“Operating System” Experimental Guide

Experiment 5: Virtual Memory Page Replacement Algorithms

Experimental Hours: 6 hours recommended

Experimental Platform: Visual C++ or Visual Studio

[Objective]

Through this experiment, we can deepen our understanding of the concept of virtual memory page replacement, and further grasp the implementation methods of FIFO(First In First Out Page Replacement), OPT(Optimal Page Replacement) and LRU(Least Recently Used Page Replacement) algorithms.

[Content]

Problem Description:

The design program simulates the working process of FIFO, OPT and LRU page replacement algorithms. Assuming that the minimum number of physical blocks allocated to each process in memory is m, the number of pages to be accessed during the process running is n, and the reference string is P1,... Pn. Using different page replacement algorithms to schedule the pages, gives the page replacement process, calculates the total number of page fault and the page fault rate of each algorithm.

The program requirements are as follows:

1) Three page replacement algorithms, FIFO, OPT and LRU, are used to simulate the page replacement process.

2) The page replacement process of three algorithms is simulated, and the memory allocation of each page is given.

3) **Input**: Minimum number of physical blocks m, number of pages n, the reference string is P1,... Pn, and the algorithms selection 1-FIFO, 2-OPT, 3-LRU.

4) **Output**: The total number of page fault and the page fault rate of each algorithm.

Implementation Tips:

- Realize three functions or objects for the page replacement algorithms: FCFS, OPT, LRU.

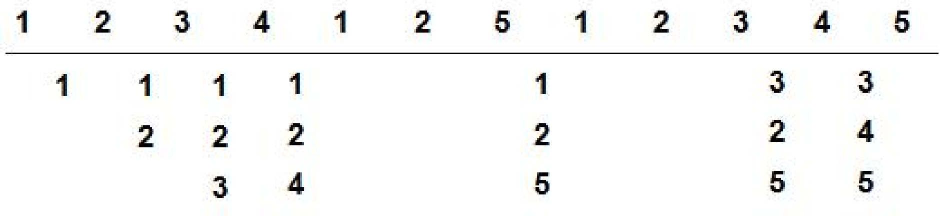
1. The process of each page replacement is as follows:

   (1) Variable initialization;

   (2) Receiving user input minimum number of physical blocks m, number of pages n, page sequence P1,... Pn, and the select number (1-FIFO, 2-OPT, 3-LRU).

   (3) Page replacement are carried out according to the algorithm selected by users, and the simulation process of page replacement algorithm is output.

   For example,



1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1			1			3	3	
	2	2	2			2			2	4	
		3	4			5			5	5	

Or

1: 1
2:1,2
3:1,2,3
4:1,2,4
1:hit
2:hit
5:1,2,5
1:hit
2:hit
3:3,2,5
4:3,4,5
5:hit

(4) Calculate the number of page fault and the rate of page fault in the selection algorithm.

(5) Return the algorithm page fault and the page fault rate.

2. The program variable definition reference (which can be added as needed) is as follows:

```
Const int MaxNumber = 100;
Int PageOrder [MaxNumber];
Int Simulate [MaxNumber] [MaxNumber];
Int PageNum;
Int TotalPageFault;
Double PageFaultRate;
Bool found;
```

- In the main function, to complete:

1. Input the minimum number of physical blocks m, number of pages n, the reference string is P1,... Pn, and the algorithms selection 1-FIFO, 2-OPI, 3-LRU.

2. Invoke the function of the page replacement algorithm that user selects, and output the page fault and the rate of the page fault of the selection algorithms.

[Requirements]:

According to the specific experimental requirements, complete the experimental report, including:

(1) Requirements analysis

- The task of programming with ambiguous statements emphasizes what the program should do. And clearly stipulates:

- The form of input and the range of input values;
- The form of output;
- The functions that the procedure can achieve;
- Test data: including correct input and output results, and incorrect input and output results.

(2) Outline design

Describes the definition of all the abstract data types used in this program, the flow of the main program, and the hierarchical (invocation) relationship between the program modules.

(3) **Detailed design**:Realize the specific algorithm of program module.

(4) Debug analysis

The recommended contents may include:

- How to solve the problems encountered in the debugging process and the analysis of the design and implementation;
- Spatio-temporal analysis of the algorithm (including the analysis of time and space complexity of basic operations and other algorithms) and improvement ideas;
- Experience.

(5) **User instructions:** Explains how to use the program you write and lists each step in detail.

(6) **Test results:**

List your test results, including input and output. The test data here should be complete and strict. Attach the screenshot of the program results.

(7) **Appendix**. Annotated source program.