissions makes it a script
 - The first line of the file specifies the language

```
cryo 1: file ~/chimera/*
al2co:      ELF 32-bit LSB executable, Intel
80386, version 1 (SYSV), for GNU/Linux 2.2.5,
dynamically linked (uses shared libs), not
stripped
chimera:    Bourne shell script text
happydoc:  a python script text
itops:     Bourne shell script text
```



man and info

- **man** displays the manual pages for a command
 - Often terse, but explains all options and error messages
- **info** is an interactive program for reading more detailed documentation

```
cryo 1: man ls
LS(1)          User Commands          LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILES
--More--
```