Lecture 4: BASH Prog. Cont

- Command substitution allows the standard output of a command to replace the command name. There are two forms:  `COMMAND` or `$(COMMAND)`.

```bash
rm `cat filelist`  # rm files in a list
txtfiles=$(ls *.txt) # capture ls display
userpass=$(getent passwd | grep '^ar63:')
passwd=$(cat /etc/passwd)  # capture file
```

Lecture 4: BASH Prog. Cont

- Arithmetic Substitution

```bash
b=12
let a=b*3      # a is 36
((a = b * 3))   # alternative to let
val=$((b**2))   # val is 144

count=0
while [ $count -le 10 ]
do
    # do stuff here
    ((count=count + 1)) # or ((count++))
done
```

Lecture 4: BASH Prog. Cont

- Here Documents

```bash
# Example with variable substitution
sftp $user@csserver <<SFTPINPUT
cd $remotedir
get $file
bye
SFTPINPUT
```

Lecture 4: BASH Prog. Cont

- Functions and Aliases

```bash
Functions only have _____ scope by default, but can be exported.

error() { echo "$*" > /dev/stderr ; }

You can create aliases for commands too:

alias ls='ls --color=tty'
alias ll='ls -l --color=tty'
alias rm='rm -i'
alias \
ls   # force rm to prompt
alias   # display aliases
\ls   # run unaliased ls
```
Lecture 4: BASH Prog. Cont
Useful Commands

- **find** will locate files and directories that satisfy certain criteria. It _________ through directories.

  ```bash
  find . -name ‘*.txt’ # all *.txt
  find . -mtime -1 # modded in last day
  # Use grep to search for strings
  # print is required here
  find . -exec grep -iq "ar63" \{} \; -print
  # Change perms on all files
  find . -type f -exec chmod 644 \{} \;
  ```

- **exec** replaces the current process with a new one (often used at the end of a script). You can also use it from within a script to redirect standard input/output/error.

  ```bash
  exec vi # “become” vi
  exec > output.txt
  echo hello there # sent to output.txt
  ```

- **bc** is an arbitrary precision ___________.

  ```bash
  # get pi to 40 places
  pi=$(echo 'scale=40; 4*a(1)' | bc -l)
  ```

- **grep** search for files containing _________ expressions (see pgs 66-68).

- **xargs** takes filenames from standard input and feeds the names to a command.

  ```bash
  ls | xargs -p -1 gzip
  ```

- **expr** is an all-purpose expression evaluator

  ```bash
  a=$(expr 5 + 3) # arithmetic
  b=$(expr length $filename) # string length
  ```

Lecture 4: BASH Prog. Cont
Useful Commands

- **export** makes a shell variable available in ____________.

  ```bash
  export myhome=/home/richardson
  ```

- **read** will read a line from standard input.

- **printf** can give fancier output than **echo**.

  ```bash
  printf “%8.4f\t\t%f\n” $val $myhome
  ```

Lecture 4: BASH Prog. Cont
Grouping

- Use the following code to read lines of input from a ____________:

  ```bash
  while read line
do
    # Do something with $line
  done < inputfile > outputfile
  ```

- Input (and output) is redirected for all commands inside the body of the while.
Lecture 4: BASH Prog. Cont

Grouping

• With () the command group is executed in a subshell (another bash process). If you set or change any variables those changes will not be seen in the __________ process.

```bash
myvar=abc
{ myvar=xyz; }
echo $myvar  # Will display xyz
( myvar=mno)
echo $myvar  # Will display xyz
```

In-Class Exercises

Lecture 4: BASH Prog. Cont

• Write a shell script named `sortusers1` that displays all system users that use a particular shell in alphabetical order. Display the username only. (Use the command `getent passwd` to generate the list of all users.)

Example calls:
```
sortusers1 /bin/bash
sortusers1 /bin/tcsh
sortusers1 /usr/sbin/nologin
```

13

Lecture 4: BASH Prog. Cont

In-Class Exercises

• Write a shell script named `sortusers2` that displays all system users that use a particular shell in order by user id. Display username followed by the user id.

Example calls:
```
sortusers2 /bin/bash
sortusers2 /bin/tcsh
sortusers2 /usr/sbin/nologin
```

15

Lecture 4: BASH Prog. Cont

In-Class Exercises

• Write a shell script named `revargs` that displays its arguments in reverse order. (The `eval` command may prove useful.)

16