

Project 4 Virtual Memory 设计文档

中国科学院大学

[姓名] 蔡润泽

[日期] 2019.12.1

1. 内存管理设计

- (1) 你设计的页表是几级页表，页表项的数据结构是什么？页表本身的数据结构是什么？

本设计采用的是是一级页表，页表项的数据结构是一个 32 位的无符号 32bit 整数。高 26 位是物理页号，低 6 位记录了 C, D, V, G。D 为脏域，V。每个进程持有一个页表，每个页表是一个由 128K 个页表项组成数组。数组的下标即为虚拟页号。

- (2) 任务 2 开发时初始化了多少个页表项，能索引到多大的物理空间，以及使用了多少个物理页框保存页表？

本设计中，页表项均为 128K 个（对应虚拟地址 0-0x80000000），页表项的物理空间索引范围为 26 位，既可以搜索 2^{26} 个物理页框，每个物理页框大小为 4KB，因此总共能检索到的物理空间为 2^8 GB。在本次实验中，一共使用了 0x1000000-0x2000000 的物理空间，因此对应的物理页框数为 0x1000。

- (3) 任务 3 中，进程的用户态栈的起始地址是多少，栈空间是多大？

本设计中，进程的用户态栈的栈顶 0x20000000。每个栈空间为 0x10000。

- (4) TLB miss 何时发生？在任务 2 中，你处理 TLB miss 的流程是怎样的？

TLB miss 在无法找到使虚地址和物理地址建立了有效映射的 TLB 表项时发生。

具体情况有：无 VPN 匹配的 TLB 项、VPN 匹配但 G 位不为 1 且 ASID 不和当前 PID 相等、有匹配项但 V 位为 0。

处理流程为：发生 TLB 例外时跳转到 TLB 例外处理入口地址 0x80000000，

再跳转到 `do_TLB_Helper` 函数。

`do_TLB_Helper()` 先利用 `context` 寄存器和 `entryhi` 的 ASID 域设置号 `entryhi` 寄存器，然后用 `tlbp` 去查询 `tlb` 表项。查询的结果会通过 `Index` 寄存器的 `P` 位体现出来，`P` 为 1 则没有找到对应的 TLB，此时进行 `refill` 例外处理，并利用 `tlbr` 进行随机充填。

如果 `P` 为 0，说明找到了 VPN 对应的 `tlb` 项，此时需要处理 `invalid` 例外。处理 `invalid` 例外需重新分配物理页框，并建立好页表对应物理页框之间的关系。将处理好的页表填充到 `entrylo0` 和 `entrylo1` 处理器中，`V` 位置 0。最后利用 `tlbwi` 填入对应 `index` 的项。

2. 缺页处理设计

- (1) 何时会发生缺页处理？你设计的缺页处理流程是怎样的？

在实验 3 中，由于页表和 TLB 均没有初始化，因此第一次检索 TLB 时会发生充填例外，而此时充填的页表也是没有初始化的，因此会发生缺页例外。缺页例外的处理在 TLB `invalid` 例外处理部分内处理。发生缺页（TLB `invalid`）时，处理函数会分配给页表物理页框，并建立好页表对应物理页框之间的关系。将处理好的页表填充到 `entrylo0` 和 `entrylo1` 处理器中，`V` 位置 0。最后利用 `tlbwi` 填入对应 `index` 的项。

- (2) 你使用什么数据结构管理物理页框，管理多少物理页框（例如管理哪些地址范围内的物理页框）？在缺页分配时，按照什么策略或原则进行物理页框分配？

物理页框管理结构如下：

```
1. typedef struct pgframe {
2.     uint32_t paddr; //physical addr
3.     uint32_t vaddr; //virtual addr
4.     pte_t * vppte;
5.     uint8_t is_valid;
6.     struct pgframe *prev;
7.     struct pgframe *next;
8. } pgframe_t;
9.
10. pgframe_t pf_group[PAGE_FRAMES_NUM];
11. queue_t emptylist;
12. queue_t fulllist;
```

其中 `paddr` 指向物理地址，`vaddr` 指向虚拟地址，`vppte` 指向对应页表项，`is_valid` 表示该页框是否有效，`prev` 和 `next` 指向前后物理页框。可以管理 `PAGE_FRAMES_NUM` 个物理页框，本实验中只需要 0x1000 个即可。缺页分配时，利用利用队列顺序查询空物理页框来进行分配。

- (3) 你的设计中是否有 `pinned` 的物理页框？若有，具体是保存什么内容的物理页框？

本设计中没有采用 `pinned` 的物理页框。

(4) 设计或实现过程中遇到的问题和得到的经验（如果有的话可以写下来，不是必需项）

在实验测试时，发现访问低 2 位不为 0 的地址时，发现地址会出现错误，这可能与运行的 MIPS 环境不支持非对齐访存有关。

3. 关键函数功能

利用 `do_TLB_Helper` 来处理 TLB 的 `refill` 和 `invalid` 例外。

参考文献

- [1] linux 内存管理---mips 基础（一）， <https://blog.csdn.net/whuzm08/article/details/79536526>
- [2] 龙芯 2F 处理器手册_v0.1