

BridgeLabz

Employability Delivered

Terminal Commands

Prerequisites to learn CLI

- CLI - Command Line Interface
- Check Java is installed ***java –version***
- Java Installation - <https://exercism.io/tracks/java/installation#linux>
- ***mkdir TerminalCommands***
- ***cd TerminalCommands/***
- Check git is installed - ***git --version***
- ***git clone <https://github.com/edurekavivekh/linux-content.git>***
- Change Dir to ***cd linux-content/*** and do ***ls***
- You will see ***linux_chit_sheet.pdf*** – for Linux Commands and ***linux_problem_sheet.pdf*** – to solve problems using linux commands

- Check your Current Working Directory
- List all files in your Current Working Directory, Top level Directory, current, parent & home Directory.
- List also the hidden files & Directory.
- List all Files & Directory in a long format
- Create a temp Folder
- Remove the temp Folder
- Create a nested temp/temp folder

File & Directory Commands

TerminalCommands — bash — 95x27

```
Narayans-MacBook-Pro:TerminalCommands narayan$ pwd  
/Users/narayan/Development/TerminalCommands
```

Current Working Directory

```
Helloworld.class          def  
Helloworld.java           def.txt  
abc                      forloop.sh  
abc.txt                  linux-content  
commands.txt             loop_commands.txt
```

```
ps_commands.txt  
temp  
test
```

List files & folders in Current Directory

```
Narayans-MacBook-Pro:TerminalCommands narayan$ ls ..  
Commands                 Programming Questions  fundooBooksOld  
DotNetDev                Python              fundooPay  
DotNetDevelopment         Temp                test  
JavaDevelopment          TerminalCommands  
NodeJSSamples            fundooBooks
```

List files & folders in Parent Directory

```
Narayans-MacBook-Pro:TerminalCommands narayan$ ls ~  
Applications             Downloads           Music  
Desktop                  Extras             Pictures  
Development              Library            Public  
Documents                Movies             iCloud Drive (Archive)
```

List files & folders in Home Directory

```
Narayans-MacBook-Pro:TerminalCommands narayan$ ls -a  
.                         abc.txt           loop_commands.txt  
..                        commands.txt        ps_commands.txt  
.DS_Store                 def               temp  
Helloworld.class          def.txt          test  
Helloworld.java           forloop.sh  
abc                      linux-content
```

List all files & folders including the hidden ones

```
[Narayans-MacBook-Pro:TerminalCommands narayan$ ls -l
total 24
-rw-r--r-- 1 narayan staff 612 Nov  1 14:58 Helloworld.class
-rw-r--r-- 1 narayan staff 222 Nov  1 14:58 Helloworld.java
-rwxr-xr-x 1 narayan staff 347 Nov  2 12:38 forloop.sh
drwxr-xr-x 9 narayan staff 288 Nov  1 16:16 linux-content
[Narayans-MacBook-Pro:TerminalCommands narayan$ mkdir temp
[Narayans-MacBook-Pro:TerminalCommands narayan$ ls
Helloworld.class          forloop.sh           temp
Helloworld.java            linux-content
[Narayans-MacBook-Pro:TerminalCommands narayan$ rm -R temp
[Narayans-MacBook-Pro:TerminalCommands narayan$ mkdir -p temp/temp
[Narayans-MacBook-Pro:TerminalCommands narayan$ ls
Helloworld.class          forloop.sh           temp
Helloworld.java            linux-content
[Narayans-MacBook-Pro:TerminalCommands narayan$ cd temp/
[Narayans-MacBook-Pro:temp narayan$ pwd
/Users/narayan/Development/TerminalCommands/temp
[Narayans-MacBook-Pro:temp narayan$ ls
temp
[Narayans-MacBook-Pro:temp narayan$ touch test
[Narayans-MacBook-Pro:temp narayan$ ls -p
temp/  test
[Narayans-MacBook-Pro:temp narayan$
```

List Files & Folder in Long Format

Create Directory temp

Remove recursively all files & folders in temp directory

Create Nested Directory

Change Directory to temp

Create a empty test file

Show all files and also show directories ending with /

File Management

- Change directory to temp
- create a empty test file.
- Copy test file into another test1 file
- move test1 file to test2 file.
- Create a link to the test file in your home dire.
- Change dir to home & remove the temp folder.
- man ls \Rightarrow to list all options
- whereis ls \Rightarrow find the location of the command

```
[temp -- bash -- 95x27]
Narayans-MacBook-Pro:temp narayan$ cp test test1
Narayans-MacBook-Pro:temp narayan$ ls -p
temp/ test test1
Copy test file into test1

Narayans-MacBook-Pro:temp narayan$ mv test1 test2
Narayans-MacBook-Pro:temp narayan$ ls -p
temp/ test test2
Move test1 file into test2

Narayans-MacBook-Pro:temp narayan$ ln -s test temp/
Narayans-MacBook-Pro:temp narayan$ ls -p temp/
test
Link test file in temp dir

Narayans-MacBook-Pro:temp narayan$ ln -s ../../linux-content/linux_chit_sheet.pdf ~/QuickLink/
Narayans-MacBook-Pro:temp narayan$ ls -l ~/QuickLink/
total 0
lrwxr-xr-x 1 narayan staff 37 Nov 3 10:03 linux_chit_sheet.pdf -> ../../linux-content/linux_chit_sheet.pdf
Link linux_chit_sheet pdf to QuickLink in home dir

Narayans-MacBook-Pro:temp narayan$ whereis ls
/bin/ls
Note on listing shows the location of actual file

Locate commands using
whereis

Narayans-MacBook-Pro:temp narayan$ ls /bin
[ dd ksh pax stty
bash df launchctl ps sync
cat echo link pwd tcsh
chmod ed ln rm test
cp expr ls rmdir unlink
csh hostname mkdir sh wait4path
date kill mv sleep zsh
Looking up bin folder in
root dir

Narayans-MacBook-Pro:temp narayan$ man ls
Lookup Command using man.
Use q to quit
```

the **-P** option.

- l** (The lowercase letter ``ell''). List in long format. (See below.) If the output is to a terminal, a total sum for all the file sizes is output on a line before the long listing.
- m** Stream output format; list files across the page, separated by commas.
- n** Display user and group IDs numerically, rather than converting to a user or group name in a long (**-l**) output. This option turns on the **-l** option.
- o** Include the file flags in a long (**-l**) output.
- o** List in long format, but omit the group id.
- P** If argument is a symbolic link, list the link itself rather than the object the link references. This option cancels the **-H** and **-L** options.
- p** Write a slash ('/') after each filename if that file is a directory.
- q** Force printing of non-graphic characters in file names as the character `?'; this is the default when output is to a terminal.
- R** Recursively list subdirectories encountered.

- change direct to etc
- list the password file
- change to previous directory.
- view the file
- browse the file line by line
- display first 10 lines of the /etc/passwd file
- display last 10 lines of /etc/passwd file.

View & Browse Files

```
[Narayans-MacBook-Pro:temp narayan$ ls -l /etc/passwd  
-rw-r--r-- 1 root wheel 6804 Feb 26 2019 /etc/passwd  
[Narayans-MacBook-Pro:temp narayan$ less /etc/passwd  
[Narayans-MacBook-Pro:temp narayan$ head /etc/passwd  
##  
# User Database  
#  
# Note that this file is consulted directly only when the system is running  
# in single-user mode. At other times this information is provided by  
# Open Directory.  
#  
# See the opendirectoryd(8) man page for additional information about  
# Open Directory.  
##  
[Narayans-MacBook-Pro:temp narayan$ tail -2 /etc/passwd  
_timed:*:266:266:Time Sync Daemon:/var/db/timed:/usr/bin/false  
_reportmemoryexception:*:269:269:ReportMemoryException:/var/db/reportmemoryexception:/usr/bin/f  
alse  
[Narayans-MacBook-Pro:temp narayan$ cat /etc/passwd
```

List in Long Format /etc/passwd file

Browsing the passwd file using less

Display first 10 lines from top

Display last 2 lines from bottom

Browsing the complete file

Pipe & Grep Commands

- cmd1 | cmd2
stdout of cmd1 to cmd2
- find all directories in the current working directory.
- find passwd file in /etc/
- find all files in the /etc/ directory

```
[Narayans-MacBook-Pro:temp narayan$ ls -p | grep /  
temp/  
[Narayans-MacBook-Pro:temp narayan$ ls -p | grep -v /  
test  
test2  
[Narayans-MacBook-Pro:temp narayan$ env  
TERM_PROGRAM=Apple_Terminal  
SHELL=/bin/bash  
TERM=xterm-256color  
TMPDIR=/var/folders/sz/zlcpnpd10qlcr6xf3frycq3h0000gn/T/  
Apple_PubSub_Socket_Render=/private/tmp/com.apple.launchd.k04qruezk78/Render  
TERM_PROGRAM_VERSION=421.2  
OLDPWD=/Users/narayan/Development/TerminalCommands  
TERM_SESSION_ID=FEB0A322-0260-4F1F-9A39-A2B3CE5D10E9  
USER=narayan  
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.kfSZSOYnZs/Listeners  
PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin  
PWD=/Users/narayan/Development/TerminalCommands/temp  
XPC_FLAGS=0x0  
XPC_SERVICE_NAME=0  
SHLVL=1  
HOME=/Users/narayan  
LOGNAME=narayan  
LC_CTYPE=UTF-8  
_=
```

/usr/bin/env

```
[Narayans-MacBook-Pro:temp narayan$ env | grep USER  
USER=narayan
```

Using Pipe i.e. | to show all Folders

Using Pipe to show all Files

Displaying Environment Variables

Using pipe to Grep USER Variable

⇒ Env Variables

- Show all environment variables
- Show only the User , home Directory of the shell.

⇒ File Size & Disk Usage

- List disk usage of each sub dir & its contents (du)
- List disk usage of a particular file or folder in a human readable format. (du -sh <>)

Env Variables & Disk Usage

```
Narayans-MacBook-Pro:TerminalCommands narayan$ echo $USER
narayan
Narayans-MacBook-Pro:TerminalCommands narayan$ echo $SHELL
/bin/bash
Narayans-MacBook-Pro:TerminalCommands narayan$ echo $HOME
/Users/narayan
Narayans-MacBook-Pro:TerminalCommands narayan$ du -sh linux-content/
 27M  linux-content/
Narayans-MacBook-Pro:TerminalCommands narayan$ du -sm * | sort -nr
27    linux-content
1      forloop.sh
1      Helloworld.java
1      Helloworld.class
0      temp
Narayans-MacBook-Pro:TerminalCommands narayan$ du -sk * | sort -n
0      temp
4      Helloworld.class
4      Helloworld.java
4      forloop.sh
27456  linux-content
Narayans-MacBook-Pro:TerminalCommands narayan$ find ~/Development -name commands.txt
/Users/narayan/Development/Commands/commands.txt
Narayans-MacBook-Pro:TerminalCommands narayan$ find . -type f -empty
./temp/test
./temp/test2
./linux-content/sample
Narayans-MacBook-Pro:TerminalCommands narayan$
```

Writing USER, SHELL and HOME variable to standard output using echo
NOTE: Variable is referred using \$

Displaying Disk Usage in Human Readable Form

Piping Disk Usage of current folder to Sort in Descending Order

Piping Disk Usage in Ascending Order
NOTE – Option n is used to Sort Numeric Value

1: Finding commands.txt file in Development folder
2: Finding empty file in current folder

Process Management

- Use nano editor to create a Helloworld java program.
- The Helloworld java program would print hello world every minute in an infinite loop
- Display the current running processes
- Display realtime the top processes.
- Identify the java process & kill it.
- start the Helloworld program in the Background
- Display the Background jobs
- Bring the most recent background job to foreground

```
Narayans-MacBook-Pro:TerminalCommands narayan$ nano -T 3 Helloworld.java
Narayans-MacBook-Pro:TerminalCommands narayan$ javac Helloworld.java
Narayans-MacBook-Pro:TerminalCommands narayan$ ls -l
total 24
-rw-r--r-- 1 narayan staff 612 Nov  3 15:06 Helloworld.class
-rw-r--r-- 1 narayan staff 224 Nov  3 15:06 Helloworld.java
-rwxr-xr-x 1 narayan staff 347 Nov  2 12:38 forloop.sh
drwxr-xr-x 9 narayan staff 288 Nov  1 16:16 linux-content
drwxr-xr-x 5 narayan staff 160 Nov  3 10:02 temp
Narayans-MacBook-Pro:TerminalCommands narayan$ java Helloworld &
[1] 68193
Narayans-MacBook-Pro:TerminalCommands narayan$ Hello world
```

1: Using nano editor to create Helloworld.java
2: NOTE: option T 3 for tabspace of 3
3: Compiling Java Program

Running the java program in Background using &

```
Narayans-MacBook-Pro:TerminalCommands narayan$ ps -elf | grep java | grep -v grep
 501 68193 57392      4006  0  31  0 8030176 30076 -
:00.16 /usr/bin/java He 3:06PM
Narayans-MacBook-Pro:TerminalCommands narayan$ jobs
[1]+  Running                  java Helloworld &
Narayans-MacBook-Pro:TerminalCommands narayan$ fg %1
java Helloworld
^Z
[1]+  Stopped                  java Helloworld
Narayans-MacBook-Pro:TerminalCommands narayan$ bg %1
[1]+ java Helloworld &
Narayans-MacBook-Pro:TerminalCommands narayan$ jobs
[1]+  Running                  java Helloworld &
Narayans-MacBook-Pro:TerminalCommands narayan$ killall java
```

Grepping the Java Process

Viewing the Jobs running in Background

Bringing the Job to the Foreground

Stopping the Job using ^Z

Starting the Job to the Foreground

Killing the Job



GNU nano 2.0.6

TerminalCommands — nano -T 3 Helloworld.java — 95x27
File: Helloworld.java

```
public class Helloworld {  
  
    public static void main(String args[]){  
        while(true){  
            System.out.println("Hello world");  
            try{  
                Thread.sleep(60000);  
            }catch(Exception ex){  
                System.out.println(ex);  
            }  
        }  
    }  
}
```

[Read 15 lines]

^G Get Help **^O** WriteOut **^R** Read File **^Y** Prev Page
^X Exit **^J** Justify **^W** Where Is **^V** Next Page

^K Cut Text **^C** Cur Pos **^U** UnCut Text **^T** To Spell

- awk '{...}'
- Use Awk command to display process ids.
 - Step 1 - print all the current running process
 - Step 2 - pipe the output as I/p to awk
 - Step 3 - awk '{print \$3}'
- echo Hello Tom but print hello Adam
 - Step 1 - echo Hello Tom
 - Step 2 - pipe the output as I/p to awk
 - Step 3 - In awk replace Tom to Adam
 - g print i.e.
 - awk '{ \$2 = "Adam"; print \$0 }'
- awk 'BEGIN{...}{...}' => Pre process
- awk '{...} END{...}' => Post process

AWK Commands

```
[Narayans-MacBook-Pro:TerminalCommands narayan$ ps -elf | grep java | grep -v grep  
 501 68416 57392      4006  0  31  0  8019936  30160 -          Grepping the Java Process  
:00.19 /usr/bin/java He  3:53PM 0  
[Narayans-MacBook-Pro:TerminalCommands narayan$ ps -elf | grep java | grep -v grep | awk '{ print $2 }'  
68416          Using awk to grep the Java Process Id  
[Narayans-MacBook-Pro:TerminalCommands narayan$ kill -9 `ps -elf | grep java | grep -v grep | awk '{ print $2 }'`  
          Using awk t& Backquotes `...` to kill java process  
[Narayans-MacBook-Pro:TerminalCommands narayan$ mypid=`ps -elf | grep java | grep -v grep | awk '{ print $2 }`  
          1: Assigning java process id to  
          variable mypid  
[Narayans-MacBook-Pro:TerminalCommands narayan$ echo $mypid  
68432          2: Printing to terminal $mypid  
[Narayans-MacBook-Pro:TerminalCommands narayan$ kill -9 $mypid  
          3: Killing the Java process  
[Narayans-MacBook-Pro:linux-content narayan$ echo Hello Tom  
Hello Tom  
[Narayans-MacBook-Pro:linux-content narayan$ echo Hello Tom | awk '{ print $0 }'  
Hello Tom  
[Narayans-MacBook-Pro:linux-content narayan$ echo Hello Tom | awk '{ print $2 }'  
Tom  
[Narayans-MacBook-Pro:linux-content narayan$ echo Hello Tom | awk '{ $2 = "Adam" } { print $0 }'  
Hello Adam  
[Narayans-MacBook-Pro:linux-content narayan$          1: echo to stdout Hello Tom  
          2: Using awk print $0 which prints Hello Tom  
          3: Using awk replace Tom with Adam and Print the  
             complete String
```

```

Narayans-MacBook-Pro:linux-content narayan$ cat data.csv
Id EmployeeName JobTitle      BasePay OvertimePay OtherPay TotalPay TotalPayBenefits
1  NATHANIEL    GM           167411   0          400184   567595   567595
2  GARY         CAPTAIN     155966   245131    137811   538909   538909
3  ALBERT       CAPTAIN     212739   106088    16452    335279   335279
4  CHRISTOPHER  MECHANIC   77916    56120     198306   332343   332343
5  PATRICK      DEPUTYCHIEF 134401   9737      182234   326373   326373
6  DAVID        ASSTDEPUTY   118602   8601      189082   316285   316285
7  ALSON        BATTALIONCHIEF 92492   89062     134426   315981   315981
8  DAVID        DEPUTYDIRECTOR 256576   0          51322    307899   307899
10 JOANNE       CHIEF       285262   0          17115    302377   302377
12 PATRICIA    CAPTAIN     99722    87082     110804   297608   297608
13 EDWARD       EXECUTIVE   294580   0          0         294580   294580

Narayans-MacBook-Pro:linux-content narayan$ cat data.csv | grep CAPTAIN | awk '{ print $2 " " $4}'
GARY 155966
ALBERT 212739
PATRICIA 99722

Narayans-MacBook-Pro:linux-content narayan$ cat data.csv | grep CAPTAIN | awk '{ sum+=$4 }END{ print sum }'
468427

Narayans-MacBook-Pro:linux-content narayan$ cat data.csv | grep CAPTAIN | awk '{ sum+=$4 }END{ print sum/NR }'
156142

Narayans-MacBook-Pro:linux-content narayan$ 
```

**Displaying Data
in data.csv file**

**Displaying the Employee Name and Base Salary
whose Job Title is Captain**

Displaying the total salary received by Captains

**Displaying the average salary received by Captains
NOTE: END is used to indicate post process and NR
is an inbuilt variable indicating Number of Records**

Conditions & Loops

- for loop

```
for arg in '....';
do
....;
....;
done
```
- eg: Move files from One folder to
their respective folders.
 - Step1: Create two empty files
abc.txt & def.txt
 - Step2: print the list of .txt files.
 - Step3: Pipe it to awk & print the
file name.
 - Step4: Do it in for loop for each
file & print the file name

```
for files in `ls *.txt`;
do echo $files | awk -F. '{print $1}'
```

done

```
Narayans-MacBook-Pro:temp narayan$ ls *.txt
abc.txt def.txt
Narayans-MacBook-Pro:temp narayan$ for files in `ls *.txt`; do echo $files; done
abc.txt
def.txt
Narayans-MacBook-Pro:temp narayan$ for file in `ls *.txt`; do folderName=`echo $file | awk -F.
'{print $1}'`; echo $folderName; done
abc
def
Narayans-MacBook-Pro:temp narayan$ for file in `ls *.txt`;
> do
>   folderName=`echo $file | awk -F. '{print $1}'`;
>   mkdir $folderName;
>   cp $file $folderName;
>   echo Copied $file to $folderName/;
> done
Copied abc.txt to abc/
Copied def.txt to def/
Narayans-MacBook-Pro:temp narayan$ ls -p
abc/           def/           folderName/
abc.txt        def.txt        temp/
Narayans-MacBook-Pro:temp narayan$ ls -p abc/
abc.txt
Narayans-MacBook-Pro:temp narayan$ ls -p def/
def.txt
Narayans-MacBook-Pro:temp narayan$
```

Using for loop to display files with ext .txt

Using for loop and awk to display file names

- 1: Using for loop to read in each file with ext .txt
- 2: Starting the loop with do
- 3: Using awk to read the file name into folderName
- 4: Making Directory with the file name
- 5: Copying the file into the folder using \$folderName
- 6: Displaying the result to stdout
- 7: indicating done to close for loop

Displaying the files in current folder and the new folders created by the for loop

Step 5 : Assign the output of 4 to
a folder variable & print
the folder name

Step 6 : Create directory using the
folder variable

Note : this is the second command
within do loop.

Step 7 : move the files to the folder
directory . This is 8rd command

Step 8 : Done .

Conditions & Loops

Conditions & Loops

- Conditional Loop
while [condition];
do
 commands...
done
- Conditional if ... then ... else
if [condition];
 then
 command;
 else
 command;
 fi



GNU nano 2.0.6

TerminalCommands — nano -T 3 forloop.sh — 95x27

File: forloop.sh

Modified

```
#!/bin/bash -x

for file in `ls *.txt`;
do
    folderName=`echo $file | awk -F. '{print $1}'`;
    #echo "checking for already existing directory started";
    if [ -d $folderName ];
    then
        rm -R $folderName;
    fi
    #echo creating foler ${folderName};
    mkdir $folderName;
    #echo copying ${file} to ${folderName};
    cp $file $folderName;

done
```

**1: Creating forloop.sh script file to copy txt files to resp dir
2: Specifying bash shell and setting debug option with -x
3: Using for loop to read each file**

**1: Using if condition to check folder exists
2: then removing the folder along with the files
3: Closing the if loop with fi**

Creating the directory with the file name and then copying the files into the corresponding directory

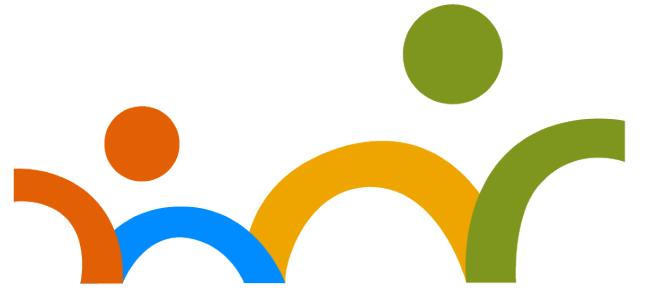
^G Get Help **^O** WriteOut **^R** Read File **^Y** Prev Page **^K** Cut Text **^C** Cur Pos
^X Exit **^J** Justify **^W** Where Is **^V** Next Page **^U** UnCut Text **^T** To Spell

[Read 16 lines]

```
Narayans-MacBook-Pro:TerminalCommands narayan$ nano -T 3 forloop.sh
Narayans-MacBook-Pro:TerminalCommands narayan$ ./forloop.sh
++ ls abc.txt def.txt
+ for file in `ls *.txt`
++ echo abc.txt
++ awk -F. '{print $1}'
+ folderName=abc
+ '[' -d abc ']'
+ rm -R abc
+ mkdir abc
+ cp abc.txt abc
+ for file in `ls *.txt`
++ echo def.txt
++ awk -F. '{print $1}'
+ folderName=def
+ '[' -d def ']'
+ rm -R def
+ mkdir def
+ cp def.txt def
Narayans-MacBook-Pro:TerminalCommands narayan$ ls -p
Helloworld.class          abc.txt           forloop.sh
Helloworld.java           def/              linux-content/
abc/                      def.txt          temp/
Narayans-MacBook-Pro:TerminalCommands narayan$
```

Running the script using ./forloop.sh

- 1: Script is run with debug statements
- 2: Indicates first abc.txt is being processed
- 3: Setting the folderName variable with abc
- 4: Checks if folder abc exists
- 5: Folder exists hence removing the folder and its contents
- 6: Creating abc directory
- 7: Copying abc.txt file to abc folder
- 8: Similarly working with the def.txt file



BridgeLabz

Employability Delivered

Thank you