1. Linux tar file downloading nodejs: wget <https://nodejs.org/download/release/v8.5.0/node-v8.5.0-linux-x64.tar.gz>
2. Extract tar file : tar –xvzf node-v8.5.0-linux-x64.tar.gz
3. mv (rename package to node)
4. Cd bin
5. Pwd
6. Create soft link: ln -s /home/puppetMaster/node/bin/node /usr/bin/node

Everything is a file in linux even directory is also a file. Kernel is the only one which touch’s the hardware. On the top of kernel(parent OS). there is shell(command interpretor in OS and sometime execute command also, it is a agency that sits between user and UNIX system)

First hardware then kernel then shell then tools.

Process using heavy heavy i/o are called I/O bound.

**No we cannot touch kernel directly that’s why it is more secure.**

**Code is Uninteractive when passing direct values in command**

1. Command to see content in number: **nl** sample.txt
2. nl –v10 –i5 sample.txt: -v will start numbering from 10 and I will make common difference: 5
3. tty :  tty is a command in Unix and Unix-like operating systems to print the file name of the terminal connected to standard input. tty stands for TeleTYpewriter
4. absolute path: etc/logstash/conf.d

relative: changing one by one.

1. Output redirection >, input <, and append >>

Cat sample.txt >> two.txt

1. Cmp: compare and show first difference and then quit: cmp file1.txt file2.txt

Filters:

1. Simple filters: cut –d(delimiter) “|” –f2 filename.txt: will display only second column divide by delimiter |
2. Advanced:

\* find filter: used to find location of file: find /home/xyz/ -name sample.txt

Find<path list><selection criteria><action>

Grep : used to find content of file: grep ‘^1002’ sample.lst, ^ is regular expr. Means value start with 1002,

$ means ends with: grep ‘1002$’ sample.lst

Egrep : extended regular expr. egrep ‘[aA]gg?[ar]+wal’ stud.lst: searching speeling of Agarwal with different combination

[aA] start with both a and A

g? means 0 or 1 time, first g means one g is compulsory

[ar] + means a or r can be repeated n number of time.

Current date and year: **`date “+%m 20%y”` , use ` (backquote or backtick) when you want to fire command inside a command,** Everything you type between backticks is evaluated (executed) by the shell before the main command

1. Eval : evaluate: **eval `date ‘+day=%d month=%m year=%Y’`**
2. **SED:** stream editor, find and replace: sed ‘s/old word/new word/’ filename.txt, here s will replace first occurrence so use **g** for global replace.

**e.**g : sed ‘s/old word/new word/gi’ filename.txt, i indicates the case sensitivity

**-**e is for multiple replace: : sed -e ‘s/old word/new word/gip’ filename.txt, p will show you only effected lines.

**sed 2d sample.**txt will not display 2nd line, sed 2q sample.txt will show you the first 2 lines

1. Ps –f : tell you the number of processes(the command that you run) running.

A shell script is also a process. PID is a process ID given to kernel. Every process, with the exception of PID 0 processes has a parent. PPID is a parent process iD.

**Chrone tab:**

1. Put the process in background mode using **&** and immediately after that command prompt is available for next command.
2. Nohub : no hangup: nohup sort &: permits execution of process even if user has logged off.
3. Kill process: kill -9 PID (order of killing the process), 15 is a default signal
4. System daemon: are the process running in a background mode which will help the kernel to manage the OS.
5. **Deamon Cron**: help to schedule the task. $ 0 18 \* \* \* /home/script.sh : will run every day at 6 pm. Week starts with Sunday.

30 15 \* 2 1 /script.sh: run script.sh every Monday every February at 3:30 pm

1. **AWK**: awk –F “|” ‘{print $0}’ sample.lst : will display the content of sample.lst., | field separator, $0 indicates entire record. Limitation of grep covered by awk. If you use $2 instead of $0 then it will display 2nd column.

awk –F “|” ‘/for/{print $0}’ sample.lst: will display value of 2nd column which have for in sentence.

awk –F “|” ‘/for/’ sample.lst : will give same output.

awk –F “|” ‘/for/{print NF}’ sample.lst: number of fields

awk –F “|” ‘$3>500{print $2, $1\*$3}’ script.lst : if 3rd column is > 500 then it will print 2nd column and multiplication of 1st and 3rd

create program with awk also: awk –F “|” ‘{

* Count=count+1
* Print $2,count}’ scriplt.lst

1. Begin and End block
2. Printing date: echo “todays date is “;date
3. Echo “1 numner”

Read no1

Echo “2 number”

Read no2

Res=`expr $no1 + $no2` **or** **let** res=no1+no2 **or** ((res=no1+no2))

Echo “the result is $res” > output.txt // here the output will go to output.txt file

1. Shell is a interpreter

cat func.sh   
add()   
{   
  no1=$1   
  no2=$2   
  #res=`expr $no1 + $no2`   
  let res=no1+no2   
  #((res=no1+no2))   
  echo "The result is $res"   
  return 0   
}

echo "Main Starts...."   
 echo "Enter First No ..."   
read n1   
echo "Enter Second No...."   
read n2   
add $n1 $n2   
echo "In Main.. The addition is $res"   
echo "Main Ends...."

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

1. Copy the file first.unix in your home directory to first.unics.

(Note: checked using ls, first.unix file also should exist along with first.unics)

1. List the contents of first.unix and first.unics with a single command.
2. Create a new directory under the temp directory.
3. From your home directory, copy all the files to the directory created under the temp sub directory.
4. Move the file first.unix to the directory temp as second.unix
5. Remove the file called first.unics from the home directory.
6. Change your directory to temp and issue the command rm \*. What do you observe?
7. Move all files whose names end with a, c and o to the HOME directory.
8. Copy all files that end with a ‘UNIX’ to the temp directory.
9. Issuing a single command, remove all the files from the directory temp and the directory itself.
10. Try commands cp and mv with invalid number of arguments and note the results.
11. Use the cat command to create a file friends, with the following data:

Madhu 6966456 09/07/68

Jamil 2345215 08/09/67

Ajay 5546785 01/04/66

Mano 7820022 09/07/68

David 8281292 09/09/60

Simmi 7864563 12/12/70

Navin 2224311 30/05/68

The fields should be separated by a tab.

1. Display contents of the file friends.
2. Copy contents of friends to newfriend without using the cp command.
3. Display contents of the file friends and newfriends in a single command.
4. Find all users currently working on the system and store the output in a file named as users.
5. Append contents of friends file to the file, users.
6. Display current system date and time and record your observations. How is the time displayed?
7. Give the execute permission for the user for a file chap1
8. Give the execute permission for user, group and others for a file add.c
9. Remove the execute permission from user, give read permission to group and others for a file aa.c
10. Give execute permission for users for a.c, kk.c, nato and myfile using single command
11. Change the directory to root directory. Check the system directories, like bin, etc, usr etc

## 4.1: Using Pipes and Filters:

1: Redirect the content of the help document ls, into a file called as lsdoc.

2: Display the content of the lsdoc page wise.

3: Display only the first 4 lines of the lsdoc file.

4: Display only the last 7 lines of the file lsdoc.

5: Remove the file lsdoc.

6: There will be B’day celebration from the friends file, find how many B’day parties will be held. If two of the friends have the B’date on the same day, then we will be having one party on that day.

7: Display the lines starting with Ma, in the file friends.

8: Display the lines starting with Ma, ending with i or ending with id, in the file friends.

9: Print all the files and the directory files from the current directory across all the sub directories, along with its path

10: Print only the Directory files.

11: Display the files starting with chap, along with its path.

12: Sort the file friends in ascending order of names.

13: Display the contents of the file friends in uppercase letters.

14: Store the contents of your home directory in a file called dir.

15: From the above file dir, display the file permissions and the name of the file only.

16: From the same dir file, store only the file names in a file called files.

17: From the same dir file, store only the permissions of files in a file called perms.

18: From the same dir file, store only the file sizes in a file called sizes.

19: Display the file names, sizes and permissions from your directory in that order.

20: Display the number of users working on the system.

21: Find out the smallest file in your directory.

22: Display the total number of lines present in the file friends.

23:Create the following fixed record format files (with “|” delimiter between fields) with the structure given below, and populate them with relevant data use these files to solve following questions

emp.lst: Empid(4),Name(18),Designation(9),Dept(10),Date of Birth(8),Salary(5)

dept.lst : Dept.Code(2),Name(10),Head of Dept’s id(4)

desig.lst: Designation Abbr.(2), Name (9)

1. Find the record lengths of each file.
2. Display only the date of birth and salary of the last employee record.
3. Extract only employee names and designations. (Use column specifications). Save output as cfile1.
4. Extract Emp.id, dept, dob and salary. (Use field specifications). Save output as cfile2.
5. Fix the files cfile1 and cfile2 laterally, along with the delimiter.
6. Sort the emp.lst file in reverse order of Emp. Names.
7. Sort the emp.lst file on the salary field, and store the result in file srtf.
8. Sort the emp.lst file on designation followed by name.
9. Sort the emp.lst file on the year of birth.
10. Find out the various designations in the employee file. Eliminate duplicate listing of designations.
11. Find the non-repeated designation in the employee file.
12. Find the number of employees with various designations in the employee file.
13. Create a listing of the years in which employees were born in, along with number of employees born in that year.

25: Write a command sequence that prints out date information in this order: time, day of week, day number, month, year:

13:44:42 IST Sun 16 Sept 1994

26: Write a command sequence that prints the names of the files in the current directory in the descending order of number of links

27: Write a command sequence that prints only names of files in current working directory in alphabetical order

28: Write a command sequence to print names and sizes of all the files in current working directory in order of size

29: Determine the latest file updated by the user

Create a file using Vi. Enter the following text:

*A network is a group of computers that can communicate with each other, share resources, and access remote hosts or other networks. Netware is a computer network operating system designed to connect, manage, and maintain a network and its services. Some of the network services are Netware Directory Services (NDS), file system, printing and security.*

1. Change the word “Netware” in the second line to “Novell Netware”.
2. Insert the text “(such as hard disks and printers)” after “share resources” in the first line.
3. Append the following text to the file:

“Managing NDS is a fundamental administrator role because NDS provides a single point for accessing and managing most network resources.”

2: Create the data files, used in the previous lab sessions using vi editor.

## : Using SED Commands

1. Create a file “Employee.dat” with text as follows.

James 76382 PACE Chennai

John 34228 GRIT Hyderabad

Peter 22321 GE Bangalore

Albert 32342 GRIT Pune

Mathew 23222 PACE Mumbai

Richard 23232 ACS Pune

1. Write a sed command to print only the lines starting at line 2 and ending with the letters “Pune”
2. Write a sed command that will display the top 5 lines from the file
3. Write a sed command that will substitute the word “Chennai” for "Pune" used in all instance of the word
4. Write a sed command that will replace occurrence of the character e with the string UNIX in all lines. (Use –e option)
5. Write a sed command to delete blank lines
6. Write a sed command to delete lines from 3 to 5

2**: Create a new file “PACE.dat which has only the lines that contain the word “PACE” from Employee.dat**

## : Using Process-Related Commands

1. Find out the PID of the processes that are activated by you
2. Find out the information about all the processes that are currently active
3. Start a different process in the background. Find out the status of the background process using the PID of the same.

## Writing awk-Scripts

1. Consider the results are stored in following format:

EmpID Name Subject Marks(/50)

E001 Nilesh Unix 30

E002 Nilesh DSA 20

Like these you have 10 records ( 5 of DSA and 5 of Unix)

Calculate the avg score secured in Unix and DSA and the first 2 topers in Unix and DSA each.

2. Write a script to get the report of the users logged on to the System in the following formats. (Records should be sorted on logging time.)

Header must include company name and Date

Records in the format

Username Logged-in-time Terminal

Tailor should include total number of the users logged in.

3. Consider a text file containing the records (colon separated fields) in the format:

## EmpName:EmpId:Subject:ObtMarks:TotMarks:Result

Write a script to get the result of “UNIX” Subject in the format (Considering the data file has TotMarks=50 for UNIX)

EmpName:ObtMark:MarksOutof35

The header of the report must contain total marks and the tailor must specify the percentage result for that subject.

Also generate another summary result containing total number of participants appeared, total number of participants passed, and Name of the participants ranked Ist IInd, IIIrd with their total score.

4 Consider the Arizona roaster as an input data file for this exercise. Using any combination of the text-processing utilities listed below, write a one-line shell command that performs each of the following tasks

(a) reports the number of players on the roster

(b) displays the roster in order by jersey number

(c) displays the roster in alphabetical order by surname

(d) displays the heaviest five players

(e) displays all players who attended Wisconsin

The Roaster is given below:-

JersyNo|Name,surname |linebackers|Weight | Date | Experience| Country

20 | Anderson, Damien |RB 5'10" | 212 | 07/17/1979 | 3 |Northwestern

30 | Ayanbadejo, Oba |FB 6'02" | 235 | 03/05/1975 | 5 |San Diego

92 | Berry,Bert |DE6'03" | 250| 08/15/1975 | 7 |NotreDame

81 | Boldin,Anquan |WR6'01" |215 | 10/03/1980 |2 | FloridaState

91 | Bryant,Wendell |DT6'04" |303| 09/12/1980 | 3 | Wisconsin

35 | Carter,Dyshod |DB5'10" |197 | 06/18/1978 |2 | Kansas State

52 | Fisher,Levar |OLB6'01" |235 | 07/02/1979 |3| NorthCarolina

11 | Fitzgerald,Larry |WR6'04" |229| 08/31/1983 |10 |Pittsburgh

63 |Garcia,Frank |G6'02" | 302| 01/28/1972|10|Washington

64 |Grace,Steven |C6'03" |295| 02/13/1979 | 3|Arizona

84 |Hamilton,Lawrence |WR6'03" |205 |08/31/1980 |2|StephenF.Austin

## Writing Shell-Scripts

1. Display the Primary and Secondary prompt. Change the primary prompt to your name: temporarily

2: As soon as you login, the prompt should be changed to your name: also the name of the home directory should be automatically displayed.

3: Check the content of the Environmental variable SHELL.

6: Write a shell script that takes the user name as input and reports whether he / she has logged in or not.

7: Write a shell script to display the file name and its contents of all the files that is there in the current directory.

8: Write a shell script, which will take a file name as argument and check whether the file exists and display its access permissions for user.

9: Pass three numbers as command line arguments and display the largest number in the given three numbers.

10: Write a shell script which will accept a pattern and a file name. The pattern will be searched in the file provided. Display appropriate messages and perform necessary validations on file.

11: To create a menu program for a) creating a file, b) Creating a directory, c) copying a file, d) moving a file. (use functions)

* 1. If the file exists already give the appropriate message
  2. If the dir exists already give the appropriate error message
  3. Source file should exist if not give a message, It should have read permission if not another message, Destination file either there or not, if not there then create it and copy it. If there, then ask whether to overwrite or not, if yes then overwrite it or else give a message file exists already and not overwritten.

12: Write a function yesno() to display question to user and accept answer as y/n. If answer to the question is y the function should return 0 otherwise 1.

Use yesno functions for asking different questions. Question will be passed as parameter to the function.

Accept filename from user check whether it is file or directory. Use yesno() function to display question do you really want to delete file? If the ans is y, then delete the file or directory.

13: Write a shell script to store names of four employees and check whether those employees are currently logged in or not. Display appropriate message.

14: Accept the user's first and last name and the echo the entire name along with some suitable comment.

15: List all files that have been modified today.

16: Display long listing of only the regular files in the current directory.

17: Display details of all files in the 2 “paths” accepted from user. The display should be screen by screen.

18: Let the script display its name and its PID.

19: Get the concatenated o/p of 2 files into a third file: Take 3 command line arguments: The first argument is the name of a destination file, and the other two arguments are names of files whose contents are to be placed in the destination file.

**Stretched Assignments:**

20: Write a menu driven shell program to:

a. Display calendar of current month

1. Search for a pattern in all the files/subdirectories from current directory.
2. Count the no. of directories / sub directories in current directory

21: Display day of week for a given date. (ddmmyyyy)

If day is Monday, display message “Monday Blues”

Friday display message “yeh! It’s week end.”

Similarly display different messages for each day of the week.

22: Display the contents of all .lst files in the current directory.

23: Design a simple calculator, which will add/subtract/multiply/divide 2 numbers.

eg. cal 10 20 + will give o/p as 30.

24: For a student file with the following fields, rollno, name, marks, Generate 2 files ‘Pass’ and ‘Fail’ containing records of student who have passed or failed. Also count the number of students who have passed or failed.

25: Accept a date string from terminal and display employees born after the input date.