12.Best fit

#include<stdio.h>

int main()

{

int fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999;

static int barray[20],parray[20];

printf("\n\t\t\tMemory Management Scheme - Best Fit");

printf("\nEnter the number of blocks:");

scanf("%d",&nb);

printf("Enter the number of processes:");

scanf("%d",&np);

printf("\nEnter the size of the blocks:-\n");

for(i=1;i<=nb;i++)

{

printf("Block no.%d:",i);

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

}

printf("\nEnter the size of the processes :-\n");

for(i=1;i<=np;i++)

{

printf("Process no.%d:",i);

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

}

for(i=1;i<=np;i++)

{

for(j=1;j<=nb;j++)

{

if(barray[j]!=1)

{

temp=b[j]-p[i];

if(temp>=0)

if(lowest>temp)

{

parray[i]=j;

lowest=temp;

}

}

}

fragment[i]=lowest;

barray[parray[i]]=1;

lowest=10000;

}

printf("\nProcess\_no\tProcess\_size\tBlock\_no\tBlock\_size\tFragment");

for(i=1;i<=np && parray[i]!=0;i++)

printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,p[i],parray[i],b[parray[i]],fragment[i]);

}

13.single level directory system

#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

int nf=0,i=0,j=0,ch;

char mdname[10],fname[10][10],name[10];

printf("enter the directory name:");

scanf("%s,mdname");

printf("enter the number of files:");

scanf("%d",&nf);

do

{

printf("enter file name to be created:");

scanf("%s",name);

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

{

if(!strcmp(name,fname[i]))

break;

}

if(i==nf)

{

strcpy(fname[j++],name);

nf++;

}

else

printf("there is already %s\n",name);

printf("do you want to enter another file(yes -1 or no - 0):");

scanf("%d",&ch);

}

while(ch==1);

printf("directory name is:%s\n",mdname);

printf("files names are:");

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

printf("\n%s",fname[i]);

getch();

return 0;

}

14. page replacement

#include <stdio.h>

#define MAX\_FRAMES 3

int main()

{

int page\_frames[MAX\_FRAMES] = {0};

int page\_faults = 0;

int page\_reference[] = {7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1};

int num\_pages = sizeof(page\_reference) / sizeof(page\_reference[0]);

for (int i = 0; i < MAX\_FRAMES; i++) {

page\_frames[i] = -1;

}

for (int i = 0; i < num\_pages; i++) {

int page = page\_reference[i];

int found = 0;

for (int j = 0; j < MAX\_FRAMES; j++) {

if (page\_frames[j] == page) {

found = 1;

break;

}

}

if (!found) {

int lru\_frame = 0;

int lru\_time = page\_frames[0];

for (int j = 1; j < MAX\_FRAMES; j++) {

if (page\_frames[j] == -1) {

lru\_frame = j;

break;

}

if (page\_frames[j] < lru\_time) {

lru\_frame = j;

lru\_time = page\_frames[j];

}

}

page\_frames[lru\_frame] = page;

page\_faults++;

}

}

printf("Number of page faults: %d\n", page\_faults);

return 0;

}