Shell Basics

Introduction

- Most modern tools have a graphical user interface (GUI)
- But command-line user interfaces (CLUIs) still have their place
- Easier to see and understand what the computer is doing on your behalf
- Most important: it's easier to combine CLUI tools than GUI tools

How to tell if you can skip this lecture

- Do you know what a shell is?
- Do you know the difference between an absolute path and a relative path?
- Do you know what a process is?
- Do you know what a pipe is?
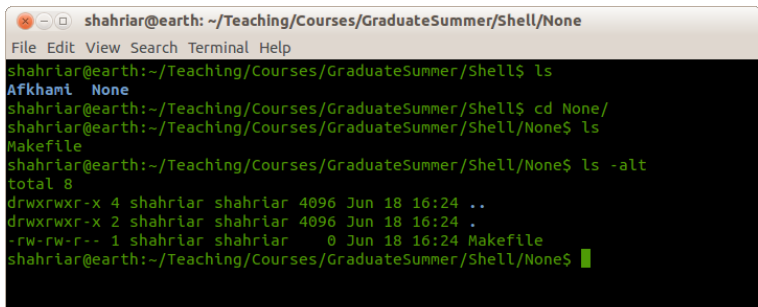- Do you know what $PATH is?
- Do you know what rwxr-xr-x means?

Kernighan & Pike 1984, still a good book despite its age

The command shell

Manages a user's interactions with the operating system by:

- Reading commands from the keyboard
- Figuring out what programs the user wants to run
- Running those programs
- Displaying their output on the screen
- Looks (and works) like an interactive terminal:

The Shell

We'll use bash:

- Output can be redirected using $>$ and $<$.
- Commands can be combined using pipelines.
- The history command can be used to view and repeat previous operations, while tab completion can be used to save re-typing.
- Directories (or folders) are nested to organize information hierarchically.
- Use grep to find things in files, and find to find files themselves.
- Programs can be paused, run in the background, or run on remote machines.
- The shell has variables like any other program, and these can be used to control how it behaves.

The File System

- The file system is the set of files and directories the computer can access
- Everything in a particular directory must have a unique name
- On Unix, the file system has a unique root directory called /
- Every other directory is a child of it, or a child of a child, etc.
- A path is a description of how to find something in a file system
- An absolute path describes a location from the root directory down
- A relative path describes how to find something from some other location

Most Unix commands have cryptic names

pwd	print working directory
ls	listing
cd	changing working directory
.	current directory
..	parent directory: directory immediately above this one

By convention, flags start with "-", as in "-c" or "-l"

ls -F
ls -a
ls -Fa
ls -s

```
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ pwd
/home/shahriar/Teaching/Courses/GraduateSummer/Shell/None
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cd ..
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ ls
Afkhami   None
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ ls -F
Afkhami/   None/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ ls -F -a
./   ../   Afkhami/   None/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ whoami
shahriar
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ date
Tue Jun 18 17:24:40 EDT 2013
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell$ cd None/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -Fa
./   ../   Makefile
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -atls
total 8
4 drwxrwxr-x 4 shahriar shahriar 4096 Jun 18 16:24 ..
4 drwxrwxr-x 2 shahriar shahriar 4096 Jun 18 16:24 .
0 -rw-rw-r-- 1 shahriar shahriar    0 Jun 18 16:24 Makefile
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ touch myfile.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -atl
total 8
drwxrwxr-x 2 shahriar shahriar 4096 Jun 18 17:28 .
-rw-rw-r-- 1 shahriar shahriar    0 Jun 18 17:28 myfile.txt
drwxrwxr-x 4 shahriar shahriar 4096 Jun 18 16:24 ..
-rw-rw-r-- 1 shahriar shahriar    0 Jun 18 16:24 Makefile
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ touch myfile.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -atl
total 8
-rw-rw-r-- 1 shahriar shahriar    0 Jun 18 17:29 myfile.txt
drwxrwxr-x 2 shahriar shahriar 4096 Jun 18 17:28 .
drwxrwxr-x 4 shahriar shahriar 4096 Jun 18 16:24 ..
-rw-rw-r-- 1 shahriar shahriar    0 Jun 18 16:24 Makefile
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$
```

mkdir	creating directory
cp	copy a file
nano	text editor
rm	delete a file
rmdir	delete a directory
mv	move or rename a file or directory
wc	word count
cat	catenate file

It's important to remember that there is no undelete.

```
shahriar@earth: ~/Teaching/Courses/GraduateSummer/Shell/None
File Edit View Search Terminal Help
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ mkdir tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -F
Makefile  myfile.txt  tmp/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ pwd
/home/shahriar/Teaching/Courses/GraduateSummer/Shell/None
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -a tmp/
.  ..
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ nano junk
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls
junk  Makefile  myfile.txt  tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -s
total 12
4 junk  0 Makefile  4 myfile.txt  4 tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -s -h
total 12K
4.0K junk     0 Makefile  4.0K myfile.txt  4.0K tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ rm tmp/
rm: cannot remove `tmp/': Is a directory
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ rmdir tmp/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -Fa
./  ../  junk  Makefile  myfile.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ mkdir tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ mv junk junk.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ mv junk.txt tmp/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cp myfile.txt tmp/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc tmp/myfile.txt
 2  6 47 tmp/myfile.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls *.txt
myfile.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cp myfile.txt file.
txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls ????.*
file.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$
```


Shell Basics

Some useful commands


```
shahriar@earth: ~/Teaching/Courses/GraduateSummer/Shell/None

File  Edit  View  Search  Terminal  Help

shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ nano myscripts.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -F
file.txt  Makefile  myfile.txt  myscripts.txt  tmp/
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cat myfile.txt
Name: Earth
Orbital Period: 365.256363004 days
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cat myscripts.txt
Name: Earth
Period: 365.26 days
Eccentricity: 0.02

Name: Venus
Period: 224.70 days
Eccentricity: 0.01
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc *.txt
  2    6   47 file.txt
  2    6   47 myfile.txt
  7   14  103 myscripts.txt
 11   26  197 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc -l *.txt
  2 file.txt
  2 myfile.txt
  7 myscripts.txt
 11 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc -w *.txt
  6 file.txt
  6 myfile.txt
 14 myscripts.txt
 26 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc -c *.txt
 47 file.txt
 47 myfile.txt
103 myscripts.txt
197 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ 
```

The real strength of the shell is its ability to combine simple utilities in useful ways.

$>$	redirect the output to a file
$<$	redirect the input to a file
|	pipe/connect processes together
*	pattern-matching wildcard
?	matches any single character

```
shahriar@earth: ~/Teaching/Courses/GraduateSummer/Shell/None
File Edit View Search Terminal Help
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc -l *.txt
  2 file.txt
  2 myfile.txt
  7 myscripts.txt
 11 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ wc -l *.txt > lengths
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls
file.txt  lengths  Makefile  myfile.txt  myscripts.txt  tmp
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ cat lengths
  2 file.txt
  2 myfile.txt
  7 myscripts.txt
 11 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ head -1 lengths
  2 file.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ tail lengths
  2 file.txt
  2 myfile.txt
  7 myscripts.txt
 11 total
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ tail lengths | head -1
  2 file.txt
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$ ls -atl
total 28
-rw-rw-r-- 1 shahriar shahriar   56 Jun 19 12:33 lengths
drwxrwxr-x 3 shahriar shahriar 4096 Jun 19 12:33 .
-rw-rw-r-- 1 shahriar shahriar  103 Jun 19 12:28 myscripts.txt
-rw-rw-r-- 1 shahriar shahriar   47 Jun 19 11:26 file.txt
drwxrwxr-x 2 shahriar shahriar 4096 Jun 19 11:14 tmp
-rw-rw-r-- 1 shahriar shahriar   47 Jun 18 17:53 myfile.txt
drwxrwxr-x 4 shahriar shahriar 4096 Jun 18 16:24 ..
-rw-rw-r-- 1 shahriar shahriar    0 Jun 18 16:24 Makefile
shahriar@earth:~/Teaching/Courses/GraduateSummer/Shell/None$
```

- Use ls to sort its output by filename extension.
- Use grep to find a pattern and in a file.
- Use diff to display the differences between two files.
- Use find command to find files.
- What do the commands pushd and popd do?

Another Exercise

Save the following:

```
####################################
#!/bin/bash

echo "Hello, $LOGNAME!"
echo "Current date is `date`"
echo "User is `whoami`"
echo "Current directory `pwd`"
chmod a+x hello.sh
echo "This file is an executable `ls -alt hello.sh`"
####################################
```

in a file called hello.sh. Run the script by invoking the command:
source hello.sh
After doing this, you can always run the script like:
./hello.sh
Explain what each line is doing by looking at the output of running
the shell script.

Some useful commands

- awk - Text processing language: Find and Replace text within file(s)
- chmod - Change access permissions
- grep - Search file(s) for lines that match a given pattern
- history - Command History
- locate - Find files - simple but fast
- whereis - Search the user's $path, man pages and source files for a program
- xargs - Execute utility, passing constructed argument list(s)