|  |  |
| --- | --- |
| **Ex.No:14.b** | **FILE ORGANIZATION TECHNIQUE** |
| **TWO LEVEL DIRECTORY** |

# AIM:

To write C program to organize the file using two level directory.

# ALGORITHM:

Step-1: Start the program.

Step-2: Declare the count, file name, graphical interface. Step-3: Read the number of files

Step-4: Read the file name

Step-5: Declare the root directory

Step-6: Using the file eclipse function define the files in a single level Step-7: Display the files

Step-8: Stop the program

# PROGRAM:

#include<stdio.h> #include<graphics.h> struct tree\_element

{

char name[20];

int x,y,ftype,lx,rx,nc,level; struct tree\_element \*link[5];

};

typedef struct tree\_element node; void main()

{int gd=DETECT,gm; node \*root; root=NULL;

clrscr(); create(&root,0,"null",0,639,320); clrscr(); initgraph(&gd,&gm,"c:\tc\bgi"); display(root);

getch(); closegraph();

}

create(node \*\*root,int lev ,char \*dname,int lx,int rx,int x)

{int i, gap;

 if(\*root==NULL)

{(\*root)=(node\*)malloc(sizeof(node)); printf("enter the name of dir file name %s",dname); fflush(stdin);

gets((\*root)->name); if(lev==0 || lev==1) (\*root)-> ftype=1; else

(\*root)->ftype=2; (\*root)->level=lev; (\*root)->y=50+lev\*50; (\*root)->x=x;

(\*root)->lx=lx ; (\*root)->rx=rx; for(i=0;i<5;i++)

(\*root)->link[i]=NULL; if((\*root)->ftype==1)

{if(lev==0 || lev==1)

{if((\*root)->level==0) printf("how many users"); else

printf(" how many files"); printf("(for %s):",(\*root)->name);

scanf("%d",&(\*root)->nc);

}else

(\*root)->nc=0; if((\*root)->nc==0) gap=rx-lx;

else

gap=(rx-lx)/(\*root)->nc; for(i=0;i<(\*root)->nc;i++)

create(&((\*root)->link[i]),lev+1,(\*root)->name,lx+gap\*i,lx+gap\*i+gap,lx+gap\*i+gap/2);

}else

(\*root)->nc=0;

}}

display(node \*root)

{int i; settextstyle(2,0,4); settextjustify(1,1); setfillstyle(1,BLUE); setcolor(14); if(root!=NULL)

{for(i=0;i<root->nc;i++)

{line(root->x,root->y,root->link[i]->x,root->link[i]->y);

}if(root->ftype==1)

bar3d(root->x-20, root->y-10,root->x+20,root->y+10,0,0);

else

fillellipse(root->x,root->y,20,20); outtextxy(root->x,root->y,root->name); for(i=0;i<root->nc;i++)

{display(root->link[i]);

}}}