

Project 4 Virtual Memory 设计文档

中国科学院大学

[姓名] 李国峰

[日期] 2021 年 1 月 26 日

1. 内存管理设计

仅实现了 TLB 的初始化 (S-core)，详情见下图代码注释。

2. 关键函数功能

```
void init_TLB(void)
{
    int i;
    uint64_t VPN2 = 0;
    uint64_t PFN = 0x10000;
    set_cp0_pagemask(0);
    for(i = 0; i < 64; i++)
    {
        set_cp0_entryhi(VPN2 << 13);
        set_cp0_entrylo0(PFN << 7 | 0b0010111);
        set_cp0_entrylo1(PFN << 7 | 0b1010111);
        set_cp0_index(i);
        tlbwi_operation();
        VPN2 += 1;
        PFN += 1;
    }
}
```

//虚页号，实现虚拟地址转换的地址的起始虚页号
//物理页号（这里抹去了最低一位，在填cp0lo寄存器时直接写了 物理页号的起点是0.5g处，因为最低0.5g留给unmapped区
//设置物理页大小 全0则为4KB
//虚页信息填入cp0hi
//偶数实页 最低为默认是0 所以PFN里没有这一位
//奇数实页
//填入板子的64项TLB中的第几项，类似数组的索引
//根据上述5个寄存器内容去填tlb

本实验请老师改了函数，因此最终结果可能会与标准有所不同。经过测试，TLB 的填充是成功的，区别在于数据来源是用户输入还是系统预置（见下图，这是老师改的函数片段，这些数据最终都可以打印出来）。

```
void rw_task1()
{
    char* argv[] = {
        "rw",
        "rw",
        "0x10000",
        "0x20000",
        "0x30000",
        "0x10000",
        "0x20000",
        "0x30000"
    };
};
```

此外，由于没有实现时钟中断，rand()函数无法发挥作用，为了验证 rw_task1()函数确实正确运行了，将 rand()改为一个具体的数，打印出的结果也为这个数，测试成功。

```
memory[i] = mem2 = 100; //rand();
```

参考文献

- [1] 中国科学院大学操作系统实验手册 Project4-MIPS (2020 秋季学期)