Exercise 1: Exploring the file system

AIM

Login and look at some files.

Issues covered

Commands: `pwd`, `ls`, `gedit`, `cd`, `cp`, `mv`, `mkdir`, `rm`, `rmdir`, `man`.
What's in "`/tmp`", "`/`" and "`/etc`"

Instructions

1. Let's get started by logging in.

a. Login to the laptop (you should have a username and password).
b. Start a terminal window.

2. Have a look around your home directory. Try the following commands.

```
pwd
ls
ls -l
ls -a
ls ..
ls shell
```

3. Let's have a look somewhere else. Change directory to acsoe.

```
$ cd ncas-isc/shell/acsoe
```

Now repeat (2)

4. Manipulating some files and directories.

a. Make a file called "`myfile`" in "`/tmp`" with gedit.
b. Make a subdirectory in "`/tmp`" called "`mydir`"
c. Rename the file "`myfile.txt`" and the subdirectory "`X`"
d. Copy "`myfile.txt`" into the "`X`" subdirectory
e. Tidy up - delete the file and subdirectory

5. Use the "`man ls`" command to find other listing options. Experiment… have a look in "`/`", and "`/etc`".

6. How not to do it:

a. Use `cd` with no arguments to jump back to your home directory.
b. Go into the "`pain`" directory
c. Use `ls` to see what files are here
d. Move them to more sensible names (if you can).

Solution 1: Exploring the file system

2.

```
$ pwd
/Users/sjp23/play/workshop_shell
$ ls
acsoe
$ ls -l
total 0
drwxr-x---  16 sjp23  staff  544 26 Feb 16:21 acsoe
$ ls -a
.    ..   acsoe
$ ls ..
badc           dataman        workshop_shell
$ ls acsoe
00README  eae-96    ease-96   freetex-96     hillcloud-96   lterm
c-130     eae-97    ease-97   freetex-98     hillcloud-97   ozprof
```

3.

```
$ cd acsoe
$ pwd
/Users/sjp23/play/workshop_shell/acsoe
$ ls
00README  eae-96    ease-96   freetex-96     hillcloud-96   lterm
c-130     eae-97    ease-97   freetex-98     hillcloud-97   ozprof
$ ls -l
total 8
-rwxr-x---  1 sjp23  staff  190 26 Feb 16:21 00README
drwxr-x---  8 sjp23  staff  272 26 Feb 16:20 c-130
drwxr-x---  8 sjp23  staff  272 26 Feb 16:20 eae-96
drwxr-x---  8 sjp23  staff  272 26 Feb 16:21 eae-97
drwxr-x---  7 sjp23  staff  238 26 Feb 16:21 ease-96
drwxr-x---  6 sjp23  staff  204 26 Feb 16:21 ease-97
drwxr-x---  6 sjp23  staff  204 26 Feb 16:21 freetex-96
drwxr-x---  6 sjp23  staff  204 26 Feb 16:21 freetex-98
drwxr-x---  8 sjp23  staff  272 26 Feb 16:21 hillcloud-96
drwxr-x---  9 sjp23  staff  306 26 Feb 16:21 hillcloud-97
drwxr-x---  6 sjp23  staff  204 26 Feb 16:21 lterm
drwxr-x---  6 sjp23  staff  204 26 Feb 16:21 ozprof
$ ls -a
.    .summary  eae-96    ease-97    hillcloud-96  ozprof
..   00README  eae-97    freetex-96 hillcloud-97  .checksums
c-130          ease-96   freetex-98 lterm
```

```
$ ls ..
acsoe
```

```
$
```

4.

```
$ cd /tmp
$ gedit myfile
$ ls
myfile
test.txt
$ mkdir mydir
$ ls -l
total 56
drwxr-xr-x  2 sjp23           wheel     68 26 Feb 17:14 mydir
-rw-r--r--  1 sjp23           wheel      7 26 Feb 17:13 myfile
$ mv myfile X
$ mv X myfile.txt
$ mv mydir X
$ cp myfile.txt  X
$ ls -l
total 56
drwxr-xr-x  3 sjp23           wheel    102 26 Feb 17:15 X
-rw-r--r--  1 sjp23           wheel      7 26 Feb 17:13 myfile.txt
$ ls -l X
total 8
-rw-r--r--  1 sjp23  wheel  7 26 Feb 17:21 myfile.txt
$ rm X/myfile.txt
$ rmdir X
$
```

6.

```
$ cd
$ cd pain
$ ls -l
total 0
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:48 -l
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:49 What the $
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:53 Ω
$ mv -l  L
$ mv What\ the\ \$ What_the_dollar
$ mv ? omega
$ ls -l
total 0
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:48 L
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:49 What_the_dollar
-rw-r--r--  1 sjp23  staff  0 20 Mar 12:53 omega
```

Exercise 2: Pipes and filters

AIM

Construct a command using pipes and filters to print just the name of the longest file.

Issues covered

Commands: `cat`, `wc`, `head`, `tail`, `cut`, `sort`, `uniq`, `|`, `*`, `?`
Using shell command completion and history.

Instructions

1. In the directory "acsoe/eae-97/macehead" construct a pipe and filter command to print the file with the most lines. (Hint: use `head`, `tail`, `wc`, `sort` and `cut`

2. Use the up arrow to edit the last command. Change the command to look for the longest file in characters.

3. Use "`*`" to look for the longest file in all the subdirectories of "acsoe/eae-97".

4. Have a play with the arrow keys and the tab key - what do they do? Try the `history` command.

Solution 2: Pipes and filters

1.

```
wc -l eae-97/macehead/* | sort -n | tail -n 2 | head -n 1 | cut -c 10-
```

As an alternative to the last command in the above pipeline, `cut -f 2 -d ' '`
will extract the second field, using space as delimiter between fields – more robust if the character width can vary.

2.

```
wc -m eae-97/macehead/* | sort -n | tail -n 2 | head -n 1 | cut -c 10-
```

3.

```
wc -m eae-97/*/* | sort -n | tail -n 2 | head -n 1 | cut -c 10-
```

4.

Up and down arrows scroll through the command history of the shell (very useful for repeating the same commands). The tab key makes suggestions for completing what you are typing. Often tab completion writes the rest of the filename after typing in the start of it. Tab key twice lists all possible completion alternatives. The `history` command lists the command history; use `!33` to run the 33rd entry in the history list.

Exercise 3: Permissions

AIM

To get comfortable with Unix permission system.

Issues covered

Commands: `chmod`, `ls -l`, `more`, `less`, `chgrp`

Instructions

1. Explain permissions to other people.

a. Change directory to "acsoe/freetex-98/Jungfrau".
b. Use `ls -l` to look at the files.
c. Run the script "./set_chmod.sh". This script will change the permissions on some of the files in this directory.

```
$ ./set_chmod.sh
```

d. Use `ls -l` again to look at the file permissions.
e. Pair up and describe to your partner what the permission mean.
f. Use the `more` (or `less`) command to see if you can access the files. Try to run the files.

2. Which do you think are most sensible set of permissions.

a. Change the files to have sensible permissions.
b. Make a new directory
c. Experiment with the permissions on the directory.

Solution 3: Permissions

1. a-d)

```
workshop_shell$ cd acsoe/freetex-98/Jungfrau
Jungfrau$ ls -l
total 33064
-rwxr-x---  1 sjp23  staff   183188 26 Feb 16:21 jf980314.em3
-rwxr-x---  1 sjp23  staff   291474 26 Feb 16:21 jf980315.em1
-rwxr-x---  1 sjp23  staff   200955 26 Feb 16:21 jf980315.em2
-rwxr-x---  1 sjp23  staff    31641 26 Feb 16:21 jf980317.nox
…
Jungfrau$ ./set_chmod.sh
Jungfrau$ ls -l
total 33064
-rwx------  1 sjp23  staff   183188 26 Feb 16:21 jf980314.em3
----rwx---  1 sjp23  staff   291474 26 Feb 16:21 jf980315.em1
-------rwx  1 sjp23  staff   200955 26 Feb 16:21 jf980315.em2
-rwxrwx---  1 sjp23  staff    31641 26 Feb 16:21 jf980317.nox
…
```

1. f)

No user permission…

```
Jungfrau$ more jf980315.em2
jf980315.em2: Permission denied
Jungfrau$ more jf980315.em1
jf980315.em1: Permission denied
```

Read permission ok…

```
Jungfrau$ more jf980318.pr1
24 1001
Monks, Paul and Zanis, Prodromos
School of Chemistry, Univesrity Leicester, Leicester, UK
Peroxy Radical Chemical Amplifier II, Free Tropospheric Experiment II, Jungfraujoch, Switzerland
FREETEX '98
```

Execute permission ok… but not really something you can execute!

```
Jungfrau$ ./jf980318.fm1
./jf980318.fm1: line 1: 24: command not found
./jf980318.fm1: line 2: Graham: command not found
./jf980318.fm1: line 3: syntax error near unexpected token `('
./jf980318.fm1: line 3: `School of Environmental Sciences, University of East Anglia (UEA), Norwich, UK'
```

Exercise 4: Needle in a haystack

AIM

Use `find` and `grep` to find the "Needle".

Issues covered

Commands: `find`, `grep`.

Instructions

1. Find the file "needle.txt" in the "acsoe" directory.

a. Change directory to acsoe.
b. Use the find command to look for the file called "needle.txt".

2. Expand your search to look for files with needle anywhere in the filename.

a. Same again but use a `*` or two

3. Use grep to find the word needle in the files under "acsoe/ease-96/Jetstream".

4. Use the `man` page for grep to work out how to do a case insensitive search for needle.

5. Use `grep` on the "js960724.ps2" file to print all lines without 1 in. (use the man page to find the right option)

6. Use `grep` on the "js960724.ps2" file to print all lines without 4 or 6 in, but does contain 33. (use a pipes to chain grep commands together)

Solution 4: Needle in a haystack

1.

```
workshop_shell$ cd acsoe
acsoe$ find . -name needle.txt
./hillcloud-96/h2/needle.txt
```

2.

```
acsoe$ find . -name '*needle*'
./ease-96/jetstream/ddddd.needle.xxx
./hillcloud-96/h2/needle.txt
```

3.

```
acsoe$ cd ease-96/jetstream
jetstream$ grep needle *
js960719.nx7:201.453308  105246 needle   2.2  .1  2.1    0   2.15    1
```

4.

```
jetstream$ grep -i needle *
js960716.jn3:198.520544    122935   26.6   .0126 NEEDLE
js960719.nx7:201.453308  105246 needle   2.2  .1  2.1    0   2.15    1
```

5.

```
jetstream$ grep -v 1 js960724.ps2
Lightman Paul
ACRU Imperial College, TTC, Silwood Park, Ascot, Berks SL5 7PW
GPS Lat & Long, Barometric Altitude
ACSOE OXICOA EASE96
Time in fractional Julian day (GMT Timebase)
4
999999 999 999 999
Time GMT hhmmss
Latitude  Decimal Degrees
Longitude Decimal Degrees
Altitude  m
5
THIS-FILE-NAME=js960724.ps2
E-MAIL-CONTACT=p.lightman@ic.ac.uk
Jday        Time GMT    Latitude    Longitude   Altitude
```

6.

```
jetstream$ grep -v 4 js960724.ps2 | grep -v 6 | grep 33
215.5025     120333     53.3098    -10.2228    592.9
215.5025     120335     53.3102    -10.2205    590.5
215.5037     120519     53.3332    -10.1023    598.3
215.5037     120521     53.3337    -10.1001    599.2
```

Exercise 5: Job control

AIM

Start and stop a sleep job. Confidence in starting and stopping jobs and familiarity with variables.

Issues covered

Commands: `set`, `export`, `echo`, `ps`, `top`, `fg`, `bg`, `jobs`, `kill`, `sleep`, `time`, `&`, `^C`, `^Z`
Writing commands in a file to make a shell script.

Instructions

1. Run "`sleep 10`". What does it do?

2. Make a "snooze.sh" file with gedit with the following content.

```
echo feeling sleepy…
sleep 10
echo wake up!
```

Run the script

```
$ ./snooze.sh
```

3. Edit the snooze.sh script to use a variable X to control the length of sleep.

4. Set `X` to 40 then run it again in the background using `&`. Use `ps` to see the process at work. Remember to `export X`.

5. Run 3 instances of the process at once.

a. Start 3 snooze jobs in the background.
b. Use the jobs command to see the processes.
c. Kill 2 of them while they sleep.
d. Bring the last one to the foreground and let it complete.

6. Run 3 instances of the process.

a. Start 2 snooze jobs in the background.
b. Start another in the foreground.
c. Use `^Z` to stop the foreground job.
d. Use `bg` to put the job in the background.
e. Bring `%1` to the foreground with the `fg` command.
f. Kill that job with `^C`.
g. Let the other jobs finish.

7. Find the difference between " " and ' '. Make a shell variable `Y` set to text of your choice. Use `echo` to print the variable. Try the following

```
echo * $Y
echo '* $Y'
echo "* $Y"
```

Solution 5: Job control

1.

```
workshop_shell$ sleep 10
```

2.

```
workshop_shell$ gedit snooze.sh
workshop_shell$ ./snooze.sh
-bash: ./snooze.sh: Permission denied
workshop_shell$ chmod 755 snooze.sh
workshop_shell$ ./snooze.sh
Feeling sleepy...
Wake up!
```

3.

```
workshop_shell$ gedit snooze.sh
workshop_shell$ cat snooze.sh
echo Feeling sleepy...
sleep $X
echo Wake up!

workshop_shell$ export X=5
workshop_shell$ ./snooze.sh
Feeling sleepy...
Wake up!
```

4.

```
workshop_shell$ export X=40
workshop_shell$ ./snooze.sh &
[1] 3509
workshop_shell$ Feeling sleepy...

workshop_shell$ ps
  PID TTY          TIME CMD
  612 ttys000    0:00.58 -bash
 3509 ttys000    0:00.00 -bash
 3510 ttys000    0:00.00 sleep 40
workshop_shell$
workshop_shell$ Wake up!

[1]+  Done                    ./snooze.sh
workshop_shell$
```

5.

```
workshop_shell$ ./snooze.sh &
```

```
[1] 3550
workshop_shell$ Feeling sleepy...

workshop_shell$ ./snooze.sh &
[2] 3552
workshop_shell$ Feeling sleepy...
./snooze.sh &
[3] 3554
workshop_shell$ Feeling sleepy...

workshop_shell$ jobs
[1]   Running                 ./snooze.sh &
[2]-  Running                 ./snooze.sh &
[3]+  Running                 ./snooze.sh &
workshop_shell$ kill %1
workshop_shell$
[1]   Terminated: 15          ./snooze.sh
workshop_shell$ kill %2
[2]-  Terminated: 15          ./snooze.sh
workshop_shell$
workshop_shell$ fg %3
./snooze.sh
Wake up!
workshop_shell$
```

6.

```
workshop_shell$ ./snooze.sh  &
[1] 11411
workshop_shell$ Feeling sleepy...

workshop_shell$ ./snooze.sh  &
[2] 11413
workshop_shell$ Feeling sleepy...
./snooze.sh
Feeling sleepy...
^Z
[3]+  Stopped                 ./snooze.sh
workshop_shell$ bg
[3]+ ./snooze.sh &
workshop_shell$ fg %1
./snooze.sh
^Cworkshop_shell$
workshop_shell$ Wake up!
Wake up!
```

```
[2]-  Done                    ./snooze.sh
[3]+  Done                    ./snooze.sh
```

Exercise 6: Wake up

AIM

What did we do yesterday?

Issues covered

Stuff from yesterday.

Instructions

1. **Make a pipe and filter command to find the third biggest file (by number of lines) in the acsoe directory. (use find, sort tail and head)**
2. **Use `$()` to store the result in a variable.**
   `X = $(my long command with |pipes and | filters)`
3. **Use echo to show the result is stored in the variable.**
   `echo $X`

Solution 6: Wake up

```
ncas-isc$ wc -l $(find shell/acsoe -type f) |sort -n  |tail -4| head -1
14421 shell/acsoe/eae-97/macehead/mh970427.cn1
ncas-isc$ X=$(wc -l $(find shell/acsoe -type f) |sort -n  |tail -4| head
-1)
ncas-isc$ echo $X
14421 shell/acsoe/eae-97/macehead/mh970427.cn1
```

Git

Aim: to start using GitHub for course examples.

On the GitHub site

1) Click to add a new repo
2) Call new repo "my-isc-work" and 3) Click the add README box
4) Click the Create repository button
5) Click "Clone or download" and copy the link.

Now in the terminal window

1) Make sure you are in your home directory
2) Write the `git clone` command and add the URL to the repository (which is different for each user)

```
$ cd
$ git clone https://<username>@github.com/<username>/my-isc-work.git
```

3) Copy some of the files from yesterday's exercises into the "my-isc-work" directory
4) Change directory to "my-isc-work"
5) Use `git status` to see what needs adding to the version control system.

```
$ git status
```

6) Add the files

```
$ git add .
```

7) Commit the changes:

```
$ git commit -m 'Add some test files'
```

8) Update GitHub repo:

```
$ git push
```

9) Look on GitHub to see your changes.

Exercise 7: Shell scripts

AIM

Use a shell script to look at output and error redirection.

Issues covered

`>`, `>>`, `which`, `tail`, `for` loops. Interpreter header lines.

Instructions

1. **Go into "acsoe/eae-97/macehead". This is a directory containing only files and no subdirectories.**
2. **Make a shell script that loops over the files in a directory printing the last line in each file. Redirect the output to a file.**
3. **Make a subdirectory to trigger an error message from the `tail` command. Append this to an error log. Run the command a few times.**
4. **Make sure your script is only executable by you and it has the right #! first line.**
5. **Use which to find the python interpreter. Try changing the first line to point to that interpreter instead. What happens now when you run it?**

Solution 7: Shell scripts

1-4.

```
$ cat ./my.sh
#!/bin/bash

for i in *
do
tail -1 $i
done



$ ./my.sh > output.txt
$ cat output.txt
150.9375  -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -
999  -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999 -999
150.593   150.349   150.838   208.8     287  88.43     204  5.6
150.592    150.351  150.832   15.938          92.504            41.925
15.106             42.969          123.124         2.592      111.401
1.188       12.862       2.393       7.397        112.808     61.752
150.592    150.351       150.832   77.378          31.374       52.149
20.287       354.200     228.634    0.000       2.486       23.683
6.034       36.593       73.681
147.10     146.85      147.35      14.3            2.17E-07
1.85E-07        7.78E-07        9.36E-08       1.63E-08       3.09E-08
147.10     146.85      147.35      14.3            1.27E-03
7.67E-05        9.77E-05        1.57E-03       7.78E-05       1.90E-04
2.77E-05
done
done
$
```

5.

```
$ which python
/usr/bin/python
$ emacs my.sh
$ ./my.sh
  File "./my.sh", line 3
    for i in *
              ^
SyntaxError: invalid syntax
```

SSH

AIM

Have a go at using `ssh`.

Issues covered

`ssh`, `scp`, `df`

Instructions

Note: User names and IP addresses may change.

1. **`ssh` to "xuser1@10.30.10.119".**
2. **Use `ls`, `pwd` and other commands to prove to yourself it's a different computer.**
3. **Logout (use exit or ^D)**
4. **Use scp to copy some files to the remote computer.**
5. **Login again and ls the files.**
6. **Logout**
7. **Use ssh to run df on the remote computer.**

Other stuff

Aim: find out about some other useful stuff

Xargs

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

Wget

2) Look at ftp://sparc-ftp1.ceda.ac.uk/sparc/hres/1_second/text/2011/03020/ in a web browser.
3) Use `wget` to download ftp://sparc-ftp1.ceda.ac.uk/sparc/hres/1_second/text/2011/03020/03020_2011010112.tgz

Copying data

4) 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.
5) Run the command again and note what is copied.
6) Add a new file to the "acsoe" directory, modify another file and delete a third. Run the command a third time.
7) Try `rsync` to the remote machine used in the scp exercise.

Globbing

8) Use glob matching in "acsoe/freetex-98/Jungfrau" to match files for dates from 980323 to 980327
9) Make a `for` loop that word counts only files from that date range.

Getting out of text editors

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 with ^X followed by ^C

Vi – get out with : followed by q.

Have a go…

/dev/null

Give it 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 that 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.

Tar

Make a tar file
```
$ tar cvf macehead.tar acsoe/lterm/macehead
```
Compress is with `gzip`
```
$ gzip macehead.tar
```
Move the file to "/tmp"
Uncompress it with `gunzip`