

北京交通大学

Operating System

Lab04:Designing a Virtual Memory Manager

学 院：	计算机与信息技术学院
学生姓名：	刘嘉鹏
学 号：	20281319

北京交通大学

2022 年 11 月

目录

1. 连续内存分配的四种算法.....错误!未定义书签。
2. 程序设计与实现 3
3. 运行结果及分析错误!未定义书签。
4. 总结与建议.....错误!未定义书签。
5. 参考材料 4

1. TLB 与内存管理

页表一般都很大，并且存放在内存中，所以处理器引入 MMU 后，读取指令、数据需要访问两次内存：首先通过查询页表得到物理地址，然后访问该物理地址读取指令、数据。为了减少因为 MMU 导致的处理器性能下降，引入了 TLB，TLB 是 Translation Lookaside Buffer 的简称，可翻译为“地址转换后援缓冲器”，也可简称为“快表”。简单地说，TLB 就是页表的 Cache，其中存储了当前最可能被访问到的页表项，其内容是部分页表项的一个副本。只有在 TLB 无法完成地址翻译任务时，才会到内存中查询页表，这样就减少了页表查询导致的处理器性能下降。TLB hit 即是第一次查找就命中，TLB miss 则是在 TLB 中没有查找到对应地址，需要到主存中查找，并且查找完毕后再将该地址存入 TLB。

2. 程序设计与实现

(1) 定义 node 节点，存储一对逻辑地址和物理地址，再用一个循环更新的 node 数组作为 TLB 的数据结构。

(2) 定义 search_tlb 函数，在 TLB 中查找是否已经存在存入了 TLB 的数据，查找成功则返回物理地址，否则返回 -1。

(3) 定义 add_tlb 函数，当发生 tlb miss 时，把对应的逻辑地址和物理地址存入 TLB，该函数需要用到 search_tlb 函数。

(4) 根据要求，运行程序时需要添加命令行参数 addresses.txt 作为读取数据的源文件，可以再添加 BACKING_STORE 作为第二个参数。

(5) 定义变量记录发生 tlb hit 和 page fault 的次数，最后计算频率。

具体实现参见代码 [manager.c](#)

3. 运行结果分析

```
● remilia@WSL:~/codes/lab05$ make  
gcc manager.c -o manager  
● remilia@WSL:~/codes/lab05$ ./manager BACKING_STORE addresses.txt > output.txt
```

用输出重定向，把结果输出到 [output.txt](#)。

与 `correct.txt` 进行比较，预期结果符合要求。

4. 总结与建议

本次实验有一定难度，重点在于对 TLB 相关算法的理解。用循环数组维护 TLB 结构，用位操作获得一个 16（32）位数据的某些位等等，都是我以前从未接触过的内容。

5. 参考材料

TLB 的工作原理：

<https://www.cnblogs.com/alantu2018/p/9000777.html>

<https://zhuanlan.zhihu.com/p/480808324>

虚拟内存管理器的实现：

<https://www.alltestanswers.com/designing-a-virtual-memory-manager-using-c-programming/#comments>