

**本科实验报告**

|  |  |
| --- | --- |
| 课程名称： | 操作系统 |
| 姓 名： | 董佳鑫 |
| 学 院： | 计算机学院 |
| 系： | 计算机系 |
| 专 业： | 计算机科学与技术 |
| 学 号： | 3210102181 |
| 指导教师： | 寿黎但 |

# Lab 3: RV64 虚拟内存管理

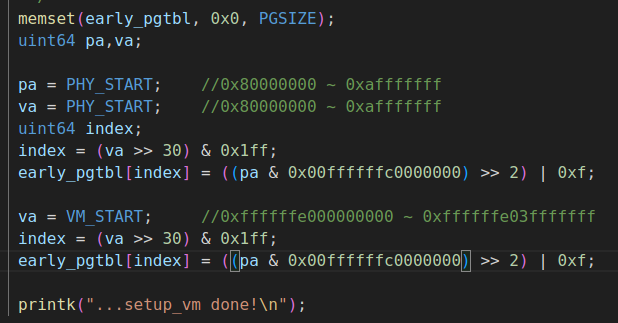
实验目的

1. 学习虚拟内存的相关知识，实现物理地址到虚拟地址的切换。
2. 了解 RISC-V 架构中 SV39 分页模式，实现虚拟地址到物理地址的映射，并对不同的段进行相应的权限设置。

实验过程和操作步骤

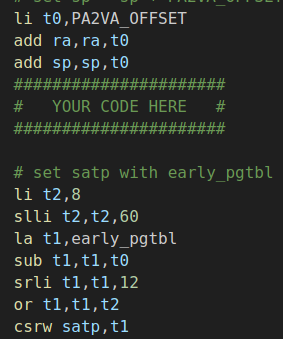
1. setup\_vm的实现

完成等值映射和偏移映射，使用项目提供的early\_pgtbl来存储映射信息。



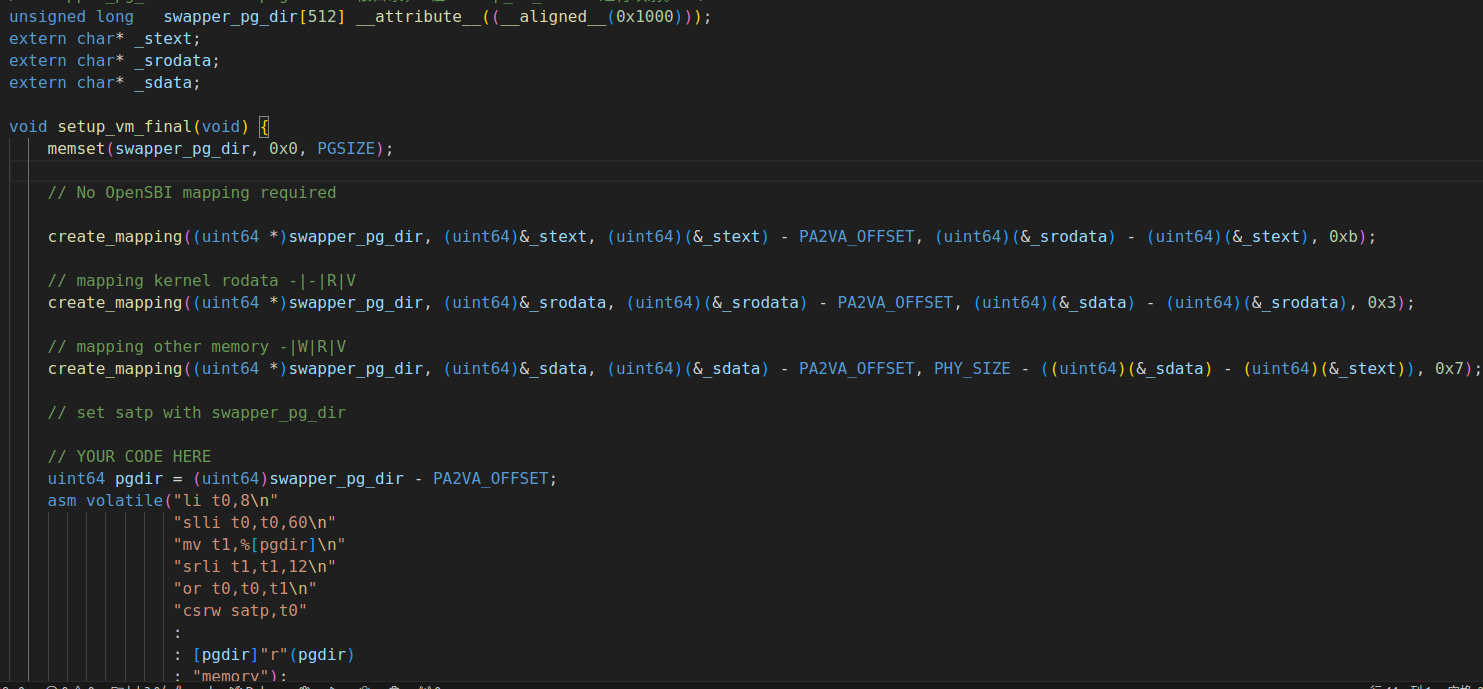
首先对early\_pgtbl进行初始化，然后按照实验要求进行映射。其中index表达式是为了取出VPN[2]作为index，early\_pgtbl[index]表达式的值是为了取出物理地址中的PPN的值放到entry对应bit位置上，然后将Page Table Entry 的权限 V | R | W | X 位设置为 1。

1. 完成relocate函数

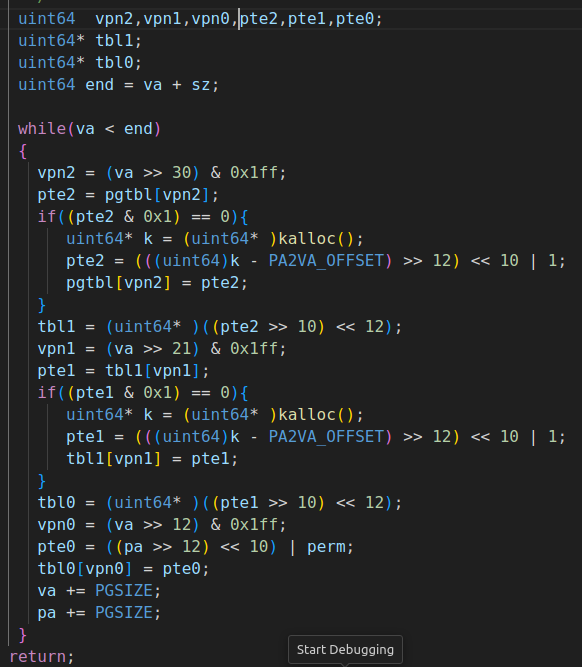


首先通过设置ra和sp寄存器的值将地址转换到虚拟地址，然后我们要设置satp寄存器的值，将mode改为8，然后将early\_pgtbl物理地址的PPN存到satp中，以便后续能通过satp寄存器找到根页表。

1. Setup\_vm\_final函数实现

 Setup\_vm\_final部分关键代码如上图。由于我们要对不同的section设置不同的映射及权限，因此需要拆分成三段分别建立映射。需要使用extern关键字引入\_stext \_srodata \_sdata等section，然后对这些section分别建立映射。然后再次设置satp寄存器的值，这里使用内联汇编实现。具体三级映射create\_mapping的实现见下一节。

1. Create\_mapping函数实现



create\_mapping三级页表的函数实现如上图。需要注意的是，我们首先要判断每一级页表是否存在，我们可以使用V （protection bits的第0位）位来判断页表是否有效，如果无效首先应该申请一块新的内存然后创建页表，并且V位置为1。在最后一级entry内容应该包含权限信息是，即perm的值。然后将内存空间内每一个page都进行一次这样的三级映射。

1. 其他文件修改

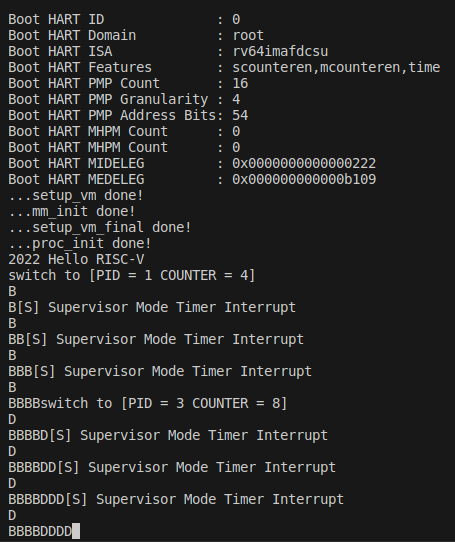
首先是head.S文件，要将调用setup\_vm 和 relocate 放到mm\_init，即内存初始化之前，而要将调用setup\_vm\_final放到mm\_init之后。

然后就是mm\_init文件中，将mm\_init函数中kfreeranged的end参数的值改为虚拟内存的结束地址。

此外就是一些关于Makefile等一些文件的细节性更改，保证项目能够编译执行。

1. 编译和测试

和lab2类似，执行make test-run后可以得到如下的结果。



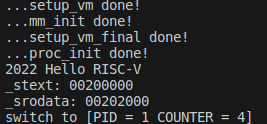
思考题

1. 根据setup\_vm\_final函数的实现我们可以知道，.text段具有执行属性和读属性，没有写属性；.rodata具有读属性，没有写属性和执行属性。我们下面来验证这些属性。

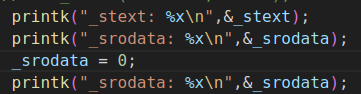
首先执行属性是比较显然的，我们的项目代码本身就是放在.text段的，项目可以执行就说明了.text具有执行属性；而.rodata只是存放常量的，没有执行功能。

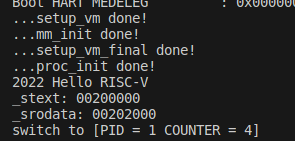


在主函数中我们添加两行代码来输出这两个section的地址。可以得到以下输出。



这个输出的值正是\_stext段和\_srodata段的起始物理地址的值，被成功读取了出来。下面我们再尝试以下写操作。同样在主函数中添加以下代码进行测试。





可以看到，并没有输出更改后的\_srodata地址，这说明\_srodata = 0这行代码不合法，因为我们没有写权限，不能对\_srodata进行写操作，因此主函数无法得到正确的结果。

2.因为在relocate函数中，当我们设置satp寄存器的值以后，我们就开启了虚拟内存，这时所有地址都会变成虚拟地址，然后通过页表映射为物理地址。PC地址也同样是如此，当设置完satp以后，PC的地址都变成了虚拟地址，然后虚拟地址再通过页表映射为物理地址。而建立等值映射的意义，就是让csrw satp,t0后面的指令的PC值能够被“找到”，即如果没有等值映射，这些地址会被当做虚拟地址，但是却找不到对应的物理地址，产生instruction fault，造成系统崩溃。

3.如果不建立等值映射也要解决上面的问题，可以通过中断处理机制来解决。在设置玩satp后的下一条指令会fault，这时跳转到trap程序，在trap程序中改变PC值：PC = PC + PA2VA\_OFFSET，然后跳转回\_start, 然后得到的PC值就可以被映射到正确的物理地址当中。

讨论心得

本次实验对我来说难度较大，主要难度还是在于理解实验要求。因为内存寻址这方面的知识本来就有些抽象，再加上本次实验需要时刻区分是物理地址还是虚拟地址，因此要正确完成不太容易。建立映射本身并不需要多么复杂的代码，但是要想清楚怎么映射以及为什么这样映射并不是一个容易的事情，经过长时间观察实验指导给出的几张示意图并且认真理清图中每一个元素代表的意义后才逐渐明白原理。

这次实验给我的收获也很大，我更加切身地体会到了虚拟内存的重要性以及实现原理，对于页表机制也有了更深的体会。此外，我对于权限控制也有了一定程度的认识，更加深刻地领会到了.text .rodata等section的本质区别。