**18. simulate the file organization techniques:**

**a.SINGLE LEVEL DIRECTORY:**

**AIM:**

Implementation of single level directory.

**DESCRIPTION:**

The simplest method is to have one big list of all the files on the disk. The entire system will contain only one directory which is supposed to mention all the files present in the file system. The directory contains one entry per each file present on the file system.

Advantages

1. Implementation is very simple.
2. If the sizes of the files are very small then the searching becomes faster.
3. File creation, searching, deletion is very simple since we have only one directory.

Disadvantages

1. We cannot have two files with the same name.
2. The directory may be very big therefore searching for a file may take so much time.
3. Protection cannot be implemented for multiple users.
4. There are no ways to group same kind of files.
5. Choosing the unique name for every file is a bit complex and limits the number of files in the system because most of the Operating System limits the number of characters used to construct the file name.

**ALGORITHM:**

**Step 1:** start

**Step 2:** master,s[20] as integer type

f[20][20][20] as character type

d[20][20] as character type

i,j as integer type

**Step 3:** read"enter number of directories:"(master)

**Step 4:**  write("enter names of directories:")

for(i=0;i<master;i++)

begin

read(d[i])

end for

**Step 5:** write("enter size of directories:")

for(i=0;i<master;i++)

begin

read(s[i])

end for

**Step 6:** write("enter the file names :")

for(i=0;i<master;i++)

begin

write"enter file names of directory %d"

for(j=0;j<s[i];j++)

begin

read(f[i][j])

end for

end for

**Step 7:** write(" directory\tsize\tfilenames\n")

write("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

for(i=0;i<master;i++)

begin

write("%s\t\t%2d\t",d[i],s[i])

for(j=0;j<s[i];j++)

begin

write("%s\n\t\t\t",f[i][j])

end for

end for

end

**Step 8:** stop

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

main()

{

int master,s[20];

char f[20][20][20];

char d[20][20];

int i,j;

printf("enter number of directorios:");

scanf("%d",&master);

printf("enter names of directories:");

for(i=0;i<master;i++)

{

scanf("%s",&d[i]);

}

printf("enter size of directories:");

for(i=0;i<master;i++)

{

scanf("%d",&s[i]);

}

printf("enter the file names :");

printf("\n");

for(i=0;i<master;i++)

{

printf("enter file names of directory %d",i+1);

for(j=0;j<s[i];j++)

{

scanf("%s",&f[i][j]);

}

}

printf(" directory\tsize\tfilenames\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for(i=0;i<master;i++)

{

printf("%s\t\t%2d\t",d[i],s[i]);

for(j=0;j<s[i];j++)

{

printf("%s\n\t\t\t",f[i][j]);

}

printf("\n");

}

printf("\t\n");

}

**OUTPUT:**

enter number of directories : 3

enter names of directories : A B C

enter size of directories : 2 3 2

enter the file names :

enter file names of directory 1 A1 A2

enter file names of directory 2 B1 B2 B3

enter file names of directory 3 C1 C2

directory size filenames

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A 2 A1

A2

B 3 B1

B2

B3

C 2 C1

C2

**b.TWO LEVEL DIRECTORY:**

**AIM:**

Implementation of two level directory.

**DESCRIPTION:**

In two level directory systems, we can create a separate directory for each user. There is one master directory which contains separate directories dedicated to each user. For each user, there is a different directory present at the second level, containing group of user's file. The system doesn't let a user to enter in the other user's directory without permission.

Characteristics of two level directory system

1. Each files has a path name as ***/User-name/directory-name/***
2. Different users can have the same file name.
3. Searching becomes more efficient as only one user's list needs to be traversed.
4. The same kind of files cannot be grouped into a single directory for a particular user.

**ALGORITHM:**

**Step 1:** start

**Step 2:** declare structure st with fields

dname[10], sdname[10][10], fname[10][10][10] as character type

ds,sds[10] as integer type

**Step 3:** declare dir[10] as structure variable of type structure st

**Step 4:** declare i , j , k , n as integer type

**Step 5:** read(enter number of directories)(n)

**Step 6:** for(i=0;i<n;i++)

begin

read(enter directory %d names:)(dir[i].dname)

read(enter size of directories:)(dir[i].ds);

**Step 7:** for(j=0;j<dir[i].ds;j++)

begin

read(enter subdirectory name and size:)(dir[i].sdname[j])(dir[i].sds[j])

for(k=0;k<dir[i].sds[j];k++)

begin

read(enter file name:)(dir[i].fname[j][k])

end for

end for

end for

**Step 8:** write"\ndirname\t\tsize\tsubdirname\tsize\tfiles"

write"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

**Step 9:** for(i=0;i<n;i++)

begin

write("%s\t\t%d",dir[i].dname,dir[i].ds)

**Step 10:** for(j=0;j<dir[i].ds;j++)

begin

write"\t%s\t\t%d\t",dir[i].sdname[j],dir[i].sds[j]

for(k=0;k<dir[i].sds[j];k++)

begin

write"%s\t",dir[i].fname[j][k]

end for

end for

end for

end

**Step 11:** stop

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

struct st

{

char dname[10];

char sdname[10][10];

char fname[10][10][10];

int ds,sds[10];

}dir[10];

void main()

{

int i,j,k,n;

printf("enter number of directories:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("enter directory %d names:",i+1);

scanf("%s",&dir[i].dname);

printf("enter size of directories:");

scanf("%d",&dir[i].ds);

for(j=0;j<dir[i].ds;j++)

{

printf("enter subdirectory name and size:");

scanf("%s",&dir[i].sdname[j]);

scanf("%d",&dir[i].sds[j]);

for(k=0;k<dir[i].sds[j];k++)

{

printf("enter file name:");

scanf("%s",&dir[i].fname[j][k]);

}

}

}

printf("\ndirname\t\tsize\tsubdirname\tsize\tfiles");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for(i=0;i<n;i++)

{

printf("%s\t\t%d",dir[i].dname,dir[i].ds);

for(j=0;j<dir[i].ds;j++)

{

printf("\t%s\t\t%d\t",dir[i].sdname[j],dir[i].sds[j]);

for(k=0;k<dir[i].sds[j];k++)

{

printf("%s\t",dir[i].fname[j][k]);

}

printf("\n\t\t");

}

printf("\n");

}

}

**OUTPUT:**

enter number of directories : 3

enter directory 1 names : A1

enter size of directories : 2

enter subdirectory name and size: A11 2

enter file name:A111

enter file name:A112

enter subdirectory name and size: A12 3

enter file name:A121

enter file name:A122

enter file name:A123

enter directory 2 names:B1

enter size of directories:3

enter subdirectory name and size: B11 1

enter file name:B111

enter subdirectory name and size: B12 2

enter file name:B121

enter file name:B122

enter subdirectory name and size: B13 1

enter file name:B131

enter directory 3 names:C1

enter size of directories:1

enter subdirectory name and size: C11 2

enter file name:C111

enter file name:C112

dirname size subdirname size files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A1 2 A11 2 A111 A112

A12 3 A121 A122 A123

B1 3 B11 1 B111

B12 2 B121 B122

B13 1 B131

C1 1 C11 2 C111 C112