UNIX File System & Permissions

1: Give the execute permission for the user for a file chap1.

- [admin@hostname01 ~]$ chmod u+x chap1

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

- [admin@hostname01 ~]$ chmod a+x add.c

3: Remove the execute permission from user, give read permission to group and others for a file aa.c

- [admin@hostname01 ~]$ chmod u-x,go+r aa.c

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

- [admin@hostname01 ~]$ 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.

- [admin@hostname01 ~]$ cd
- [admin@hostname01 ~]$ ls -d /bin /etc /usr

  /bin /etc /usr

Using Pipes and Filters

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

- [admin@hostname01 ~]$ man ls | col -b > lsdoc

2: Display the content of the lsdoc page wise.

- [admin@hostname01 ~]$ more lsdoc

```
[admin@hostname01 ~]$ more lsdoc
LS(1)                                                   User Commands

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).
       nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
```

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

- [admin@hostname01 ~]$ head -4 lsdoc

```
[admin@hostname01 ~]$ head -4 lsdoc
LS(1)                                   User Commands                                   LS(1)

NAME
       ls - list directory contents
```

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

- [admin@hostname01 ~]$ tail -7 lsdoc

```
[admin@hostname01 ~]$ tail -7 lsdoc
       law.

SEE ALSO
       Full documentation <https://www.gnu.org/software/coreutils/ls>
       or available locally via: info '(coreutils) ls invocation'

GNU coreutils 8.32                              December 2024                           LS(1)
```

5: Remove the file lsdoc.

- [admin@hostname01 ~]$ 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.

- [admin@hostname01 ~]$ cut -d' ' -f2 friends | sort | uniq | wc -l

  4

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

- [admin@hostname01 ~]$ grep "^Ma" friends

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

- [admin@hostname01 ~]$ grep '^Ma.*\([i]\ | [i]d\)' friends

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

- [admin@hostname01 ~]$ find . -print

```
[admin@hostname01 ~]$ find . -print
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/77o4snp9.default-default
./.mozilla/firefox/77o4snp9.default-default/times.json
./.mozilla/firefox/77o4snp9.default-default/.parentlock
./.mozilla/firefox/77o4snp9.default-default/lock
./.mozilla/firefox/77o4snp9.default-default/compatibility.ini
./.mozilla/firefox/77o4snp9.default-default/cookies.sqlite
./.mozilla/firefox/77o4snp9.default-default/prefs.js
./.mozilla/firefox/77o4snp9.default-default/sessionCheckpoints.json
./.mozilla/firefox/77o4snp9.default-default/permissions.sqlite
./.mozilla/firefox/77o4snp9.default-default/storage.sqlite
./.mozilla/firefox/77o4snp9.default-default/storage
./.mozilla/firefox/77o4snp9.default-default/storage/permanent
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/.metadata-v2
```

10: Print only the Directory files.

- [admin@hostname01 ~]$ find . -type d -print

```
[admin@hostname01 ~]$ find . -type d -print
.
./.mozilla
./.mozilla/extensions
./.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
./.mozilla/plugins
./.mozilla/firefox
./.mozilla/firefox/77o4snp9.default-default
./.mozilla/firefox/77o4snp9.default-default/storage
./.mozilla/firefox/77o4snp9.default-default/storage/permanent
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.files
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.files/journals
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/3561288849sdhlie.files
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/1451318868ntouromlalnodry--epcr.files
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/1657114595AmcateirvtiSty.files
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/2823318777ntouromlalnodry--naod.files
./.mozilla/firefox/77o4snp9.default-default/storage/permanent/chrome/idb/2918063365piupsah.files
./.mozilla/firefox/77o4snp9.default-default/storage/temporary
./.mozilla/firefox/77o4snp9.default-default/storage/default
```

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

- [admin@hostname01 ~]$ find . -name "chap*" -print

```
[admin@hostname01 ~]$ find . -name "chap*" -print
./chap0a
./new_dir/chap0a
./new_dir/temp/new_dir/chap0a
./chap1
```

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

- [admin@hostname01 ~]$ sort friends > friends_sorted

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

- [admin@hostname01 ~]$ sort friends > friends_sorted OR
- [admin@hostname01 ~]$ cat friends | tr 'a-z' 'A-Z'

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

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

- [admin@hostname01 ~]$ ls -lR ~ > dir

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

- [admin@hostname01 ~]$ awk ' {print $1, $9} ' dir

```
[admin@hostname01 ~]$ awk ' {print $1, $9} ' dir
/home/admin:
total
-rw-r--r--. aa.c
-rw-r--r--. {a.c,
-rwxr--r--. a.c
-rwxr-xr-x. add.c
drwxr-xr-x. chap0a
-rw-r--r--. Chap0a
-rwxr--r--. chap1
d---------. demo
-rw-rw----. demofile
drwxr-xr-x. Desktop
-rw-r--r--. dir
drwxr-xr-x. Documents
drwxr-xr-x. Downloads
-rw-r--r--. errorlog.txt
-rw-r--r--. filename.txt
-rw-r--r--. first.unix
-rw-r--r--. friends
-rw-r--r--. friends_sorted
```

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

- [admin@hostname01 ~]$ awk ' {print $9} ' dir > files

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

- [admin@hostname01 ~]$ awk ' {print $1} ' dir > perms

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

- [admin@hostname01 ~]$ awk ' {print $5} ' dir > sizes

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

- [admin@hostname01 ~]$ awk ' {print $9, $5, $1} ' dir

```
[admin@hostname01 ~]$ awk ' {print $9, $5, $1} ' dir
  /home/admin:
  total
aa.c 0 -rw-r--r--.
{a.c, 0 -rw-r--r--.
a.c 0 -rwxr--r--.
add.c 0 -rwxr-xr-x.
chap0a 6 drwxr-xr-x.
Chap0a 0 -rw-r--r--.
chap1 0 -rwxr--r--.
demo 20 d---------.
demofile 0 -rw-rw----.
Desktop 6 drwxr-xr-x.
dir 0 -rw-r--r--.
Documents 6 drwxr-xr-x.
Downloads 50 drwxr-xr-x.
errorlog.txt 41 -rw-r--r--.
filename.txt 391 -rw-r--r--.
first.unix 0 -rw-r--r--.
```

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

- [admin@hostname01 ~]$ who | wc -l

  2

21: Find out the smallest file in your directory.

- [admin@hostname01 ~]$ ls -ls | tail -n1

  0 drwxr-xr-x. 2 admin admin   6 Aug 25 2022 Videos   OR

- [admin@hostname01 ~]$ ls -lS | head -1 | awk '{print $9}'

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

- [admin@hostname01 ~]$ wc -l friends

  18 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)

- [admin@hostname01 ~]$ touch emp.list dept.lst desig.lst
- [admin@hostname01 ~]$ vi emp.list
- [admin@hostname01 ~]$ vi dept.lst
- [admin@hostname01 ~]$ vi desig.lst

1. Find the record lengths of each file.
   - [admin@hostname01 ~]$ awk '{print length}' emp.list | head -n 1
     58
   - [admin@hostname01 ~]$ awk '{print length}' dept.lst | head -n 1
     36
   - [admin@hostname01 ~]$ awk '{print length}' desig.lst | head -n 1
     22

2. Display only the date of birth and salary of the last employee record.
   - [admin@hostname01 ~]$ tail -n 1 emp.list | awk -F '|' '{print $5, $6}'
     30-02-2000 45000

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

   Save output as cfile1.

   - [admin@hostname01 ~]$ awk -F '|' '{print $2, $3}' emp.list > cfile1

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

   - [admin@hostname01 ~]$ awk -F '|' '{print $1, $4, $5, $6}' emp.list > cfile2

5. Fix the files cfile1 and cfile2 laterally, along with the delimiter.
    - [admin@hostname01 ~]$ paste -d'|' cfile1 cfile2 > fixed_file
6. Sort the emp.lst file in reverse order of Emp. Names.
    - [admin@hostname01 ~]$ sort -t'|' -k2,2r emp.list

      Empid | Name | Designation | Dept | Date of Birth | Salary

      1003 | Smith | Programmer | IT002 |30-02-2000 | 45000

      1002 | Mike | Analyst | IT002 | 16-10-1977 |48000

      1001 | John | Manager | CS001 | 01-03-1975 | 50000
7. Sort the emp.lst file on the salary field, and store the result in file srtf.
    - [admin@hostname01 ~]$ awk -F '|' '{print $6}' emp.list | sort -n > srtf
8. Sort the emp.ls t file on designation followed by name.
    - [admin@hostname01 ~]$ sort -t'|' -k3,3 -k2,2 emp.list

      Empid | Name | Designation | Dept | Date of Birth | Salary

      1002 | Mike | Analyst | IT002 | 16-10-1977 |48000

      1001 | John | Manager | CS001 | 01-03-1975 | 50000

      1003 | Smith | Programmer | IT002 |30-02-2000 | 45000
9. Sort the emp.lst file on the year of birth.
    - [admin@hostname01 ~]$ sort -t'|' -k5,5 emp.list

      Empid | Name | Designation | Dept | Date of Birth | Salary

      1001 | John | Manager | CS001 | 01-03-1975 | 50000

      1002 | Mike | Analyst | IT002 | 16-10-1977 |48000

      1003 | Smith | Programmer | IT002 |30-02-2000 | 45000

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

- [admin@hostname01 ~]$ awk -F '|' '{print $3}' emp.list | sort | uniq > unique_desig
- [admin@hostname01 ~]$ cut -d'|' -f3 emp.list | sort | uniq
  Analyst
  Manager
  Programmer

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

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

  Analyst

  Manager

  Programmer

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

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

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.list | cut -c1-4 | sort | uniq -c

  1 01-

  1 16-

  1 30-0

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.list | sort | uniq | nl -v100 -i5

  100 Analyst

  105 Manager

  110 Programmer

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' m_emp.lst
  1001 | Manu Sharma | Manager | CS | 19700101 | 80000

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

- [admin@hostname01 ~]$ grep 'Manu Sharma' m_emp.lst | awk -F'|' '{print $1, $4}'
  1001 CS

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

- [admin@hostname01 ~]$ grep 'Manager' p_emp.lst | cat -
  1001 | Manu Sharma | Manager | CS | 19700101 | 80000
  1005 | Ashish Tripathi | Manager | Admin | 19680715 | 90000
- [admin@hostname01 ~]$ grep 'Manager' m_emp.lst | cat -
  1001 | Manu Sharma | Manager | CS | 19700101 | 80000
  1005 | Ashish Tripathi | Manager | Admin | 19680715 | 90000
- [admin@hostname01 ~]$ grep 'Manager' ttc_emp.lst | cat -

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

- [admin@hostname01 ~]$ grep 'S.E.' p_emp.lst | wc -l
  1
- [admin@hostname01 ~]$ grep 'S.E.' m_emp.lst | wc -l
  1
- [admin@hostname01 ~]$ grep 'S.E.' ttc_emp.lst | wc -l
  1

5. List only the Line Numbers and Employee names of employees in 'H/W' in Pune file.

- [admin@hostname01 ~]$ awk -F'|' '$4 == "HW" {print NR, $2}' p_emp.lst

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' m_emp.lst > nonhr

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

- [admin@hostname01 ~]$ awk -F'|' '!($3 ~ /Manager/) {birthdays[$5] = $2} END {min = asort(birthdays); print birthdays[min]}' p_emp.lst m_emp.lst ttc_emp.lst

Vikas Gupta

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

  m_emp.lst

  p_emp.lst

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

- [admin@hostname01 ~]$ grep -i 'saxena\|saksena' m_emp.lst

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

- [admin@hostname01 ~]$ grep 'Manager' p_emp.lst | awk -F'|' '$6 >= 50000 && $6 <= 99999' | wc -l

  2

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 ~]$ awk -F'|' '$1 >=2000 && $1 <= 2999' p_emp.lst | awk '{print $2}'

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

- [admin@hostname01 ~]$ grep -v 'HR' p_emp.lst | grep -v 'Admin' > F1

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

- [admin@hostname01 ~]$ grep -i 'Dwived\|Trivedi\|Chaturve' p_emp.lst

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

- [admin@hostname01 ~]$ grep 'HR' p_emp.lst | cat -
- [admin@hostname01 ~]$ grep 'Admin' p_emp.lst | cat -

  1005 | Ashish Tripathi | Manager | Admin | 19680715 | 90000

- [admin@hostname01 ~]$ grep 'Recr' p_emp.lst | sort -r

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

- [admin@hostname01 ~]$ date +"%H:%M:%S %Z %a %d %b %Y"

  01:28:16 IST Wed 15 Jan 2025

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

- [admin@hostname01 ~]$ ls -l | awk '{print $1, $NF}' | sort -nr | cut -d' ' -f2-

  100

  add.c

  nato

  myfile

  kk.c

  chap1

  a.c

  demofile

  users

  unique_desig

  ttc_emp.lst

  ttc

  srtf

  sizes

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

- [admin@hostname01 ~]$ ls |sort

  =

  aa.c

  {a.c,

a.c

add.c

cfile1

cfile2

chap0a

Chap0a

chap1

demo

demofile

dept.lst

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

- admin@hostname01 ~]$ ls -l |awk '{print $9, $5}'

= 0

aa.c 0

{a.c, 0

a.c 0

add.c 0

cfile1 76

cfile2 140

chap0a 6

Chap0a 0

chap1 0

demo 20

demofile 0

dept.lst 147

desig.lst 66

29: Determine the latest file updated by the user.

- [admin@hostname01 ~]$ ls -1t | head -n1

  F1