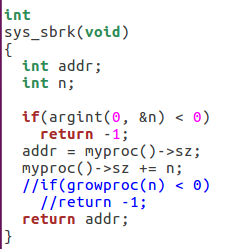
## Part 1

Homework 要求修改 sbrk syscall，对系统调用sbrk的实际实现sys\_sbrk进行修改，只将进程的内存空间大小增加n而不进行实际的分配，即把调用growproc()函数的部分注释掉并增加内存大小。修改后的sys\_sbrk()函数如下。



其中，返回的地址是新分配的地址空间的开头，在此处就是原来地址空间的末尾。我们增加了myproc()->sz，但实际却并未增加进程大小。  
启动xv6后输入echo hi命令，得到如下错误提示：



因为在shell中运行echo时，需要调用malloc函数。运行malloc的时候，虽然返回是成功了的，但是当程序试图操作cmd指向的内存区域的时候，发现该内存区域不是当前进程所有的，因为在sys\_sbrk中根本没分配。

## Part 2

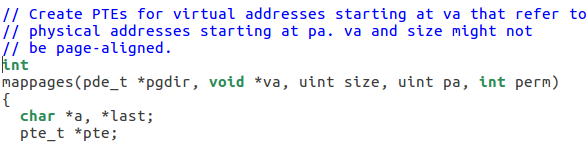
修改trap.c中的代码，使得系统可以处理上述错误。

在trap.c中，当发现是page fault错误的时候，可以按照当前进程的myproc()->sz来实际的分配内存，注意这个时候的sz的大小不是实际的大小，而是希望的大小值。

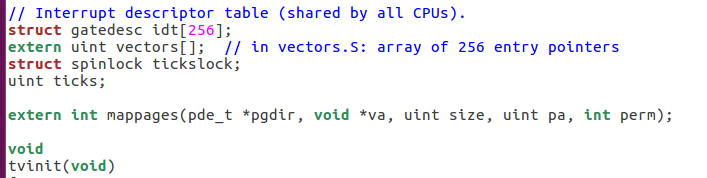
所以首先应该获取发生page fault时刻的虚地址，那个地址之后的部分就应该是本来应该分配但是实际没有分配的，而实际需要分配多少，应该根据myproc()->sz的大小来定。

因为在发生page fault的时候的地址，是我们在malloc之后返回给进程的地址，而这个地址又是原来的myproc()->sz，所以该虚地址的大小就应该是实际的内存的大小。

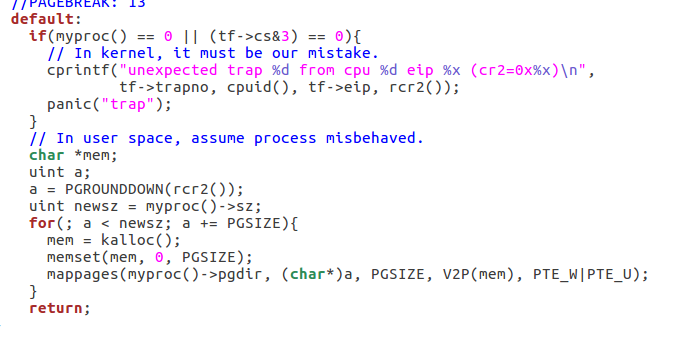
需要在trap.c中调用vm.c中的int mappages()函数，所以要去除原本的static关键字。



在trap.c中在调用之前使用extern关键字声明int mappages()函数:



在trap.c中的void trap(struct trapframe \*tf)的defaut部分添加以下代码，放置的位置为if模块后:



再次编译运行，得到正确结果。

