Міністерство освіти і науки України

Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет прикладної математики

Кафедра Спеціалізованих комп’ютерних систем

Лабораторна робота №3

З дисципліни «Операційні системи» :

«Керування пам’яттю»

Варіант №16

Виконав:

студент групи КВ-91

Федай Г. Г.

Перевірив:

Зайцев В. В.

Київ 2012

**Завдання:**

При моделировании алгоритмов без использования внешней памяти считать что исходные адреса каждого незагруженного процесса начинается с 0000..00, размеры каждого из процессов должны задаваться произвольно. Каждому адресу незагруженного процесса по запросу следует указать реальный адрес памяти если процесс загружен.

Загрузку и выгрузку ( при необходимости) процессов выполнять в соответствии с конкретно заданной очередью.

При моделировании алгоритмов с использованием внешней памяти следует задать виртуальное адресное пространство каждого процесса. Обращение к памяти выполнять по виртуальному адресу, осуществлять при необходимости загрузку и выгрузку соответствующих частей процессов, распечатывая значения конкретного физического адреса, который соответствует заданному виртуальному.

При моделировании алгоритмов работы КЭШ-памяти адресное пространство основной и КЭШ-памяти может задаваться произвольно при соотношении объемов КЭШ-памяти и основной не менее 1:10. Поиск, запись и замещение информации в КЭШ-памяти должно выполняться путем задания искомых адресов основной памяти.

## Разработать модель алгоритма поиска и замещения данных в КЭШ - памяти для

## детерминированного отображения данных на КЭШ. Согласование данных основной и КЭШ-памяти выполнить по принципу обратной записи.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace OS3

{

class MemCell

{

public int num\_req;

public int addr;

public int cash\_addr;

public int size;

public int val;

public bool modified;

public MemCell()

{

cash\_addr = -1;

addr = -1;

modified = false;

val = -1;

}

public MemCell(MemCell mc)

{

this.addr = mc.addr;

this.cash\_addr = mc.cash\_addr;

this.val = mc.val;

this.modified = mc.modified;

}

}

class Mems

{

int mem\_size;

int cash\_size;

int capacity;

int amount;

MemCell[] main\_memory;

MemCell[] cash\_memory;

public Mems()

{

capacity = 16;

mem\_size = 2048;

main\_memory = new MemCell[mem\_size / capacity];

Console.WriteLine("number is {0}", mem\_size / capacity);

for (int i = 0; i < main\_memory.Length; i++)

{

main\_memory[i] = new MemCell();

main\_memory[i].addr = i \* capacity;

main\_memory[i].val = i;

}

cash\_size = 256;

cash\_memory = new MemCell[cash\_size / capacity];

Console.WriteLine("cash number is {0}", cash\_size / capacity);

for (int i = 0; i < cash\_memory.Length; i++)

{

cash\_memory[i] = new MemCell();

cash\_memory[i].addr = -1;

cash\_memory[i].cash\_addr = i \* capacity;

}

amount = mem\_size / cash\_size;

}

public void PushToCash(MemCell cell) //input data to cash memory

{

int idx = cell.addr / (capacity\*amount);

if (cash\_memory[idx].addr > 0 && cash\_memory[idx].modified == true )

{

cash\_memory[idx].modified = false;

main\_memory[cash\_memory[idx].addr / capacity].val = cash\_memory[idx].val;

}

cash\_memory[idx].addr = cell.addr;

cash\_memory[idx].val = cell.val;

}

public bool DCHangeInCash(int value, int addr)

{

int idx = addr / (capacity\*amount);

if (cash\_memory[idx].addr == addr)

{

cash\_memory[idx].val = value;

cash\_memory[idx].addr = addr;

cash\_memory[idx].modified = true;

return true;

}

return false;

}

public int ReadFromCash(int addr)

{

int idx = addr / (capacity \* amount);

if (cash\_memory[idx].addr == addr)

return cash\_memory[idx].val;

return -1;

}

public void ReadData(int addr)

{

int num;

if ( ( num = ReadFromCash(addr)) > 0)

{

Console.WriteLine("Кеш попадання\nRead from cash {0}", num);

return;

}

int idx = addr / capacity;

Console.WriteLine("Кеш промах\nRead data from memory {0}", main\_memory[idx].val);

PushToCash(main\_memory[idx]);

Console.WriteLine("added new record to cash\n");

}

public void ChangeData(int value, int addr)

{

if (DCHangeInCash(value, addr))

{

Console.WriteLine("Кеш попадання");

return;

}

Console.WriteLine("Кеш промах");

int idx = addr / capacity;

PushToCash(main\_memory[idx]);

Console.WriteLine("added new record to cash\n");

}

public void PrintCash()

{

for (int i = 0; i < cash\_size / capacity; i++)

Console.WriteLine("cash addr {0} addr {1} val {2} modified {3}", cash\_memory[i].cash\_addr, cash\_memory[i].addr, cash\_memory[i].val, cash\_memory[i].modified);

}

public void PrintMem()

{

for (int i = 0; i < mem\_size / capacity; i++)

Console.WriteLine("addr {0} val {1} ", main\_memory[i].addr, main\_memory[i].val);

}

}

class Program

{

static void Main(string[] args)

{

Mems memory = new Mems();

memory.PrintMem();

memory.ReadData(128);

memory.ReadData(544);

memory.ReadData(32);

memory.PrintCash();

memory.ChangeData(678, 128);

memory.ReadData(128);

memory.ReadData(144);

memory.PrintCash();

memory.PrintMem();

}

}

}