The Unix Shell

More tricks!

More Tricks

- These are small exercises to tell you things you need to know.

xargs

- This does not work

```
$ find acsoe | ls
acsoe         presentations
$
```

- Find pipes a list of files to ls.
- ls ignores input and just does a normal listing of the current working directory.
- Lots of commands expect a list of arguments, not standard input. Is there anything to help?

xargs

- The "xargs" command runs the same command on all files specified in the input.
- Usually used with "find" output, e.g.:

```
find . -name '*.nc' | xargs chmod u=rwx
```

Changes permissions on all .nc files.

xargs

- by default splits the file list into *batches*:

```
chmod 644 file1 file2 … file100
chmod 644 file101 file102 …
```

- use "`-n 1`" if the command can only process one file at a time:

```
find . -name '*.tar' | xargs -n 1 tar -tvf
```

  - displays contents of all 'tar' files found

xargs exercise

Use find piped to xargs to do something (wc, ls –l , head -1, etc)

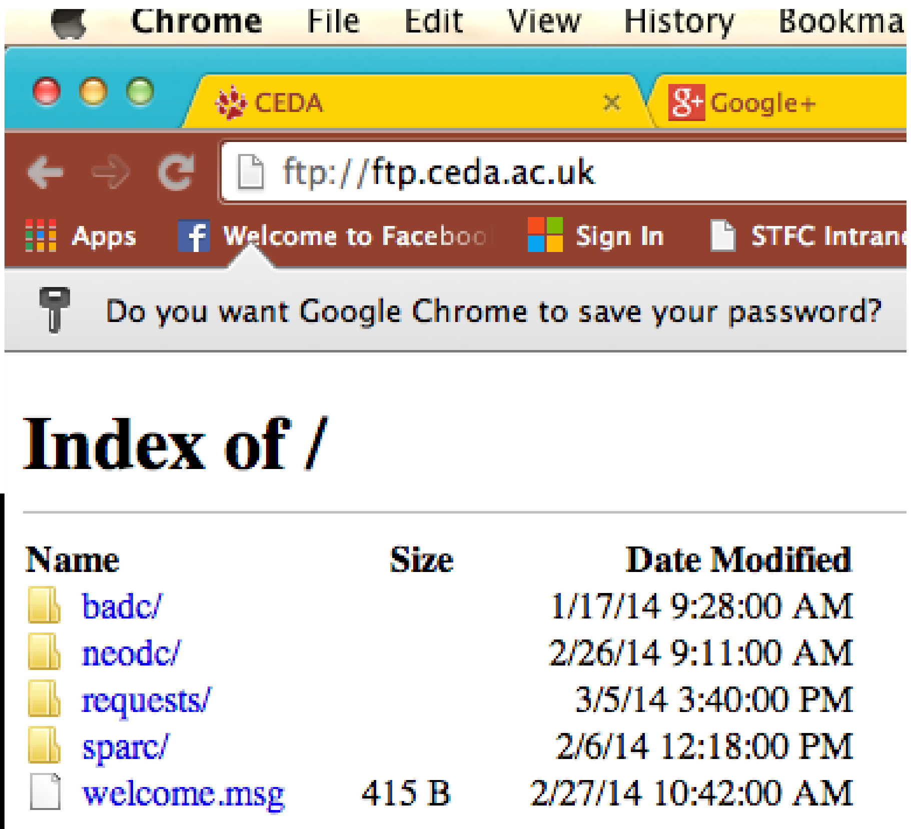
Other ways to move data around

There are a lot of tools to help you move data from one machine to another. Common ones are:

- FTP
- SFTP
- Rsync
- Wget
- Curl

FTP

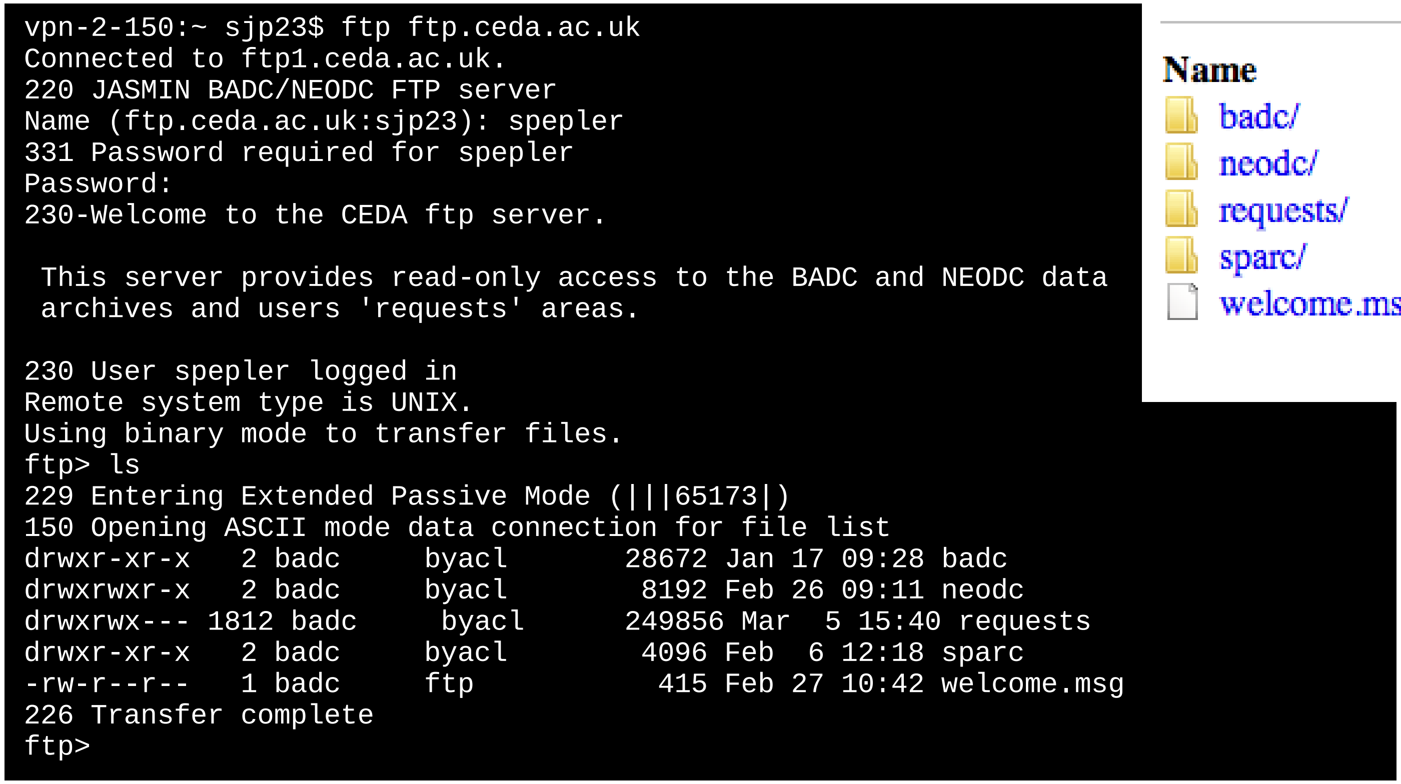
- Can use most browsers to ftp files
- Can also use a command line interface too (easy to script)



Do you want Google Chrome to save your password?

Index of /

Name	Size	Date Modified
badc/		1/17/14 9:28:00 AM
neodc/		2/26/14 9:11:00 AM
requests/		3/5/14 3:40:00 PM
sparc/		2/6/14 12:18:00 PM
welcome.msg	415 B	2/27/14 10:42:00 AM



```
vpn-2-150:~ sjp23$ ftp ftp.ceda.ac.uk
Connected to ftp1.ceda.ac.uk.
220 JASMIN BADC/NEODC FTP server
Name (ftp.ceda.ac.uk:sjp23): spepler
331 Password required for spepler
Password:
230-Welcome to the CEDA ftp server.

 This server provides read-only access to the BADC and NEODC data
 archives and users 'requests' areas.

230 User spepler logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||65173|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 badc     byacl       28672 Jan 17 09:28 badc
drwxrwxr-x   2 badc     byacl        8192 Feb 26 09:11 neodc
drwxrwx--- 1812 badc     byacl      249856 Mar  5 15:40 requests
drwxr-xr-x   2 badc     byacl        4096 Feb  6 12:18 sparc
-rw-r--r--   1 badc     ftp           415 Feb 27 10:42 welcome.msg
226 Transfer complete
ftp>
```

Transferring data with sftp

- Like scp, this uses ssh. However, gives an interactive interface like ftp.
- Usage (Linux):
  - "`sftp` *`host`*" or "`sftp` *`username@host`*"
  - ftp commands e.g. `cd`, `lcd`, `put`, `get`
- Windows:
  - psftp (in PuTTY suite) works similarly from command line
  - also Filezilla GUI
- As before, set up ssh keys first.

wget

- `wget` makes it easy to grab resources from a http or ftp address.
- (curl is a similar tool)

Transfering data exercise

- Have a look at the following address in a web browser. Note it's not a http address.
- ftp://sparc-ftp1.ceda.ac.uk/sparc/hres/1_second/text/2011/03020/
- Get one of the files with wget from the command line.

rsync

- copies files over the network (or locally)
- where destination files already exist, copies only what is required to update any differences
- push / pull files over ssh:

```
rsync -e ssh user@host:remote_path local_path      ← pull
rsync -e ssh local_path user@host:remote_path      ← push
```

  - requires no special configuration (though remember to set up ssh keys)
  - similar to scp syntax, e.g. remote path is relative to home directory unless starts with /

Transferring data with rsync (continued)

- Useful flags for rsync:

  `-r` (recursive) – go down the directory tree copying stuff.

  `-c` (checksum) – when deciding what files to send, look not only at size and timestamp but if necessary also file contents

  `--delete` – remove files from destination not present at source end. *(Test with -n first!)*

  `-v` (verbose) – list files that are transferred (or deleted)

  `-n` (dry run) – go through the motions but do not actually transfer (or delete) files. Useful with -v.

  - `-a` (archive) – copy recursively and try to copy permissions, ownership, etc.

rsync exercise

- Copy the data in the acsoe directory to an acsoe2 directory with `rsync`. Use the `–v` (verbose) option so you can see what is happening.
- Run the command again and note what is copied.
- Add a new file to acsoe directory, modify another file and delete a third. Run the command a third time.
- Try `rsync` to the remote machine used in the scp exercise.

Pattern matching: globs

- Unix shells recognises various wildcards in filenames. We have seen these two:

  * matches any number of characters

  ? matches one character

- These filename matching patterns, known as "globs", are replaced with a list of matching filenames before the command is executed.

```
$ ls
1    3    5    a1   b1   c1   d1
2    4    a    b    c    d


$ ls *1
1 a1 b1  c1   d1


$ ls ??
a1 b1 c1 d1
```

Pattern matching: globs

- Here is another glob for you

  […] matches any of the characters listed (or range of characters, e.g. [0-9])

```
$ ls [a-c]*
a a1 b b1 c c1
```

Pattern matching: globs

- And another glob

  `{fred, barny, wilma}` matches any of the comma separated names listed.

  For example `ls *.{jpg,png}` will list all your jpg and png files.

Glob exercise

- Use glob matching in acsoe/freetex-98/jungfrau
- Make a for loop that word counts only files from that date range

I'm a terminal based editor get me out of here!

- Some editors use the terminal window.
- The default editor used by some commands means you need to know how to get out of them sometimes.
- If you are not used to them you can get stuck.
- Emacs – get out with ^X ^C   (maybe need ^G^X^C)
- Vi – get out with escape, then :q! then enter.

Have a go!

Some standard environment variables you might like to know about

- DISPLAY sets the display windowed programs attempt to use.
- HOME your home directory.
- PATH Where your shell looks for programs to run.
- EDITOR If you run a program that needs a text editor it will look in here to see which one to use.
- PS1 Your command line prompt.

/dev/null

- If you don't need the stdout or the stderr you can dump it.
- For example, a program produces a lot of output and a few error messages mixed in. If you can't find the error messages then redirect the output to /dev/null

Give if a go with

```
$ head -1 `find acsoe/freetex-98 -type f`
Too much output to notice the errors.

$ head -1 `find acsoe/freetex-98 -type f` > /dev/null
```

Sourcing files

Try this:

Make a script file which sets a variable

```
Z=Dino
```

Run the file and then use echo to look at the Z variable.

Try again but this time do this

```
$ . ./myscript
```

This is called sourcing a file is runs it in the current shell instead of starting a new one.

Compression and aggregation tools

- Zip (and unzip) – makes a zip file (compression and aggregation)
- Gzip (and ungzip) – compresses a file. (just compression)
- Tar – make an tar file. An aggregation. Often used with gzip.

Compression and aggregation tools

- Make a tar file

```
$ tar cvf macehead.tar acsoe/lterm/macehead
```

- Compress it with gzip

```
$ gzip macehead.tar
```

- Move the file to /tmp
- Uncompress it with gunzip
- Untar the file

```
$ tar xvf macehead.tar
```