Text parsing

Tutor: Lưu Thanh Trà

Email: lttra@hoasen.edu.vn

Outline

- Shell variable
-

grep

- grep –h printf *.c
  - -h: do not show the filename
- grep –c printf *.c
  - Count the number of lines containing the key in each file
- grep –l printf *.c
  - Print the filename only
- grep –q findsomething datafile > /dev/null
  - $? = 0 if found
- grep –i something file
  - Case in-sensitive
- grep –v donotprintme *.c
  - -v ignore the line containing a word

Some complex patterns

- “.” : matches any characters
  - grep abc.xyz
- * : repeat zero or more occurrences of the previous character
  - grep A*
- .* :
  - grep .*
- ^   : at the beginning/end
  - grep ^abc file
- $: at the end
  - grep abc$ file

- [AaBbCcij]: any character in the list
- [^AaBbCcij]: none of character in the list
- \{n\}: n number of repetition
- \{n,m\}: n to m number of repetition
  - grep ‘[0-9]’\{3\}-\{0,1\}[0-9]\{2\}-\{0,1\}[0-9]\{4\}’

  ⇒ match 123-45-6789

awk

- `awk '{print $1}' myinput.file`
- `awk '{print $1}' < myinput.file`
- `cat myinput.file | awk '{print $1}'`
- `ls -l | awk '{print $1, $NF}'`
  - NF: number of parameters
  - `$NF`: the last parameter

- awk '{

  ```
  > for (i=NF; i>0; i--) {
  > printf "%s ", $i;
  > }
  > printf "\n"
  ```

  - }‘
- Summing the fifth field of the output
  - $ ls -l | awk '{sum += $5} END {print sum}'

  => END: run the command just one time when the program ends

- Count the number of files owned by various users

```
#pro.awk
NF > 7 {
user[$3]++ }
END {
for (i in user)
{ printf "%s owns %d files\n", i, user[i] }}
```

- Output:

```
$ ls -lR /usr/local | awk –f pro.awk
bin owns 68 files
albing owns 1801 files
root owns 13755 files
man owns 11491 files
```

An history application with awk

```
#
# cookbook filename: hist.awk
#
function max(arr, big)
{
big = 0;
for (i in user)
{
if (user[i] > big) { big=user[i];}
}
return big
}
NF > 7 {
user[$3]++
}
END {
# for scaling
maxm = max(user);
for (i in user)
{
scaled = 60 * user[i] / maxm ;
printf "%-10.10s [%8d]:", i, user[i]
for (i=0; i<scaled; i++) {
printf "#“;
}
printf "\n“;
}}
```

- Result of the application

```
$ ls -lR /usr/local | awk -f hist.awk
bin          [ 68]:#
albing [ 1801]:#######
root   [ 13755]:##################################################
man    [ 1491]:##########################################
$
```

Sort

- sort file1.txt file2.txt myotherfile.xyz
- *somecommands* | *sort*
- sort
  - -n: sort number
  - -f : ingnorecase
  - -r : reverse order

- sort with “uniq –c”
  - cut -d':' -f7 /etc/passwd | sort | uniq -c | sort -rn

```
20 /bin/sh
10 /bin/false
2 /bin/bash
1 /bin/sync
```

- IP address
  - $ sort -t. -n +3.0 ipaddr.list
  - 10.0.0.2
  - 192.168.0.2
- $ sort -t . -k 1,1n -k 2,2n -k 3,3n -k 4,4n ipaddr.list
  - 10.0.0.2
  - 10.0.0.5
  - 10.0.0.20
  - 192.168.0.2
  - 192.168.0.4
  - 192.168.0.12