

四川大学计算机学院、软件学院

实验报告

学号： 2022141460180 姓名： 封欢欢 专业： 计算机科学与技术 班级： 行政4班 第 12-13 周

课程名称	操作系统课程设计	实验课时	4 小时
实验项目	内存管理实验	实验时间	第 12 周到第 13 周
实验目的	1) 了解 Windows 及 Linux 内存管理机制。 2) 掌握页面虚拟存储技术。 3) 了解内存分配原理，特别是以页面为单位的虚拟内存分配方法。 4) 学会使用 Windows 下内存管理的基本 API 函数。 5) 了解进程中内存分配与虚内存的区别；		
实验环境	Clion2024.1.1		
实验内容（算法、程序、步骤和方法）	<p>一、对实验结果的基本理解：</p> <p>1. 比较 virtualalloc 函数和 malloc 函数分配内存的区别</p> <p>(1) VirtualAlloc 是低级别的内存分配函数，提供更多控制和灵活性。 malloc 是高级别的内存分配函数，使用简单，适合一般用途。</p> <p>(2) VirtualAlloc 允许详细控制内存页面的权限和状态。malloc 不提供对内存页面的权限控制，只进行基本的分配和释放。</p> <p>(3) VirtualAlloc 适用于需要大块连续内存、特殊内存保护和直接控制内存分配的场景。malloc 适用于通用程序的动态内存分配，尤其是频繁的小块分配。</p> <p>(4) VirtualAlloc 分配和释放内存的开销较高。malloc 通过运行时库进行优化，适合频繁的内存分配和释放。</p> <p>2.对虚拟内存大小，物理内存大小，页面文件的理解</p> <p>物理内存是计算机中实际存在的硬件内存，即 RAM，大小由计</p>		

算机安装的 RAM 模块的容量决定。

虚拟内存是操作系统提供的一种内存管理技术，它允许程序使用比实际物理内存更大的地址空间。虚拟内存的大小可以远大于物理内存，具体取决于操作系统和处理器架构。

页面文件是操作系统在硬盘上分配的一块区域，用作虚拟内存的一部分当物理内存不足时，操作系统会将不常用的数据从物理内存移动到页面文件，以腾出物理内存空间用于当前需要的数据。这种机制称为“分页”

**3.virtumem: 各指令执行后虚拟内存大小的变化**

虚存的保留与提交	虚存大小减小，与申请大小相匹配
虚存的除配	虚存大小不变，
虚存的除配并释放虚存空间	虚存大小增加，比申请大小更大
改变虚存内存页的保护	虚存大小不变
锁定虚存内存页	虚存大小不变
虚存的保留	虚存大小减小，与申请大小相匹配

**二、选做实验**

实验要求：

在页式虚拟存储器中，一个程序由 P1-P8 共 8 个页面组成，程序执行过程中依次访问的页面如下，  
P3,P4,P2,P6,P4,P3,P7,P4,P3,P6,P3,P4,P8,P4,P6。

假设系统分配给这个程序的主存有 3 个页面，采用 LRU 算法对这 3 个页面进行调度，求缺页次数和命中率。（输出过程）

函数：

```
#include <iostream>
```

```

using namespace std;

int main() {
    // 页面访问序列
    int page_sequence[] = {3, 4, 2, 6, 4, 3, 7, 4, 3, 6, 3,
4, 8, 4, 6};
    int sequence_length = sizeof(page_sequence) /
sizeof(page_sequence[0]);
    int num_pages = 3; // 内存中可以存放的页面数
    int num_faults = 0; // 缺页次数
    int hits = 0; // 命中次数

    int memory[3] = {-1, -1, -1}; // 用于存放当前内存中的页面
    int last_used[3] = {0, 0, 0}; // 记录每个页面上次使用的时间

    for (int time = 0; time < sequence_length; ++time) {
        int page = page_sequence[time];
        bool hit = false;

        // 检查页面是否在内存中
        for (int i = 0; i < num_pages; ++i) {
            if (memory[i] == page) {
                hits++;
                last_used[i] = time; // 更新该页面的上次使用时间
                hit = true;
                break;
            }
        }

        if (!hit) {
            // 页面未命中，缺页
            num_faults++;

            // 找到最近最少使用的页面进行替换
            int lru_index = 0;
            for (int i = 1; i < num_pages; ++i) {
                if (memory[i] == -1) { // 找到空闲位置
                    lru_index = i;
                    break;
                }
                if (last_used[i] < last_used[lru_index]) {
                    lru_index = i;
                }
            }

            memory[lru_index] = page; // 替换页面
            last_used[lru_index] = time; // 更新使用时间
        }
    }
}

```

	<pre>    }      // 输出当前内存状态     cout &lt;&lt; "Time " &lt;&lt; time &lt;&lt; ": Page " &lt;&lt; page &lt;&lt; " =&gt; Memory: ";     for (int i = 0; i &lt; num_pages; ++i) {         if (memory[i] == -1) {             cout &lt;&lt; "[] ";         } else {             cout &lt;&lt; "[" &lt;&lt; memory[i] &lt;&lt; "]" ";         }     }     cout &lt;&lt; endl; }  double hit_rate = static_cast&lt;double&gt;(hits) / sequence_length; double fault_rate = static_cast&lt;double&gt;(num_faults) / sequence_length;  cout &lt;&lt; "缺页次数: " &lt;&lt; num_faults &lt;&lt; endl; cout &lt;&lt; "命中率: " &lt;&lt; hit_rate &lt;&lt; endl;  return 0; }</pre>
数据记录 和计算	输出结果:

	 <pre>Time 0: Page 3 =&gt; Memory: [] [3] [] Time 1: Page 4 =&gt; Memory: [] [3] [4] Time 2: Page 2 =&gt; Memory: [2] [3] [4] Time 3: Page 6 =&gt; Memory: [2] [6] [4] Time 4: Page 4 =&gt; Memory: [2] [6] [4] Time 5: Page 3 =&gt; Memory: [3] [6] [4] Time 6: Page 7 =&gt; Memory: [3] [7] [4] Time 7: Page 4 =&gt; Memory: [3] [7] [4] Time 8: Page 3 =&gt; Memory: [3] [7] [4] Time 9: Page 6 =&gt; Memory: [3] [6] [4] Time 10: Page 3 =&gt; Memory: [3] [6] [4] Time 11: Page 4 =&gt; Memory: [3] [6] [4] Time 12: Page 8 =&gt; Memory: [3] [8] [4] Time 13: Page 4 =&gt; Memory: [3] [8] [4] Time 14: Page 6 =&gt; Memory: [6] [8] [4] 缺页次数: 9 命中率: 0.4  Process finished with exit code 0</pre>
结 论 (结 果)	成功完成实验
小 结	通过本次实验，加深了我对分页存储管理下进程创建，进程结束时的内存空间的分配与回收的理解；学会了使用 Windows 下内存管理的基本 API 函数；对进程中内存分配与虚内存的区别也有进一步认知；进程中内存分配与虚内存的区别。
指导老师评议	成绩评定：  指导教师签名：