

## 操作系统原理第九章作业

姓名：马福泉 学号：23336179 截止日期：2025 年 5 月 26 日

完成日期：2025 年 5 月 23 日

**Question 1: 9.5** 假设有一个按需调页存储器，页表放在寄存器中。处理一个页错误，当有空的帧可用或被置换的帧没有被修改过时要 用 8ms，当被置换的帧被修改过时用 20ms。存储器存取时间为 100ns。假设被置换的页中有 70% 被修改过，有效存取时间不超过 200ns 时最大可以接受的页错误率是多少？

**Answer 1:**

$$\begin{aligned} \text{平均页错误处理时间} &= 0.3 \times 8\text{ms} + 0.7 \times 20\text{ms} = 16.4\text{ms} \\ \text{有效存取时间} &: (1-p) \times 100\text{ns} + p \times 16400000\text{ns} \leq 200 \\ \text{解该方程得} & P \leq \frac{100}{16399900} \approx 6.09 \times 10^{-6} \\ \text{即最大可接受页错误率约为} & 6 \times 10^{-6} \end{aligned}$$

**Question 2: 9.10** 假设一个具有下面时间利用率的按需调页系统：

CPU 利用率      20%

分页磁盘        97.7%

其他 IO 设备    5%

试说明下面哪一项可能提高 CPU 的利用率，为什么？

- a. 安装一个更快的 CPU
- b. 安装一个更大的分页磁盘
- c. 提高多道程序的程度
- d. 降低多道程序的程度
- e. 安装更多内存
- f. 安装个更快的硬盘，或对多个硬盘用多个控制器
- g. 加入预约式页面调度算法预取页
- h. 增加页面大小

**Answer 2:**

(1) CPU 利用率低 (20%)：CPU 大部分时间处于空闲状态。分页磁盘利用率高 (97.7%)：分页磁盘几乎一直在工作，说明系统频繁进行页面交换 (swap in/out)。其他 I/O 设备利用率低 (5%)：其他 I/O 设备使用较少。

从这些数据可以看出，系统的瓶颈在于分页磁盘。由于分页磁盘几乎满负荷运行，而 CPU 利用率很低。

(2) a. CPU 利用率只有 20%，更快的 CPU 只会更多地处于空闲状态，不能提高利用率。

b. 安装一个更大的分页磁盘可以存储更多数据，更大的磁盘不会直接减少缺页率或加速缺页处理，反而会增加寻址的时间，因此不能提高 CPU 利用率。

c. 提高多道程序的程度

增加多道程序的程度意味着同时运行更多进程，增加内存压力，可能导致更多缺页。会进一步降低 CPU 利用率，因为更多进程等待 I/O。

d. 降低多道程序的程度，减少同时运行的进程数量会，减少内存压力，从而减少缺页，分页磁盘利用率可能下降，从而减少 I/O 等待时间 CPU 可以更多时间运行进程，利用率提高。

e. 安装更多内存可以：减少缺页率，减少对分页磁盘的依赖，CPU 等待 I/O 的时间减少，利用率提高。

f. 安装更快的硬盘，或对多个硬盘用多个控制器可以加速缺页处理，减少 I/O 队列等待时间，CPU 利用率提高。

g. 加入预约式页面调度算法预取页可以提前加载可能需要的页面，减少缺页。

如果实现得当，可以减少缺页，提高 CPU 利用率。

h. 增加页面大小：更大的页面可以减少缺页率。但也可能导致内部碎片增加。

所以可能减少缺页，提高 CPU 利用率，但效果取决于具体场景。

**Question 3: 9.14** 假设一个请求调页系统具有一个平均访问和传输时间为 20ms 的分页磁盘。地址转换是通过在主存中的页表来进行的，每次内存访问时间为 1 $\mu$ s。这样，每个通过页表进行的内存访问都要访问内存两次。为了提高性能，加入一个相关存储器，当页表项在相关存储器中时，可以减少对内存的访问次数。假设 80% 的访问发生在相关存储器中，而且剩下中的 10% (或总的 2%) 会导致页错误。内存的有效访问时间是多少？

Answer 3:

$$TLB \text{命中: } 0.8 \times 1 \mu s = 0.8 \mu s$$

$$TLB \text{缺失且无页错误: } 0.2 \times 0.9 \times 2 \mu s = 0.36 \mu s$$

$$TLB \text{缺失且有页错误: } 0.2 \times 0.1 \times (2000 + 1) \mu s = 400.02 \mu s$$

$$\text{故有效存取时间 } 0.8 + 0.36 + 400.02 = 401.18 \mu s$$

**Question 4:** 9.15 颠簸的原因是什么?系统怎样检测颠簸?一旦系统检测到颠簸,系统怎样来消除这个问题?

Answer 4:

- (1) 分配的页数少于进程所需的最小页数时发生颠簸,并迫使它不断地页错误。
- (2) 该系统可通过对比多道程序的程度来估计 CPU 利用率的程度,以此来检测颠簸。
- (3) 解决方法:降低多道程序程度。增加物理内存,预取页面等。

**Question 5:** 请求分页管理系统中,假设某进程的页表内容如下表所示

页号	页框 (Page Frame) 号	有效位 (存在位)
0	101H	1
1	—	0
2	254H	1

页面大小为 4KB,一次内存的访问时间是 100ns,一次快表 (TLB) 的访问时间是 10ns,处理一次缺页的平均时间为  $10^8$ ns (已含更新 TLB 和页表的时间),进程的驻留集大小固定为 2,采用最近最少使用 (LRU) 置换算法和局部淘汰策略。假设: ① TLB 初始为空; ② 地址转换时先访问 TLB,若 TLB 未命中,再访问页表 (忽略访问页表后的 TLB 更新时间); ③ 有效位为 0 表示页面不在内存,产生缺页中断,缺页中断处理后,返回到产生缺页中断的指令处重新执行。设有虚地址访问序列 2362H, 1565H, 25A5H, 请问:

- 1) 依次访问上述三个虚拟地址,各需多少时间? 给出计算过程。
- 2) 基于上述访问序列,虚地址 1565H 的物理地址是多少? 请说明理由。

Answer 5:

07.2362H: TLB为空,未命中  $\rightarrow$  访问页表,页号为2

$$10ns(TLB) + 100ns(页表) + 100ns(内存) = 210ns$$

1565H: TLB未命中  $\rightarrow$  访问页表,缺页  $\rightarrow$  缺页处理

访问时间主要用于缺页处理  $10^8 ns$

25ASH: TLB命中(LRU保留)

$$访问时间 10(TLB) + 100(内存) = 110ns$$

(2) ①缺页处理, 页号1不在内存,

根据LRU, 置换页号0. 页框号101H分配给页号1

$$②计算: 101H \ll 12 + 565H = 101565H$$