**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,g+r,o+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~cd /**

**[admin@hostname01 ~]$ ls -d/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 > lsdoc**

**2: Display the content of the lsdoc page wise.**

**[admin@hostname01 ~]$ less lsdoc**

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

**Ans[admin@hostname01 Desktop]$ head -n 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 Desktop]$ tail -n 7 lsdoc**

**There is NO WARRANTY, to the extent permitted by 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)**

**[admin@hostname01 Desktop]$**

**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**

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

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

**Ans [admin@hostname01 ~]$ grep '^Ma' friends**

**madhu\t6966456\t09/07/68**

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

**Ans [admin@hostname01 ~]$ 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**

**[admin@hostname01 ~]$ find . -type f**

**10: Print only the Directory files.**

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

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

**[admin@hostname01 Desktop]$ find . -type f -name "chap\*"**

**[admin@hostname01 Desktop]$ ls chap\***

**ls: cannot access 'chap\*': No such file or directory**

**[admin@hostname01 Desktop]$ cd ~**

**[admin@hostname01 ~]$ ls chap\***

**chap1**

**[admin@hostname01 ~]$ find . -type f -name "chap\*"**

**./chap1**

**find: ‘./demo’: Permission denied**

**[admin@hostname01 ~]$ sudo find . -type f -name "chap\*"**

**[sudo] password for admin:**

**Sorry, try again.**

**[sudo] password for admin:**

**./chap1**

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

**[admin@hostname01 ~]$ sort friends**

**sort: cannot read: friends: No such file or directory**

**[admin@hostname01 ~]$ cd Desktop/**

**[admin@hostname01 Desktop]$ sort friends**

**Ajay 5546785 01/04/66**

**David 8281292 09/09/60**

**Jamil 2345215 08/09/67**

**Madhu 6966456 09/07/68**

**Mano 7820022 09/07/68**

**Navin 2224311 30/05/68**

**Simmi 7864563 12/12/70:**

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

**[admin@hostname01 Desktop]$ 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 Desktop]$ ls -l ~ >dir**

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

**[admin@hostname01 Desktop]$ awk '{print $1, $9}' dir**

**total**

**-rw-r--r--. aa.c**

**-rw-r--r--. add**

**-rwxr-xr-x. add.c**

**-rwxr--r--. chap1**

**d---------. demo**

**-rw-r--r--. demofile**

**drwxr-xr-x. Desktop**

**drwxr-xr-x. Documents**

**drwxr-xr-x. Downloads**

**-rw-r--r--. errorlog.txt**

**-rw-r--r--. lsdoc**

**drwxr-xr-x. Music**

**-rw-r--r--. network\_services.txt**

**-rw-r--r--. newfriend**

**drwxr-xr-x. Pictures**

**drwxr-xr-x. Public**

**drwxr-xr-x. Templates**

**-rw-r--r--. users**

**drwxr-xr-x. Videos**

**-rwxr-xr-x. working\_shell.sh**

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

**[admin@hostname01 Desktop]$ awk '{print $1, $9}' dir**

**total**

**-rw-r--r--. aa.c**

**-rw-r--r--. add**

**-rwxr-xr-x. add.c**

**-rwxr--r--. chap1**

**d---------. demo**

**-rw-r--r--. demofile**

**drwxr-xr-x. Desktop**

**drwxr-xr-x. Documents**

**drwxr-xr-x. Downloads**

**-rw-r--r--. errorlog.txt**

**-rw-r--r--. lsdoc**

**drwxr-xr-x. Music**

**-rw-r--r--. network\_services.txt**

**-rw-r--r--. newfriend**

**drwxr-xr-x. Pictures**

**drwxr-xr-x. Public**

**drwxr-xr-x. Templates**

**-rw-r--r--. users**

**drwxr-xr-x. Videos**

**-rwxr-xr-x. working\_shell.sh**

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

**:[admin@hostname01 Desktop]$ awk '{print $9}' dir >files**

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

**[admin@hostname01 Desktop]$ awk '{print $5}' dir > sizes**

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

**[admin@hostname01 Desktop]$ awk '{print $5}' dir > sizes**

**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 -n 1**

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

**:[admin@hostname01 Desktop]$ wc -l friends**

**7 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.**

**[admin@hostname01 ~]$ awk '{print length}' emp.lst**

**47**

**[admin@hostname01 ~]$ awk '{print length}' dept.lst**

**12**

**[admin@hostname01 ~]$ awk '{print length}' desig.lst**

**12**

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

**[admin@hostname01 ~]$ tail -n 1 emp.lst | awk -F'|' '{print $5, $6}'**

**180513. Extract only employee names and designations. (Use column specifications).**

**Save output as cfile1.**

**[admin@hostname01 ~]$ cut -d'|' -f2,3 emp.lst > cfile1**

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

**cfile2.**

**[admin@hostname01 ~]$ cut -d'|' -f2,3 emp.lst > cfile1**

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

**[admin@hostname01 ~]$ paste -d'|' cfile1 cfile2**

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

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

**1002 |Riri |Engineer |GET |187500101 |50000998 50000**

**7. 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**

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

**[admin@hostname01 ~]$1002 |Riri |Engineer |GET |187500101 |50000**

**9. 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**

**.**

**Engineer**

**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**

**2. 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**

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

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

**4. 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**

**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**

**25: Write a command sequence that prints out date information in this order: time,**

**day of week, day number, month, year :**

**[admin@hostname01 ~]$ date "+%T %A %d %b %Y"**

**00:24:54 Tuesday 14 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 | sort -k2 -n -r | awk '{print $9}'**

**Desktop**

**styles**

**Styles**

**Videos**

**Templates**

**Public**

**Pictures**

**Music**

**Documents**

**Downloads**

**demo**

**dev**

**fixed\_file**

**sizes**

**networrk.txt**

**pune.lst**

**ttc.lst**

**nonhr**

**mumbai.lst**

**27: Write a command sequence that prints only names of files in current working**

**directory in alphabetical order.**

**[admin@hostname01 ~]$ ls -l | sort**

**d---------. 2 root root 21 Jan 12 18:06 demo**

**drwxr-xr-x. 10 admin admin 184 Jan 13 18:42 Desktop**

**drwxr-xr-x. 2 admin admin 50 Aug 25 2022 Downloads**

**drwxr-xr-x. 2 admin admin 6 Aug 25 2022 Documents**

**drwxr-xr-x. 2 admin admin 6 Aug 25 2022 Music**

**drwxr-xr-x. 2 admin admin 6 Aug 25 2022 Pictures**

**drwxr-xr-x. 2 admin admin 6 Aug 25 2022 Public**

**drwxr-xr-x. 2 admin admin 6 Aug 25 2022 Templates**

**total 148**

**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 -lS | awk '{print $9, $5}'**

**dev 64470**

**dir 1086**

**networrk.txt 523**

**perms 246**

**Desktop 184**

**files 158**

**fixed\_file 57**

**sizes 57**

**29: Determine the latest file updated by the user.**

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

**nonhr**