**Week – 10**

**Aim:**

Simulate frame allocation Methods.

a) minimum Number of frames b) equal allocation c) proportional allocation

**Code:**

#include<stdio.h>

#include<unistd.h>

void mnf(int fs, int n) {

int mf, p;

printf("Enter min no of Frames allocated: ");

scanf("%d", &mf);

printf("Enter no of processes: ");

scanf("%d", &p);

int pr[p], rf[p];

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

printf("Enter process %d size: ", i);

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

rf[i] = pr[i] / fs;

if(pr[i] % fs != 0)

rf[i] += 1;

}

printf("Process\tProcessSize\tReqFrames\tallocatedFrames\n");

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

printf("%d\t%d\t%d\t%d\n", i, pr[i], rf[i], mf);

}

int k = n - (p \* mf);

printf("left over frames:%d\n", k);

}

void ea(int fs, int n) {

int p;

printf("Enter no of processes: ");

scanf("%d", &p);

int pr[p], rf[p];

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

printf("Enter process %d size: ", i);

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

rf[i] = pr[i] / fs;

if(pr[i] % fs != 0)

rf[i] += 1;

}

Int mf = n / p;

printf("Process\tProcessSize\tReqFrames\tallocatedFrames\n");

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

printf("%d\t%d\t%d\t%d\n", i, pr[i], rf[i], mf);

}

int k = n - (mf \* p);

printf("Left Over Frames:%d\n", k);

}

void pa(int fs, int n) {

int p, s = 0;

printf("Enter no of processes: ");

scanf("%d", &p);

int pr[p], rf[p], mf[p];

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

printf("Enter process %d size: ", i);

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

rf[i] = pr[i] / fs;

if(pr[i] % fs != 0)

rf[i] += 1;

s += pr[i];

}

int k = n;

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

mf[i] = (pr[i] \* n) / s;

k -= mf[i];

}

printf("Process\tProcessSize\tReqFrames\tallocatedFrames\n");

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

printf("%d\t%d\t%d\t%d\n", i, pr[i], rf[i], mf[i]);

}

printf("Left Over Frames:%d\n", k);

}

int main() {

int n, fs, x;

printf("Enter Frame size: ");

scanf("%d", &fs);

printf("Enter no of Frames: ");

scanf("%d", &n);

do {

printf("1.MNF 2.EA 3.PA\n");

scanf("%d", &x);

switch(x) {

case 1:

mnf(fs, n);

break;

case 2:

ea(fs, n);

break;

case 3:

pa(fs, n);

break;

case 4:

break;

}

} while(x != 4);

}

**Output:**

