**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

по дисциплине: «Операционные системы»

на тему: **«**Алгоритмы замещения страниц**»**

Выполнил: студент гр. ИТП-41

Лисицин Н.С.

Принял: преподаватель-стажёр

Карась О.В.

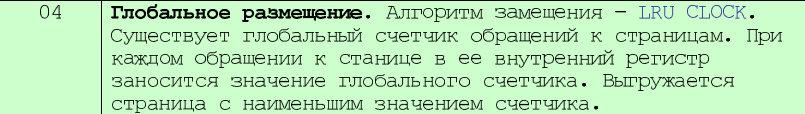
Гомель 2022

**Цель:** изучить способы создания алгоритмов замещения страниц.

**Задача:**

Разработать программу реализующая заданный алгоритм замещение страниц в память

**Вариант 4:**

****

**Ход работы.**

**.**

*LRU* – это один из приближенных к алгоритму ОРТ является алгоритм, исходящий из эвристического правила, что недавнее прошлое – хороший ориентир для прогнозирования ближайшего будущего

Для реализации этой утилиты был написана программа на языке программирования *C*. При первом запуске программы пользователя просят ввести количество фреймов, количество страниц и строку.

Результат представлен на рисунке 1.



Рисунок 1 – Результат ввода данных

В результате мы получаем количество страниц и информацию о их готовности и расположении в фрейме памяти.

Результат представлен на рисунке 2.

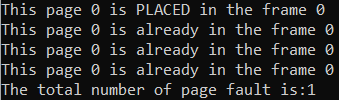


Рисунок 2 – Результат вывода информации о страницах

В данной лабораторной работе было необходимо реализовать систему управления виртуальной памятью в виде консольного приложения, которое и было создано. Данное приложение чрезвычайно просто, удобно и в то же время очень многофункционально, умеет выполнять задание лабораторной работы и даже больше (например умеет выключаться без выключения компьютера и без нажатия *Alt+F4*).

**Вывод:** в ходе выполнения данной лабораторной работы были получены навыки по созданию консольного приложения на языке *С*++ которое выполняет алгоритм замещения страниц *LRU* *ClOCK*.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Листинг программы**

***Lab*10.*cpp*:**

#*include*<*iostream*>

*using* *namespace* *std*;

*int* *max*(*int* *r*[],*int* *n*)

{

*int* *pos*,*i*,*l*,*max*=0;

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

{

*if*(*r*[*i*]>*max*)

{

*l*=*r*[*i*];

*max*=*r*[*i*];

*pos*=*i*;

}

}

*return* *pos*;

}

*void* *read*(*int* *a*[],*int* *n*)

{

*int* *i*;

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

{

*cin*>>*a*[*i*];

}

}

*void* *display*(*int* *b*[],*int* *n*)

{

*int* *i*;

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

{

*cout*<<*b*[*i*]<<" ";

}

}

*void* *timer*\_*confg*(*int* *tmr*[],*int* *pos*)

{

*int* *j*=*pos*;

*switch*(*pos*)

{

*case* 0:

*tmr*[*j*]=0;

*tmr*[*j*+1]=*tmr*[*j*+1]++;

*tmr*[*j*+2]=*tmr*[*j*+2]++;

*break*;

*case* 1:

*tmr*[*j*]=0;

*tmr*[*j*-1]=*tmr*[*j*-1]++;

*tmr*[*j*+1]=*tmr*[*j*+1]++;

*break*;

*case* 2:

*tmr*[*j*]=0;

*tmr*[*j*-1]=*tmr*[*j*-1]++;

*tmr*[*j*-2]=*tmr*[*j*-2]++;

*break*;

}

}

*int* *main*()

{

*int* *pg*[20],*tmr*[20],*i*,*j*,*n*,*pf*=0,*rs*[20],*m*,*avail*=0;

*cout*<<"*Enter* *number* *of* *frames*: ";

*cin*>>*n*;

*for*(*i*=0;*i*<20;*i*++)

{

*pg*[*i*]=-1;

}

*cout*<<"*Enter* *the* *number* *of* *pages*:";

*cin*>>*m*;

*cout*<<"*Enter* *the* *reference* *string*: ";

*read*(*rs*,*m*);

*int* *pos*;

*tmr*[0]=2;

*tmr*[1]=1;

*tmr*[2]=0;

*for*(*i*=0;*i*<*m*;*i*++)

{

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

{

*if*(*pg*[*j*]==-1)

{

*pg*[*j*]=*rs*[*i*];

*avail*=1;

*pf*++;

*cout*<<"*This* *page* "<<*rs*[*i*]<<" *is* *PLACED* *in* *the* *frame* "<<*j*<<*endl*;

*break*;

}

*else*

{

*if*(*pg*[*j*]==*rs*[*i*])

{

*pos*=*j*;

*timer*\_*confg*(*tmr*,*pos*);

*avail*=1;

*cout*<<"*This* *page* "<<*rs*[*i*]<<" *is* *already* *in* *the* *frame* "<<*j*<<*endl*;

*break*;

}

*else*

{

*if*(*avail*==0 || *j*==(*n*-1))

{

*pos*=*max*(*tmr*,*n*);

*pg*[*pos*]=*rs*[*i*];

*pf*++;

*timer*\_*confg*(*tmr*,*pos*);

*cout*<<"*The* *page* "<<*rs*[*i*]<<" *is* *replaced* *with* *frame* "<<*pos*<<*endl*;

*break*;

}

}

}

}

}

*cout*<<"*The* *total* *number* *of* *page* *fault* *is*:"<<*pf*;

*return* 0;

}