**Week –**

Simulate the following File Organization Techniques

a) Single level directory b) Two level Directory

**Code:**

#include <stdio.h>

#include <string.h>

char a[100][100];

char u[100][100];

char f[100][100][100];

int i = 0, l = 0, m = 0;

int search(char f[]) {

for (int j = 0; j <i; j++) {

if (strcmp(a[j], f) == 0) {

return (j);

}

}

return (-1);

}

int searchuser(char f[]) {

for (int j = 0; j < l; j++) {

if (strcmp(u[j], f) == 0) {

return (j);

}

}

return (-1);

}

void createfile(char f[]) {

if (search(f) == -1) {

strcpy(a[i], f);

i++;

} else {

printf("invalid");

}

}

void createuser(char f[]) {

if (searchuser(f) == -1) {

strcpy(u[l], f);

l++;

} else {

printf("invalid");

}

}

void traverse(void) {

for (int j = 0; j <i; j++) {

printf("%s", a[j]);

}

}

void traverseuser(void) {

for (int j = 0; j < l; j++) {

printf("%s", u[j]);

}

}

void renam(char f[], char g[]) {

int k = search(f);

strcpy(a[k], g);

}

void renamuser(char f[], char g[]) {

int k = searchuser(f);

strcpy(u[k], g);

}

void delet(char f[]) {

int j = search(f);

for (int k = j; k <i - 1; k++) {

strcpy(a[k], a[k + 1]);

}

i--;

}

void deletuser(char f[]) {

int j = searchuser(f);

for (int k = j; k < l - 1; k++) {

strcpy(u[k], u[k + 1]);

}

l--;

}

int searchuf(char user[], char p[]) {

int k = searchuser(user);

int j;

for (j = 0; j < m; j++) {

if (strcmp(f[k][j], p) == 0)

return (j);

}

return (-1);

}

void createuf(char user[], char p[]) {

if (searchuser(user) == -1) {

printf("invalid");

} else {

if (searchuf(user, p) == -1) {

int k = searchuser(user);

strcpy(f[k][m], p);

m++;

} else

printf("invalid");

}

}

void traverseuf(char user[]) {

int k = searchuser(user);

for (int j = 0; j < m; j++) {

printf("%s", f[k][j]);

}

}

void renamuf(char user[], char p[], char t[]) {

int k = searchuser(user);

int v = searchuf(user, p);

strcpy(f[k][v], t);

}

void deletuf(char user[], char p[]) {

int k = searchuser(user);

int v = searchuf(user, p);

for (int n = v; n < m - 1; n++) {

strcpy(f[k][n], f[k][n + 1]);

}

m--;

}

void sld(void) {

int y;

char s[100], s2[100];

while (y) {

int x;

printf("enter 1:create 2: traverse 3: search 4:delet 5:rename 6:exit");

scanf("%d", &x);

switch (x) {

case 1:

printf("enter file");

scanf("%s", &s);

createfile(s);

break;

case 2:

traverse();

break;

case 3:

printf("enter file");

scanf("%s", &s);

if (search(s) == -1)

printf("false");

else

printf("true");

break;

case 4:

printf("enter file");

scanf("%s", &s);

delet(s);

break;

case 5:

printf("enter old name");

scanf("%s", &s);

printf("enter new");

scanf("%s", &s2);

renam(s, s2);

break;

}

printf("want to cont");

scanf("%d", &y);

}

}

void tld(void) {

int y;

char s[100], s2[100], s1[100];

while (y) {

int x;

printf("enter 1:createu 2: traverseu 3: searchu 4:deletu 5:renameu 6:createuf 7:searchuf 8: traverseuf 9:deletuf 10:renamuf");

scanf("%d", &x);

switch (x) {

case 1:

printf("enter user");

scanf("%s", &s);

createuser(s);

break;

case 2:

traverseuser();

break;

case 3:

printf("enter file");

scanf("%s", &s);

if (searchuser(s) == -1)

printf("false");

else

printf("true");

break;

case 4:

printf("enter file");

scanf("%s", &s);

deletuser(s);

break;

case 5:

printf("enter old name");

scanf("%s", &s);

printf("enter new");

scanf("%s", &s2);

renamuser(s, s2);

break;

case 6:

printf("enter user");

scanf("%s", &s);

printf("enter file");

scanf("%s", &s2);

createuf(s, s2);

break;

case 7:

printf("enter user");

scanf("%s", &s);

printf("enter file");

scanf("%s", &s2);

if (searchuf(s, s2) != -1)

printf("found");

else

printf("not found");

break;

case 8:

printf("enter user");

scanf("%s", &s);

traverseuf(s);

break;

case 9:

printf("enter user");

scanf("%s", &s);

printf("enter file");

scanf("%s", &s2);

deletuf(s, s2);

break;

case 10:

printf("enter user");

scanf("%s", &s);

printf("enter old file");

scanf("%s", &s2);

printf("enter new file");

scanf("%s", &s1);

renamuf(s, s2, s1);

break;

}

printf("want to cont");

scanf("%d",&y);

}

}

void main() {

int n;

printf("enter 1:sld 2:tld");

scanf("%d", &n);

switch (n) {

case 1:

sld();

break;

case 2:

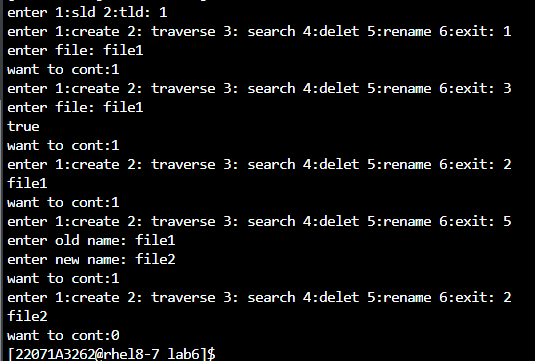
tld();

break;

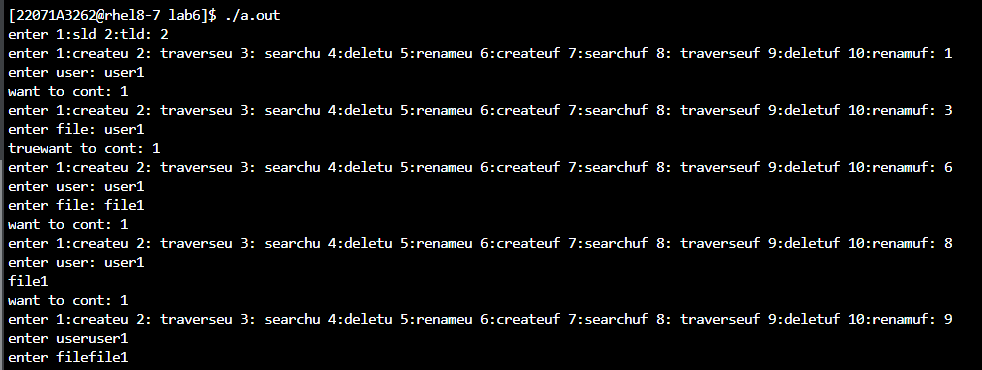
}

}

**Output for Single Level directory:**

****

**Output for Two Level directory:**

****