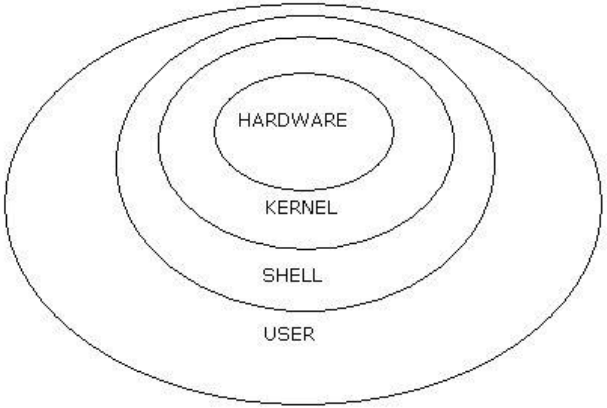
Shell Script

2019/02/25 @TsungHan

Outline

- Introduction to shell script
- Build a script
  - Variable
  - Flow control
  - Test
- Useful tools
  - regex
  - grep, cut, awk, sed
- Exercise : AAAAA+

Introduction to shell script



- Shell : The user interface to kernel
- There are a bunch of different CLI shells
  - zsh, bash ...
- Shell Scripts :
  - Small program that is run or executed by the Unix shell
  - Commands in the shell script are also programs
  - Support if-else, loop, function ...

Start to build a script !

```
#!/bin/env bash
# first line is shebang
# your code starts here
echo “Hello, world”
```

1. Open the editor and write down the lines on the left.

2. Save the code to a file (ex : hello.sh), and now you have two ways to run it.
   - Run `sh hello.sh`
   - Give file execution permission and run `./hello.sh`

Variable

```
# assign value
course="nasa"
## no space between "="

# take value
echo "$course"
echo "{$course}-2019"
## {} use to specify var
```

```
# command
all_dir=$(ls .)
all_dir=`ls .`

# Warning !!!
echo "I am at $(pwd)"
echo 'I am at $(pwd)'
## They are not the same
```

Something you need to know

Variable

```
# args
echo $#  # number of args
echo $0  # command itself
echo $1, $2, $3 # N-th args
```

```
# variable as number
a=1
b=$((2))
c=$a+$b       ## c = 1+2
d=$(($a+$b)) ## d = 3
```

Flow Control

```
# if-else
if condition; then
    commands
elif condition; then
    commands
else
    commands
fi
```

```
# case
case VARIABLE in
    pattern1)
        commands
        ;;
    pattern2)
        commands
        ;;
esac
```

Flow Control

```
# for
for VARIABLE in 1 2 3 4 5 .. N;
do
    commands
done

for VARIABLE in $(COMMAND);
do
    commands
done
```

```
# while
while condition-A; do
    if condition-B; then
        continue
    else
        break
    fi
    commands
done
```

Test

```
# test if file exists
if [ -f FILENAME ];
then
    processing...
fi
## Usage : [ <Expression> ]
```

```
# test if args >= 3
if [ $# -ge 3 ];
then
    processing...
fi
```

See full rules on the lecture notes.

Useful Tools

- regex
- grep
- cut
- awk
- Sed
- Others

Regular Expressions

A regular expression (regex) is an expression describing some form of texts.

A simple word such as “apple” is actually a valid regex.

There are many regexs, and sed uses Basic Regular Expression (BRE). You can freely combine regex to match almost everything!

BRE syntax

Expression	Description
a single character, such as a, b, or c	matches itself
*	matches zero or more instances of previous regex
.	matches any character
^	matches the beginning of the pattern space, such as the beginning of a file
$	like ^, but matches the end
[list]	matches any of the character in the list
[^list]	matches any of the character not in the list

BRE syntax

Expression	Description
\+	like *, but matches one or more
\?	like *, but matches zero or one
\{i\}	matches exactly i times ($0 \leq i \leq 255$)
\{i,j\}	matches i to j times (inclusive)
\{i,\}	matches more than or equal to i times

regex - example

```
‘a\{3\}b’ -- matches ‘aaab’
‘[aeiou]’ -- matches vowels
‘.*’      -- matches any string
‘$’       -- matches the end of string
‘\$’      -- matches a dollar sign, ‘\’ is for ‘escaping’
‘\\$’     -- matches strings ending with a backslash
```

grep - print lines matching a pattern

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

# options:
-A, B, C : number of lines after / before context
-n : prints line number
-o : prints only the matching part of the lines
-e : regex pattern
-v : select lines those are not matching
...
```

grep - example1

```
[root@nasa-hw0 ~]# ps aux | grep -v root | grep beast
b079021+  5033  0.0  0.0   4216    36 pts/43   S+    2月15   0:08 ./beast
b067050+ 27678  0.0  0.0   4216    40 ?        S     2月17   0:03 ./beast
```

grep - example2

```
[root@mx1 ~]# mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-------
4A92730BE3DC    11773 Tue Feb 19 02:40:02
Return.EID406046f7.Job@news.cite.com.tw
      (connect to jan.csie.ntu.edu.tw[140.112.90.77]:25: Connection timed
out)
                                                joller@jan.csie.ntu.edu.tw

...
```

grep - example2

```
[root@mx1 ~]# mailq  | grep "^[A-Z0-9]\{10\}"
3406D30D3DD2*   30179 Thu Feb 14 20:56:13  MAILER-DAEMON
EED41308A7AE*  797749 Tue Feb 19 11:21:55  keke432111@gmail.com
4322B30D3DAC     9499 Thu Feb 14 22:05:13  MAILER-DAEMON
4CDBA30D3DD3     5686 Tue Feb 19 09:31:01  epaper@most.gov.tw
...
```

cut - remove sections from each line of files

```
$ cut OPTION... [FILE]...

# options
-c      : specifies character positions
-d, -f  : cut by delimiter and select f-th one
```

cut - example1

```
# uptime
[root@nasa-hw0 ~]# uptime
 20:48:26 up 31 days,  7:05, 22 users,  load average: 1.51, 1.24, 1.15
```

```
# my uptime
[root@nasa-hw0 ~]# echo "load average: \
> $(cat /proc/loadavg | cut -d ' ' -f 1-3)"
load average: 1.51, 1.24, 1.15
```

Last Update: Wed Feb 20 11:12:53 CST 2019

*: Reserved for daily use. DO NOT RUN COMPUTING-INTENSIVE JOBS!

Host	CPU (%)	1m Load	5m Load	15m Load	Free Mem (MB)	Swap (%)	tmp2 Free	Uptime (days)	Users
bsd1	1	0.29	0.27	0.20	13465	0.25	67.1 GiB	7	35
linux1	1	0.74	0.55	0.47	55597	0.99	748.1 GiB	7	29
linux2	0	0.48	0.39	0.31	63224	0.28	792.2 GiB	7	9
linux3	3	1.30	1.38	1.42	57511	0.61	777.3 GiB	7	4
linux4	3	3.28	2.98	2.99	50949	3.54	792.2 GiB	7	7
linux5	36	9.29	9.89	9.54	66500	48.05	692.7 GiB	7	3
linux6	2	0.24	0.24	0.23	126906	0.00	2.0 TiB	7	5
linux7	49	6.80	7.35	8.26	72642	18.92	930.4 GiB	7	7
linux8	2	0.24	0.26	0.27	127478	0.00	943.4 GiB	7	3
linux9	32	7.16	7.18	7.18	8263	0.31	2.0 TiB	7	4
linux10	18	6.70	8.13	8.12	70547	33.68	2.0 TiB	7	3
linux11	14	3.19	3.96	5.40	85551	17.29	938.2 GiB	7	24
linux12	16	1.97	2.51	2.76	72491	97.55	2.0 TiB	7	16
linux13	26	6.32	6.35	6.26	37874	4.68	2.0 TiB	7	26
linux14	13	5.69	5.52	5.59	89218	10.12	2.0 TiB	7	1
linux15	4	0.39	0.57	0.60	68785	0.42	2.0 TiB	7	7
oasis1 (*)	12	0.09	0.15	0.17	6965	0.00	60.1 GiB	8	25
oasis2 (*)	16	0.39	0.30	0.28	5695	0.36	34.2 GiB	8	9
oasis3 (*)	0	0.00	0.00	0.00	0	0.00	0	0	0

monitor.csie.ntu.edu.tw

cut - example2

```
[root@nasa-hw0 big_rock]# ls -al | cut -c 1-10
總計 16
dr-xr-xr-x
dr-xr-xr-x
-r-xr-xr-x
-r-xr-xr-x
...
```

awk - pattern scanning and processing language

```
$ awk '{AWK language}'


# basic usage
awk '{print $1}'
awk -F ':' '{print $1 $6}' ## -F : specify delimiter
```

awk - example

```
[root@nasa-hw0 ~]# cat /etc/passwd | awk -F ":" '{print $1":"$6}'
...
piepie01:/home/NASA{D3sc3nd4n7_0f_G0d}/piepie01
b04505009:/home/NASA{D3sc3nd4n7_0f_G0d}/b04505009
b05701244:/home/NASA{D3sc3nd4n7_0f_G0d}/b05701244
b05609053:/home/NASA{D3sc3nd4n7_0f_G0d}/b05609053
...

# same as 'cut -d : -f 1,6'
```

sed - stream editor for filtering and transforming text

```
$ sed [OPTION] [SCRIPT]...

# Try to read the manual by yourselves!
# You can find it here:
# https://www.gnu.org/software/sed/manual/sed.html#sed-commands-list
```

sed - example1

```
# print a.txt with line 2 to 5 deleted
cat a.txt | sed ‘2,5d’

# print a.txt with ‘apple’ inserted next to line 2 (on line 3)
cat a.txt | sed ‘2a apple’

# like the previous but before line 2
cat a.txt | sed ‘2i apple’
```

sed - example2

```
# delete all bugs in the file
sed -i 's/[bB][Uu][Gg][Ss]//g' SINKING_SHIP


# replace all ‘aab’ to ‘bba’
cat a.txt | sed ‘s/a\{2\}b/bba/g’

# replace all strings starting with a ‘#’ to empty lines
cat a.txt | sed ‘s/#.*$//g’
```

Learn more about sed

Others

```
# Basic commands
ls, pwd, cd
chmod, chown
ps, top, kill
rm, mv, cp, mkdir, touch
cat, less, head, tail, echo, printf
...

# Advanced commands
https://hacker-tools.github.io
```

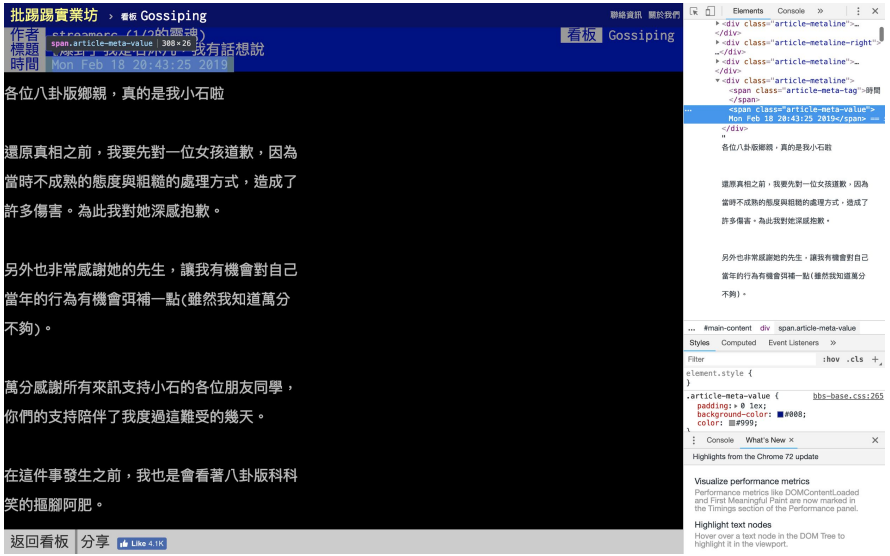
Exercise - AAAAA+

- Crawl some ptt articles which is published on 2/19
- Calculate the number of AAAAA\+ (5 or more consecutive uppercase A) in these articles' html file
- List the # of AAAAA\+, url and author

Exercise - AAAAA+

Hints :

- Find all articles’ urls from the article lists (“/bbs/Gossiping/M.***.A.***.html”)
- Check whether “Tue Feb 19” is in the article
- You might need to know how to add new element into shell array (ref)



Exercise - AAAAA+

```
# 1. Download the material
$ wget https://www.csie.ntu.edu.tw/~patrickwu2/parser.sh

# 2. Finish all TODO in the incomplete shell script
for i in {1..2};
do
    search_url="https://www.ptt.cc/bbs/Gossiping/search?page=$i&q=我是石沐凡，我有話想說"
    search_result=`curl --cookie "over18=1" $search_url --silent`
    # TODO 1 : get all articles' url
    # Hint   : lookup all "/bbs/Gossiping/......"
...
```

Exercise - AAAAA+

```
# 3. Test Correctness
$ sh parser.sh > output.txt

$ cat output.txt

有關 { [爆卦] 我是石沐凡，我有話想說 , 2/19 } 的相關搜尋結果如下：
0   https://www.ptt.cc/bbs/Gossiping/M.1550550269.A.253.html asd95955 (Znus)
0   https://www.ptt.cc/bbs/Gossiping/M.1550549737.A.558.html chadcooper (還在尋找穩健的下一步)
0   https://www.ptt.cc/bbs/Gossiping/M.1550548736.A.053.html ODFans (只發中肯文)
...
0   https://www.ptt.cc/bbs/Gossiping/M.1550505910.A.536.html ejijojo (小羊羊)
0   https://www.ptt.cc/bbs/Gossiping/M.1550505690.A.84F.html NavyWind (NavyWind)
共有 177 個 AAAAA
```

Other Information

- Lab 2 補 demo 表單
- TA Hour
  - 3/4 13:00 ~ 14:00 @R217
  - 3/7 11:00 ~ 12:00 @R217