CEG2722: Data Analysis II

Command Line Data Processing

- Lecture 3 : Obtaining Data -

Achraf Koulali

Geospatial Engineering

December 13, 2021

Newcastle University

Lessons from Practical 1

Running a command-line from the terminal

- Syntax: $ command options/arguments

```
$ cd dir
$ cp file1.txt file2.txt
$ grep word documentfile.txt
```

Read returnd errors

Running a command-line from the terminal

Example

```
$ more file1.txt
more: stat of file1.txt failed: No such file or directory
```

ls usage

When using `ls -l`, you get something similar to:

```
$ ls -l
total 8
-rw-rw-r-- 1 koulali koulali    0 Nov 26 15:11 file1.txt
drwxrwxr-x 2 koulali koulali 4096 Nov 26 15:11 test1
drwxrwxr-x 2 koulali koulali 4096 Nov 26 15:11 test2
describe each column
```

What's the difference between absolute and relative paths?

```
$ ls /home/user/data/
$ ls ./data/
```

Obtaining Data

At the end of this session you should be able to:

- Download data from the Internet
- Decompress files
- Extract data from spreadsheets
- Query relational databases
- Call web APIs

Obtaining data

The data used in this session is available in the `Examples/Session2/` directory:

```
$ cd ~/ceg2722/Examples/Session3/
$ ls
```

Downloading from the Internet

- The easiest way to use `curl` is to specify a URL as a command-line argument.
- Let's download the list of all Ordonance Survey (OS) GNSS network:

```
$ curl https://www.ordnancesurvey.co.uk/documents/resources/osnet-coordinates-file.txt
```

```
# 2020-11-06
# ***********************************************************************
# ***** OS Net coordinates were all updated on 00:00:00 2016-Aug-26 *****
# ***** Coordinates are now known as 'OS Net v2009'                *****
# ***** Previous coordinates are 'OS Net v2001' and in new section  *****
# ***********************************************************************
#
# Changes:
# 2020-11-06. Correction - Antenna at KING was changed from LEIAR25 to LEIAR20 on 2019-02-12
```

Saving data from `curl`

- By defaults `curl` prints the content on the terminal. To write in a local file (-O option):

```
$ curl https://www.ordnancesurvey.co.uk/documents/resources/osnet-coordinates-file.txt -O
```

- Or, we can use the redirect option (as we have shown in Lecture 2)

```
$ curl https://www.ordnancesurvey.co.uk/documents/resources/osnet-coordinates-file.txt >f1.txt
```

Decompressing Files

- Often, large datasets are distributed in a compressed format.
- Common file extensions of compressed archives are: .tar.gz, .zip, and .rar.
- The command line tools such as : `tar`, `gunzip` and `unrar` are used to decompress the archives.

Decompressing Files

- Using compressed files with `tar.gz` (pronounced as “gzipped tarball”) as an example.
- To extract the archive `logs_ex2.tar.gz`:

```
$ tar -xvzf logs_ex2.tar.gz
```

```
camb_20081001.log
easi_20090217.log
liar_20200930.log
nott_20080429.log
```

Decompressing Files

```
$ tar -xvzf logs_ex2.tar.gz
```

x → extract

v → verbose

z → gzipped file

f → archive file

Decompressing Files

- To uncrompress the files into a specified directory `logs`, we use the option `-C`:

```
$ mkdir logs
$ tar -xvzf logs_ex2.tar.gz -C logs
# to check the number of files uncrompressed in ./logs
$ ls ./logs/ | wc -l
```

Explore Data files

- Sometimes, we receive datsets with different extension types:
  - Excel: `.csv` or `.xlsx`...
  - ArcGIS: `.shp`, `.dbf`, ...
  - Point cloud: `.las`,...
- If we want to specify just a few files, we use wildcards:

```
$ cd Examples/Session3/rinex
$ ls *o
# this displays all files ending with the letter o
$ ls site00?.01o
# ? matches one character only
$ ls *.??n
# combining * and ??
```

Explore Data files

- “[]” matches exactly one of the chars (or range) in brackets

```
$ cd Examples/Session3/
$ ls site00[1-3].01o
# lists sites from day 001 to 003
$ ls site00[abc].01o
# lists files : site00a.01o  site00b.01o  site00c.01o
```

Explore Data files

- Sometimes we want to specify a particular list. → Brace expansion

```
$ ls -l doc.{inp,out,err}
-rw-rw-r-- 1 koulali koulali 0 Jul  7 15:03 doc.err
-rw-rw-r-- 1 koulali koulali 0 Jul  7 15:03 doc.inp
-rw-rw-r-- 1 koulali koulali 0 Jul  7 15:03 doc.out
```

- We can use a range of integers or characters:

```
$ ls site{001..008}.01o
# lists files from day 001 to 008
```

Test your knowledge

Quiz 3.1: GNSS orbit files are ditributed with the format `cccwwwwd.sp3`, where `ccc` is the IGS centre, `wwww` the GPS week and `d` is the day of the week.

- `cd` into `quiz3.1` directory, then list files from the centers: esa and gfz using expansion braces.

Displaying Files: cat, head and tail

cat displays the contents of a whole file (or several)

```
# e.g. to display the file text.txt
$ cd Session3
cat text1.txt
This is line 1
This is line 2
This is line 3
This is line 4
# e.g. displays the first 2 lines of the file text.txt
head -n 2 text1.txt
This is line 1
This is line 2
# e.g. displays the last 2 lines of the file text.txt
tail -n 2 text1.txt
```

Calling Web APIs

- Data can be accessible through the Internet in the form of API.
- API stands for **A**pplication **P**rogramming **I**nterface.
- web APIs often return data in a structured format, such as JSON or XML.

Web APIs are a way to strip away all the extraneous visual interface that you don't care about and get the data that you want.

Calling Web APIs

Example: The GNSS Interferometric Reflectometry API returns data in this format:

```
{"acknowledgement":"http://gnss-reflections.org",
"amp":8,
"archive":"unavco",
"azim1":0,
"azim2":360,
"createdAt":"2021-07-07 14:41:59Z",
...}
```

- For the station p038, the year 2020, day of year 135, we can display setup using:

```
# jq is a light JSON processor, which is not a standard Linux tool.
curl 'http://gnss-reflections.org/api?station=p038&year\
=2020&doy=135&archive=unavco&jsononly=True' | jq '.'
```

Creating scripts

- If you need to repeat the command-line tools on a regular basis ⟹ wraping one-liners into a script.
- During this course we will use the GNU “nano” editor (terminal based text editor).

To launch nano

```
$ nano
```

Creating scripts

- Example of a bash script to display the current date/time.

```bash
#!/bin/bash
# the line above is called shebang
# it instructs the system which executable to interpret the commands.
echo "The current date :"
date
```

Creating scripts

- Example of a bash script to display the current date/time.
- You need to make your script executable before running:

```
$ chmod +x myscrip.sh
# to run your script
$ ./myscript
current date:
Thu  8 Jul 10:26:46 BST 2021
```

Creating scripts

Adding arguments:

- Let's modify the previous script to add the format of the date:

```
#!/bin/bash
#
echo "The current date:"
date +$1
# $1 refers to the first argument
```

re-run with the date format year-month-day

```
$ ./myscript %Y-%m-%d
current date:
2021-07-08
```

Test your knowledge

Quiz 3.2: Write a script that displays the average daily weather report for a given location (argument).

To display the weather in your terminal use the command-line:

```
curl https://wttr.in/location*
```

Test your knowledge

Which wildcard represents all files?

1. all
2. "*"
3. “?”

Test your knowledge

What the "rm -f *" command-line does?

1. removes all files
2. removes all files and directories
3. removes all files with one character name

Test your knowledge

What the following command-line does?

```
cp /data/rinex/*/nslg*
```

1. copies all files starting with “nslg” in the rinex directory
2. copies all files starting with “nslg” in the all sub-directories of rinex dir.
3. copies all files in the system starting with nslg

Test your knowledge

Explain what the following command line does.

```
cp ./data/rinex/2014/03{2..8}/14{d,m,n,g}/nslg* ./qc/
```