

Lab 2

Echo

What are the differences among the following commands? Explain with screenshots.

a) `echo cal` prints cal

```
gpolyak@cscd-linux01:~$ echo cal
cal
gpolyak@cscd-linux01:~$
```

b) `echo $(cal)` prints the current calendar variable

```
gpolyak@cscd-linux01:~$ echo $(cal)
January 2018 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31
gpolyak@cscd-linux01:~$
```

c) `echo $cal` prints the contents of the variable named cal

```
gpolyak@cscd-linux01:~$ echo $cal
gpolyak@cscd-linux01:~$
```

d) `echo "$(cal)"` prints the current calendar, but formatted

```
gpolyak@cscd-linux01:~$ echo "$(cal)"
January 2018
Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
gpolyak@cscd-linux01:~$
```

e) `echo `cal`` prints the current calendar variable

```
gpolyak@cscd-linux01:~$ echo `cal`
January 2018 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31
gpolyak@cscd-linux01:~$
```

f) `echo `echo `cal``` prints a variable that has been assigned to the output of `echo `cal``

```
gpolyak@cscd-linux01:~$ echo `echo `cal``
January 2018 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31
```

Environment Variables

What command will show you all the environment variables? What command will display the environment variable named PATH? Show both with screenshot.

`env`

```
gpolyak@cscd-linux01:~$ env
TERM=xterm-256color
SHELL=/bin/bash
XDG_SESSION_COOKIE=65f8860d8b52bdc2ca29c8e56eae1a1-1516910225.205246-2054022413
SSH_CLIENT=10.104.177.77 48366 22
SSH_TTY=/dev/pts/1
USER=gpolyak
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:
su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.t
az=01;31:*.lzh=01;31:*.lma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:
.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31
:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cp
o=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35
:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.sv
gz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=0
1;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:
.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;
35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.o
gx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00
;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga=00;36:*.spx=00;36:*.xspf=00;36:
TMOUT=1200
MAIL=/var/mail/gpolyak
PATH=/usr/local/java/bin:/usr/local/java/jre/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sb
in:/bin:/usr/games:/usr/local/games
PWD=/home/EASTERN/gpolyak
JAVA_HOME=/usr/local/java
LANG=en_US.UTF-8
KRB5CCNAME=FILE:/tmp/krb5cc_900746857_R8KitA
SHLVL=1
HOME=/home/EASTERN/gpolyak
LOGNAME=gpolyak
SSH_CONNECTION=10.104.177.77 48366 146.187.134.22 22
LESSOPEN=| /usr/bin/lesspipe %s
LESSCLOSE=/usr/bin/lesspipe %s %s
_/usr/bin/env
gpolyak@cscd-linux01:~$
```

echo \$PATH

```
gpolyak@cscd-linux01:~$ echo $PATH
/usr/local/java/bin:/usr/local/java/jre/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sb
in:/usr/games:/usr/local/games
gpolyak@cscd-linux01:~$
```

I/O

What are the differences among the following commands. Explain with examples and Screenshot.

cat waits for input

```
gpolyak@cscd-linux01:~$ cat
^C
gpolyak@cscd-linux01:~$
```

cat < filename prints the contents of filename

```
gpolyak@cscd-linux01:~$ cat < filename
-bash: filename: No such file or directory
gpolyak@cscd-linux01:~$ ls
Assignment CSCD240 Documents Music Pictures Templates
Assignment1 Desktop Downloads netstorage Public Videos
gpolyak@cscd-linux01:~$ cat >filename
^C
gpolyak@cscd-linux01:~$ ls
Assignment CSCD240 Documents filename netstorage Public Videos
Assignment1 Desktop Downloads Music Pictures Templates
gpolyak@cscd-linux01:~$ cat < filename
cat: filename: No such file or directory
gpolyak@cscd-linux01:~$
```

cat > filename creates or overwrites filename with user input

(see above screenshot)

cat >> filename appends user input to filename

```
gpolyak@cscd-linux01:~$ cat >> filename
this line will be appended to filename
^C
gpolyak@cscd-linux01:~$ cat < filename

this line will be appended to filename
gpolyak@cscd-linux01:~$
```

Write a command that counts the total number of lines the string “bird” exists in a file named “The Rhyme of Ancient Mariner” in your current directory.

grep -c “bird” “The Rhyme of Ancient Mariner”

Write a command that searches the string “line” in all .c and .txt files starting from your current directory and all sub directories.

find -name “*.c” -or -name “*.txt” | xargs grep “line”

Metacharacters in Regular Expression

What will the following patterns match? Explain.

a) ^bags\$

Only 1 exact match: bags

b) ^...\$

Any 3 characters

c) l.g

Includes an l followed by any character, followed by a g

d) ^\.

Anything that starts with a literal .

Grep/Find/Pipe

Consider the following file named “FruitsList.txt”. Try the following commands and explain each output with screenshot.

a) grep “[A-Z]e” FruitsList.txt

Searches in the file for a line that contains a capital letter followed by an e

```
gpolyak@cscd-linux01:~$ grep "[A-Z]e" FruitsList.txt
gpolyak@cscd-linux01:~$
```

b) grep -i “[A-Z]e” FruitsList.txt

Same as the above command except that this ignores case

```
gpolyak@cscd-linux01:~$ grep -i "[A-Z]e" FruitsList.txt
apple
Orange
Pineapple
lemon
berry
gpolyak@cscd-linux01:~$
```

c) `grep "[^A-Z]e" FruitsList.txt`

Searches the file for lines containing a lowercase letter followed by e

```
gpolyak@cscd-linux01:~$ grep "[^A-Z]e" FruitsList.txt
apple
Orange
Pineapple
lemon
berry
gpolyak@cscd-linux01:~$
```

d) `grep -i "^[A-Z]e" FruitsList.txt`

Searches the file for lines that start with a letter followed by e

```
gpolyak@cscd-linux01:~$ grep -i "^[A-Z]e" FruitsList.txt
lemon
berry
gpolyak@cscd-linux01:~$
```

e) `grep "^le" FruitsList.txt`

Searches the file for lines that start with le

```
gpolyak@cscd-linux01:~$ grep "^le" FruitsList.txt
lemon
gpolyak@cscd-linux01:~$
```

f) `grep "le$" FruitsList.txt`

Searches the file for lines that end with le

```
gpolyak@cscd-linux01:~$ grep "le$" FruitsList.txt
apple
Pineapple
gpolyak@cscd-linux01:~$
```

Suppose you are in your home directory. What are the differences between the following commands? Explain with screenshot.

`find . -name "*.txt"`

Recursively finds and prints the paths of every file whose name ends with .txt starting from root

```
gpolyak@cscd-linux01:~$ find . -name "*.txt"
.
./.bash_history
./.dmrc
./.xsession-errors
./.AMD
./.AMD/GLCache
./.AMD/GLCache/264b93aebdcecf_13.bin
./.AMD/GLCache/5dec8f3a46c2b5b6_13.bin
./.AMD/GLCache/4f8fc111a7d8117_13.idx
./.AMD/GLCache/cd93fc235638ede0_13.bin
./.AMD/GLCache/d5806ae70c8210b1_13.idx
./.AMD/GLCache/5dec8f3a46c2b5b6_13.idx
./.AMD/GLCache/cd93fc235638ede0_13.idx
./.AMD/GLCache/4f8fc111a7d8117_13.bin
./.AMD/GLCache/d5806ae70c8210b1_13.bin
./.AMD/GLCache/264b93aebdcecf_13.idx
./.ICEauthority
./.bash_logout
./.gnome2
./.gnome2/acceles
./netstorage
./Desktop
./CSCD240
./CSCD240/Assignment
```



```
find ~ -name "*.txt"
```

Recursively finds and prints the paths of every file ending in .txt starting from home directory

```
gpolyak@cscd-linux01:~$ find ~ -name "*.txt"
/home/EASTERN/gpolyak/CSCD240/calendar2017.txt
/home/EASTERN/gpolyak/.mozilla/firefox/v9cwejet.default/SiteSecurityServiceState.txt
/home/EASTERN/gpolyak/.mozilla/firefox/v9cwejet.default/revocations.txt
/home/EASTERN/gpolyak/.cache/tracker/miner-applications-locale.txt
/home/EASTERN/gpolyak/.cache/tracker/db-version.txt
/home/EASTERN/gpolyak/.cache/tracker/last-crawl.txt
/home/EASTERN/gpolyak/.cache/tracker/db-locale.txt
/home/EASTERN/gpolyak/.cache/tracker/first-index.txt
/home/EASTERN/gpolyak/FruitsList.txt
gpolyak@cscd-linux01:~$
```

Write a command that finds all text files in your home directory and subdirectory and shows the long listing.

```
find ~ -name "*.txt" | xargs ls -l
```

What will the following commands do? Explain with screenshots.

a) `ls -l | grep '^.....rw'`

Takes the output of `ls -l` and uses that to search for files that can be read and written to by users not included in the group and/or not included as an owner

```
gpolyak@cscd-linux01:~$ ls -l | grep '^.....rw'
-rwxrwxrwx 1 gpolyak IT-GenericLinuxGroup 15 Jan 18 09:47 netstorage -> /mnt/ns-gpolyak
gpolyak@cscd-linux01:~$
```

b) `grep -n variable *. [ch]`

Looks for a file in the current directory that has the string “variable” in the name and ends with either .c or .h

```
gpolyak@cscd-linux01:~$ grep -n variable *. [ch]
grep: *. [ch]: No such file or directory
gpolyak@cscd-linux01:~$
```

Processes and Jobs

What is process? How will you differentiate processes from jobs?

A process is any running program with its own address space. A job has to be interactively started and might not have its own address space.

What are the difference between the following commands: Explain with screenshot.

`ps` and `ps -aux`.

`ps` selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal as the invoker.

`ps -aux` shows every process on the system using BSD syntax.