**操作系统大作业2**

提交截止日：12月7日零时

**总体要求**

在github上创建os-assignment2项目，提供（1）虚存管理模拟程序源代码及结果（存成文本文件）；（2）实验报告（word/pdf），包含所有实验的基本过程描述。

1. **虚存管理模拟程序，40分**

1.1 Chapter 10. Programming Projects: Designing a Virtual Memory Manager（OSC 10th ed.），30分。

1. 保持为vm.c，使用如下测试脚本test.sh，进行地址转换测试，并和correct.txt比较。

|  |
| --- |
| #!/bin/bash -e  echo "Compiling"  gcc vm.c -o vm  echo "Running vm"  ./vm BACKING\_STORE.bin addresses.txt > out.txt  echo "Comparing with correct.txt"  diff out.txt correct.txt |

注：本小题不要求实现Page Replacement，TLB分别实现FIFO和LRU两种策略。

1. 实现基于LRU的Page Replacement；使用FIFO和LRU分别运行vm（TLB和页置换统一策略），打印比较Page-fault rate和TLB hit rate，给出运行的截屏。提示：通过getopt函数，程序运行时通过命令行指定参数。
   1. 编写一个简单trace生成器程序，可以用任意语言，报告里面作为附件提供。运行生成自己的addresses-locality.txt，包含10000条访问记录，体现内存访问的局部性（参考Figure 10.21, OSC 10th ed.），绘制类似图表（数据点太密的话可以采样后绘图），表现内存页的局部性访问轨迹。然后以该文件为参数运行vm，比较FIFO和LRU策略下的性能指标，最好用图对比。给出结果及分析，10分。
2. **xv6-lab-2020页表实验（**[**Lab:page tables**](https://pdos.csail.mit.edu/6.828/2020/labs/pgtbl.html)**），20分**

完成Print a page table任务。要求按图1格式打印页表内容；其中括号内表示页表项权限，R表示可读，W表示可写，X表示可执行，U表示用户可访问。物理页后的数字（pa 32618）表示第几个物理页帧。要求在报告中提供实现所需的源代码和运行截屏，代码要求有充分注释。然后，回答接下来的6个问题（分别对应代码注释行中的标签）。

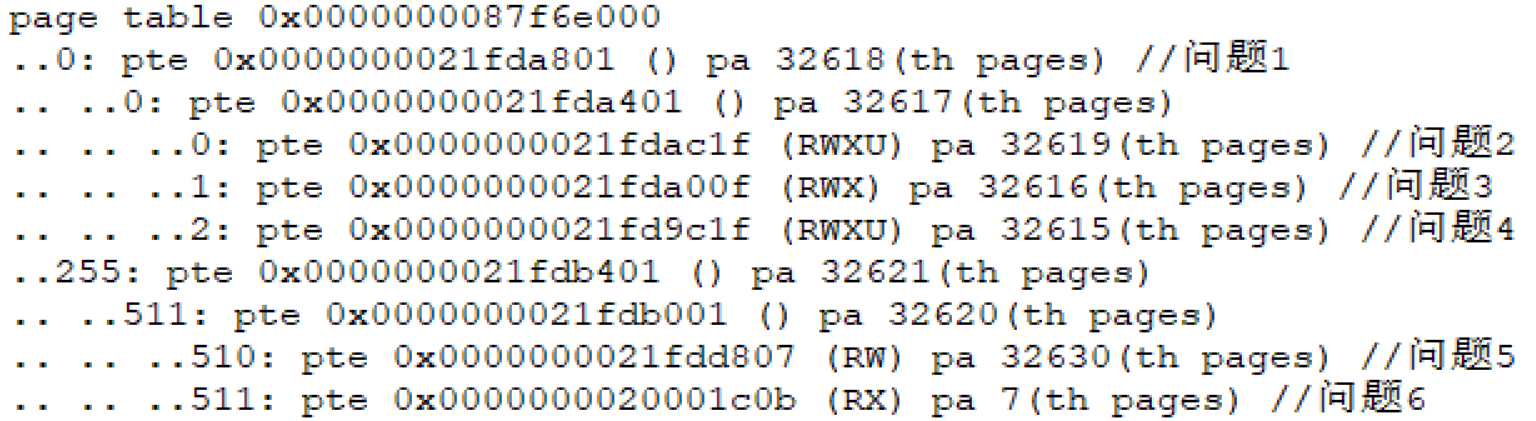


图1. init进程的页表内容

问题1：为什么第一对括号为空？32618在物理内存的什么位置，为什么不从低地址开始？结合源代码内容进行解释。

问题2：这是什么页？装载的什么内容？结合源代码内容进行解释。

问题3：这是什么页，有何功能？为什么没有U标志位？

问题4：这是什么页？装载的什么内容？指出源代码初始化该页的位置。

问题5：这是什么页，为何没有X标志位？

问题6：这是什么页，为何没有W标志位？装载的内容是什么？为何这里的物理页号处于低地址区域（第7页）？结合源代码对应的操作进行解释。

$ echo hi

usertrap(): unexpected scause 0x000000000000000f pid=3

sepc=0x0000000000001238 stval=0x0000000000004008

va=0x0000000000004000 pte=0x0000000000000000

panic: uvmunmap: not mapped

1. **xv6-lab-2020内存分配实验 (**[**Lab: xv6 lazy page allocation**](https://pdos.csail.mit.edu/6.828/2020/labs/lazy.html)**)，40分**

3.1 完成Lazy allocation子任务

，要求echo hi正常运行，报告中可以描述自己的尝试过程，以及一些中间变量。

3.2 完成Lazytests and Usertests子任务。对于Lazytests，要求屏幕输出如下图所示；对于usertests任务，要求通过所有除sbrkarg之外的测试。给出运行截屏。

在阅读报告中提供代码修改片段，说明针对哪些文件，哪些函数进行了修改，新代码加上充分注释；可以写一些体会。

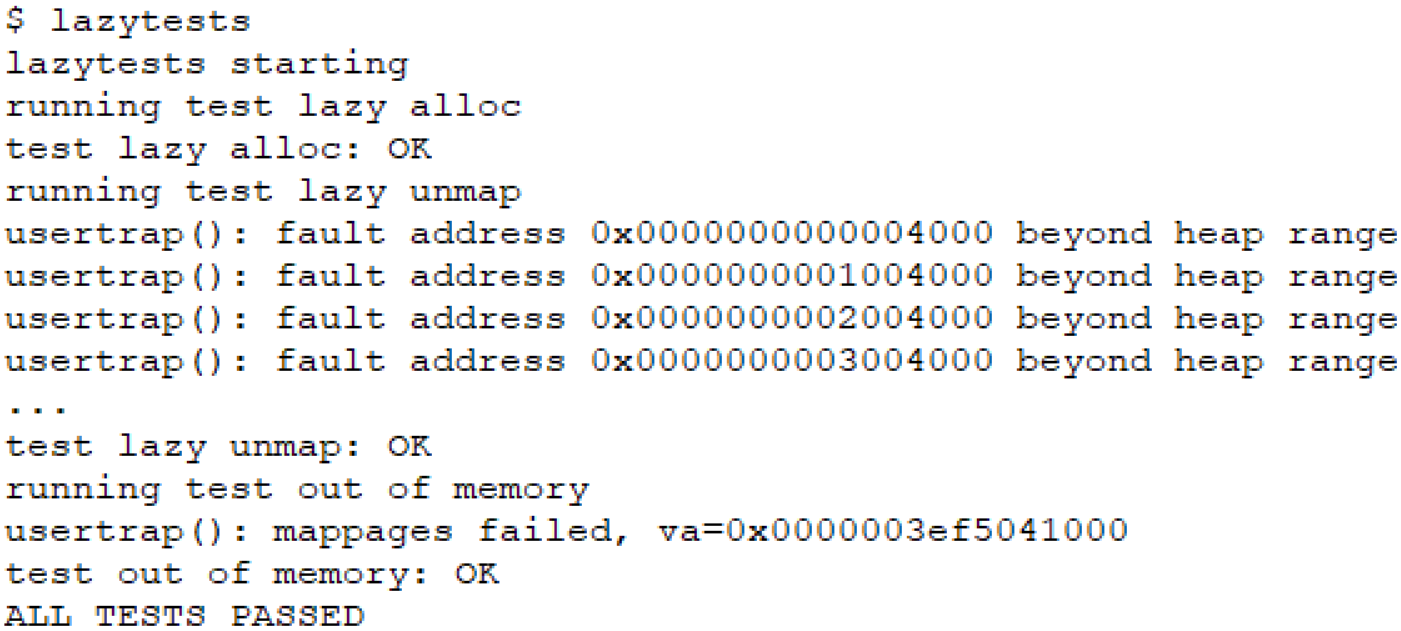


图2. Lazytests运行输出示例