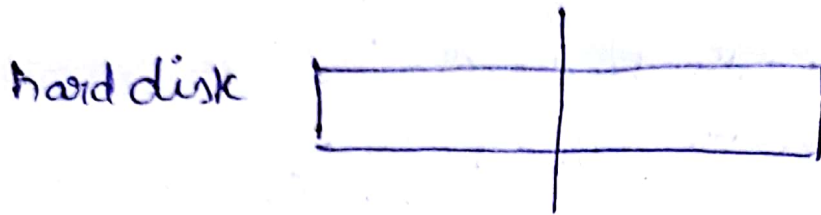


## Directory and its structure



divide into 2 partitions based on our requirement

partitions

volumes

minidisk

In order to give information about each file each volume has at least one directory.

Directory :- Directory can be viewed as a symbol table which translates file names into its entries.

(a) Directory is a special type of file which contain information about the file

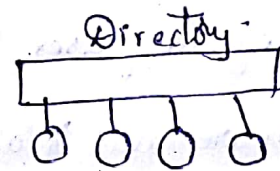
Directory contain information about the file like name, type of file, access rights, date of creation of files.

different types of operations that <sup>on</sup> a directory performs

- (i) search
- (ii) create
- (iii) list
- (iv) traverse (if no. of levels  $\uparrow$  then traversing should be there)
- (v) delete
- (vi) rename

Different types of structures of directories.

1. Single level
2. two level
3. tree based structured
4. Acyclic graph directory
5. general graph directory



① Single level directory :- In this directory there is only one directory, all the file information is there in the directory.

advantages:- Creation is simple, searching is easy as it is a single directory, deletion is also easy.

disadvantages:- as the no. of files  $\uparrow$  then there will be a naming conflict when you want to create a new file.

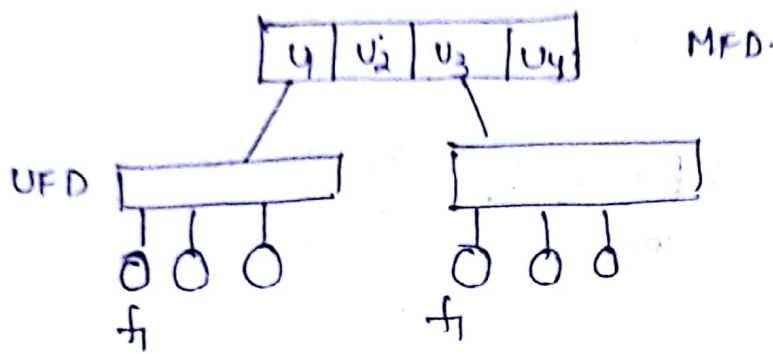
(i) as there is a unique directory for all users, there is a threat to my files. So, protect our files from other users.

(ii) we can't separate different users.

MS-DOS allows 11 characters for naming.

UNIX has 255 characters for naming.

(ii) Two level directory structure:-



MFD is master file directory contain information of different users and his respected user file directory. once any user login the MFD checks the login credentials if they are valid it allows the respective user to his own directory (UFD) so, he can access his only file so protection problem is solved.

advantage:- different users can have same naming files as the one user can't access the others file so, protection is there to files.

disadvantage:- It does not allow to share files b/w different users.

E: mostly main problem is there with the system files as the system files are not there in the directory of user so it is a problem

sol<sup>n</sup> ①: Copying the system files in each directory.

It will be waste of memory so, go to sol<sup>n</sup> ②



Solution ②: All the system files are placed in MFD. then.

if user gives a command if the file with <sup>the given</sup> name is present in the UFD then it takes and runs it if not there it goes to the MFD or SFD which is present in MFD and executes it.

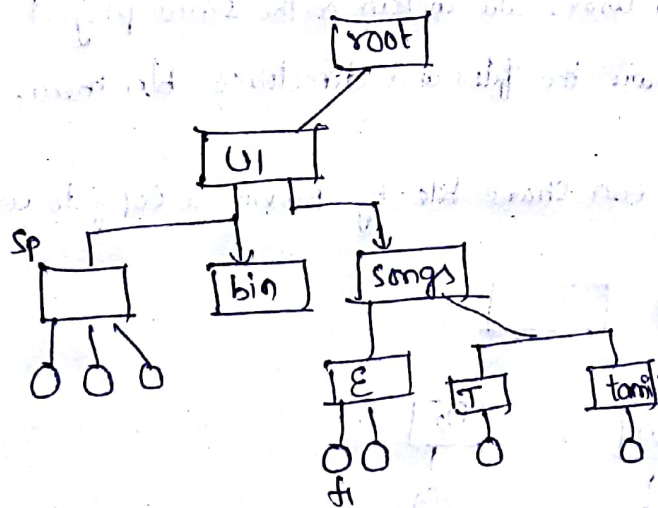
It is.

### ⑤ Tree structured directory :-

If the user wants to store different files in ~~some~~ <sup>one</sup> directory. So, there is no way to place in two level directory. So we are going with tree structured directory.

It contains "root" directory. contains user level directory.

The user can create files as well as directories which contain file and subdirectories also.



here files can be present at any where so, in order to access a

file we are going with path. There are two types of paths.

- ① Absolute path
- ② relative path

Absolute path :-

If specifying the file from the root to the file then it is absolute path.

root / u1 / songs / E / fi.type

Relative path :- If we give path from the current working directory.

let user is in songs then path is E / fi.type

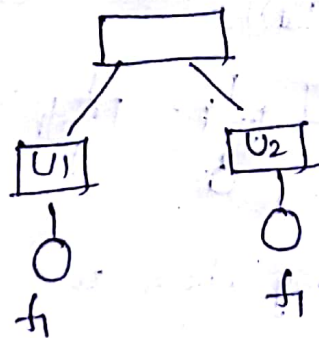
advantages :- user can create his own directory and subdirectories, "sharing of the files is done using permissions"

disadvantages :- Searching file is difficult as the no of <sup>levels</sup> ~~file~~ increases.

(iv) Acyclic graph directory :-

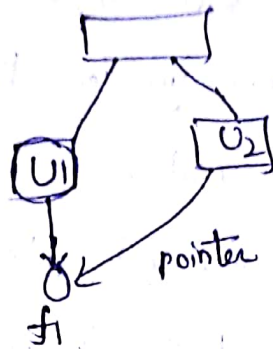
If two users are working on the same project that is necessary to share the files and directories b/w them.

Solution :- I can share file by providing a copy to user 2



if  $f_1$  has modified it does not effect in file in the user 2  
so, it is not workout.

Solution 2: make one copy of file and make no. of people to make copy point to the respective file.



advantages :- here what changes are made in the shared file will be there (or) reflected in both users. So, sharing is achieved.

as it is a closed fig we can't say it as a tree, so, it is a graph but it is directed so it is Directed Acyclic graph.

disadvantages:- If one file is pointed by many no. of users or directories if any one of the user deleted it but if the user (Other) pointing to the same address but there is no file it is called "dangling problem". So, in order to avoid this we follow a data structure implementation. whenever there is a file pointed by pointer directories

by using ds all the pointers are placed so that I can know all the pointers. So, before deleting my file I delete all the pointers so that there is no problem in the deleting the file.



If the no. of files in the d's ↑ es. it may be a wastage of memory.

So, we go with solution ~~at~~ implementation is make one counter where ever you see a link increase the counter value. If any link is deleted decrement the counter value. if the counter value is '0' then I can know that all the ~~for~~ links to the files are deleted so, I can delete file there will be no problem.