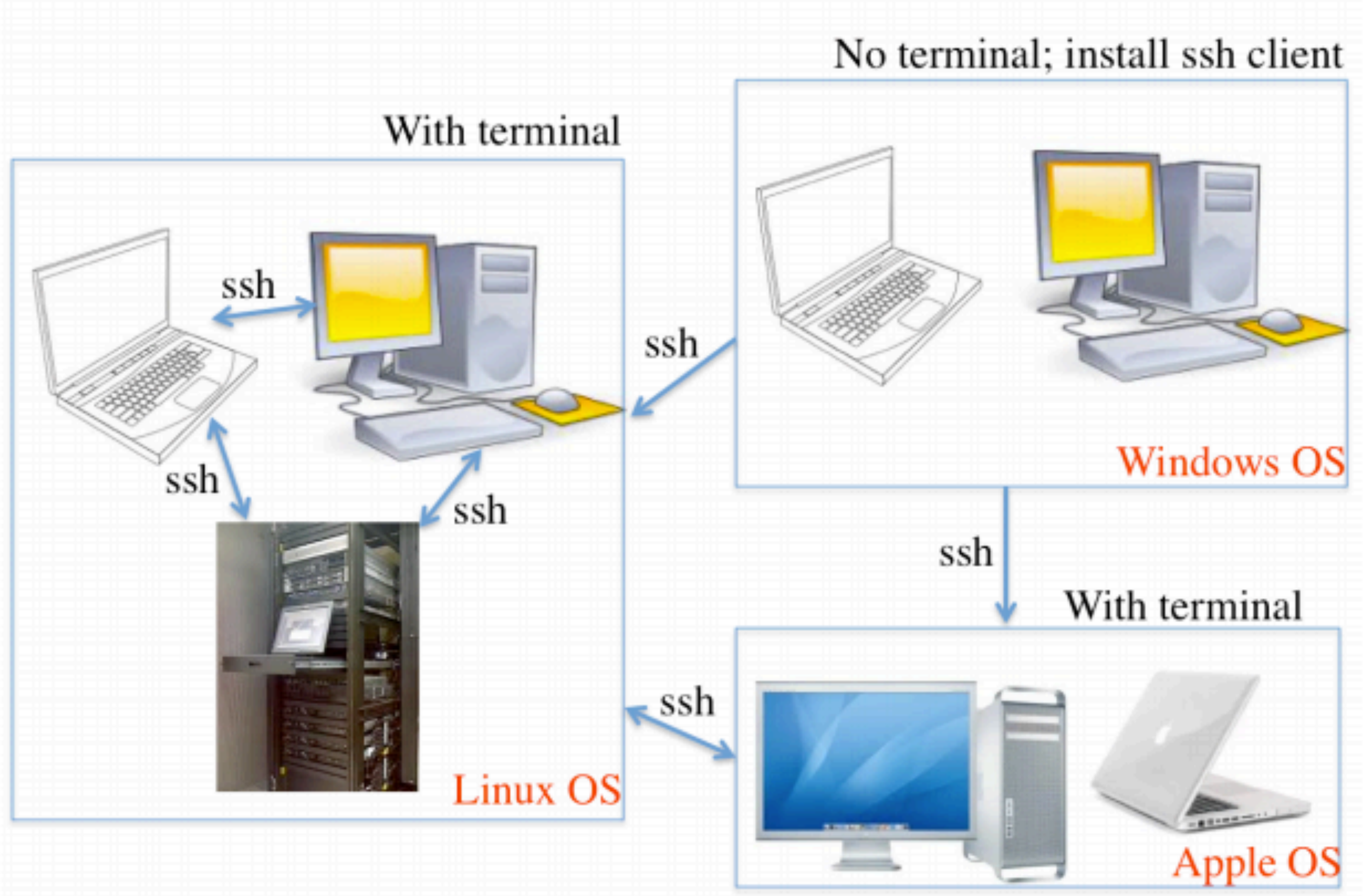
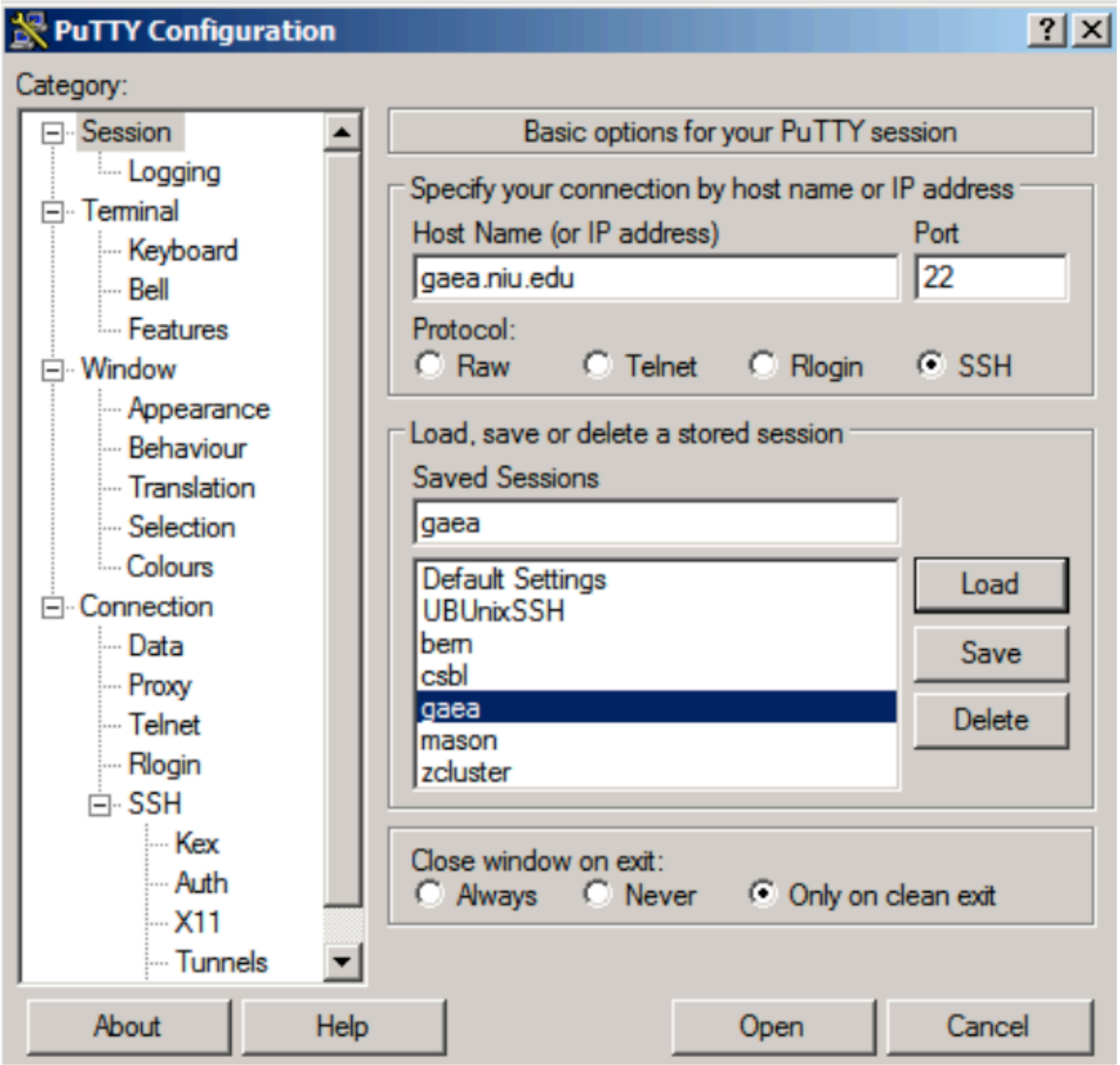
Linux command line basics

Yanbin Yin

Spring 2013

Figure 1.5 Different ways to connect to Unix terminal environment





Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): gaea.niu.edu

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions: gaea

Default Settings
UBUnixSSH
bem
csbl
gaea
mason
zcluster

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 1.1 Find the terminal application in Ubuntu

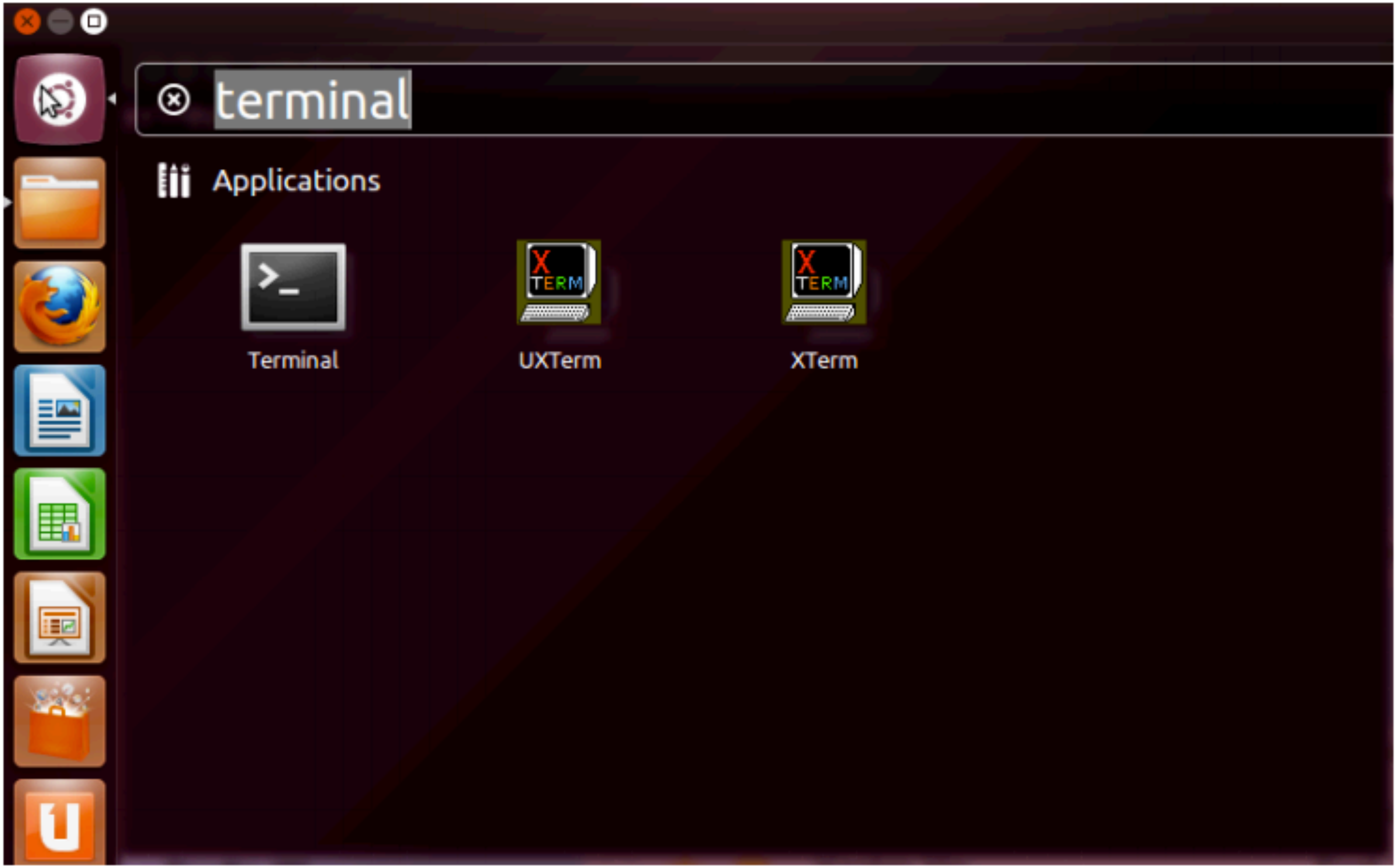
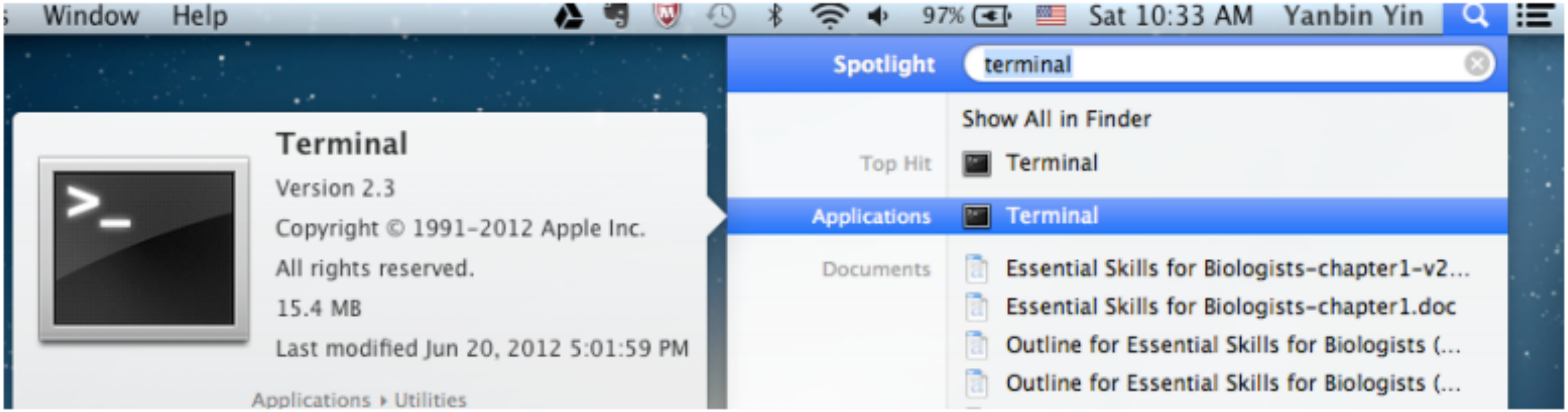


Figure 1.2 Find the terminal application in MAC OS



Terminal

Version 2.3

Copyright © 1991-2012 Apple Inc.

All rights reserved.

15.4 MB

Last modified Jun 20, 2012 5:01:59 PM

Applications ▸ Utilities

Terminal

Terminal

Essential Skills for Biologists-chapter1-v2...

Essential Skills for Biologists-chapter1.doc

Outline for Essential Skills for Biologists (...

Outline for Essential Skills for Biologists (...

Terminal window tricks

(may not work on all VNC or ssh clients…)

- ❑ Helpful tricks to avoid excessive command typing
  - ➢ Use **copy/paste**. Any text "mouse-selected" while holding the left mouse button is copied to clipboard. It may then be pasted, e.g., into a command, by clicking the **right** mouse button (PuTTy) or the **middle** button (when working through the console in 625 Rhodes).
  - ➢ Use **Up/Down arrow keys** – this will cycle through recently executed commands.
  - ➢ Use the **TAB key** – this will often present you with a list of choices after typing a part of a command – no need to remember everything.
  - ➢ **history** command: list all recently used commands – can copy a desired command and paste it to execute again, or refer to a command by its index

Examples:

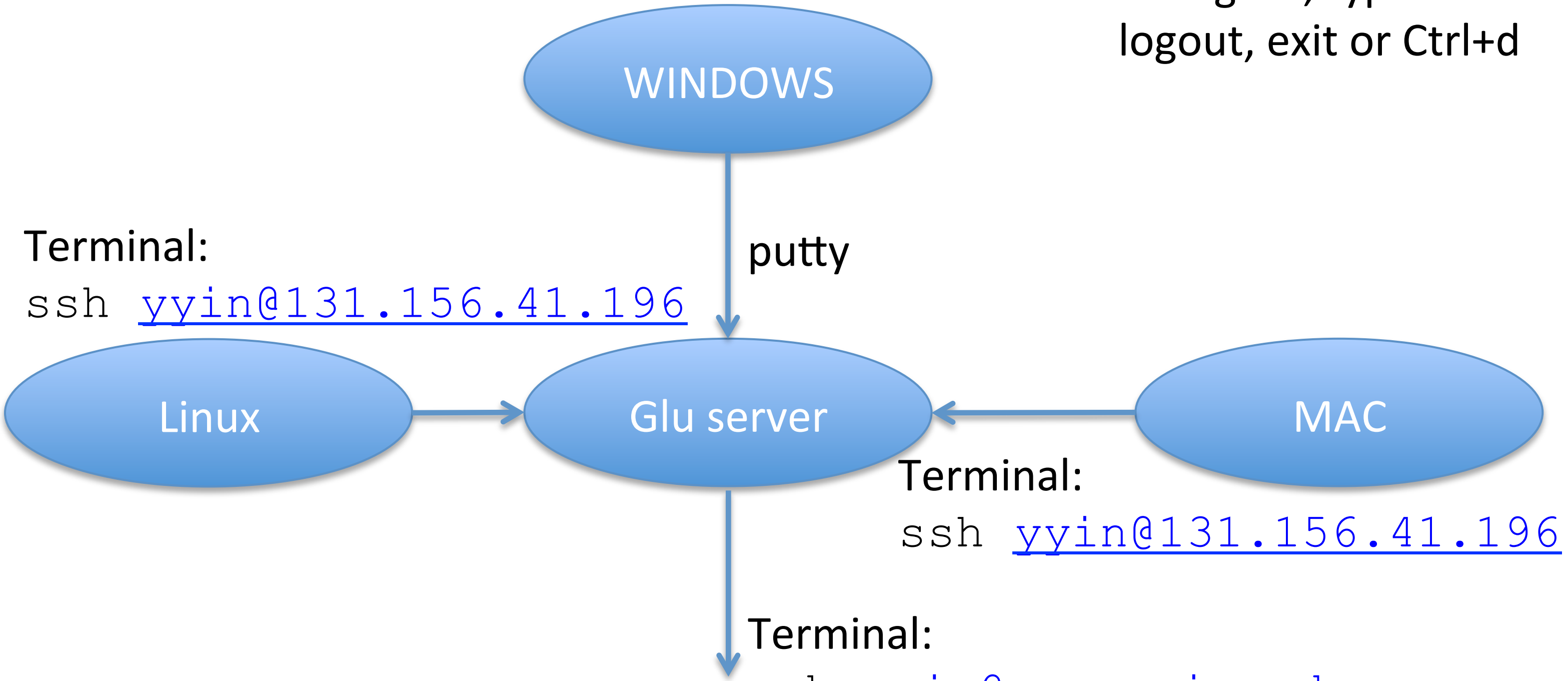
history (*list all remembered commands)*

history | more *(list all remembered commands page by page)*

history | grep workdir *(list all remembered commands containing string "workdir")*

Log in

To logout, type
logout, exit or Ctrl+d



Terminal:

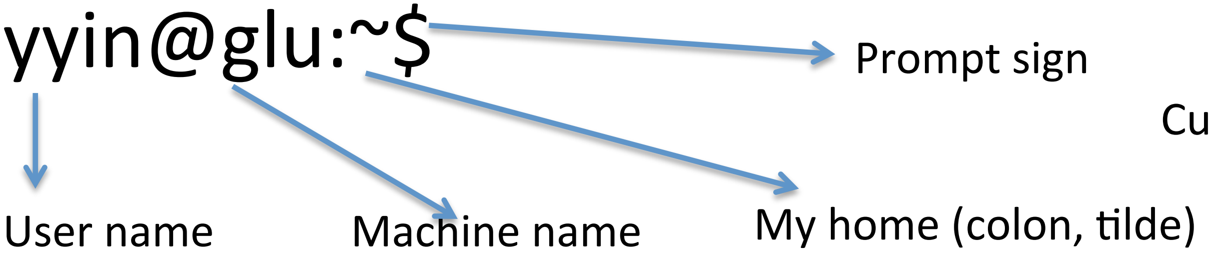
```
ssh yyin@131.156.41.196
```

Terminal:

```
ssh yyin@131.156.41.196
```

Terminal:

```
ssh yyin@gaea.niu.edu
ssh yinyb@zcluster.rcc.uga.edu
…
```



```
yyin@glu:~$
```

Prompt sign

User name

Machine name

My home (colon, tilde)

Cursor sits right after prompt

Interactive command line terminal

In the graphical interface of Windows, Linux and MAC, you click your mouse to make something happen.

However, under command line terminal (or console) interface, you always type in a command using the keyboard and hit Enter to let something happen.

Ctrl+c
Ctrl+z

The command you typed in is interpreted and executed by the Shell, the Linux software and interface to connect the users and the OS.

The output of the command is printed on the screen by default or written to a file. (together with error msg if any)

Commands

Commands can be a single word/letter or a few words separated by space; always hit Enter after you finished typing

The first word/letter is often a Shell command or an external program or your own script name

Other words can be command options, files, folders etc.

Examples:

```
yyin@glu:~$ pwd
/home/yyin
```

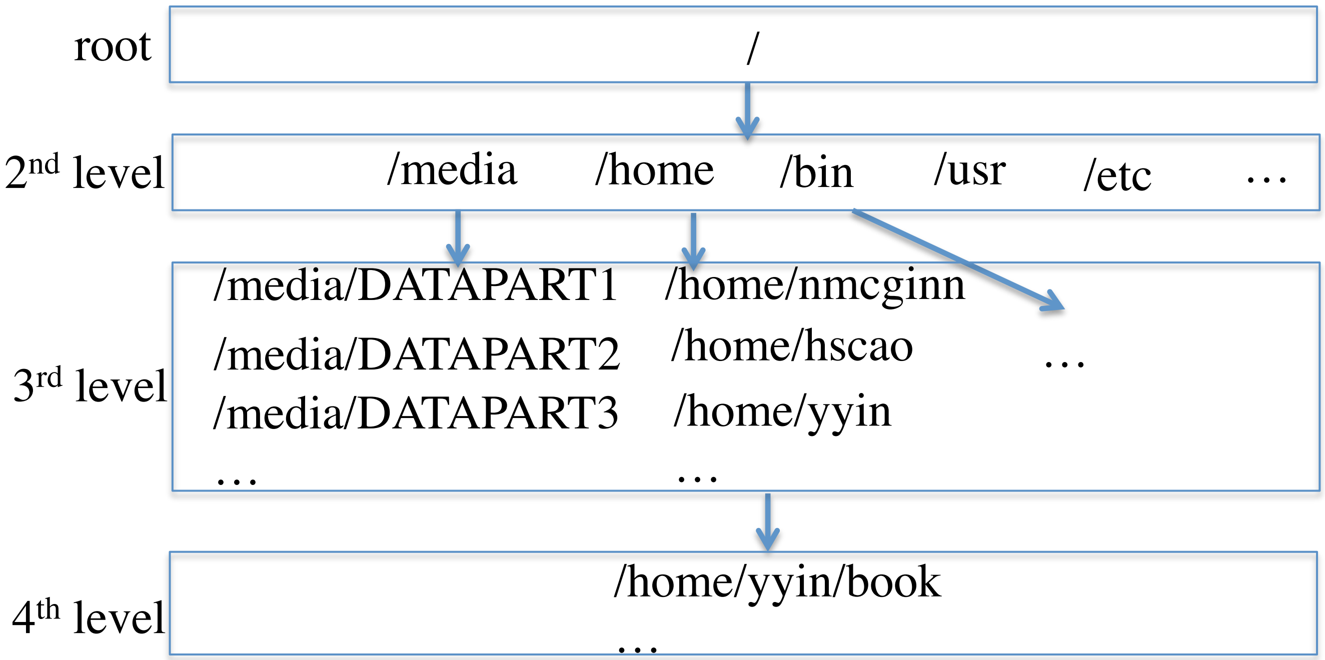
```
yyin@glu:~$ ls /media/DATAPART4/z1003529/
aspergillus  mkdir.txt
```

Everything is a file or a folder/directory

A folder contains files or/and other folders (subdirectories)

A subdirectory can contain other files or folders

So the file system is a tree-like structure (multi-furcation)



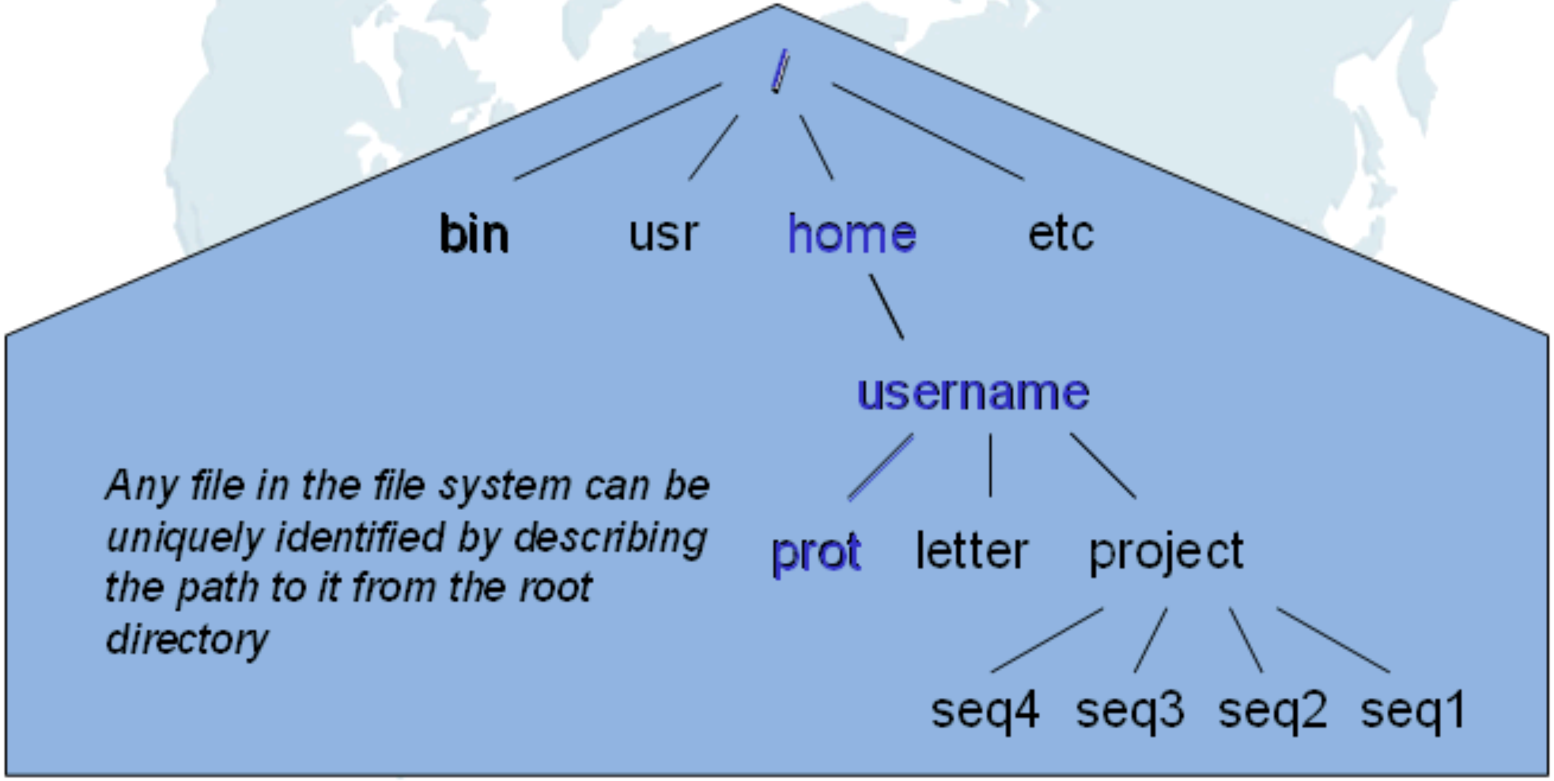
root
/

2nd level
/media /home /bin /usr /etc …

3rd level
/media/DATAPART1
/media/DATAPART2
/media/DATAPART3
…
/home/nmcginn
/home/hscao
/home/yyin
…
…

4th level
/home/yyin/book
…

Organisation of the file system



Any file in the file system can be uniquely identified by describing the path to it from the root directory

```
/home/username/prot
```

http://edu.isb-sib.ch/mod/resource/view.php?id=306

File/folder/program names

Case sensitive!!!

`File and file are two different files`

Allow to contain letters, numbers, underscore (_), dot (.), dash (-), plus (+) but not to use other special characters and spaces

Example:

```
[yyin@gaea fungal]$ ls
all.hmmlib_1.75.fungal-dockerin.domain.fa.nobac.n2+.fa.id.source.nrgi.ps.col.lab
```

For programs, better use the correct file extension (.sh, .pl, .c, .py), e.g. `run-blast.sh`

Home folder (dir)

After you login a remote Linux machine or you opened a terminal in your Ubuntu or MAC computer, you are at your home

You can create or download folders, files, bioinformatics softwares

Make it organized and clean by creating folders that have meaningful names, such as tools, data, work, project, scripts etc.

You can also create subfolders, e.g. blast under tools, or project1 under project

Don’t put everything in your home directory; you will easily accumulate too many files/folders/programs/scripts that you will have a headache to find or remember things

Directory commands

pwd	find out where you are (your current directory or working directory)
cd	change directory

Remember?

Don’ need to type everything in
Always use TAB key to auto-complete a word

Relative path and absolute path

Find out where you are:

```
yyin@glu:~$ pwd
/home/yyin
```

Absolute path or full path

If I want to change directory to /home/

```
yyin@glu:~$ cd ..
yyin@glu:/home$
```

If I want to go back to my home

```
yyin@glu:/home$ cd yyin
yyin@glu:~$
```

Relative path, relative to your current dir: /home

I can also use the absolute path to go back

```
yyin@glu:/home$ cd /home/yyin
yyin@glu:~$
```

Special denotations:

tilde	~ or ~/	your home
	. or ./	your current directory
	.. or ../	the dir one level above
	../..	?

Try:

```
cd ~

cd ..
pwd

cd ../..
pwd
```

If you think you are lost in the file system and don’t know where you are, always run

```
pwd
```

From anywhere to go back to you home

```
cd
cd ~
```

More directory commands

mkdir	create a directory
rmdir	delete an empty directory (have no subdirectories or files)

At your home, try

```
mkdir bioinfo
```

Remember case sensitive and no special characters and space!!!

List command

ls list what files and directories are there in a folder

Try

These are options, used to enrich the functionalities of a command

```
ls
ls -l              list in long listing format
ls -la             list all including hidden files/folders
ls -lt             list according to modification time
ls -ltr            list according to time in reverse order
ls ..              list one level up
ls -l /home/yyin   list things under a given folder
Ls -l | less       if there are too many files to display in
                   one page, use pipe and less to show page
                   by page (will explain shortly)
```

The manual command

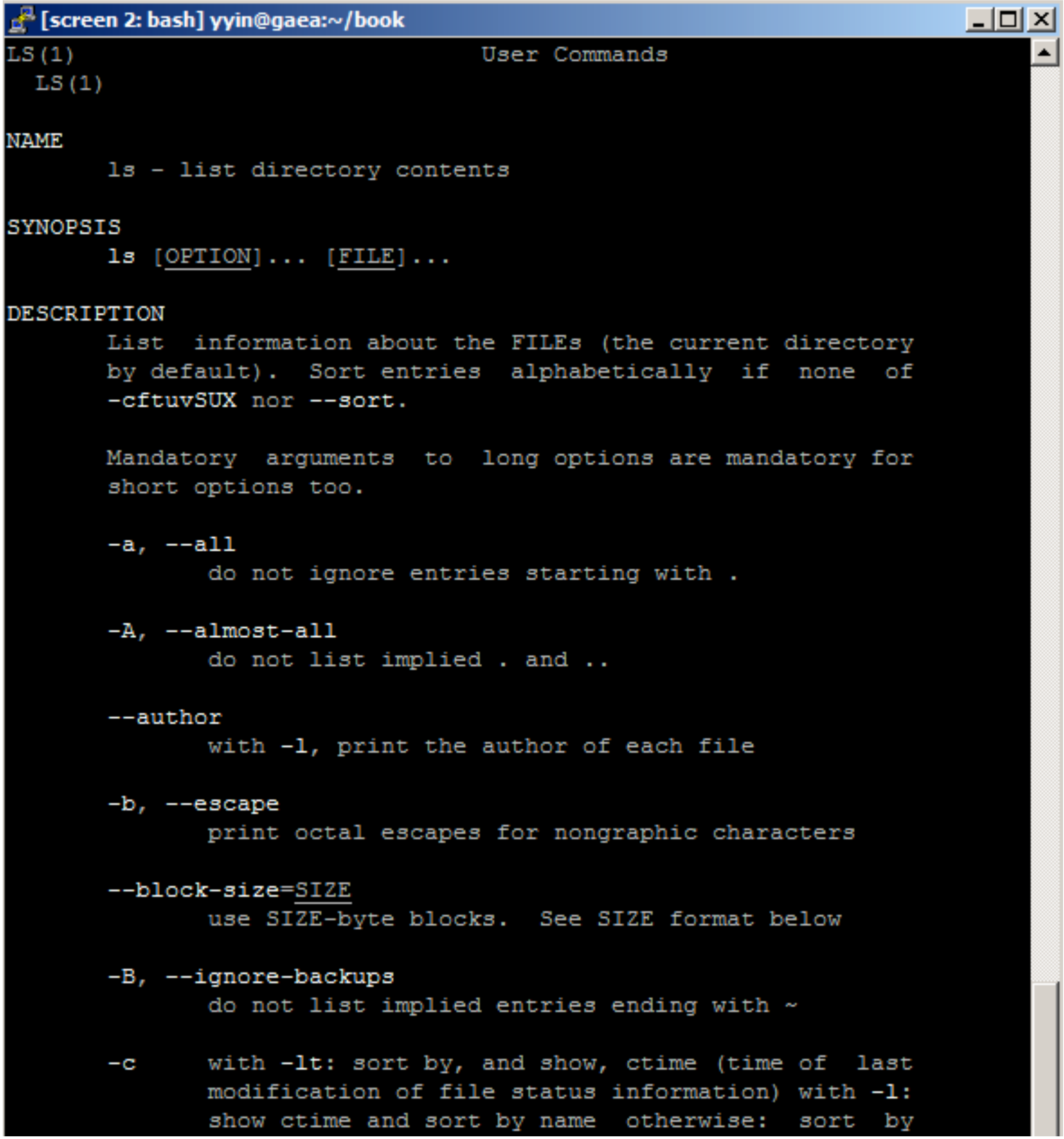
```
man ls
```

If the manual is more than one page, hit space or PgDn key to page down, PgUp key or u to page up, hit q to exit

Man followed by any Linux command to display the manual of that command

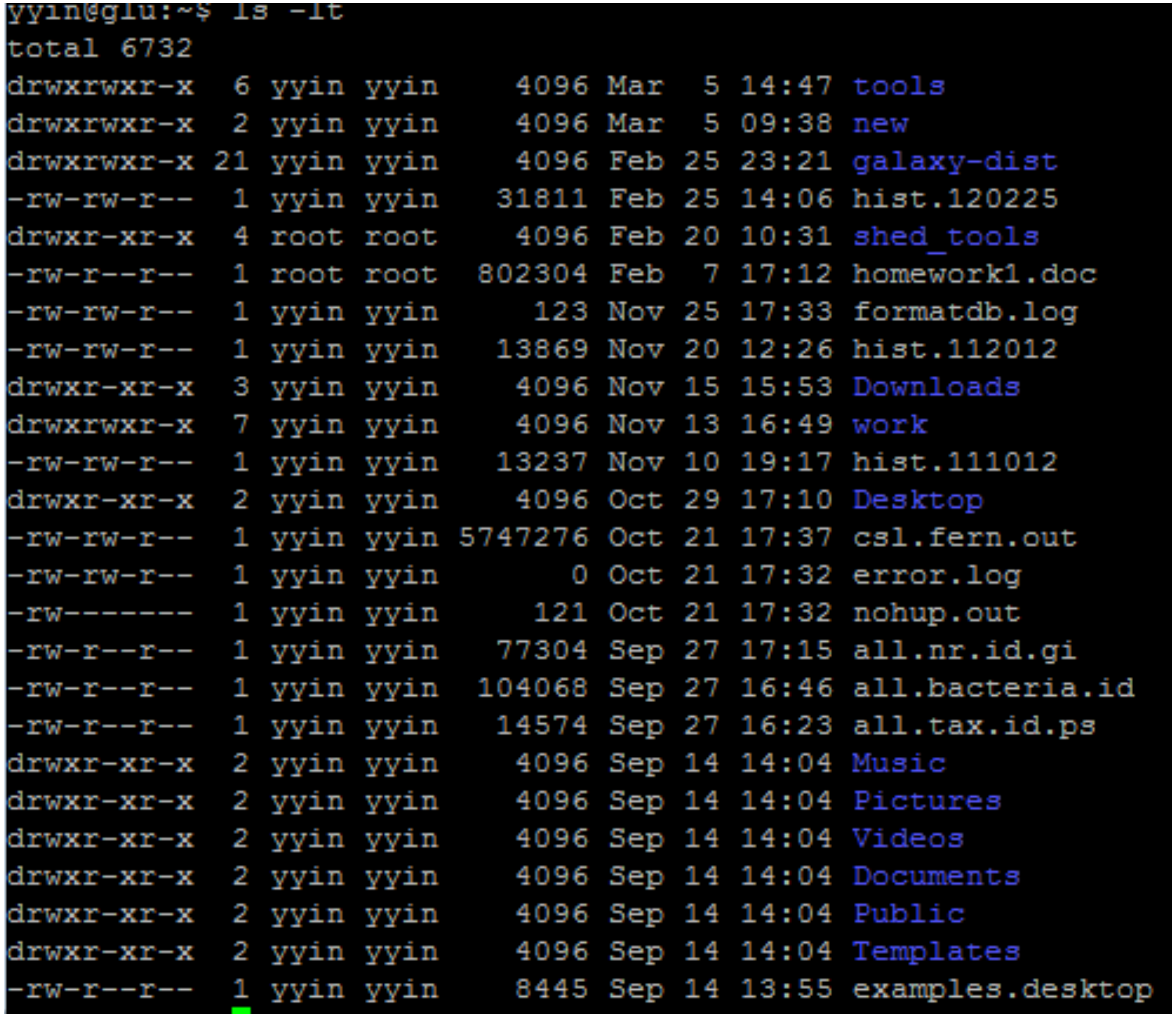
```
man pwd
man cd
```

…



```
LS(1)                            User Commands
  LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information about the FILEs (the current directory
       by default).  Sort entries alphabetically if  none  of
       -cftuvSUX nor --sort.

       Mandatory  arguments  to  long options are mandatory for
       short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print octal escapes for nongraphic characters

       --block-size=SIZE
              use SIZE-byte blocks.  See SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of  last
              modification of file status information) with -l:
              show ctime and sort by name  otherwise:  sort  by
```

```
ls -l
```



```
yyin@glu:~$ ls -lt
total 6732
drwxrwxr-x  6 yyin yyin    4096 Mar  5 14:47 tools
drwxrwxr-x  2 yyin yyin    4096 Mar  5 09:38 new
drwxrwxr-x 21 yyin yyin    4096 Feb 25 23:21 galaxy-dist
-rw-rw-r--  1 yyin yyin   31811 Feb 25 14:06 hist.120225
drwxr-xr-x  4 root root    4096 Feb 20 10:31 shed_tools
-rw-r--r--  1 root root  802304 Feb  7 17:12 homework1.doc
-rw-rw-r--  1 yyin yyin     123 Nov 25 17:33 formatdb.log
-rw-rw-r--  1 yyin yyin   13869 Nov 20 12:26 hist.112012
drwxr-xr-x  3 yyin yyin    4096 Nov 15 15:53 Downloads
drwxrwxr-x  7 yyin yyin    4096 Nov 13 16:49 work
-rw-rw-r--  1 yyin yyin   13237 Nov 10 19:17 hist.111012
drwxr-xr-x  2 yyin yyin    4096 Oct 29 17:10 Desktop
-rw-rw-r--  1 yyin yyin 5747276 Oct 21 17:37 csl.fern.out
-rw-rw-r--  1 yyin yyin       0 Oct 21 17:32 error.log
-rw-------  1 yyin yyin     121 Oct 21 17:32 nohup.out
-rw-r--r--  1 yyin yyin   77304 Sep 27 17:15 all.nr.id.gi
-rw-r--r--  1 yyin yyin  104068 Sep 27 16:46 all.bacteria.id
-rw-r--r--  1 yyin yyin   14574 Sep 27 16:23 all.tax.id.ps
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Music
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Pictures
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Videos
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Documents
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Public
drwxr-xr-x  2 yyin yyin    4096 Sep 14 14:04 Templates
-rw-r--r--  1 yyin yyin    8445 Sep 14 13:55 examples.desktop
```

d
means
folder

Permission

User and Group

size

Modification time

File and folder permission can be changed

Let’s look at one of the files under /home/yyin/:

```
-rw-r--r--  1 yyin yyin  104068 Sep 27 16:46 all.bacteria.id
```

The first one “-“ means it is a file; it will be “*d*” if it is a folder (directory).

The following nine columns indicate the permission of read (*r*), write (*w*) and execute (*x*) granted for the user (first three columns), the group (middle three) and others (last three). “-“ means no permission

```
-rwxrwxrwx                the file could be read, write, and
execute                      by anybody
-r--------                can only be read by the user
```

```
yyin@glu:~$ chmod go-r all.bacteria.id
yyin@glu:~$ ll all.bacteria.id
-rw------- 1 yyin yyin 104068 Sep 27 16:46 all.bacteria.id
```

Now you can not read that file in my home

How to view files

In order for you to read my file:

```
yyin@glu:~$ chmod go+r all.bacteria.id
yyin@glu:~$ ll all.bacteria.id
-rw-r--r-- 1 yyin yyin 104068 Sep 27 16:46 all.bacteria.id
```

At your home, try

```
less /home/yyin/all.bacteria.id
```

Similar commands:

```
more
head
tail
cat
```

Unlike in Windows, text files of any size can be viewed

Only text files can be viewed

There are also binary files, zipped files and tarred files that can not be viewed. For example,

- Executables (e.g., blast, samtools, bwa, bowtie)
- Data in binary format (e.g, BAM files, index files for BWA or Bowtie, formatted BLAST databases)
- Compressed files (usually *.gz, *.zip, *.bz2,…, but extensions not necessary)

How to copy and move files/folders

At you home, try

```
cp /home/yyin/all.bacteria.id .
```

Remember what dot (.) means?

Now try,

```
mv all.bacteria.id bioinfo
```

You created this folder in your home moments ago

What about?

```
mv all.bacteria.id all.bacteria.id2
```

Difference?

```
mv file folder
mv file file
```

What if you want to copy a folder?

```
cp -r bioinfo bioinfo2
```

How to delete files/folders?

`rm file`	`remove file`
`rm folder`	`remove empty folder`
`rm -rf folder`	`remove not empty folder`

For example,

```
rm -rf bioinfo2
```

WARNING: deleted things can not be recovered in Linux!!!

To be asked before deletion,

```
yyin@glu:~$ rm -i all.bacteria.id
rm: remove regular file `all.bacteria.id'? n
```

Create or edit files

Text editors:

nano

pico

vi

Suppose you are at your home:

1. Copy a file to your home/bioinfo

```
cp /home/yyin/work/SRR043594/454Isotigs.txt bioinfo
```

2. Try nano (Intuitive user interface)

```
nano 454Isotigs.txt
```

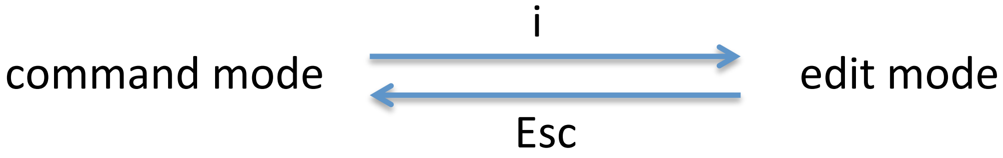
3. Try vi (command-driven interface, but much more power)

```
vi 454Isotigs.txt
```

Create a file from scratch using vi.

1) you type `vi filename` and hit *enter*
2) after you are in *vi*, type *i* to get into edit mode and copy & paste content in *vi*
3) hit *Esc* to exit edit mode and then *:x* to save the file and exit *vi*.

Vi basics



The following commands **operate in command mode (hit Esc before using them)**

x	delete one character at cursor position
u	undo
dd	delete the current line
G	go to end of file
1G	go to beginning of file
10G	go to line 10
$	go to end of line
1	go to beginning of line
:q!	exit without saving
:w	save (but not exit)
:wq or :x	save and exit
Arrow keys:	move cursor around (in both modes)

http://cbsu.tc.cornell.edu/ww/1/Default.aspx?wid=36

Search and substitution in vi

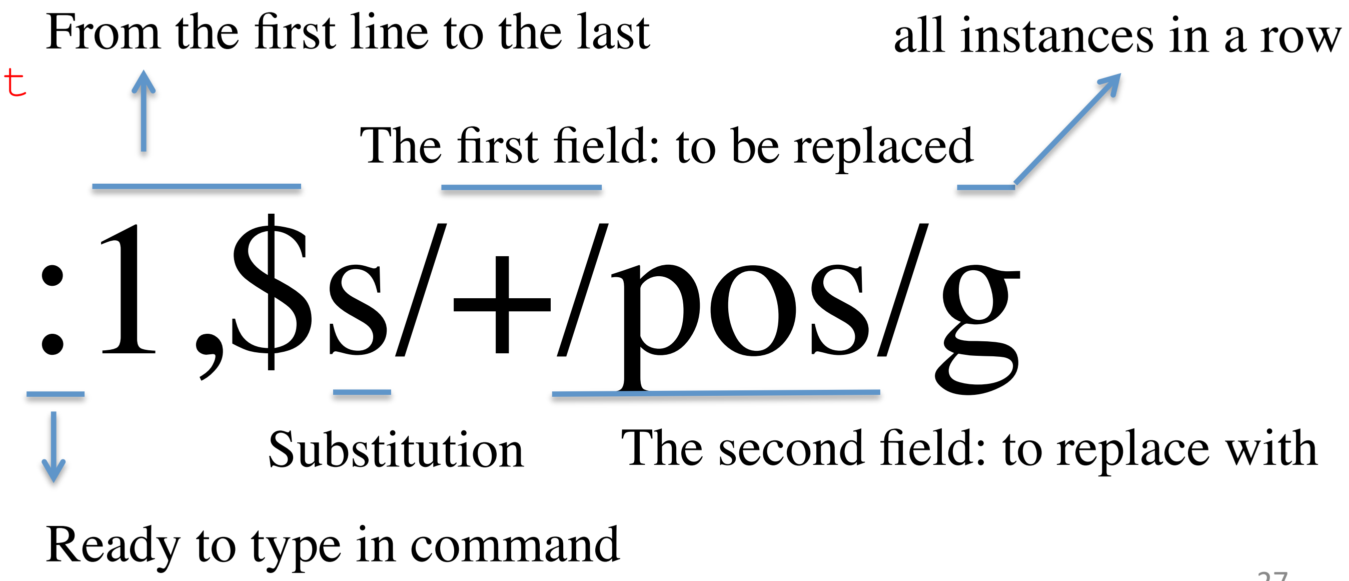
In command mode, you can do a number of fancy things. The most useful are:

- Search: hit slash ("/") to get the cursor to the left-bottom corner; you can type any word or letter to search it; type *n* to go to the next instance

- Replace: hit *Esc* (at any time, hitting *Esc* to get back to the default status is the safest thing to do) and type "*:1,$s/+/pos/g*" and then enter will replace all "+" to "pos".

Try this in

```
vi 454Isotigs.txt
```



From the first line to the last

all instances in a row

The first field: to be replaced

```
:1,$s/+/pos/g
```

Substitution

The second field: to replace with

Ready to type in command

Get data from remote ftp/http website

ftp
sftp
ncftp
lftp

```
lftp addr   command to connect to a remote ftp server
cd dir      change to the directory
cd ..       change to the upper folder (..)
ls          list files and folders in the current directory at once
ls dir      list files and folders in dir at once
ls | less   list page by page (good if the list is too long)
get file    get a file
mirror dir  get a folder
zmore file  view the file content
by or bye   exit lftp
```

wget

wget is a program useful for downloading files from both FTP and HTTP sites.

wget is non-interactive: you simply enter the necessary options and arguments on the command line and the file is downloaded for you.

Often you browse a http webpage or a ftp site and locate the remote files/folders you want to download and then go to the terminal and type

```
wget http-or-ftp-address
```

For example,

```
wget -q http://cys.bios.niu.edu/yyin/teach/PBB/cesa-pr.fa
wget -q ftp.ncbi.nih.gov/blast/db/FASTA/yeast.aa.gz
wget -r -q ftp.ncbi.nih.gov/genomes/Bacteria/

wget -q
ftp.ncbi.nih.gov:/blast/executables/LATEST/ncbi-
blast-2.2.27+-x64-linux.tar.gz
wget
ftp://emboss.open-bio.org/pub/EMBOSS/emboss-latest.tar.gz
```

Input and output redirection

Unix has a special way to direct input and output from commands or programs.

By default, the input is from keyboard (called standard input, *stdin*): you type in a command and Shell takes the command and executes it.

The standard output by default is to the terminal screen (*stdout*);

if the command or program failed, you will also have standard errors dumped to the terminal screen (*stderr*).

However, if you do not want the output dumped to the screen, you can use ">" to redirect/write the output into a file. For example, try

```
head -50 454Isotigs.txt
head -50 454Isotigs.txt > 454Isotigs.txt.head50
head -50 454Isotigs.tx
head -50 454Isotigs.tx 2> 454Isotigs.txt.head50
```

"2>" to dump the error message
No space here!

Archive and compress files/folders

To save disk space, we can compress large files if we do not intend to use them for a while. A lot of files downloaded from the web are compressed and need to be uncompressed before any processing can take place.

Common compressed formats:

- gzip (gz)

  gzip my_file *(compresses file* my_file*, producing its compressed version,* my_file.gz*)*

  gzip –d my_file.gz *(decompress* my_file.gz*, producing its original version* my_file*)*

- bzip2

  bzip2 my_file *(compresses file* my_file*, producing its compressed version,* my_file.bz2*)*

  bunzip2 my_file.bz2 *(decompress* my_file.bz2*, producing its original version* my_file*)*

Common compressed formats (continued):

- zip

  zip my_file.zip my_file1 my_file2 my_file3 *(create a compressed archive called* my_files.zip*, containing three files:* my_file1, my_file2, my_file3*)*

  zip -r my_file.zip my_file1 my_dir *(if* my_dir *is a directory, create an archive* my_file.zip *containing the file* my_file1 *and the directory* my_dir *with all its content)*

  zip –l my_file.zip *(list contents of the zip archive* my_file.zip*)*

  unzip my_files.zip *(decompress the archive into the constituent files and directories*

- tar

  tar -cvf my_file.tar my_file1 my_file2 my_dir *(create a compressed archive called* my_files.tar*, containing files* my_file1, my_file2 *and the directory* my_dir *with all its content)*

  tar –tvf my_file.tar *(list contents of the tar archive* my_file.tar*)*

  tar - xvf my_files.tar *(decompress the archive into the constituent files and directories)*

Use man tar to learn more

Common compressed formats (continued):

- tgz (also, tar.gz – essentially a combo of “tar” and “gzip”)

  tar -czvf my_file.tgz my_file1 my_file2 my_dir *(create a compressed archive called* my_files.tgz*, containing files* my_file1, my_file2 *and the directory* my_dir *with all its content)*

  tar –tzvf my_file.tgz *(list contents of the tar archive* my_file.tar*)*

  tar -xzvf my_files.tgz *(decompress the archive into the constituent files and directories)*

Check disk usage

Disk space is a limited resource, and you want to frequently monitor how much disk space you have used. To check the disk space usage, use the *du* (disk usage) command

```
yyin@glu:~$ du -hs .
22G
```

To check how much space left on the entire storage file system, use the *df* command

```
yyin@glu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       894G  665G  184G  79% /
udev             12G  8.0K   12G   1% /dev
tmpfs           4.8G  848K  4.8G   1% /run
none            5.0M     0  5.0M   0% /run/lock
none             12G  304K   12G   1% /run/shm
/dev/sdb1       1.8T  196M  1.7T   1% /media/DATAPART5
/dev/sdf1       917G  3.0G  868G   1% /media/DATAPART4
/dev/sde1       917G  2.7G  868G   1% /media/DATAPART3
/dev/sdd1       917G  4.5G  866G   1% /media/DATAPART2
/dev/sdc1       917G  4.3G  867G   1% /media/DATAPART1
```

history

w

write
wc
passwd

top
ps
kill
jobs
zless
cat

Ctrl+z, Ctrl+c, Ctrl+d

Find

Sort, cut, unique, join, paste, sed, grep, awk

Homework #7

Create a folder under your home called data

- Find and wget NCBI nonredundant protein database
- Find and wget Pfam-A hmm database
- Uncompressed the downloaded files

Next class: Install bioinfo softwares on Linux