Lab4实验报告

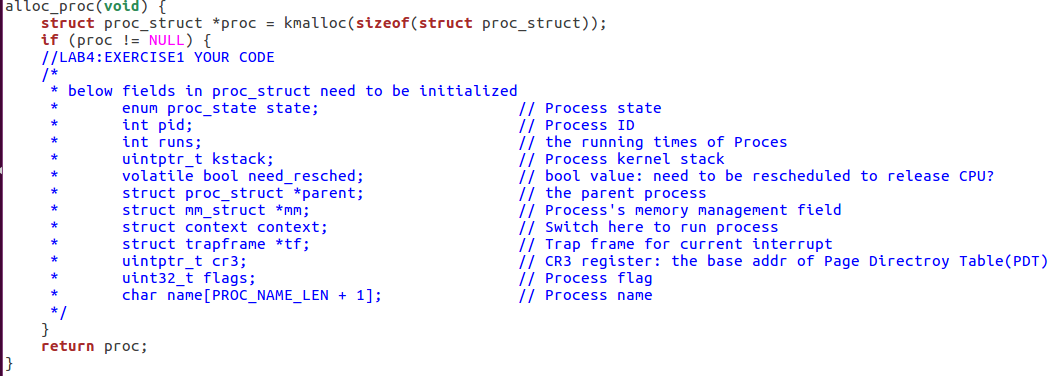
班级：计算机科学与技术2班

姓名：许金曼

学号：171491220

1.分配并初始化一个进程控制块（需要编码）

在kern/process/proc.c中找到alloc\_proc函数部分，注释说明我们要对struct proc\_struct进行初始化



proc->state = PROC\_UNINIT; //进程所处的状态，有PROC\_UNINIT（未初始状态）、PROC\_SLEEPING（睡眠）、PROC\_RUNNABLE（运行与就绪态）、PROC\_ZOMBIE四种（僵死状态）

proc->pid = -1;

proc->runs = 0;

proc->kstack = 0; //每个线程都有内核栈，并且位于内核地址空间的不同位置

proc->need\_resched = 0;

proc->parent = NULL; //父进程

proc->mm = NULL; //内存管理信息，包括内存映射列表、页表指针等

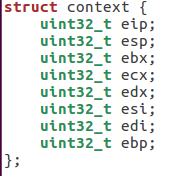
memset(&(proc->context),0,sizeof(proc->context)); //进程上下文，保存寄存器，用于进程切换

proc->tf = NULL; //中断帧，记录了进程在中断前的状态

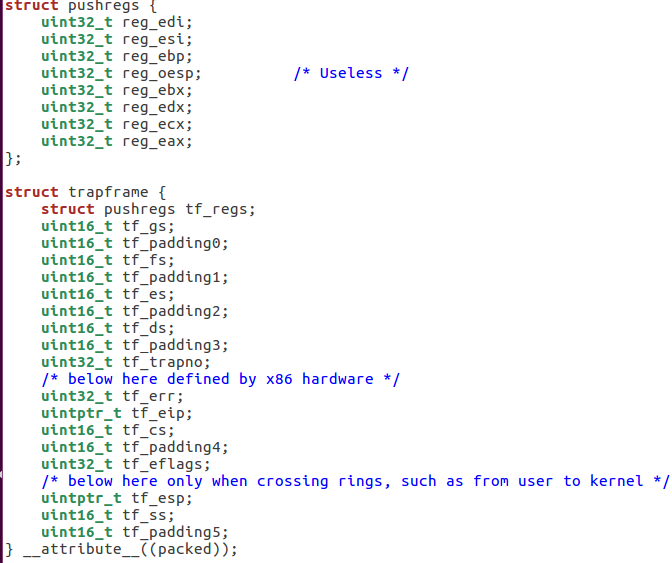
proc->cr3 = boot\_cr3;

proc->flags = 0;

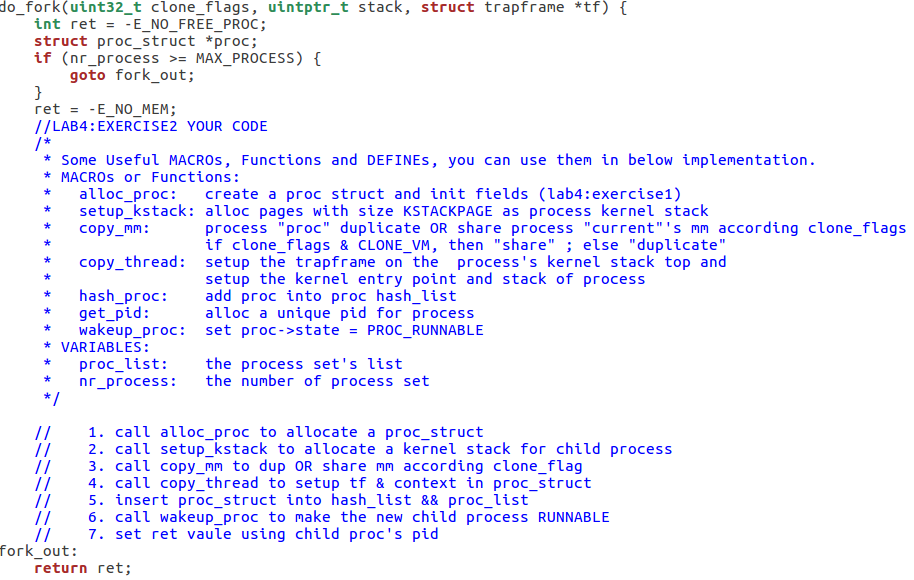
memset(proc->name,0,PROC\_NAME\_LEN);



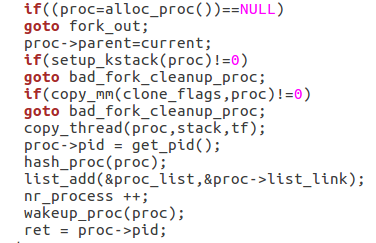
context 完成切换

tf 用来保存中断帧

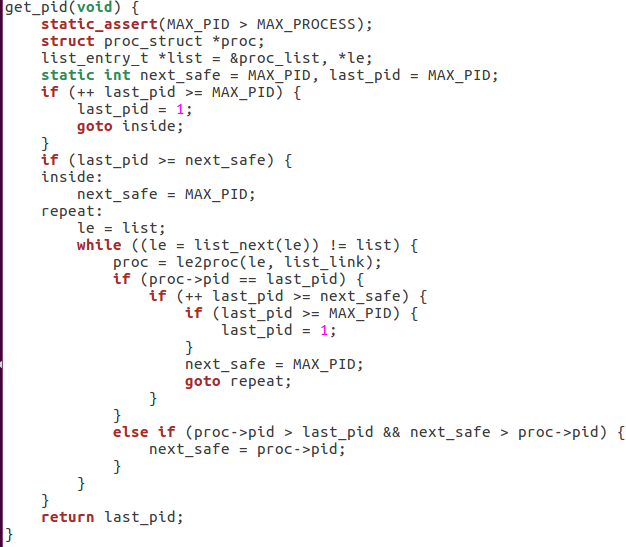
2.为新创建的内核线程分配资源（需要编码）



根据注释，完成代码：

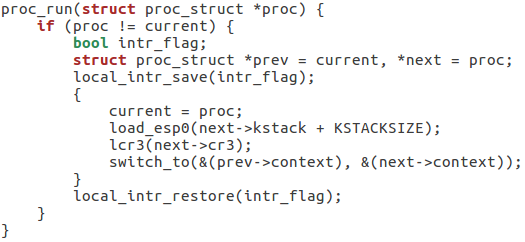


根据get\_pid()函数：



如果proc\_pid与last\_pid相等，则将last\_pid+1，如果last\_pid>=MAX\_PID,将其变为1，确保了last\_pid<pid<next\_safe,所以get\_id()为每个调用fork的线程返回不同的id

3.阅读代码，理解proc\_run函数和它调用的函数如何完成进程切换的。



首先判断要调度的进程是不是当前进程，如果不是的话，则关闭中断，将当前进程设为待调度进程，加载待调度进程的内核栈基地址，将当前的cr3寄存器改为需要运行进程的页目录表，进行上下文切换，保存原线程的寄存器并恢复调度线程的寄存器，再恢复中断。