Operating systems lab

Ex 09 – paging techniques

Sreyas v

Cse c 185001162

QUEUE.H

typedef int Data;

typedef struct Node{

Data d;

struct Node \*next;

}Node;

typedef Node\* Queue;

int isEmpty(Queue front, Queue rear){

if (front == NULL)

return 1;

return 0;

}

int size(Queue front, Queue rear){

if(isEmpty(front, rear))

return 0;

int c = 0;

Node \*tmp = front;

while(tmp){

tmp = tmp -> next;

c++;

}

return c;

}

void enqueue(Queue \*front, Queue \*rear, const Data d){

Node\* new = (Node\*)malloc(sizeof(Node));

new -> d = d;

new -> next = NULL;

if(isEmpty(\*front, \*rear))

(\*front) = (\*rear) = new;

else{

(\*rear) -> next = new;

(\*rear) = new;

}

}

Data dequeue(Queue \*front, Queue \*rear){

Data rVal =0;

if(isEmpty(\*front, \*rear))

return rVal;

Node \*tmp = (\*front);

rVal = (\*front) -> d;

if (\*front == \*rear)

(\*rear) = NULL;

(\*front) = (\*front) -> next;

free(tmp);

return rVal;

}

void display(Queue front, Queue rear){

if(isEmpty(front, rear)){

printf(" Empty Queue!\n");

return;

}

Queue tmp = front;

while(tmp){

printf("% d", tmp -> d);

tmp = tmp -> next;

}

printf("\n");

}

MAIN PROGRAM

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#define MAX 10

typedef struct Process{

unsigned int pid;

unsigned int pages[MAX];

unsigned int n\_pages;

}Process;

#include "Queue.h"

void allocate(Queue\*, Queue\*, Process\*, int, int, int);

void deallocate(Queue\*, Queue\*, Process\*, const int);

void pagetable(Process\*, const int);

int main(){

Process p[MAX];

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

p[i].pid = -1;

p[i].n\_pages=0;

}

Queue front = NULL,

rear = NULL;

int phy\_mem = -1,

frame\_size = -1,

choice = -1,

no\_frames = -1,

pid = -1,

mem\_reqd = -1;

printf(" Enter the Size of Physical Memory: ");

scanf("%d", &phy\_mem);

printf(" Enter the Page Size: ");

scanf("%d", &frame\_size);

no\_frames = phy\_mem / frame\_size;

for(int i = 0; i < 12; i++)

enqueue(&front, &rear, (random() % no\_frames) +1);

while(1){

printf("\t\t\t\tPAGING TECHNIQUES\n");

printf(" 1 - Process Request\n");

printf(" 2 - De-Allocation\n");

printf(" 3 - Page table for all input processes\n");

printf(" 4 - Free Frame List\n");

printf(" 0 - Exit\n");

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

printf(" Enter your choice: ");

scanf("%d", &choice);

switch(choice){

case 0:

exit(0);

case 1:

printf("\n Enter the PID of the Process & Memory Required: ");

scanf("%d %d", &pid, &mem\_reqd);

allocate(&front, &rear, p, pid, mem\_reqd, frame\_size);

break;

case 2:

printf("\n Enter the PID of Process to De-Allocate: ");

scanf("%d", &pid);

deallocate(&front, &rear, p, pid);

break;

case 3:

for(int i = 0; i < MAX; i++)

pagetable(p, i);

break;

case 4:

printf("\n The List of Free Frames:\n");

display(front, rear);

break;

default:

printf(" Invalid Input!\n");

break;

}

}

}

void allocate(Queue \*front, Queue \*rear, Process \*p, int pid, int mem\_reqd, int page\_size){

int no\_pages = mem\_reqd / page\_size;

if(no\_pages > size(\*front, \*rear)){

printf(" Insufficient Memory!\n");

return;

}

if(p[pid].pid != -1){

printf(" Duplicate PID!\n");

return;

}

printf("\n Process is divided into %d Pages", no\_pages);

p[pid].pid = pid;

p[pid].n\_pages = no\_pages;

for(int i = 0; i < no\_pages; i++)

p[pid].pages[i] = dequeue(front, rear);

pagetable(p, pid);

printf(" Successfully Allocated Pages!\n");

}

void deallocate(Queue \*front, Queue \*rear, Process \*p, const int pid){

if(!p[pid].n\_pages){

printf(" No Such Process exists!\n");

return;

}

p[pid].pid = -1;

int n\_pages = p[pid].n\_pages;

p[pid].n\_pages = 0;

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

enqueue(front, rear, p[pid].pages[i]);

printf(" Successfully De-Allocated Process!\n");

}

void pagetable(Process\* p, const int pid){

if(p[pid].n\_pages == 0)

return;

printf("\n\n Page Table for Process %d\n", pid);

for(int i = 0; i < p[pid].n\_pages; i++)

printf(" Page %-2d : Frame %-2d\n", i, p[pid].pages[i]);

printf("\n");

}

OUTPUT

C:\Users\sreya\Desktop\labs\OS\Paging>gcc paging.c -o m

C:\Users\sreya\Desktop\labs\OS\Paging>m

Enter the Size of Physical Memory: 32

Enter the Page Size: 1

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

10 4 31 5 2 13 23 15 19 17 10 18

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 1

Enter the PID of the Process & Memory Required: 1 4

Process is divided into 4 Pages

Page Table for Process 1

Page 0 : Frame 10

Page 1 : Frame 4

Page 2 : Frame 31

Page 3 : Frame 5

Successfully Allocated Pages!

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

2 13 23 15 19 17 10 18

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 1

Enter the PID of the Process & Memory Required: 2 3

Process is divided into 3 Pages

Page Table for Process 2

Page 0 : Frame 2

Page 1 : Frame 13

Page 2 : Frame 23

Successfully Allocated Pages!

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

15 19 17 10 18

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 2

Enter the PID of Process to De-Allocate: 1

Successfully De-Allocated Process!

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

15 19 17 10 18 10 4 31 5

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 1

Enter the PID of the Process & Memory Required: 3 6

Process is divided into 6 Pages

Page Table for Process 3

Page 0 : Frame 15

Page 1 : Frame 19

Page 2 : Frame 17

Page 3 : Frame 10

Page 4 : Frame 18

Page 5 : Frame 10

Successfully Allocated Pages!

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

4 31 5

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 1

Enter the PID of the Process & Memory Required: 5 2

Process is divided into 2 Pages

Page Table for Process 5

Page 0 : Frame 4

Page 1 : Frame 31

Successfully Allocated Pages!

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 4

The List of Free Frames:

5

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 3

Page Table for Process 2

Page 0 : Frame 2

Page 1 : Frame 13

Page 2 : Frame 23

Page Table for Process 3

Page 0 : Frame 15

Page 1 : Frame 19

Page 2 : Frame 17

Page 3 : Frame 10

Page 4 : Frame 18

Page 5 : Frame 10

Page Table for Process 5

Page 0 : Frame 4

Page 1 : Frame 31

PAGING TECHNIQUES

1 - Process Request

2 - De-Allocation

3 - Page table for all input processes

4 - Free Frame List

0 - Exit

------------------------------------------

Enter your choice: 0