**CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY AND RESEARCH** | | | | | |
| **Department of Computer Science & Engineering** | | | | | |
| **Student ID** | : | 19DCS098 | **Student Name** | : | PARTH NITESHKUMAR PATEL |
| **Subject Code** | : | CS350 | **Subject Name** | : | OPERATING SYSTEM |
| **Date of exam** | : | 16 NOV 2021 |  |  |  |

**DEFINITION:**

Simulate Optimal Page Replacement Algorithm

**PROGRAM CODE:**

#include<stdio.h>

int

main ()

{

int numberOfFrames;

int numberOfPages;

int frames[10];

int pages[30];

int temp[10];

int flag\_1, flag\_2, flag\_3;

int i, j, k;

int pos;

int max;

int faults = 0;

printf ("ENTER THE TOTAL NUMBER OF FRAMES: ");

printf ("\n-------------------------------------\n");

scanf ("%d", &numberOfFrames);

printf ("\n-------------------------------------\n");

printf ("ENTER THE NUMBER OF THE PAGES: ");

printf ("\n-------------------------------------\n");

scanf ("%d", &numberOfPages);

printf ("\n-------------------------------------\n");

printf ("ENTER THE PAGE REFERENCE STRING: ");

printf ("\n-------------------------------------\n");

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

{

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

}

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

{

frames[i] = -1;

}

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

{

flag\_1 = flag\_2 = 0;

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

{

if (frames[j] == pages[i])

{

flag\_1 = flag\_2 = 1;

break;

}

}

if (flag\_1 == 0)

{

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

{

if (frames[j] == -1)

{

faults++;

frames[j] = pages[i];

flag\_2 = 1;

break;

}

}

}

if (flag\_2 == 0)

{

flag\_3 = 0;

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

{

temp[j] = -1;

for (k = i + 1; k < numberOfPages; ++k)

{

if (frames[j] == pages[k])

{

temp[j] = k;

break;

}

}

}

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

{

if (temp[j] == -1)

{

pos = j;

flag\_3 = 1;

break;

}

}

if (flag\_3 == 0)

{

max = temp[0];

pos = 0;

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

{

if (temp[j] > max)

{

max = temp[j];

pos = j;

}

}

}

frames[pos] = pages[i];

faults++;

}

printf ("\n");

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

{

printf ("%d\t", frames[j]);

}

}

printf ("\n-------------------------------------\n");

printf ("\n\nTotal Page Faults = %d", faults);

printf ("\n-------------------------------------\n");

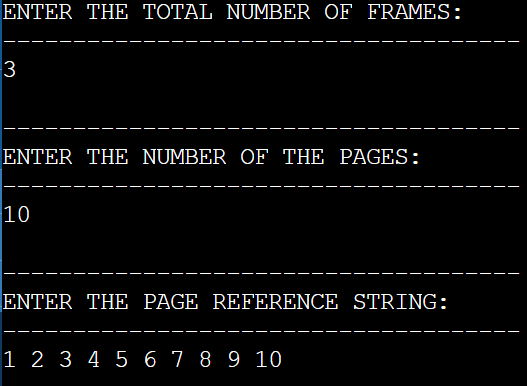
printf ("\nPARTH PATEL\n19DCS098\n");

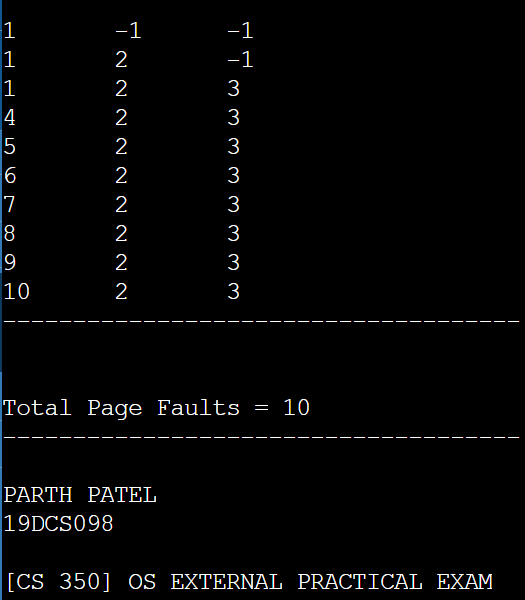
printf ("\n[CS 350] OS EXTERNAL PRACTICAL EXAM");

return 0;

}

**OUTPUT:**





**DEFINITION:**

Write a C program in LINUX to perform Worst Fit Memory allocation algorithms and Calculate Internal and External Fragmentation.

**PROGRAM CODE:**

#include<stdio.h>

#define MAX 30

void

main ()

{

static int bf[MAX];

static int ff[MAX];

int fragment[MAX];

int blocks[MAX];

int f[MAX];

int i, j;

int numberOfBlocks, numberOfFiles;

int temp;

printf ("WORST FIT");

printf ("\n------------------------------------------\n");

printf ("\nENTER THE TOTAL NUMBER OF THE BLOCKS : ");

printf ("\n------------------------------------------\n");

scanf ("%d", &numberOfBlocks);

printf ("\n------------------------------------------\n");

printf ("ENTER THE NUMBER OF FILES : ");

printf ("\n------------------------------------------\n");

scanf ("%d", &numberOfFiles);

printf ("\n------------------------------------------\n");

printf ("\nENTER THE SIZE OF THE BLOCKS :\n");

printf ("\n------------------------------------------\n");

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

{

printf ("BLOCK->%d : ", i);

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

}

printf ("\n------------------------------------------\n");

printf ("ENTER THE SIZE OF THE FILES : \n");

printf ("\n------------------------------------------\n");

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

{

printf ("FILE->%d : ", i);

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

}

printf ("\n------------------------------------------\n");

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

{

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

{

if (bf[j] != 1)

{

temp = blocks[j] - f[i];

if (temp >= 0)

{

ff[i] = j;

break;

}

}

}

fragment[i] = temp;

bf[ff[i]] = 1;

}

printf

("\n----------------------------------------------------------------------------\n");

printf ("\nFileNumber\tSizeOfFile\tBlockNumber\tSizeOfBlock\tFragements");

printf

("\n------------------------------------------------------------------------------\n");

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

printf ("\n%d\t\t%d\t\t%d\t\t%d\t\t\t%d\n", i, f[i], ff[i],

blocks[ff[i]], fragment[i]);

printf ("\n------------------------------------------\n");

printf ("\nPARTH PATEL\n19DCS098");

printf ("\n------------------------------------------\n");

printf ("\n[CS 350] OS EXTERNAL PRACTICAL EXAM");

printf ("\n------------------------------------------\n");

}

**OUTPUT:**

