
LUT
University
Operating Systems and System Programming
LAND
OF THE
CURIOUS

Experiment 10 Project5

- Project 5 Simulate the page frame replacement
- Important data structures
  - Page table
  - typedef struct
  - {
  - int vmn; // virtual page number
  - int pmn; //physical page number
  - int exist; //Whether to occupy a physical block
  - int time;
  - }vpage_item;

Experiment 10 Project5

- time item is added to the page table for the replacement algorithm to select victim pages. In different replacement algorithms, time has different meanings.
  - In the FIFO algorithm, time is the time when the virtual page enters the memory. It is only set when the virtual page is brought into memory from external memory. When select victim pages, select the virtual page with the smallest time, that is, the earliest virtual page that enters memory.
  - In the LRU algorithm, time is the most recent access time. Every time the virtual page is accessed, time is set as the current access time. When select victim pages, select the one with the smallest time value, that is, the one that has not been used for the longest time.
  - In the OPT algorithm, time has no meaning.

Experiment 10 Project5

```
vm.c
#define VM_PAGE 7      /*Number of virtual pages*/
#define PM_PAGE 4       /* Number of memory blocks allocated to the job */
#define TOTAL_INSERT 18 /*Length of  page reference string*/
typedef struct
{
        int vmn;
        int pmn;
        int exist;
        int time;
}vpage_item;
```

Experiment 10 Project5

- vpage_item page_table[VM_PAGE];
- vpage_item* ppage_bitmap[PM_PAGE];
- int vpage_arr[TOTAL_INSERT] = { 1,2,3,4,2,6,2,1,2,3,7,6,3,2,1,2,3,6 };

Experiment 10 Project5

```
void init_data(){ //initialize data
        for (int i = 0; i<VM_PAGE; i++){
                page_table[i].vmn = i + 1;
                page_table[i].pmn = -1;
                page_table[i].exist = 0;
                page_table[i].time = -1;
        }
        for (int i = 0; i<PM_PAGE; i++) /*initialize the physical page map*/
        {
                ppage_bitmap[i] = NULL;
        }
}
```

Experiment 10 Project5

```
void FIFO()/*FIFO page replacement algorithem*/
{
        int k = 0;
        int i;
        int sum = 0;
        int missing_page_count = 0;
        int current_time = 0;
        bool isleft = true; /* Whether there are remaining physical blocks */
```

Experiment 10 Project5

```
while (sum < TOTAL_INSERT){
    if (page_table[vpage_arr[sum] - 1].exist == 0)
    {   missing_page_count++;
        if (k < 4)
        {   if (ppage_bitmap[k] == NULL) /*find a free block*/
            {   ppage_bitmap[k] = &page_table[vpage_arr[sum] - 1];
                ppage_bitmap[k]->exist = 1;
                ppage_bitmap[k]->pmn = k;
                ppage_bitmap[k]->time = current_time;
                k++;
            }
        }
```

Experiment 10 Project5

```
else{
        int temp = ppage_bitmap[0]->time;
        int j = 0;
        for (i = 0; i < PM_PAGE; i++){
            if (ppage_bitmap[i]->time < temp){
                    temp = ppage_bitmap[i]->time;
                    j = i;}
         }
         ppage_bitmap[j]->exist = 0;
         ppage_bitmap[j] = &page_table[vpage_arr[sum] - 1]; /*update page table */
         ppage_bitmap[j]->exist = 1;
         ppage_bitmap[j]->pmn = j;
         ppage_bitmap[j]->time = current_time;
}//else
```

Experiment 10 Project5

- }//if
- current_time++;
- sum++;
- }//while
- printf("The number of page faults of FIFO is:%d\t Page fault rate:%f\t The number of replacement:%d\tReplacement rate:%f", missing_page_count, missing_page_count / (float)TOTAL_INSERT, missing_page_count - 4, (missing_page_count - 4) / (float)TOTAL_INSERT);
- }//FIFO

Experiment 10 Project5

```
main
do{
        printf("Please choose page replacement algorithm：1.FIFO\t2.LRU\t3.OPT\t0. quit\n");
        scanf_s("%d", &a);
        switch (a){
             case 1:init_data();
                    FIFO();
                    break;
              case 2:init_data();
                     LRU();
                    break;
            case 3:init_data();
                       OPT();
                       break;
           }
      } while (a != 0);
```

Experiment 10 Project5

- Task
  - Complete the LRU and OPT algorithm code
- Submission
  - 1) Report: post the experimental results; analyze the reasons for the experimental results, and analyze the differences in the page fault rates of the three replacement algorithms.
  - 2) The code for this project.