**Week-5**

Implementation of the following Memory Allocation Methods for fixed partition

a) First Fit b) Worst Fit c) Best Fit

**Code:**

#include<stdio.h>

void firstfit(int a[], int n, int b[], int n1){

int i, j, s = 0;

char c[n1];

for(i = 0; i< n1; i++){

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

if(b[i] <= a[j]){

printf("%d %d %c %d\n", i, b[i], 'y', a[j]);

a[j] = a[j] - b[i];

c[i] = 'y';

break;

}

}

if(c[i] != 'y')

printf("%d %d %c\n", i, b[i], 'n');

}

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

s = s + a[i];

}

printf("fragmentation %d\n", s);

}

void desc(int a[], int n){

int i, j, t;

for(i = 0; i< n - 1; i++){

for(j = i + 1; j < n; j++){

if(a[i] < a[j]){

t = a[i];

a[i] = a[j];

a[j] = t;

}

}

}

}

void asen(int a[], int n){

int i, j, t;

for(i = 0; i< n - 1; i++){

for(j = i + 1; j < n; j++){

if(a[i] > a[j]){

t = a[i];

a[i] = a[j];

a[j] = t;

}

}

}

}

void bestfit(int a[], int n, int b[], int n1){

int i, j,s=0;

char c[n1];

for(i = 0; i< n1; i++){

asen(a, n);

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

if(b[i] <= a[j]){

printf("%d %d %c %d\n", i, b[i], 'y', a[j]);

a[j] = a[j] - b[i];

c[i] = 'y';

break;

}

}

if(c[i] != 'y')

printf("%d %d %c\n", i, b[i], 'n');

}

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

s = s + a[i];

}

printf("fragmentation %d\n", s);

}

void worstfit(int a[], int n, int b[], int n1){

int i, j,s=0;

char c[n1];

for(i = 0; i< n1; i++){

desc(a, n);

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

if(b[i] <= a[j]){

printf("%d %d %c %d\n", i, b[i], 'y', a[j]);

a[j] = a[j] - b[i];

c[i] = 'y';

break;

}

}

if(c[i] != 'y')

printf("%d %d %c\n", i, b[i], 'n');

}

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

s = s + a[i];

}

printf("fragmentation %d\n", s);

}

int main(){

int n, n1, i,x,f=0;

printf("Enter number of holes: ");

scanf("%d", &n);

printf("Enter number of processes: ");

scanf("%d", &n1);

while(1){

int a[n], b[n1];

printf("Enter hole sizes: ");

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

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

}

printf("Enter process sizes: ");

for(i = 0; i< n1; i++) {

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

}

printf("select 1:firstFit 2:worstFit 3:bestFit : ");

scanf("%d",&x);

switch(x){

case 1:firstfit(a,n,b,n1);

break;

case 2:worstfit(a,n,b,n1);

break;

case 3:bestfit(a,n,b,n1);

break;

default:printf("re enter");

}

printf("want to continue 1/0: ");

scanf("%d",&f);

if(f==0)

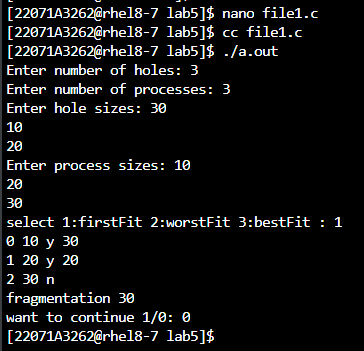
break;

}

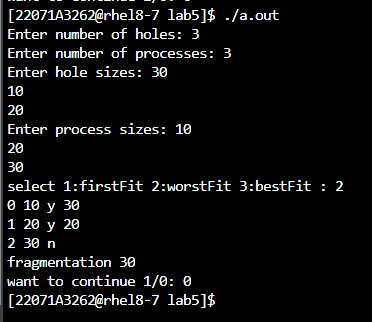
return 0;

}

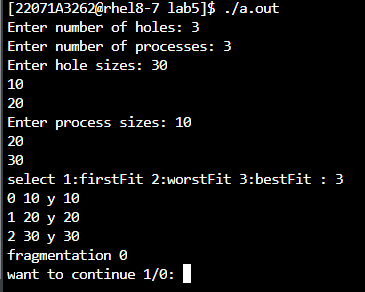
**Output for first fit:**

****

**Output for worst fit:**

****

**Output for best fit:**

****