**实验报告**

**Lab 3**

姓名：施君豪

班级：20 人工智能

学号：20307140008

**实验报告填写要求**

1.请在每个exercise之后简要叙述实验原理，详细描述实验过程。可以使用中文进行描述，不对语言做要求。

2.请将你认为的关键步骤附上必要的截图。

3.有需要写代码的实验，必须配有代码、注释以及对代码功能的说明。

4.你还可以列举包括但不局限于以下方面:实验过程中碰到的问题你是如何解决的、实验之后你还留有哪些疑问和感想。

5.如果实验附有练习，请在每个练习之后作答，这是实验报告评分的重要部分。

6.Challenge为加分选作题。每个lab可能有多个challenge,我们会根据完成情况以及难度适当加分，具体情况会在课上说明。这部分的实验过程描述应该比exercise更加详细。

7.切勿抄袭亦或是去互联网复制粘贴答案。

【练习题模板】

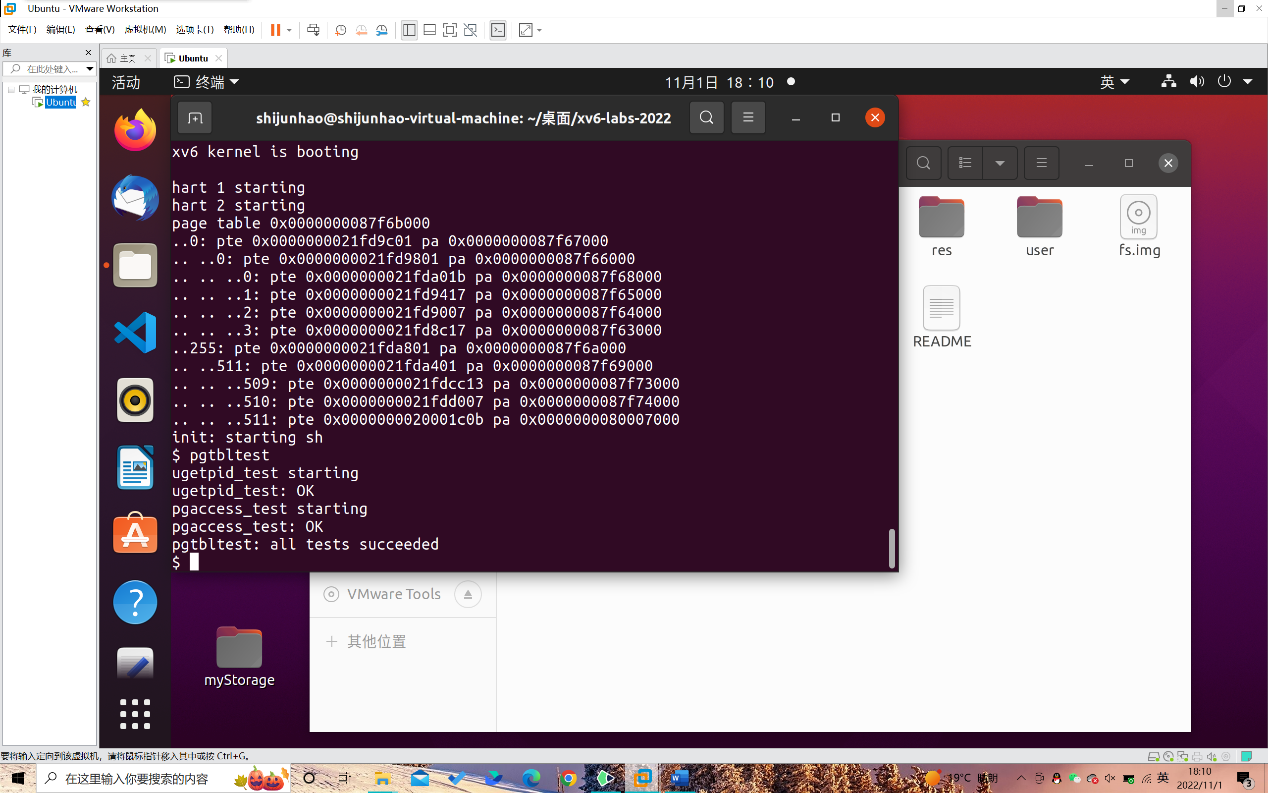
1. Question
2. Code
3. Screenshot
4. Difficulties and solutions

实验练习一：**Speed up system calls (easy)**

**Some operating systems (e.g., Linux) speed up certain system calls by sharing data in a read-only region between userspace and the kernel. This eliminates the need for kernel crossings when performing these system calls. To help you learn how to insert mappings into a page table, your first task is to implement this optimization for the getpid() system call in xv6.**

1. 代码

修改文件较多，见附件

1. 运行截图：

三、困难与解决办法

按照提示中的内容对usyscall申请了页面，并proc\_pagetable中依葫芦画瓢来进行地址的映射，随后对usyscall的内容补充初始化与释放相关内容。对mappages中的PTE\_X等系列指令一开始没有看懂（那个valid和user can access有点抽象），查阅相关帮助后顺利解决。

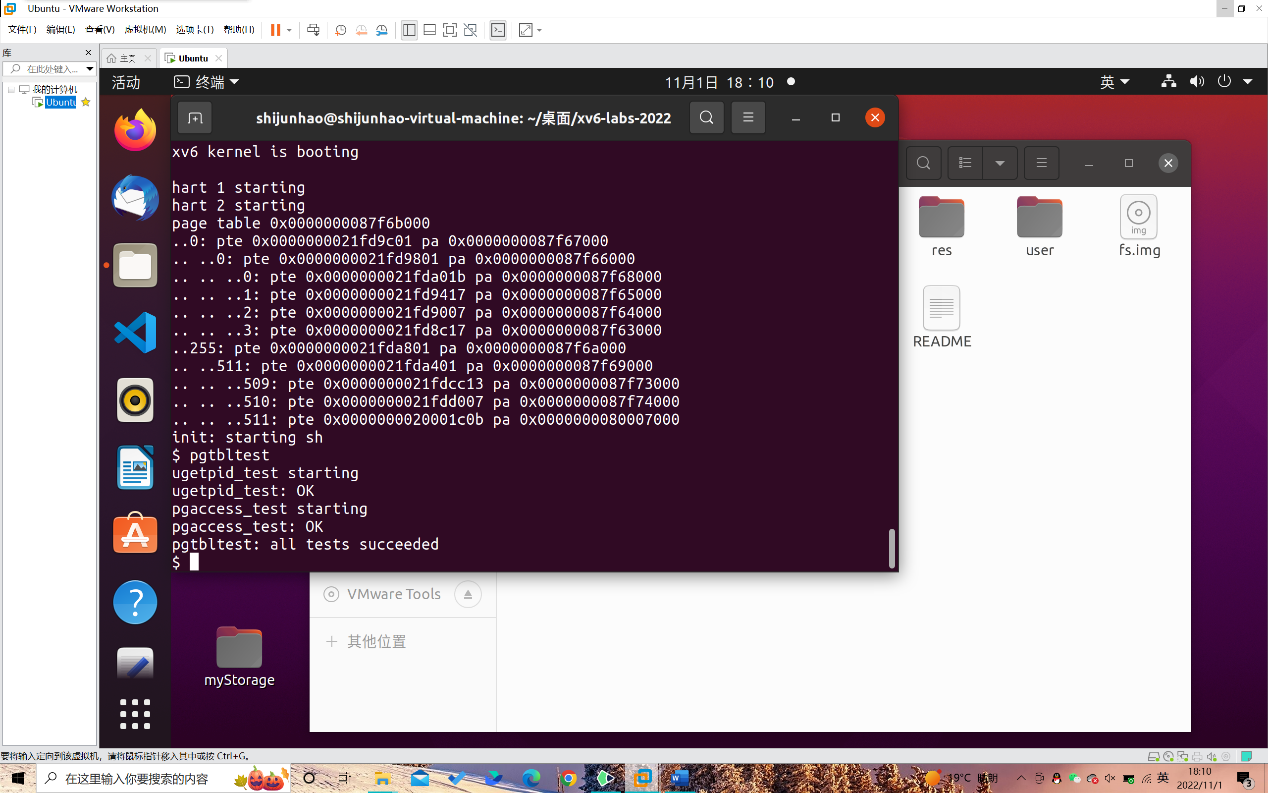
实验练习二：**Print a page table (easy)**

**To help you visualize RISC-V page tables, and perhaps to aid future debugging, your second task is to write a function that prints the contents of a page table.**

**Define a function called vmprint(). It should take a pagetable\_t argument, and print that pagetable in the format described below. Insert if(p->pid==1) vmprint(p->pagetable) in exec.c just before the return argc, to print the first process's page table. You receive full credit for this part of the lab if you pass the pte printout test of make grade.**

一、代码：

详见附件

1. 屏幕截图
2. 困难与解决办法

由于页表是类似于一个树的分级结构，故使用递归的方法进行遍历。按照提示在exec.c中添加相关内容，并仿照freewalk编写类似结构。在查看页表是否可行时一开始遗漏了pte & (PTE\_X|PTE\_W|PTE\_R) 的后半部分，在调试修改后顺利解决问题。

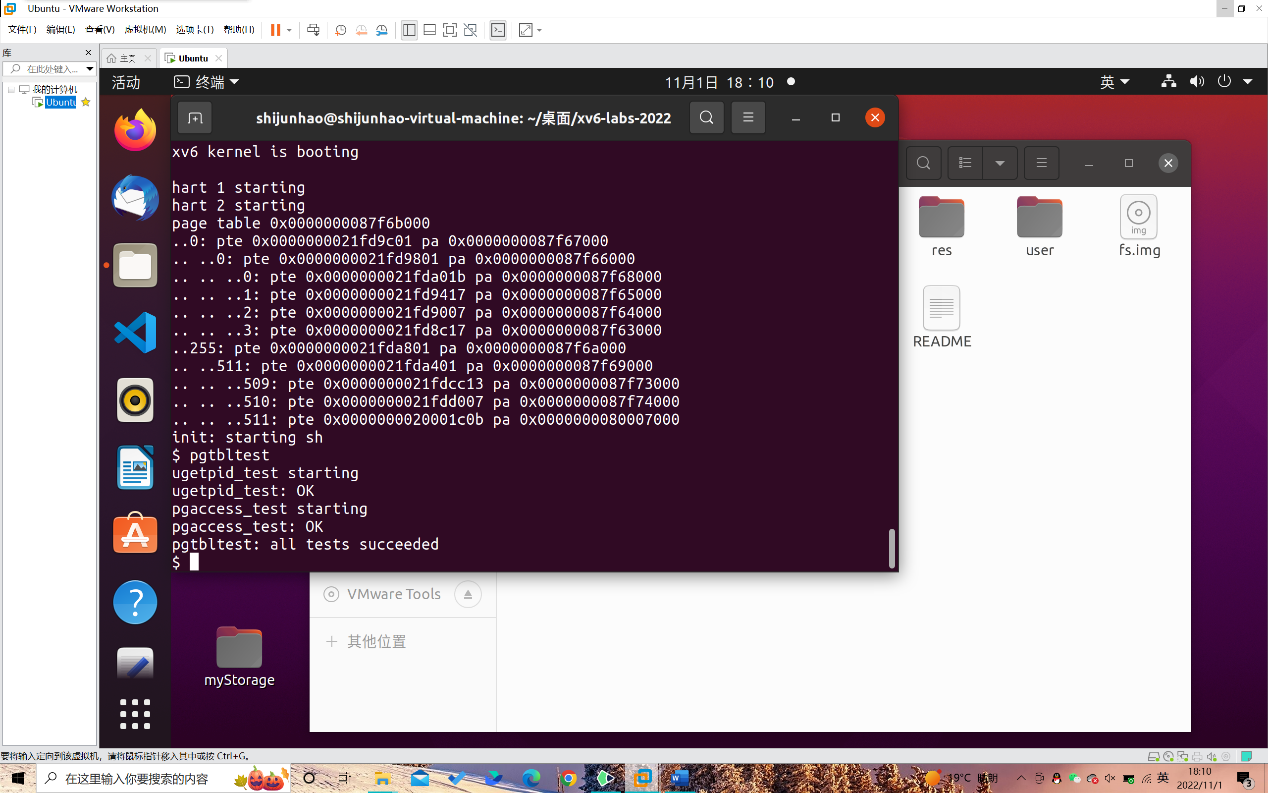
实验练习三：**Detect which pages have been accessed (hard)**

**Some garbage collectors (a form of automatic memory management) can benefit from information about which pages have been accessed (read or write). In this part of the lab, you will add a new feature to xv6 that detects and reports this information to userspace by inspecting the access bits in the RISC-V page table. The RISC-V hardware page walker marks these bits in the PTE whenever it resolves a TLB miss.**

**Your job is to implement pgaccess(), a system call that reports which pages have been accessed. The system call takes three arguments. First, it takes the starting virtual address of the first user page to check. Second, it takes the number of pages to check. Finally, it takes a user address to a buffer to store the results into a bitmask (a datastructure that uses one bit per page and where the first page corresponds to the least significant bit). You will receive full credit for this part of the lab if the pgaccess test case passes when running pgtbltest.**

一、代码：

详见附件

1. 屏幕截图
2. 困难与解决办法

首先查阅riscv手册，了解PTE\_A的定义并将其加入头文件中。随后按照提示完成sys\_pageaccess，利用argaddr和argint完成对参数的处理。随后在proc.c中编写pageaccess函数，读取当前虚拟地址，找到对应页表，找到其对应PTE\_A位置。遇到的困难在于对walk函数的使用不太熟悉，以及有点忘了最后copyout函数的用法，最后都顺利解决。（以及又忘记defs.h中添加函数名字了）。

**问题回答：**

（1）在 Part A加速系统调用部分，除了 getpid()系统调用函数，你还能想到哪 些系统调用函数可以如此加速？

加速系统调用的原理位通过在用户空间和内核之间共享只读区域中的数据来消除在执行这些系统调用时对内核交叉的需要，适合那些本身运行时间很短，对时间精确性要求高的系统调用。可行的例子：对目前时间的调用（类似gettimeofday），读取，打开某个文件/文件夹，查看进程状态等

（2）虚拟内存有什么用处？

虚拟内存为每个进程提供了一个大的、一致的、私有的地址空间。

1、相较于实际的物理内存，虚拟内存为每个进程提供了一致的地址空间，每个进程的虚拟地址均为从0开始，进程的地址使用与实际使用的物理地址无关，简化了内存管理。

2、虚拟内存保护了每个进程的地址空间不被其他进程破坏，进程之间互不干扰，同时保有各自的隐私。

3、虚拟内存可以将进程之中并未使用的内存分配给别的进程使用，使内存的实际使用更为高效。

4、虚拟内存可以使合作的进程公用同样的代码，节省内存空间。

（3）为什么现代操作系统采用多级页表？

相比于只使用一级页表，多级页表只需要首级页表存在主存里，多级页表可以在首次使用时再创建，这样就大大减少了常驻主存的页表数量。除此以外，使用多级页表可以使得页表在内存中离散存储，不用保证页目录项和页表项连续。

（4）简述 Part C的 detect 流程。

sys\_pageaccess：首先利用argaddr和agrint获取参数，随后调用proc.c中的pageaccess函数。

pageaccess：首先获取进程（myproc（）），获取进程中的页表项。随后对每一页进行遍历，利用walk函数获取对应的实际页表项目。随后对是否使用进行判断，若正在使用则将第i位置为1，最后利用copyout函数将数据传输给用户。

在输入pgtbltest后，通过中断将指令传入内核，随后syscall调用sys\_pageaccess，随后依次完成上述内容，最终获得数据。

**Part 3：The Ending**

通过本实验我了解了xv6系统中对于虚拟内存以及TLB的设置，同时对操作系统中虚拟内存的内容有了更深入的理解。在过程中也可以亲身体会到整个实验流程是循序渐进的，前面实验所学到的东西在后面的实验中会有所体现。