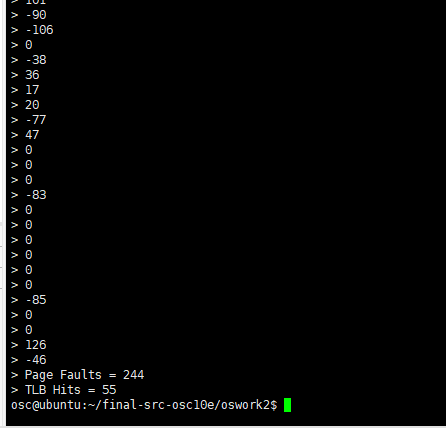
操作系统大作业2

自动化2班 吴天然 17364077

1. 虚存管理模拟程序
2. 代码为vm.c
3. 运行结果：
4. Linux内存管理实验
5. Slab主要为slab分配器，其作用为：对于频繁地分配和释放的数据结构，会缓存它；频繁分配和回收比如导致内存碎片，为了避免，空闲链表的缓存会连续的存放，已释放的数据结构又会放回空闲链表，不会导致碎片；让部分缓存专属单个处理器，分配和释放操作可以不加SMP锁。Slab层把不同的对象划分为高速缓存组，每个高速缓存组都存放不同类型的对象，每个对象类型对应一个高速缓存，并且每个高速缓存都是用kmem\_cache结构来表示，slab层会把高速缓存组释放到内核中的页的不同分区中。

内核把页划分在不同的区，有三个区，ZONE\_DMA（DMA使用的页，物理内存<16MB）, ZONE\_NORMAL（可正常寻址的页 物理内存16~896）, ZONE\_HIGHMEM（动态映射的页 物理内存>896）。free\_pages为页释放函数。页中的不同区的数据经过页释放函数被释放到内存管理区然后再被释放到物理内存中，同时，进程页表和内核页表中的数据也可以直接被释放到物理内存中。

物理内存只有进程真正去访问虚拟地址，发生缺页中断时，才分配实际的物理页面，建立物理内存和虚拟内存的映射关系，应用程序操作的是虚拟内存；而处理器直接操作的却是物理内存。当应用程序访问虚拟地址，必须将虚拟地址转化为物理地址，处理器才能解析地址访问请求，物理内存是通过分页机制实现的。

虚拟内存转化为真是物理内存时，虚拟进程空间：通过查询进程页表，获取实际物理内存地址；虚拟内核空间：通过查询内核页表，获取实际物理内存地址；物理内存映射区：物理内存映射区与实际物理去偏移量。

用户空间中，用户空间中的进程的内存被称为进程地址空间。内核空间中包含的对象有代码段（可执行文件代码），数据段（可执行文件的已初始化全局变量），bss段（程序中未初始化的全局变量，零页映射），以及任何内存映射 文件和任何共享内存段。4G进程地址空间被划分两部分，内核空间和用户空间。用户空间从0到3G，内核空间从3G到4G。用户进程通常情况只能访问用户空间的虚拟地址，不能访问内核空间虚拟地址。只有用户进程进行系统调用(代表用户进程在内核态执行)等情况可访问到内核空间，内核空间是由内核负责映射，不会跟着进程变化；内核空间地址有自己对应的页表，用户进程各自有不同额页表。

内核空间中还存在虚拟内存区域。mm\_struct为内存描述符，其中mmap所描述的对象是相同的，以链表形式存放，利于高效地遍历所有元素，其中mmap中可能还含有缓存虚拟内存区域的地址。Task\_struct代表进程的进程描述符。内核线程直接使用前一个进程的内存描述符，仅仅使用地址空间中和内核内存相关的信息。

应用程序操作的对象时映射到物理内存之上的虚拟内存，而处理器直接操作的是物理内存。故应用程序访问一个虚拟地址时，需要将虚拟地址转换为物理地址，然后处理器才能解析地址访问请求，这个转换工作通过查询页表完成。顶级页表为页全局目录（PGD），指向二级页目录；二级页表为中间页目录（PMD），指向PTE中的表项；最后一级为页表（PTE），指向物理页面。

所有进程都必须占用一定数量的内存，这些内存用来存放从磁盘载入的程序代码，或存放来自用户输入的数据等。内存可以提前静态分配和统一回收，也可以按需动态分配和回收。

采用虚拟内存管理技术，每个进程都有各自独立的进程地址空间(即4G的线性虚拟空间)，无法直接访问物理内存。这样起到保护操作系统，并且让用户程序可使用比实际物理内存更大的地址空间。

2.get\_free\_pages为页分配函数，分配页后，会返回指向其逻辑地址的指针；kmalloc函数返回的是一个指向内存块的指针，其内存块大小至少为size,所分配的内存在**物理内存**中连续且保持原有的数据(不清零)；

Vmalloc函数返回的是一个指向内存块的指针，其内存块大小至少为size,所分配的内存是**逻辑上连续**的。

5.（1）**brk是将数据段(.data)的最高地址指针\_edata往高地址推，mmap是在进程的虚拟地址空间中（堆和栈中间，称为文件映射区域的地方）找一块空闲的虚拟内存。**

（2）进程所面对的虚拟内存地址空间，只有按页映射到物理内存地址，才能真正使用。且受物理存储容量限制，整个堆虚拟内存空间不可能全部映射到实际的物理内存。

（3）malloc所申请的内存主要从Heap区域分配，虚拟内存和物理内存通过页表的映射关系转换。