UNIX File System & Permissions

1: Give the execute permission for the user for a file chap1. 🡪 chmod u+x chap1

2: Give execute permission for user, group and others for a file add.c 🡪 chmod a+x add.c

3: Remove the execute permission from user, give read permission to group and others for a file aa.c 🡪chmod u-x, g+r,o+r aa.c

4: Give execute permission for users for a.c, kk.c, nato and myfile using single command.

🡪 chmod u+x a.c kk.c nato myfile

5: Change the directory to root directory. Check the system directories, like bin, etc, usr etc.

🡪 cd / 🡪 ls

*\*\*\*\*Already pasted the screenshots for the above section in previous assignment.\*\*\*\**

Using Pipes and Filters

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

🡪 man ls | col -b > lsdoc OR :r !man ls | col -b 🡪 :w lsdoc 🡪 :q! 🡪 cat lsdoc

2: Display the content of the lsdoc page wise.

🡪 less lsdoc

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

🡪 head -n 4 lsdoc

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

🡪 tail -n 7 lsdoc

5: Remove the file lsdoc.

🡪 rm 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.

🡪 sort friends | uniq | wc -l

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

🡪 grep ‘^Ma’ friends

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

🡪 grep ‘^Ma.\*\(i\|id\)$’ friends

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

🡪 find. -print

10: Print only the Directory files.

🡪 find. -type d -print

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

🡪 find. -type f -name’chap\*’ -print

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

🡪 sort friends > friends\_sorted

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

🡪 cat friends | tr ‘[a-z]’ ‘[A-Z]’ OR cat friends | tr ‘[:lower:]’ ‘[:upper:]’

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

🡪 ls ~ > dir

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

🡪 ls -l ~ > dir

awk ‘{print $1, $9}’ dir

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

🡪 cat dir | awk ‘{print $9}’ dir > files

cat files

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

Awk.

🡪 cat dir | awk ‘{print $1}’ dir > perms

cat perms

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

🡪 cat dir | awk ‘{print $5}’ dir > sizes

cat sizes

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

🡪 ls -l | awk ‘{print $9, $5, $1}’

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

🡪 who | wc -l

21: Find out the smallest file in your directory.

🡪 ls -lS | grep ‘^-‘ | awk’$5 != “0”’ | head -n 1

The grep filter just includes the files and not directories as we have been asked to find the smallest file. The awk filter removes the empty files.

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

🡪 cat friends | wc -l

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)

**🡪 Create emp.lst file**: echo -e “1234|John Doe | Developer | IT |19980105 |60000\n1253|Ron Williams | Manager |Sales|19920303 |80000\n1786|Severus Snape |Analyst|Finance |20000101|38000” > emp.lst

**Create Dept.lst file:** echo -e “01 |Sales |1253\n02|IT| |1234\n03|Finance |1786 “ > dept.lst

**Create design.lst file:** echo -e”MG |Manager \nDV |Developer\nAN |Analyst

1. Find the record lengths of each file.

🡪 wc -c emp.lst dept.lst design.lst

1. Display only the date of birth and salary of the last employee record.

🡪 tail -n 1 emp.lst | awk -F‘|’‘{print $5, $6}’

3. Extract only employee names and designations. (Use column

Specifications. Save output as cfile1.

🡪 awk -F ‘|’ ‘{print $2, $3}’ emp.lst > cfile1

1. Extract Emp.id, dept, dob and salary. (Use field specifications). Save output as cfile2.

🡪[admin@hostname01 ~]$ cut -d'|' -f1,4,5,6 emp.lst > cfile2

1. Fix the files cfile1 and cfile2 laterally, along with the delimiter.

🡪[admin@hostname01 ~]$ paste -d'|' cfile1 cfile2 > fixed\_file

1. Sort the emp.lst file in reverse order of Emp. Names.

🡪admin@hostname01 ~]$ sort -t'|' -k2,2r emp.lst

1. Sort the emp.lst file on the salary field, and store the result in file srtf.

🡪[admin@hostname01 ~]$ sort -t'|' -k6,6n emp.lst > srtf

1. Sort the emp.ls t file on designation followed by name.

🡪[admin@hostname01 ~]$ sort -t'|' -k3,3 -k2,2 emp.lst

1. Sort the emp.lst file on the year of birth.

🡪[admin@hostname01 ~]$ sort -t'|' -k5,5 emp.lst

10. Find out the various designations in the employee file. Eliminate duplicate listing of designations.

🡪[admin@hostname01 ~]$ cut -d'|' -f3 emp.lst | sort | uniq

Engineerx

11. Find the non-repeated designation in the employee file.

🡪[admin@hostname01 ~]$ cut -d'|' -f3 emp.lst | sort | uniq -u

12. Find the number of employees with various designations in the employee file.

🡪[admin@hostname01 ~]$ cut -d'|' -f3 emp.lst | sort | uniq -c

3 Engineer

13. Create a listing of the years in which employees were born in, along with number of employees born in that year.

🡪[admin@hostname01 ~]$ cut -d'|' -f5 emp.lst | cut -c1-4 | sort | uniq -c

1 1998

14. Use nl command to create a code table for designations to include designation code (Start with dept. code 100, and subsequently 105, 110 …).

🡪[admin@hostname01 ~]$ cut -d'|' -f3 emp.lst | sort | uniq | nl -v100 -i5

100 Engineer

24: PCS has its offices at Pune, TTC and Mumbai. The employees’ data is stored

separately for each office. Create appropriate files (with same record structure as

in previous assignment) and populate with relevant data.

1. List details about an employee ‘Manu Sharma’ in the Mumbai office.

🡪[admin@hostname01 ~]$ grep "Manu Sharma" mumbai.lst

5001|Manu Sharma |Engineer |H&W |19939830|72000

1. List only the Emp.Id. And Dept. of Manu Sharma.

🡪[admin@hostname01 ~]$ grep "Manu Sharma" mumbai.lst | awk -F'|' '{print $1, $4}'

5001 H&W

1. List details of all managers in all offices. (O/P should not contain file names.).

🡪admin@hostname01 ~]$ grep -i "manager" \* | cut -d':' -f2-

1. Find the number of S.E. in each office.

🡪[admin@hostname01 ~]$ grep -i "S>E" \* | cut -d':' -f1 | sort | uniq -c

5. List only the Line Numbers and Employee names of employees in ‘H/W’ in

Pune file.

🡪[admin@hostname01 ~]$ grep -r -n "H/W" , | grep "pune" |cut -d: -f1,2

6. Obtain a listing of all employees other than those in ‘HR’ in the Mumbai file

and save contents in a file ‘nonhr’.

🡪[admin@hostname01 ~]$ grep -v "HR" mumbai.lst > nonhr

7. Find the name and designation of the youngest person who is not a manager.

🡪[admin@hostname01 ~]$ grep -v "manager" mumbai.lst | sort -t'|' -k6,6n | head -n 1 | awk -F'|' '{print $2, $3}'

Manu Sharma Engineer

8. Display only the filename(s) in which details of employee by the name

‘Seema Sharma’ can be found.

🡪[admin@hostname01 ~]$ grep -l "seema sharma" \*.lst

9. Locate the lines containing saxena and saksena in the Mumbai office.

🡪[admin@hostname01 ~]$ grep -i "saxena\|saksena" mumbai.lst

10. Find the number of managers who earn between 50000 and 99999 in the Pune

office.

🡪[admin@hostname01 ~]$ grep -i "Manager" pune.lst | awk -F'|' '$6 >= 50000 && $6 <= 99999 {print $0}' | wc -l

1

11. List names of employees whose id is in the range 2000 – 2999: in Pune

Office; in all offices.

🡪[admin@hostname01 ~]$ grep -r -E "^[2][0-9]{3}" \*.lst | awk -F'|' '{print $2}'

12. Locate people having same month of birth as current month in Pune office.

🡪[admin@hostname01 ~]$ cuurent\_month=$(date +%m) grep "pune" pune.lst | awk -F'|' -v month="$current\_month" '{if(substr($5,6,2) == month) print $2, $3}'

13. List details of all employees other than those of HR and Admin in file F1.

🡪[admin@hostname01 ~]$ grep -v -E "HR|Admin" F1.lst

14. Locate for all Dwivedi, Trivedi, Chaturvedi in Pune file.

🡪[admin@hostname01 ~]$ grep -i -E "Dwivedi|Trivedi|Chaturvedi" pune.lst

15. Obtain a list of people in HR, Admin and Recr. depts. sorted in reverse order

of the dept.

🡪[admin@hostname01 ~]$ grep -i -E "HR|Admin|Recr." \*.lst | sort -t'|' -k4,4r

pune.lst:3002|Sushil Mahtre |Manager |admin |19939320|35460

[admin@hostname01 ~]$

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

* Date “+%T%A%d%b%Y”

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

🡪 ls -l | sort -k2 -n -r | awk ‘{print $9}’

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

🡪 ls -l |awk ‘{print $9}’ | sort

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

* ls -lS | awk ‘{print $9, $5}’

29: Determine the latest file updated by the user.

* ls -1t |head -n 1