

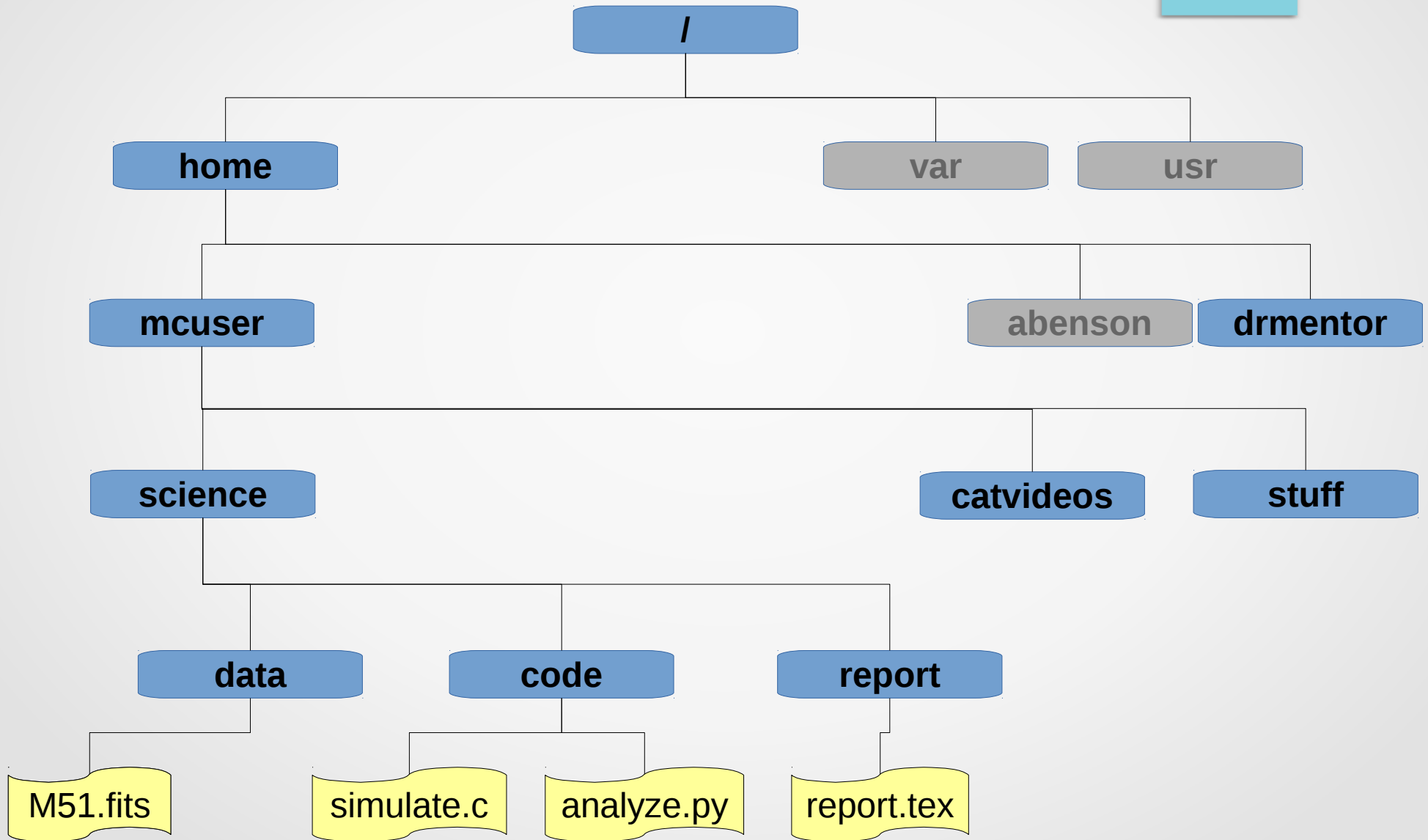
Introduction to UNIX Command Line

- Files and directories
- Some useful commands (echo, cat, grep, find, diff, tar)
- Redirection
- Pipes
- Variables
- Background processes
- Remote connections (e.g. ssh, curl)
- Scripts

The Command Line

- What is it?
 - An interface to UNIX
 - You type commands, things happen
- Also referred to as a “shell”
- We'll use the bash shell – check you're using it by typing (you'll see what this means later):
 - `echo $SHELL`
 - If it doesn't say “bash”, then type `bash` to get into the bash shell

Files and Directories



Files and Directories

- Get a pre-made set of directories and files to work with
- We'll talk about what these commands do later
- The “\$” is the command prompt (yours might differ). Type what's listed after it, then press enter.

```
$ curl -L -o playground.tar 'http://bit.ly/2swZ1eF'  
$ tar xvf playground.tar
```

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

Files and directories

“Present Working Directory”

Shows the full path of your current location in the filesystem.

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

“Change Directory”

Change to a new directory – relative to current location unless you give a full path

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  scienc
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

“LiSt”

List the content of the current directory.
Shows both directories and files.

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

“MaKe DIRectory”

Make a new directory in the current directory

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

“MaKe DIRectory”

Make a new directory in the current directory

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

Options “-l” to ls requests a long listing.

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

Options “-l” to ls requests a long listing.

Files and directories

```
$ pwd
/home/abenson
$ cd playground
$ pwd
/home/abenson/playground
$ ls
animals  documents  science
$ mkdir mystuff
$ ls
animals  documents  mystuff  science
$ cd animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
```

Permissions

User

Group

Size in bytes

Creation time

File name

Files and directories

```
$ ls -a
.  ..  badger.txt  porcupine.txt
$ cd .
$ pwd
/home/abenson/playground/animals/mammals
$ cd ..
$ pwd
/home/abenson/playground/animals
$ cd
$ pwd
/home/abenson
$ cd ~
$ pwd
/home/abenson
```

Files and directories

```
$ cd ~/playground/animals/mammals/
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ chmod go+w badger.txt
$ ls -l
total 8
-rw-rw-rw-. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ cp badger.txt super_badger.txt
$ ls
badger.txt  porcupine.txt  super_badger.txt
$ mv super_badger.txt super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt  super_mellivorinae.txt
```

Files and directories

```
$ cd ~/playground/animals/mammals/
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson
-rw-r--r--. 1 abenson abenson
$ chmod go+w badger.txt
$ ls -l
total 8
-rw-rw-rw-. 1 abenson abenson
-rw-r--r--. 1 abenson abenson
$ cp badger.txt super_badger.txt
$ ls
badger.txt  porcupine.txt  super_badger.txt
$ mv super_badger.txt super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt  super_mellivorinae.txt
```

“CHange MODe”

Changes permissions for the file:

u = user

g = group

o = other

+ -= – add, remove, set

rx – read, write, execute

Files and directories

```
$ cd ~/playground/animals/mammals
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ chmod go+w badger.txt
$ ls -l
total 8
-rw-rw-rw-. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ cp badger.txt super_badger.txt
$ ls
badger.txt  porcupine.txt  super_badger.txt
$ mv super_badger.txt super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt  super_mellivorinae.txt
```

“CoPy”

Make a copy of a file – you give the name of the original file, followed by the new file.

Files and directories

```
$ cd ~/playground/animals/mammals/
$ ls
badger.txt  porcupine.txt
$ ls -l
total 8
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ chmod go+w badger.txt
$ ls -l
total 8
-rw-rw-rw-. 1 abenson abenson 1944 May 31 18:03 badger.txt
-rw-r--r--. 1 abenson abenson 1347 May 31 18:05 porcupine.txt
$ cp badger.txt super_badger.txt
$ ls
badger.txt  porcupine.txt  super_badger.txt
$ mv super_badger.txt super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt  super_mellivorinae.txt
```

“MoVe”

Move (rename) a file – you give the name of the original file, followed by the new name.

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badget.txt
Badgers are short-legged omnivores...
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badget.txt
Badgers are short-legged omnivores..
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

"echo"

Write the given text to screen (or somewhere else – more of this later).

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badget.txt
Badgers are short-legged omnivores...
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

“conCATenate”

Take the content of all files listed,
join them together, and write it out.

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badger.txt
Badgers are short-legged omnivores...
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

“conCATenate”

Take the content of all files listed,
join them together, and write it out.

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badger.txt
Badgers are short-legged omnivores...
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

"more"

Display contents of a file to terminal with pagination (use return or space to step through).

Useful Commands

```
$ echo "Hello? Can you hear me?"
Hello? Can you hear me?
$ cat badger.txt
Badgers are short-legged omnivores...
$ cat badger.txt porcupine.txt
Badgers are short-legged omnivores...
Porcupines are rodents with a coat of sharp spines...
$ more badger.txt
Badgers are short-legged omnivores...
$ less badger.txt
Badgers are short-legged omnivores...
$ man less
NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
```

"less"

Fancy version of more. Scroll forward and backward. Search ("/"), etc.

Useful Commands

```
$ echo "Hello? Can you hear me?"
```

```
Hello? Can you hear me?
```

```
$ cat badger.txt
```

```
Badgers are short-legged omni
```

```
$ cat badger.txt porcupine.tx
```

```
Badgers are short-legged omnivores...
```

```
Porcupines are rodents with a coat of sharp spines...
```

```
$ more badger.txt
```

```
Badgers are short-legged omnivores...
```

```
$ less badget.txt
```

```
Badgers are short-legged omnivores...
```

```
$ man less
```

```
NAME
```

```
less - opposite of more
```

```
SYNOPSIS
```

```
less -?
```

```
less --help
```

"MANual"

Show documentation for a command.

Useful Commands

```
$ grep skunk badger.txt
```

```
members of the skunk family, placing
```

```
$ grep rodent *.txt
```

```
porcupine.txt:Porcupines are rodents
```

```
porcupine.txt:Porcupines are the third-largest of the rodents,
```

```
$ cd
```

```
$ find playground -name "*.txt"
```

```
playground/animals/mammals/super_mellivorinae.txt
```

```
playground/animals/mammals/badger.txt
```

```
playground/animals/mammals/porcupine.txt
```

```
playground/animals/reptiles/caiman.txt
```

```
$ find playground -type d
```

```
playground
```

```
playground/documents
```

```
playground/animals
```

```
playground/animals/mammals
```

```
playground/animals/reptiles
```

```
playground/science
```

Useful Commands

```
$ grep skunk badger.txt
```

```
members of the skunk family, placing
```

```
$ grep rodent *.txt
```

```
porcupine.txt:Porcupines are rodents
```

```
porcupine.txt:Porcupines are the third-largest of the rodents,
```

```
$ cd
```

```
$ find playground -name "*.txt"
```

```
playground/animals/mammals/super
```

```
playground/animals/mammals/badge
```

```
playground/animals/mammals/porcu
```

```
playground/animals/reptiles/caiman.txt
```

```
$ find playground -type d
```

```
playground
```

```
playground/documents
```

```
playground/animals
```

```
playground/animals/mammals
```

```
playground/animals/reptiles
```

```
playground/science
```

"Get Regular Expression"

Search for a word (or words) in a file.

Useful Commands

```
$ grep skunk badger.txt
members of the skunk family, placing
$ grep rodent *.txt
porcupine.txt:Porcupines are rodents
porcupine.txt:Porcupines are the third-largest of the rodents,
$ cd
$ find playground -name "*.txt"
playground/animals/mammals/super_mellivorinae.txt
playground/animals/mammals/badger.txt
playground/animals/mammals/porcupine.txt
playground/animals/reptiles/caiman.txt
$ find playground -type d
playground
playground/documents
playground/animals
playground/animals/mammals
playground/animals/reptiles
playground/science
```

Wildcards

"*" matches any sequence of characters
"?" matches any single character

Useful Commands

```
$ grep skunk badger.txt
members of the skunk family, placing
$ grep rodent *.txt
porcupine.txt:Porcupines are rodents
porcupine.txt:Porcupines are the third-largest of the rodents,
$ cd
$ find playground -name "*.txt"
playground/animals/mammals/super_mellivorinae.txt
playground/animals/mammals/badger.txt
playground/animals/mammals/porcupine.txt
playground/animals/reptiles/caiman.txt
$ find playground -type d
playground
playground/documents
playground/animals
playground/animals/mammals
playground/animals/reptiles
playground/science
```

“find”

Search a path for files/directories that match some specification

Redirection

```
$ rm super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt
$ cd playground/animals/mammals/
$ cat badger.txt > another_badger.txt
$ ls
another_badger.txt  badger.txt  porcupine.txt
$ cat > list_of_mammals.txt
bat
cat
ocelot
camel
^D
$ cat list_of_mammals.txt
bat
cat
ocelot
camel
```

Useful Commands

```
$ tar cvf playground.tar playground
playground/
playground/documents/
playground/mystuff/
playground/animals/
playground/animals/mammals/
playground/animals/mammals/super_mellivorinae.txt
playground/animals/mammals/badger.txt
playground/animals/mammals/porcupine.txt
playground/animals/reptiles/
playground/animals/reptiles/caiman.txt
playground/science/
$ ls
playground  playground.tar
```

Useful Commands

```
$ tar cvf playground.tar playground
```

```
playground/
```

```
playground/documents/
```

```
playground/mystuff/
```

```
playground/animals/
```

```
playground/animals/mammals/
```

```
playground/animals/mammals/su
```

```
playground/animals/mammals/ba
```

```
playground/animals/mammals/po
```

```
playground/animals/reptiles/
```

```
playground/animals/reptiles/c
```

```
playground/science/
```

```
$ ls
```

```
playground  playground.tar
```

“Tape ARchive”

Combine all files (and directories) into one archive file. Useful for sharing data, archiving etc.

Use:

```
tar xvf playground.tar
```

to re-extract.

Redirection

```
$ rm super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt
$ cd playground/animals/mammals/
$ cat badger.txt > another_badger.txt
$ ls
another_badger.txt  badger.txt
$ cat > list_of_mammals.txt
bat
cat
ocelot
camel
^D
$ cat list_of_mammals.txt
bat
cat
ocelot
camel
```

“ReMove”

Delete a file. THERE IS NO UNDO!!!

Add -r option to recursively remove all files in a directory.

THERE IS NO UNDO!!!!!!!!!!!!!!!

Redirection

```
$ rm super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt
$ cd playground/animals/mammals/
$ cat badger.txt > another_badger.txt
$ ls
another_badger.txt  badger.txt  porcupine.txt
$ cat > list_of_mammals.txt
bat
cat
ocelot
camel
^D
$ cat list_of_mammals.txt
bat
cat
ocelot
camel
```

Redirection

“>” sends output to named file instead of to the terminal. So, this was a way to make a copy of our file.

Redirection

```
$ rm super_mellivorinae.txt
$ ls
badger.txt  porcupine.txt
$ cd playground/animals/mammals/
$ cat badger.txt > another_badger.txt
$ ls
another_badger.txt  badger.txt  porcupine.txt
$ cat > list_of_mammals.txt
bat
cat
ocelot
camel
^D
$ cat list_of_mammals.txt
bat
cat
ocelot
camel
```

Redirection

No input file given..... Terminal waits for you to type input. Whatever you type goes into the file.

Redirection

```
$ echo "More badger stuff here later." >> another_badger.txt
$ cat badger.txt
.
.
.
More badger stuff to go here later.
$ diff badger.txt another_badger.txt
7a8
> More badger stuff to go here later.
```

Redirection

```
$ echo "More badger stuff here later." >> another_badger.txt
```

```
$ cat badger.txt
```

```
.  
.
.
```

```
More badger stuff to go here later.
```

```
$ diff badger.txt another_badger.txt
```

```
7a8
```

```
> M
```

Append

Like ">", but appends instead of over-writing.

```
More later.
```

Redirection

```
$ echo "More badger stuff here later." >> another_badger.txt
$ cat badger.txt
.
.
.
More badger stuff to go here later.
$ diff badger.txt another_badger.txt
7a8
> More badger stuff to go here later.
```

“DIFFerence”

Show differences between two files.

Redirection and Pipes (and Sorting)

```
$ sort list_of_mammals.txt
bat
camel
cat
ocelot
$ sort < list_of_mammals.txt
bat
camel
cat
ocelot
$ cat list_of_mammals.txt | sort
bat
camel
cat
ocelot
```

Redirection and Pipes (and Sorting)

```
$ sort list_of_mammals.txt
bat
camel
cat
ocelot
$ sort < list_of_mammals.txt
bat
camel
cat
ocelot
$ cat list_of_mammals.txt | sort
bat
camel
cat
ocelot
```

“Sort”

Sort input into alphanumerical order.

Redirection and Pipes (and Sorting)

```
$ sort list_of_mammals.txt
bat
camel
cat
ocelot
$ sort < list_of_mammals.txt
bat
camel
cat
ocelot
$ cat list_of_mammals.txt | sort
bat
camel
cat
ocelot
```

Redirect input

Redirects input from a file into a command.

Redirection and Pipes (and Sorting)

```
$ sort list_of_mammals.txt
bat
camel
cat
ocelot
$ sort < list_of_mammals.txt
bat
camel
cat
ocelot
$ cat list_of_mammals.txt | sort
bat
camel
cat
ocelot
```

Pipe

“|” takes output from first command and sends it as input to second.

Variables

```
$ echo $SHELL
/bin/bash
$ echo $USER
abenson
$ echo $HOME
/home/abenson
$ workPath=$HOME/playground
$ cd $workPath
$ pwd
/home/abenson/playground
```

Variables

```
$ echo $SHELL
/bin/bash
$ echo $USER
abenson
$ echo $HOME
/home/abenson
$ workPath=$HOME
$ cd $workPath
$ pwd
/home/abenson/
```

Environment Variables

“\$” means replace following with the value of the named variable.

The shell sets up many environment variables, such as those shown.

Variables

```
$ echo $SHELL
/bin/bash
$ echo $USER
abenson
$ echo $HOME
/home/abenson
$ workPath=$HOME/playground
$ cd $workPath
$ pwd
/home/abenson/playground
```

User Variables

Define your own variables and then use them in the command line.

Background Processes

```
$ ./job.sh  
$ ^C  
$ ./job.sh  
$ ^Z  
$ jobs  
$ fg  
$ ^Z  
$ bg  
$ jobs
```

Background Processes

```
$ ./job.sh  
$ ^C  
$ ./job.sh  
$ ^Z  
$ jobs  
$ fg  
$ ^Z  
$ bg  
$ jobs
```

A script

we'll learn more about scripts soon –
this one just writes annoying messages

Background Processes

```
$ ./job.sh
$ ^C
$ ./job.sh
$ ^Z
[1]+  Stopped
$ jobs
[1]+  Stopped
$ fg
$ ./job.sh
$ ^Z
[1]+  Stopped
$ bg
[1]+ ./job.sh &
$ jobs
[1]+  Running
$ ./job.sh &
[2] 21194
```

Cancel

^C cancels the running command

./job.sh

./job.sh &

Background Processes

```
$ ./job.sh
$ ^C
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ jobs
[1]+  Stopped                  ./job.sh
$ fg
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ bg
[1]+  ./job.sh &
$ jobs
[1]+  Running                  ./job.sh &
$ ./job.sh &
[2] 21194
```

Suspend

^Z suspends the running command

Background Processes

```
$ ./job/.sh
$ ^C
$ ./job/.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ jobs
[1]+  Stopped                  ./job.sh
$ fg
$ ./job/.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ bg
[1]+ ./job.sh &
$ jobs
[1]+  Running                  ./job.sh &
$ ./job/.sh &
[2] 21194
```

Jobs

Shows a list of jobs running or suspended

Background Processes

```
$ ./job.sh
$ ^C
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ jobs
[1]+  Stopped                  ./job.sh
$ fg
$ ./job.sh
$ ^Z
[1]+  Stopped
$ bg
[1]+ ./job.sh &
$ jobs
[1]+  Running                  ./job.sh &
$ ./job.sh &
[2] 21194
```

“ForeGround”

Bring the last suspended job back into the foreground (i.e. start running again).

Background Processes

```
$ ./job.sh
$ ^C
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ jobs
[1]+  Stopped                  ./job.sh
$ fg
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ bg
[1]+ ./job.sh &
$ jobs
[1]+  Running
$ ./job.sh &
[2] 21194
```

“BackGround”

Run the last suspended job in the background.

Background Processes

```
$ ./job.sh
$ ^C
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ jobs
[1]+  Stopped                  ./job.sh
$ fg
$ ./job.sh
$ ^Z
[1]+  Stopped                  ./job.sh
$ bg
[1]+ ./job.sh &
$ jobs
[1]+  Running
$ ./job.sh &
[2] 21194
```

"&"

Start the command in the background.

Network and Remote Connections

```
$ hostname
dhcp225.obs.carnegiescience.edu
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] hostname
obs-ssh.ad.ciw.edu
testuser@mies ~] exit
Connection to mies.stanford.edu closed.
$ cd animals/mammals
$ scp badger.txt testuser@mies.stanford.edu:myname.txt
badger.txt      100% 1944      1.9KB/s   00:00
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] ls
badger.txt
testuser@mies ~] rm myname.txt
testuser@mies ~] exit
Connection to mies.stanford.edu closed.
```

Network and Remote Connections

```
$ hostname
dhcp225.obs.carnegiescience.edu
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] hostname
obs-ssh.ad.ciw.edu
testuser@mies ~] exit
Connection to ssh.obs.carnegiescience.edu closed.
$ cd animals/mammals
$ scp badger.txt testuser@mies.stanford.edu:
badger.txt      100% 1944
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] ls
badger.txt
testuser@mies ~] rm badger.txt
testuser@mies ~] exit
Connection to ssh.obs.carnegiescience.edu closed.
```

“Secure Shell”

Open a shell on a remote server.

Password is: HvtTc6Bw

Network and Remote Connections

```
$ hostname
dhcp225.obs.carnegiescience.edu
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] hostname
obs-ssh.ad.ciw.edu
testuser@mies ~] exit
Connection to ssh.obs.carnegiescience.edu closed.
$ cd animals/mammals
$ scp badger.txt testuser@mies.stanford.edu:
badger.txt      100% 1944      1.9KB/s   00:00
$ ssh testuser@mies.stanford.edu
Last login: Thu Jun  2 14:03:01 2016 from ociw.edu
testuser@mies ~] ls
badger.txt
testuser@mies ~] rm badger.txt
testuser@mies ~] exit
Connection to ssh.obs.carnegiescience.edu closed.
```

“Secure CoPy”

Copy a file to a remote server. Or from remote server to your laptop

Password is: HvtTc6Bw

Network and Remote Connections

```
$ curl -L -o playground1.tar 'http://bit.ly/2swZ1eF'
```

curl

Download a file from a URL. Useful for downloading big datasets on a remote server.

Scripts

```
#!/bin/sh

echo This is a script...
echo Hello user $USER
echo Here are all the files I can find...
find . -type f
exit
```

Scripts

```
$ cd ~/playground
$ ./script.sh
This is a script...
Hello user abensonca
Here are all the files I can find...
./script.sh
./animals/mammals/badger.txt
./animals/mammals/porcupine.txt
./animals/reptiles/caiman.txt
```

Scripts: .bashrc

- The script in `~/ .bashrc` gets run every time you log in
- Use it to configure the environment the way you want it to be
- For example, you could add an alias:

```
$ alias up='cd ..'  
$ cd ~/playground/animals  
$ pwd  
/home/abenson/playground/animals  
$ up  
$ pwd  
/home/abenson/playground
```

Bonus: Text Editing with EMACS

- There are many text editors in Unix
- Use whichever you prefer
- One option available everywhere is EMACS

```
$ emacs playground/animals/reptiles/caiman.txt &  
[1] 29575
```

Bonus: Text Editing with EMACS

Open file

^X^F

Save file

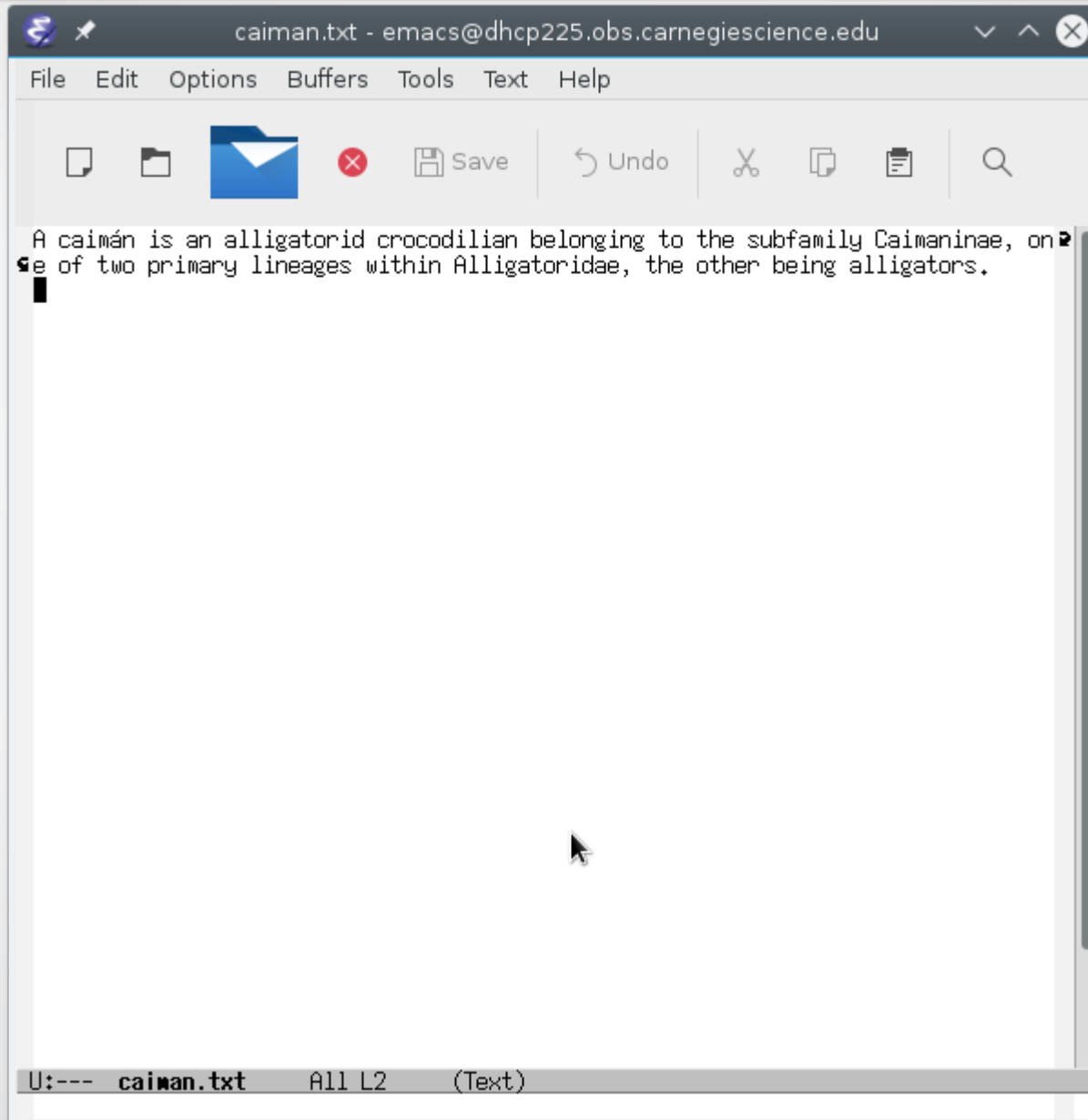
^X^S

Close file

^X^K

Exit EMACS

^X^C



The screenshot shows the Emacs text editor window titled "caiman.txt - emacs@dhcp225.obs.carnegiescience.edu". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Text", and "Help". The toolbar contains icons for opening a file, saving, undo, cut, copy, paste, and search. The main text area contains the following text: "A caimán is an alligatorid crocodilian belonging to the subfamily Caimaninae, one of two primary lineages within Alligatoridae, the other being alligators." The status bar at the bottom shows "U:--- caiman.txt A11 L2 (Text)".

Cut

^W

Paste

^Y

Search forward

^S

Search backward

^R