

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。**](#_Toc120521506)

[2. 程序设计与实现 3](#_Toc120521507)

[3. 运行结果及分析 **错误!未定义书签。**](#_Toc120521508)

[4. 总结与建议 **错误!未定义书签。**](#_Toc120521509)

[5. 参考材料 4](#_Toc120521510)

# 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。

# 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 falut的次数，最后计算频率。

具体实现参见代码<manager.c>

# 运行结果分析

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。

# 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。

# 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>