

CS 344 – UNIX OS

Fundamentals – Lecture #10

Purushotham Bangalore
Department of Computer and Information
Sciences
University of Alabama at Birmingham (UAB)

“echo” command – miscellaneous

- Using " " and ' ' with echo

```
$ echo 'Hello $USER'
```

Hello \$USER

```
$ echo "Hello $USER"
```

Hello puri
- When the shell interprets ' it stops interpretation until a matching ' is found
- To eliminate newline with echo use

```
echo -n <text>
```

echo -n "Enter your choice: "

“getopts” function

- Used to retrieve options and option-arguments from a list of parameters
- Usage: `getopts options variable`
- Example: `while getopts abc option`
- `getopts` function examines the argument list for options following a dash and then assigns the first option that matches to the variable specified
- If one of the options requires an argument, it must be followed by a colon (e.g., `getopts a:b:c option`)
- If no matching option is found, the variable specified will be set to `?` character

4/11/2005

3

“getopts” example

```
$ cat getopts.sh
while getopts abo: c
do
  case $c in
    a | b) echo "Option selected is $c" ;;
    o)     echo "Option selected is $c and it's arguments are $OPTARG" ;;
    \?)   echo $USAGE; exit 2;;
  esac
done
$ ./getopts.sh -o xxx,z,yy -b -a filename
Option selected is o and it's arguments are xxx,z,yy
Option selected is b
Option selected is a
$ ./getopts.sh -abo "xxx,z,yy" filename
Option selected is a
Option selected is b
Option selected is o and it's arguments are xxx,z,yy
```

Note: You can use `*` as the default value in case statements

4/11/2005

4

Using “sed”

- To quit after matching a specified line number – sed '20 q' filename (first 20 lines are printed on the screen)
- To quit after the first matching of a given pattern – sed '/pattern/ q' filename
- To delete specific lines – sed '1,10 d' filename (lines 1-10 are not displayed)
- To delete lines that match a pattern – sed '/pattern/ d' filename
 - sed '/xyz/ d' filename
 - sed '3,/abc/ d' filename
- Regular expressions can also be used as patterns for quit and delete
 - sed '/^[A-Z].*[0-9]\$/ q' filename
 - sed '/^\$/ d' filename

4/11/2005

5

Using “sed”

- To replace the first instance of a specified pattern – sed 's/abc/ABC/' filename
- To replace multiple instances of a specified pattern – sed 's/abc/ABC/g' filename
- To suppress output from sed use –n option
 - sed –n '/pattern/' filename (only the first matching line is displayed)
 - sed –n '3,6 p' filename (print lines 3 through 6, without –n option all lines will be printed once while lines 3-6 will be printed twice)
 - sed –n '1,15 s/abc/ABC/ p' filename (the first line between 1 and 15 that matches the pattern will be printed) **[Linux Only]**

4/11/2005

6

Using “sed”

- To print lines that do not match use !p
 - sed -n '/^[0-9]/ !p' filename (don't print lines starting with a number)
 - sed -n '/^\$/ !p' filename (don't print blank lines)
- Complex substitution using contextual addresses **[Linux Only]**
 - sed -n '/pattern1/ s/pattern2/replace/g p' file
 - sed -n '/[1-5]/ s/abc/ABC/g p' filename
 - sed -n '5,/abc/ s/xyz/XYZ/' filename

4/11/2005

7

Reading and Writing Files from sed

- To read in a file at a specified location
 - sed '5 r inputfile' filename (insert inputfile after line 5 in file “filename”)
 - sed '/abc/ r inputfile' filename (after each line that contains “abc” insert the file “inputfile”)
- To write specific lines to another file
 - sed '1,10 w outfile' filename (write lines 1-10 from file “filename” to the file “outfile”)
 - sed '/^[a-zA-Z]/ w outfile' filename (write lines that start with a alphabet to the file “outfile”)

4/11/2005

8

Passing Multiple Instructions to sed

- To pass multiple instructions to sed on the command line use `-e` option
 - `sed -e 's/abc/ABC/' -e 's/xyx/XYZ/' filename`
 - `sed -e '/^[a-z]/ p' -e 's/[0-9]/XXX/g p' file`
- We can also include the different instructions in a separate file and specify this commands file as input to sed with `-f` option
 - `sed -f optscript filename`

```
$ cat > optscript  
s/root/ROOT/g  
s/\bin\csh/\bin\tcsh/g  
$ cat /etc/passwd | sed -f optscript
```

4/11/2005

9

Insert/Append Text with sed

- To insert text before a specified line use
 - `sed '5 i some text' filename`
 - `sed '/abc/i some text' filename`
- To insert text after a specified line use
 - `sed '5 a some text' filename`
 - `sed '/abc/a some text' filename`
- When using a script file as input to sed, the script file should have

```
/abc/i  
Text To Be Inserted Before The Line  
/abc/a\  
Text To Be Appended After The Line
```

4/11/2005

10

“find” utility

- Used to search for files in a directory hierarchy
- Usage: file path expression
- Examples:
 - Print entire directory hierarchy from current directory:
find . or find . -print
 - Print files with write access for other:
find . -perm -o+w
 - Print all file names in the current directory and below, but skip CVS directories:
find . -name CVS -prune -o -print
 - Remove all files in your home directory named a.out or *.o that have not been accessed for a week:
find \$HOME \(-name a.out -o -name '*.o' \) \
-atime +7 -exec rm {} \;

4/11/2005

11

gzip and tar utilities

- To compress files use gzip
- Usage: gzip filename (a file filename.gz is created)
- To expand a compressed file use gunzip
- Usage: gunzip filename.gz (a file filename is created)
- To create an archive use tar
- Usage: tar cvf archivename directory/files
- To expand the archive use tar xvf archive
- “tar” utility can be used to create a compressed archive, use tar cvfz archivename files

```
$ tar cvfz examples.tar.gz examples
$ file examples.tar.gz
examples.tar.gz: gzip compressed data, from Unix
$ cp examples.tar.gz /tmp/
$ cd !$
$ gunzip -c examples.tar.gz | tar xvf -
```

4/11/2005

12