Introduction to Unix

University of Massachusetts Medical School

October, 2014


CAN'T YOU DO
ANYTHING RIGHT?

DISCLAIMER

For the sake of clarity, the concepts mentioned in these slides have been simplified significantly.

Most of the command line tools and programs we explain have many more features that can't be found here. Unix command line tools are very well documented and their manual pages are freely available.

These slides are prepared for the users of MGHPCC. MGHPCC system has features that don't exist in standard Unix systems such as `loading modules`.

A rigorous treatment of topics on Unix and Bash can be found in various books in several levels.

Some References

Unix:
Unix in a Nutshell , Arnold Robbins, ISBN-13: 978-0596100292
Learning the bash Shell: Unix Shell Programming, Cameron Newham, ISBN-13: 978-0596009656

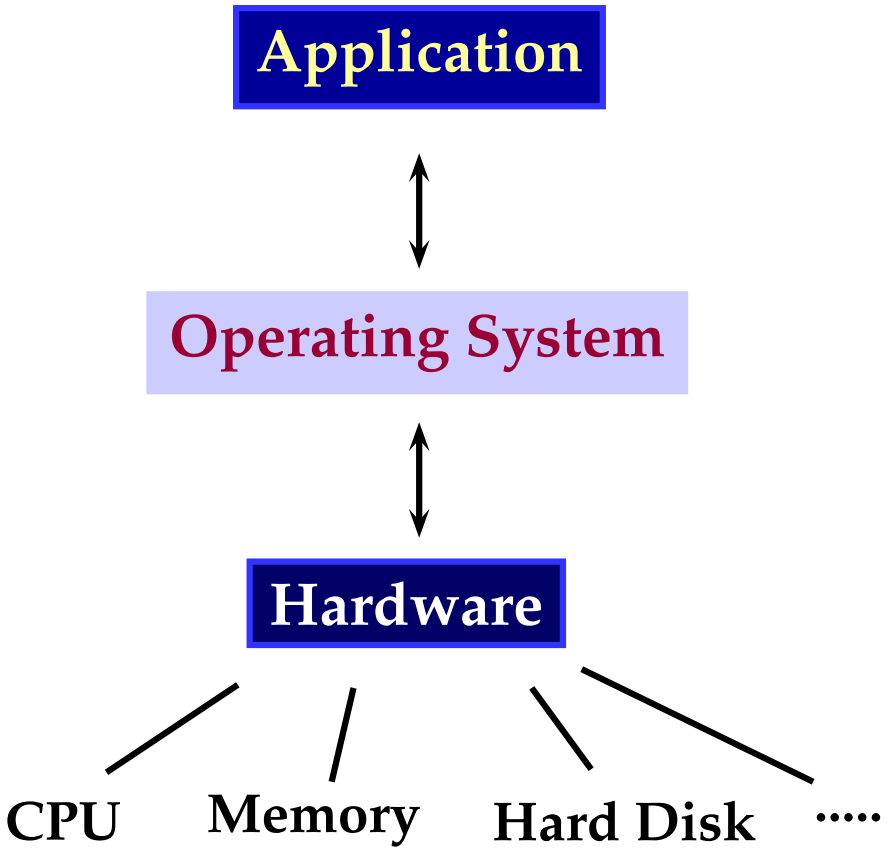
Umass High Perfotmance Computing:
http://wiki.umassrc.org/

Bioinformatics:
UCSC genome bioinformatics: https://genome.ucsc.edu/
Sam file format: samtools.github.io/hts-specs/SAMv1.pdf
Samtools: http://samtools.sourceforge.net/
Bedtools: http://bedtools.readthedocs.org/en/latest/
bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml
tophat: http://ccb.jhu.edu/software/tophat/index.shtml

What is an Operating System?

An operating system is a collection of software that manages system resources (such as memory, cpu(s), display and etc.) and provide applications a simpler interface to the system hardware.

Operating System



Unix

Developed in the 1960's by Ken Thompson, Dennis Ritchie. It became very popular in the 70's and 80's.

"Contrary to popular belief, Unix is user friendly. It just happens to be very selective about who it decides to make friends with."
–anonymous

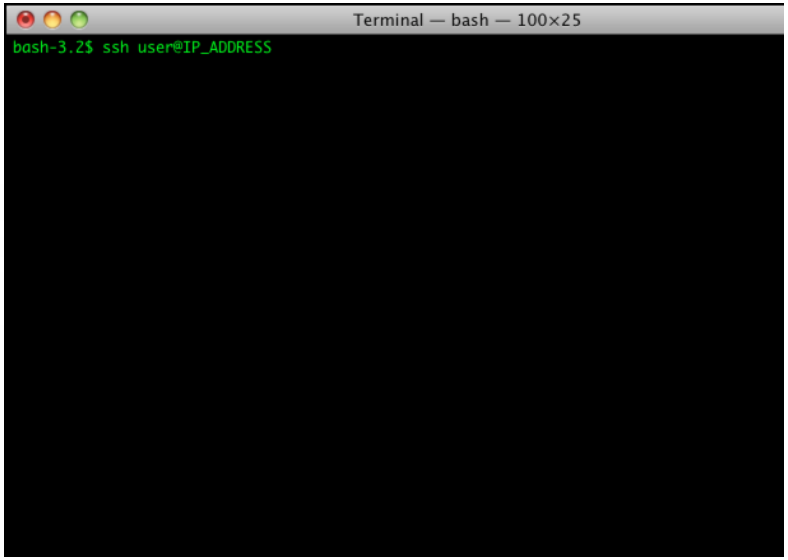
Unix-like operating systems

Unix-like operating systems *"behave"* like the original Unix operating system and comply (at least partially) with POSIX (portable operating system interface) standards.

Examples: Linux, OS X, Free BSD, Solaris, Hurd.

Unix $\equiv$ Unix-like $\equiv$ Unix-clone

What is a Terminal?

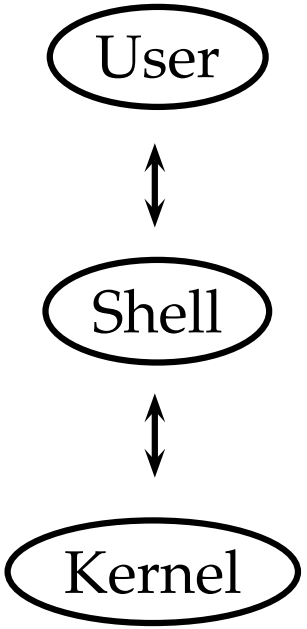


A terminal is a software that emulates a teletype writer terminal used in the early days of Unix.

Unix Shell

A shell is a software that runs inside a terminal and interprets and executes user commands.

One of the most popular shells being used today is called **BASH** (Bourne Again Shell). It is also the default shell in the UMass HPC cluster, OS X and many Linux distributions.



Convention

$ denotes the Bash command line prompt. It is not meant to be typed. All Bash commands will appear in a gray box.

```
$ bashcommand
```

Hello Bash

We connect to MGHPCC server.

```
$ ssh username@ghpcc06.umassrc.org
```

Hello Bash

We connect to MGHPCC server.

```
$ ssh username@ghpcc06.umassrc.org
```

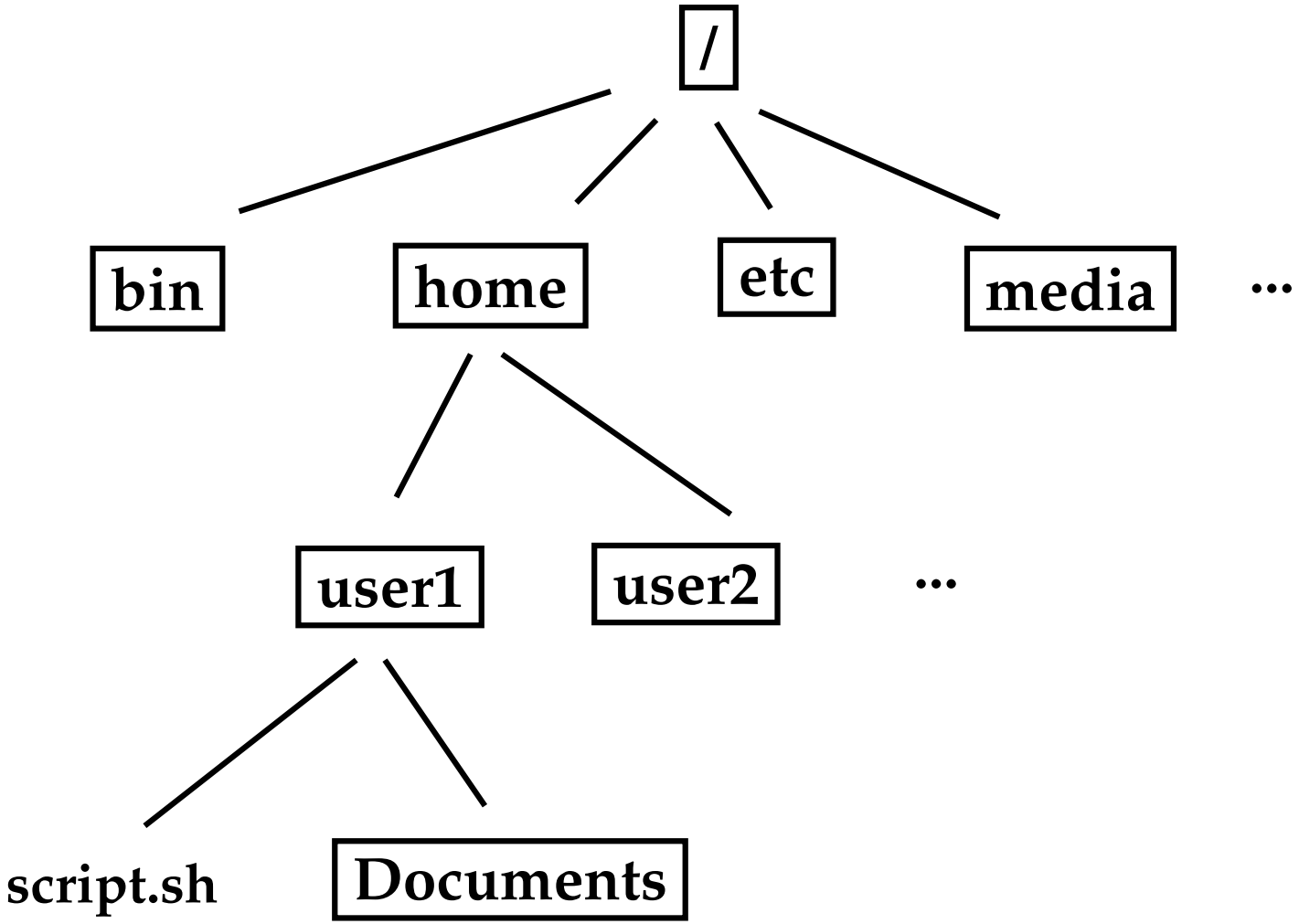
Let's verify that we are at the right place.

```
$ hostname
```

To print something on the screen, we use echo

```
$ echo Hello World
```

Navigating the Unix File System



Exploring the Directories

Bash always has a working directory. To see the working directory,

```
$ pwd
```

Exploring the Directories

Bash always has a working directory. To see the working directory,

```
$ pwd
```

To see the contents of the working directory,

```
$ ls
```

Exploring the Directories

Bash always has a working directory. To see the working directory,

```
$ pwd
```

To see the contents of the working directory,

```
$ ls
```

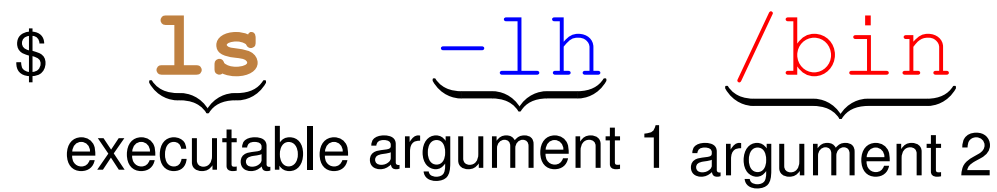
To see the contents of a particular directory, provide the path of the directory.

```
$ ls /bin
```

displays th contents of the directory `/bin`

Command Line Arguments

A program's execution and / or output format can be modified, according to our needs, by providing command line arguments.



More on ls

We can get more information about the files or directories by running `ls` with additional arguments.

Compare

```
$ ls
```

with

```
$ ls -l
```

A closer look at ls

`ls` list the given directory contents.
To see the complete list of options,

```
$ man ls
```

This will display the manual page of `ls`.

A closer look at ls

`ls` list the given directory contents.
To see the complete list of options,

```
$ man ls
```

This will display the manual page of `ls`.

Some commonly used options are:

-l	:	list contents in more detail
-A	:	list all files in the direcory
-h	:	when used with the `-l` option, prints file sizes in KB, MB, GB, and etc. to reduce the number of digits on the screen.

These options can be grouped

```
$ ls -lh
```

Let's first have some files and folders to experiment with.

```
$ /tmp/bootcamp.pl
```

Upon success, you will see
/home/user_name/bootcamp

Changing the working directory

First, we see the working directory

```
$ pwd
```

Now, lets change the directory to the `bootcamp` directory.

```
$ cd ~/bootcamp
```

We can verify the change by

```
$ pwd
```

Creating Directories

```
$ mkdir directory_path
```

Note that

```
$ mkdir /home/username/bootcamp/my_new_directory
```

is the same as

```
$ mkdir ~/bootcamp/my_new_directory
```

Copying Files

```
$ cp sourcefile(s) destination_path
```

Let's copy our sample files into the home folder.

```
$ cp ~/bootcamp/transcriptomics.tar.gz ~/transcriptomics.tar.gz
```

Copying Directories

```
$ cp source_directory destination_directory
```

will not work in general.
You need to provide the **-r** parameter.

```
$ cp -r source_directory destination_directory
```

Another useful option is `-v` which prints the files being copied.

```
$ cp -rv source_directory destination_directory
```

Moving Files

```
$ mv source_file destination_file
```

Deleting Files

`rm` : Remove comand

```
$ rm file_path
```

To remove non-empty directories, use rm with the `-r` option.

```
$ rm -r directory
```

rm is a dangerous command

There is no undoing of rm!
You may want to use it in the ineractive mode not to delete accidentally files

```
$ rm -i file
```

File Path

~ : Home directory

. : current directory

.. : parent directory

- If a file path begins with **/**, then it is an ***absolute path***. It doesn't depend on the current working directory.
- If a file path begins **WITHOUT /**, then it is a ***relative path***. The path is defined according to the current directory. So, the path **depends** on the current working directory.

Examples

`/tmp/bootcamp/bootcamp.pl` : Absolute path.

`bootcamp.pl` : Relative path. This is the `botcamp.pl` file in the **working** directory.

`../bootcamp.pl` : Relative path. This is the `botcamp.pl` file in the **parent** directory of the working directory.

`~/bootcamp.pl` : This is the file in the **home** directory of the current user.

Files

Technically, every file is a binary file.

Yet, it is common practice to group the files into two:

`Text Files`	:	Contains only printable characters.
`Binary Files`	:	Files that are not text files.

Binary files are generally more compact and therefore smaller in size. They are read and processed by specific applications.

Viewing text files

Text files are of extreme importance in bioinformatics.
There are many tools to view and edit text files.
`less` is a very useful program.

```
$ less text_file
```

For details read the manual page

```
$ man less
```

Editing Text Files

For a very simple text editor:

```
$ nano
```

vi and **emacs** are very powerful text editors with sophisticated features.

File Permissions

Let's try to run a bash script.

```
$ ./hello.sh
```

Let's try another one

```
$ ./other_hello.sh
```

Why didn't this work?

File Permissions

Let's try to spy on Manuel and see what he is up to.

```
$ ls /home/mg44w
```

File Permissions

Let's try to spy on Manuel and see what he is up to.

```
$ ls /home/mg44w
```

In Unix, users can only access (read, write or execute) files that they have permission to.

Seeing File Permissions

A typical output of

```
$ ls -lh
```

is

-rw-r–r–	1	ho86w	moore	11M	Nov 5	00:41	RPKM.csv
-rwxrwxrwx	1	ho86w	moore	143B	Oct 27	02:53	my_script.sh
drwxr-xr-x	21	ho86w	moore	714B	Jan 24	13:52	sam_files

r : read
w : write
x : execute

User : Has read, write and execute access.

Group : Has read and execute access but can't write to the file

Others : Has only read access.

Changing File Permissions

`chmod:` Change file mode bits.

```
$ chmod filemode file
```

To give only read access to other users,

```
$ chmod o=r myscript.sh
```

To give read and execution access together,

```
$ chmod o=rx myscript.sh
```

What is a Process?

A computer process is an instance of a program. It takes up space in the memory and its instructions are executed in the CPU.

Processes

You can list **your** running processes

```
$ ps -u username
```

where `username` is your username. You can see them in more detail with the `-F` option

```
$ ps -u username -F
```

Terminating a process

You can terminate a process by

```
$ kill -9 processID
```

where `processID` is the process id that can be obtained by the `ps` command.

Background Processes

To send a process to the background, first put the process on hold by pressing `ctrl` and z, and then send it to the background by

```
$ bg
```

You can check your processes by

```
$ jobs
```

To put a back to the foreground, first get the process id by

```
$ jobs
```

and then

```
$ fg % processNumber
```

To start a process on the background, put & at the end of your command

```
$ command &
```

Working with Compressed Files

A common type of binary files is zipped files. We can save space by zipping text files.
We use
`gzip` to compress / decompress files
`tar` to pack files into one file for archiving purposes.
To compress a file

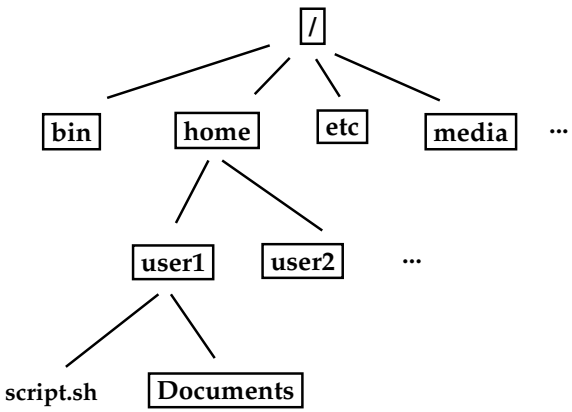
```
$ gzip file
```

To decompress, `reads.fastq.gz`,

```
$ gzip -d reads.fastq.gz
```

tar files

Say we collect our results on a project in one directory. We want to send this directory in an email attachment to a collaborator. How do we do that?



tar files

To pack a directory

```
$ tar -cvf arcive.tar directory
```

To pack a directory in a zipped tar file

```
$ tar -czvf archive.tar.gz directory
```

To get our directory back from the tar file

```
$ tar -xvf archive.tar
```

To get our directory back from the tar.gz file

```
$ tar -xvzf archive.tar.gz
```