**UNIX File System & Permissions**

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

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

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

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

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

**Using Pipes and Filters**

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

**ls --help > lsdoc**

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

**more 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 lsdo**

**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.**

**awk '{print $3}' friend | sort | uniq | wc -l**

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

**grep '^ma' friend**

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

**grep '^ma.\*\(i\|id\)$' friend**

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

**find . -type f**

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

**find . -type d**

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

**find . -type f -name 'chap\***

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

**sort friend**

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

**tr ‘a-z’ ‘A-Z’ < friend**

**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.**

**-l $(cat dir) | awk '{print $1, $9}'**

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

**ls -l $(cat dir) | awk '{print $9}' > files**

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

**ls -l $(cat dir) | awk '{print $1}' > perms**

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

**ls -l $(cat dir) | awk '{print $5}' > 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 | tail -n 1**

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

**wc -l 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.**

**awk -F ‘|’ ‘{print length($0)}’ emp.lst**

**awk -F ‘|’ ‘{print length($0)}’ dept.lst**

**awk -F ‘|’ ‘{print length($0)}’ desig.lst**

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

**tail -n 1 emp.lst | awk -F ‘|’ ‘{print $4, $5}**

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

**Save output as cfile1.**

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

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

**cfile2.**

**awk -F ‘|’ ‘{print $1, $4, $5, $6}’ emp.lst > cfile2**

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

**Paste -d ‘,’ cfile1 cfile2 > merged\_file**

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

**sort -t ‘|’ -k2,2r emp.lst**

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

**Sort -t ‘|’ -k6,6n emp.lst > srtf**

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

**sort -t ‘|’ -k3,3 -k2,2 emp.lst**

**9. Sort the emp.lst file on the year of birth.**

**sort -t ‘|’ -k5,5 emplis**

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

**awk -F ‘|’ ‘{print $3}’ emp.lst | sort | uniq**

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

**awk -F ‘|’ ‘{print $3}’ emp.lst | sort |uniq -u**

**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.**

**awk -F ‘|’ ‘{print $3}’ emp.lst | sort | uniq -c**

**14. Use nl command to create a code table for designations to include designation**

**code (Start with dept. code 100, and subsequently 105, 110 …).**

**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 %Z %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 | 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 -lt –time=modify**