Course outline

```
wk date  topic
01 02/09 python (lecture1)
02 09/09 python (lecture1)
03 16/09 object orientation (lecture2)
04 23/09 reg ex (lecture2)
05 30/09 reg ex (lecture3)
06 07/10 reg ex + bash (lecture4)
07 14/10 --cancel--
08 21/10 --strike--
09 28/10 bash (lecture5) <<
10 04/11 webservers (lecture6)
11 11/11 webservers + web apis (lecture6, lecture 7)
12 18/11 web apis (lecture7)
13 25/11 web api work (lecture8)
```

The bash Command line interface (continued)

- Topics today include...
- the pipe
- bash scripts
  - really simple little programs to do something repetitive.
- grep
  - lets you use regular expressions on the commandline
- wget
  - download stuff off the internet
- etc...

Pipe

- The **pipe** | lets you send text from one command to another.
- E.g.

```
cat test.py | head -n 4
```

- outputs the first 4 lines of `test.py`.
- **grep** lets you do regular expressions. This command displays the first 20 words in the dictionary that start with q (which don't have apostrophes in them).

```
cat /etc/dictionaries-common/words | grep "^q[^']*$" | head -n 20
```

(all on one line)

Astronomy Picture of the Day (APOD) Example

- Example:
  - For the astronomy pictures of the day for the first week in September, how many links went to:
    - wikipedia
    - nasa
    - elsewhere?

APOD Example

- Step One: Download all of the first week of apods:
  - We can download a web resource using wget - a really handy command!

```
wget http://apod.nasa.gov/apod/ap140901.html
wget http://apod.nasa.gov/apod/ap140902.html
wget http://apod.nasa.gov/apod/ap140903.html
wget http://apod.nasa.gov/apod/ap140904.html
```

    - : etc :
  - We should never have to do something repetitive - that's what the computer's for.

  - Hint: During lectures, if you don't have internet access, you can access these files on my laptop during the lecture at:

```
wget http://10.10.?.?/sept_apods/ap140904.html
```

APOD Example

We can loop using this syntax: (all on one line!)

```
for ((i=1;i<=7;++i)); do
  wget http://apod.nasa.gov/apod/ap14090$i.html; done
```

- The details of bash script syntax are beyond this course - but if you need it in future, you'll usually be able to find a clue while searching online on how to do the thing you want.
- One bit of coursework might require some lines of bash, but you won't be expected to know anything beyond the very basics (cd, ls, pwd, etc) for the exam.

APOD Example

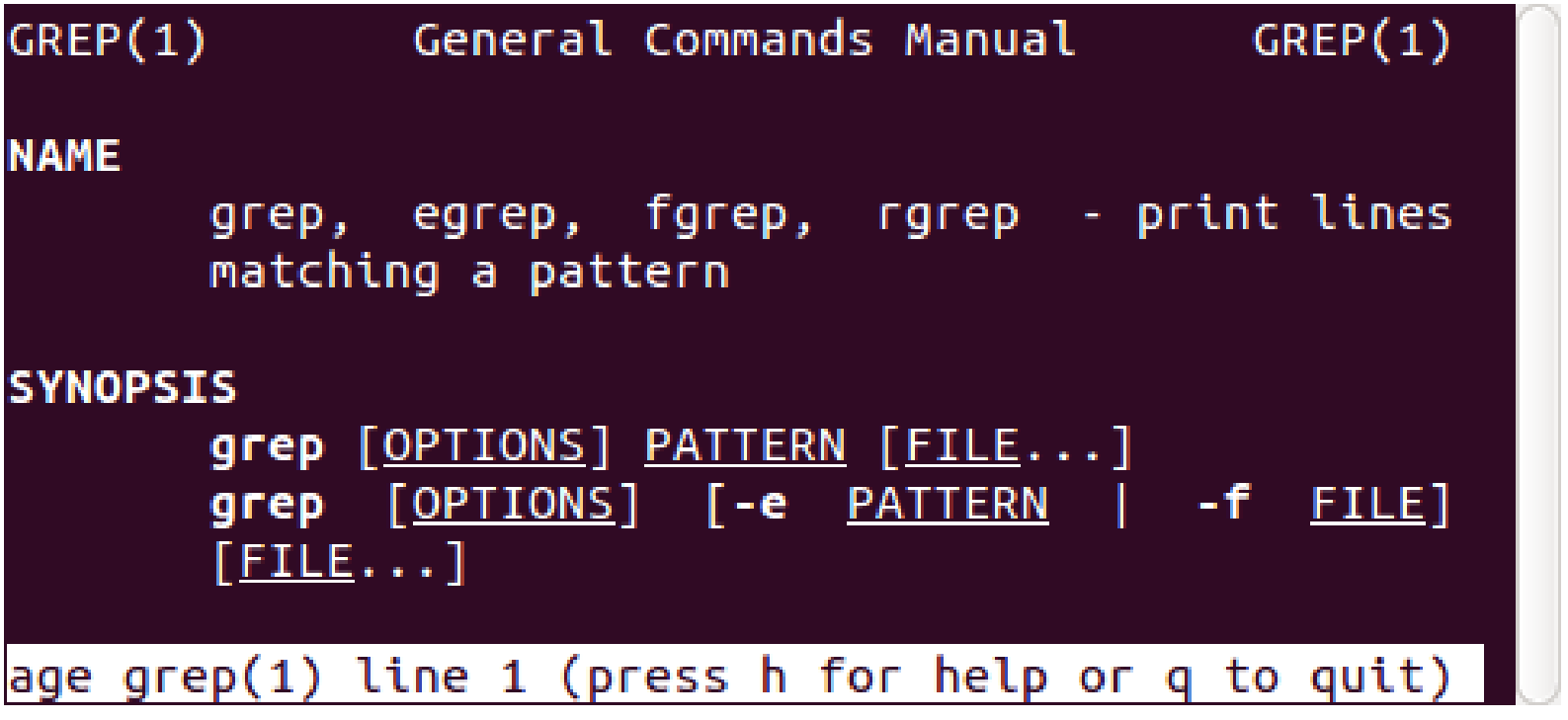
Use **`ls`** to see the files we've downloaded:

```
lionfish@atlas:~/teaching/apods$ ls
ap140901.html  ap140904.html  ap140907.html
ap140902.html  ap140905.html
ap140903.html  ap140906.html
lionfish@atlas:~/teaching/apods$
```

APOD Example

Step Two:

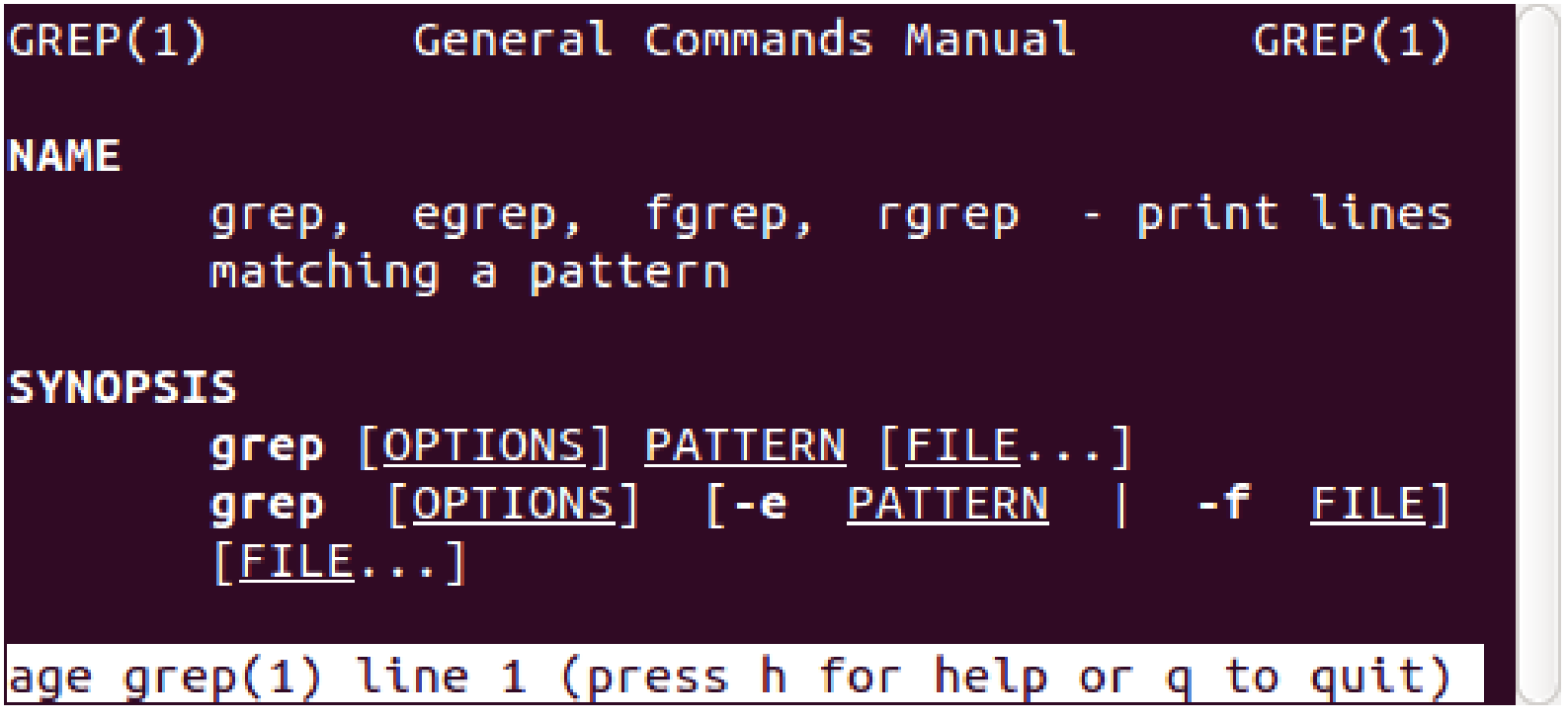
- We now need to search these files (using a regular expression!) to find how many links are to wikipedia.
- To do this we need to use grep.
  - Type in `man grep`



APOD Example

`grep [OPTIONS] PATTERN [FILE...]`

- `PATTERN` is something like `"<a href="` and `FILE` is all the files we downloaded. In my case this is all the files in the current directory, so I can just use `*`
- `grep "<a href=" *`



APOD Example

Step Three: Searching for wikipedia links

- We can modify the regex to be more specfic:

```
grep "<a href=\"[^\"]*wikipedia" *
```

APOD Example

Step Four: Searching for nasa links

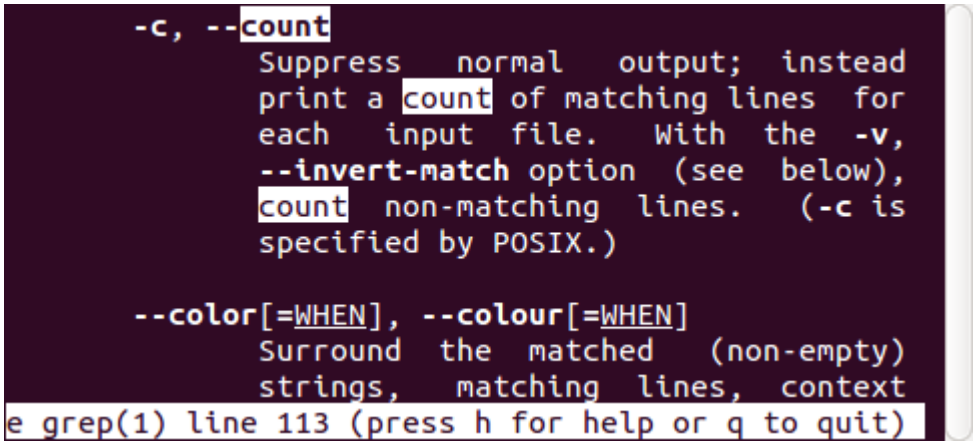
- We can modify the previous expression

```
grep "<a href=\"[^\"]*nasa" *
```

APOD Example

Step Five: Counting

- We need grep to return the number of times it's found the pattern. Look in the manual (man pages), with: `man grep`
- hint: You can search them by pressing / and typing in the search key. In this example we're searching for 'count'.



APOD Example

Step Five: Counting

- So we can just add -c to our grep command:
- `grep -c "<a href=\"[^\"]*nasa" *`

```
lionfish@atlas:~/teaching/apods$ grep -c "<a href=\"[^\"]*nasa" *
ap140901.html:7
ap140902.html:6
ap140903.html:6
ap140904.html:9
ap140905.html:6
ap140906.html:7
ap140907.html:7
lionfish@atlas:~/teaching/apods$
```

APOD Example

Step Five: Counting

- Not quite what we wanted!
- Quick hack, we can also pipe text into grep

```
cat * | grep -c "<a href=\"[^\"]*nasa"
```

- NASA gets: 48 links
- wikipedia gets: 14 links
- out of a total of: 233 links
- note: To use subexpressions and similar, need to use extended grep (`egrep`)

Packet sniffing

- Want to know what's happening on your internet connection?
- As an illustration of why encryption can be important, we consider the example.html page on the laptop server.

- Several tools to allow us to see the raw packet data. I've piped the output of one of these tools (tcpflow) to the **rawpacketdata.log** file.

```
tail -f rawpacketdata.log
```

Packet sniffing

- The form sends the password back in plaintext. If you look at the html code you'll see that this parameter is **`pwd`**.
- So we'll look for that in the stream of data...

```
tail -f rawpacketdata.log | grep 'pwd'
```

Homework!

- Look in the man page for `grep` (`man grep`), and find out how to make it ignore upper/lower case.
- Download the world page of the guardian's website using **`wget`**: http://www.theguardian.com/world
- Use the `-c` argument with grep to count the number of times the following words are on the page:
  - asia
  - africa
  - europe
  - calm
  - fight

Homework!

- Put the wget and the five grep commands all into a single bash script, call it **`checkworld.sh`**
- First line:

  `#!/bin/bash`
- Make it executable: **`chmod 755 checkworld.sh`**
- Run it with **`./checkworld.sh`**
- The output should be something like:

```
lionfish@atlas:~$ ./checkworld.sh
--2014-10-28 15:22:32--  http://www.theguardian.com/world
Resolving www.theguardian.com (www.theguardian.com)... 185.31.18.207
Connecting to www.theguardian.com (www.theguardian.com)|185.31.18.207|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 197002 (192K) [text/html]
Saving to: ‘world’

100%[======================================================================>] 197,002      653B/s   in 26s

2014-10-28 15:23:05 (7.37 KB/s) - ‘world’ saved [197002/197002]

10
16
8
0
4
lionfish@atlas:~$
```

Homework!

- Send me your checkworld.sh scripts:
  msmith@cit.ac.ug
- This is 5% of your course grade