**操作系统大作业2**

提交截止日：12月18日零时

1. **总体要求**

在github上创建os-assignment2项目，项目包括2个题目的结果:

1. 虚存管理模拟程序
2. Linux内存管理实验程序

该目录下，同时存放1个pdf/word文件，作为实验报告。

注：附加题的分数单独计算，累加到正常分数上面，最后总分不超过100分。

**2. 虚存管理模拟程序，50分+10分（附加题）**

1. Chapter 10. Programming Projects: Designing a Virtual Memory Manager（OSC 10th ed.）
2. 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：本小题不要求实现页置换（Page Replacement），TLB用简单的FIFO策略。30分。

1. 实现LRU的TLB，8分。
2. 实现基于LRU的Page Replacement，8分。
3. 代码可读性，4分。
4. 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。
5. （**附加题10分**）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1万条访问记录，体现内存访问的局部性（参考Figure 10.21, OSC 10th ed.），绘制类似图表，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。
6. **Linux内存管理实验，50分+10分（附加题）**

阅读Linux内存管理相关代码片段，提供程序和阅读报告，描述关键数据结构中和内存相关的成员的意义，以及指针指向关系。涉及的数据结构包括（但不限于）task\_struct，mm\_struct, vm\_area\_struct, vm\_operations\_struct, page等

1. 分析图1（注：图1是2级页表，对应于IA-32位系统），解释图中**每一类方框**和箭头的含义，在代码树中寻找相关数据结构片段，做简单解释。30分。
2. 参考图2解释内核层不同内存分配接口的区别，包括\_\_get\_free\_pages，kmalloc，vmalloc等，3分。
3. 参考[Anatomy of a Program in Memory](https://manybutfinite.com/post/anatomy-of-a-program-in-memory/)和[User-Level Memory Management](https://linuxdevices.org/ldfiles/misc/Linux_Programming_by_Example_ch03.pdf)中例程，写一个实验程序mtest.c，生成可执行程序mtest；打印代码段、数据段、BSS，栈、堆等的相关地址；需要创建自己的例子，不允许简单照搬，8分。
4. 参考[How The Kernel Manages Your Memory](https://manybutfinite.com/post/how-the-kernel-manages-your-memory/)，通过/proc/pid\_number/maps，分析mtest各个内存段（参考[链接](https://blog.csdn.net/lijzheng/article/details/23618365)）。绘制图表，解释输出的每一段的各种属性，包括每一列的内容。为了让mtest程序驻留内存，可以在程序末尾加上长时睡眠，并将mtest在后台运行，即./mtest & 6分。
5. 参考[A Malloc Tutorial](https://wiki-prog.infoprepa.epita.fr/images/0/04/Malloc_tutorial.pdf)以及相关资料（如[链接](https://blog.csdn.net/gfgdsg/article/details/42709943)）回答以下问题：3分
6. 用户程序的内存分配涉及brk/sbrk和mmap两个系统调用，这两种方式的区别是什么，什么时候用brk/sbrk，什么时候用mmap？
7. 应用程序开发时，为什么需要用标准库里的malloc而不是直接用这些系统调用接口？malloc额外做了哪些工作？
8. malloc的内存分配，是分配的虚拟内存还是物理内存？两者之间如何转换？
9. （**附加题，10分**）模仿malloc接口，实现一对简单的函数，命名为myalloc/myfree，实现堆上的动态内存分配和释放，并提供测试函数。相关代码以myalloc.c文件提供在项目目录下面。在自己的机器上进行实验，观察随着malloc/free的行为，/proc/pid\_number/maps中如何反映堆内存的变化情况，给出截屏和解释。实现基本功能5分，在内存块管理方面进行专门优化5分。

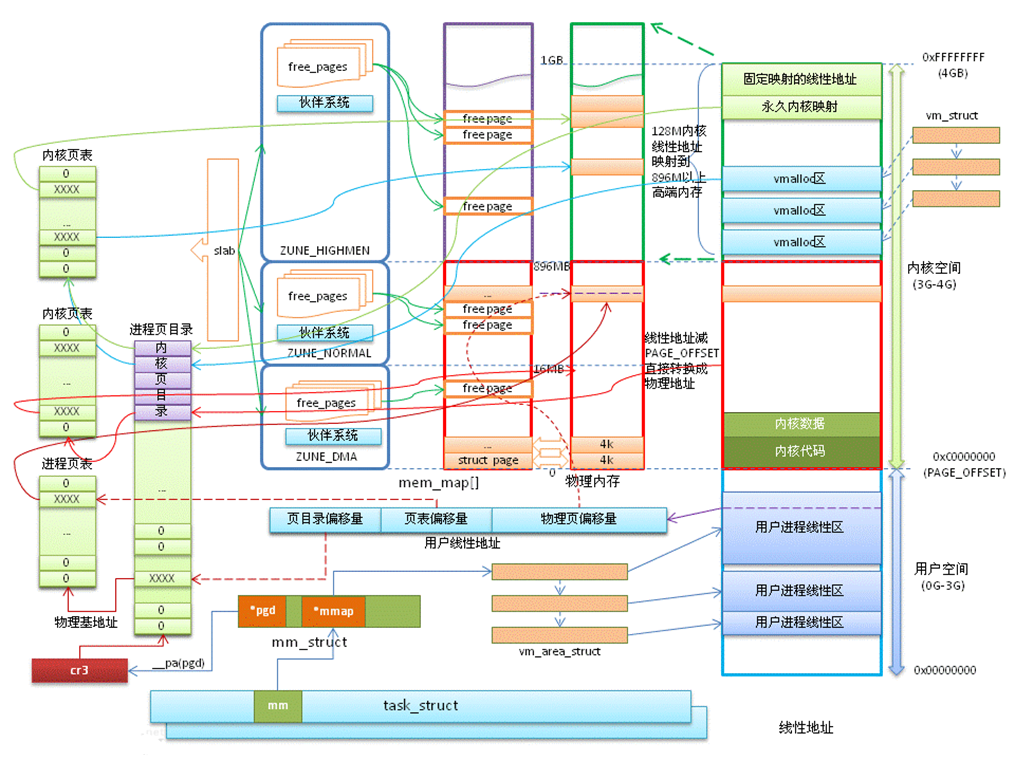
****

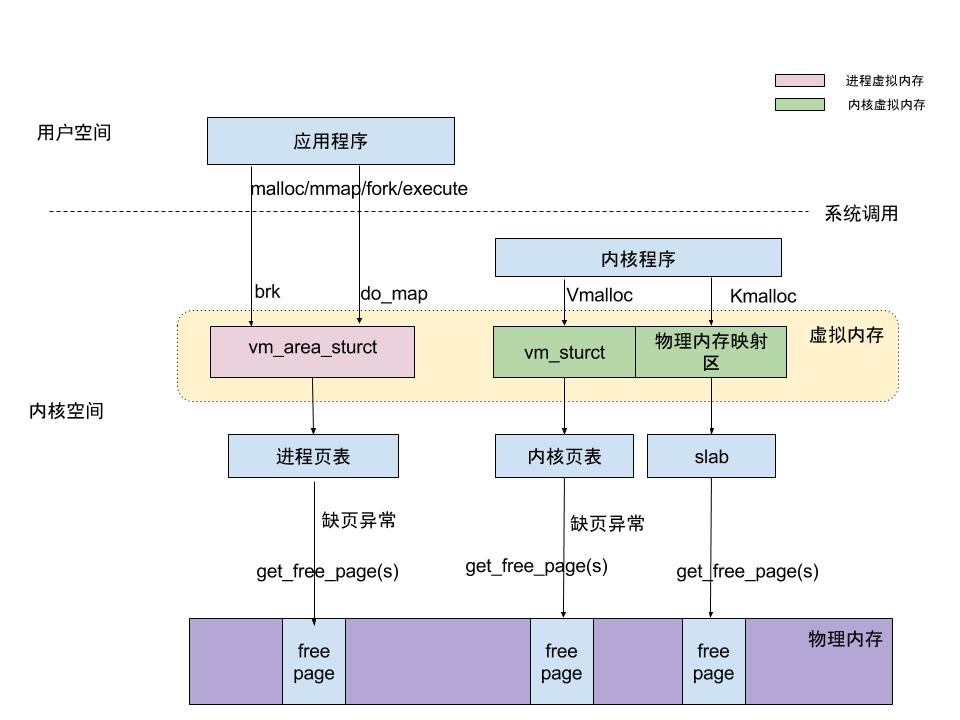
图1. Linux内核内存管理示意图（IA\_32）

图2. 用户空间和内核空间内存分配示意图