

Ex. No. 14a Single-Level Directory**Date:****Aim**

To organize files in a single level directory structure, I.e., without sub-directories.

Algorithm

1. Get name of directory for the user to store all the files
2. Display menu
3. Accept choice
4. If choice =1 then
 - Accept filename without any collision
 - Store it in the directory
5. If choice =2 then
 - Accept filename
 - Remove filename from the directory array
6. If choice =3 then
 - Accept filename
 - Check for existence of file in the directory array
7. If choice =4 then
 - List all files in the directory array
8. If choice =5 then
 - Stop

Program

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

struct
{
    char dname[10];
    char fname[25][10];
    int fcnt;
}dir;

main()
{
    int i, ch;
    char f[30];
    clrscr();
    dir.fcnt = 0;
    printf("\nEnter name of directory -- ");
    scanf("%s", dir.dname);

    while(1)
    {
        printf("\n\n 1. Create File\t2. Delete File\t3. Search
        File \n4. Display Files\t5. Exit\nEnter your choice--");
        scanf("%d", &ch);
```

```

switch(ch)
{
    case 1:
        printf("\n Enter the name of the file -- ");
        scanf("%s", dir.fname[dir.fcnt]);
        dir.fcnt++;
        break;

    case 2:
        printf("\n Enter the name of the file -- ");
        scanf("%s", f);
        for(i=0; i<dir.fcnt; i++)
        {
            if(strcmp(f, dir.fname[i]) == 0)
            {
                printf("File %s is deleted ",f);
                strcpy(dir.fname[i], dir.fname[dir.fcnt-1]);
                break;
            }
        }
        if(I == dir.fcnt)
            printf("File %s not found", f);
        else
            dir.fcnt--;
        break;

    case 3:
        printf("\n Enter the name of the file -- ");
        scanf("%s", f);
        for(i=0; i<dir.fcnt; i++)
        {
            if(strcmp(f, dir.fname[i]) == 0)
            {
                printf("File %s is found ", f);
                break;
            }
        }
        if(I == dir.fcnt)
            printf("File %s not found", f);
        break;

    case 4:
        if(dir.fcnt == 0)
            printf("\n Directory Empty");
        else
        {
            printf("\n The Files are -- ");
            for(i=0; i<dir.fcnt; i++)
                printf("\t%s", dir.fname[i]);
        }
        break;
}

```

```

        default:
            exit(0);
    }
}
getch();
}

```

Output

Enter name of directory -- CSE

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 1

Enter the name of the file -- fcfs

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 1

Enter the name of the file -- sjf

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 1

Enter the name of the file -- lru

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 3

Enter the name of the file -- sjf

File sjf is found

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 3

Enter the name of the file -- bank

File bank is not found

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 4

The Files are -- fcfs sjf lru bank

1. Create File 2. Delete File 3. Search File
4. Display Files 5. Exit

Enter your choice -- 2

Enter the name of the file -- lru

File lru is deleted

Result

Thus files were organized into a single level directory.

Ex. No. 14b Two-Level Directory**Date:****Aim**

To organize files as two-level directory with each user having his own user file directory (UFD).

Algorithm

1. Display menu
2. Accept choice
3. If choice =1 then
 - Accept directory name
 - Create an entry for that directory
4. If choice =2 then
 - Get directory name
 - If directory exist then accept filename without collision else report error
5. If choice =3 then
 - Get directory name
 - If directory exist then Get filename
 - If file exist in that directory then delete entry else report error
6. If choice =4 then
 - Get directory name
 - If directory exist then Get filename
 - If file exist in that directory then Display filename else report error
7. If choice =5 then Display files directory-wise
8. If choice =6 then Stop

Program

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

struct
{
    char dname[10], fname[10][10];
    int fcnt;
}dir[10];

main()
{
    int i, ch, dcnt, k;
    char f[30], d[30];
    clrscr();
    dcnt=0;
    while(1)
    {
        printf("\n\n 1. Create Directory\t 2. Create File\t 3.
Delete File");
        printf("\n 4. Search File \t \t 5. Display \t 6. Exit \n
Enter your choice -- ");
```

```

scanf("%d", &ch);
switch(ch)
{
    case 1:
        printf("\n Enter name of directory -- ");
        scanf("%s", dir[dcnt].dname);
        dir[dcnt].fcnt = 0;
        dcnt++;
        printf("Directory created");
        break;

    case 2:
        printf("\n Enter name of the directory -- ");
        scanf("%s", d);
        for(i=0; i<dcnt; i++)
            if(strcmp(d,dir[i].dname) == 0)
            {
                printf("Enter name of the file -- ");
                scanf("%s", dir[i].fname[dir[i].fcnt]);
                dir[i].fcnt++;
                printf("File created");
                break;
            }
        if(i == dcnt)
            printf("Directory %s not found",d);
        break;

    case 3:
        printf("\nEnter name of the directory -- ");
        scanf("%s", d);
        for(i=0; i<dcnt; i++)
        {
            if(strcmp(d,dir[i].dname) == 0)
            {
                printf("Enter name of the file -- ");
                scanf("%s", f);
                for(k=0; k<dir[i].fcnt; k++)
                {
                    if(strcmp(f, dir[i].fname[k]) == 0)
                    {
                        printf("File %s is deleted ", f);
                        dir[i].fcnt--;
                        strcpy(dir[i].fname[k],
                            dir[i].fname[dir[i].fcnt]);
                        goto jmp;
                    }
                }
                printf("File %s not found",f);
                goto jmp;
            }
        }
    }
}

```

```

        printf("Directory %s not found",d);
    jmp : break;

    case 4:
        printf("\nEnter name of the directory -- ");
        scanf("%s", d);
        for(i=0; i<dcnt; i++)
        {
            if(strcmp(d,dir[i].dname) == 0)
            {
                printf("Enter the name of the file -- ");
                scanf("%s", f);
                for(k=0; k<dir[i].fcnt; k++)
                {
                    if(strcmp(f, dir[i].fname[k]) == 0)
                    {
                        printf("File %s is found ", f);
                        goto jmp1;
                    }
                }
                printf("File %s not found", f);
                goto jmp1;
            }
        }
        printf("Directory %s not found", d);
    jmp1: break;

    case 5:
        if(dcnt == 0)
            printf("\nNo Directory's ");
        else
        {
            printf("\nDirectory\tFiles");
            for(i=0;i<dcnt;i++)
            {
                printf("\n%s\t\t",dir[i].dname);
                for(k=0;k<dir[i].fcnt;k++)
                    printf("\t%s",dir[i].fname[k]);
            }
        }
        break;

    default:
        exit(0);
    }
}
getch();
}

```

Output

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 1

Enter name of directory -- CSE

Directory created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 1

Enter name of directory -- ECE

Directory created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 2

Enter name of the directory -- ECE

Enter name of the file -- amruth

File created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 2

Enter name of the directory -- CSE

Enter name of the file -- kowshik

File created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 2

Enter name of the directory -- CSE

Enter name of the file -- pranesh

File created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 2

Enter name of the directory -- ECE

Enter name of the file -- ajith

File created

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 5

Directory Files

CSE kowshik pranesh

ECE amruth ajith

1. Create Directory 2. Create File 3. Delete File 4. Search File 5. Display 6. Exit

Enter your choice -- 3

Enter name of the directory -- ECE

Enter name of the file -- ajith

File ajith is deleted

Result

Thus user files have been stored in their respective directories and retrieved easily.