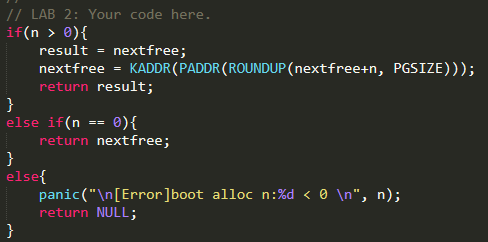
**PartA**

PartA要求实现虚拟内存的初始化，本质上就是设置好管理页的一些元数据，需要实现五个函数，实现思路及具体代码如下所示：

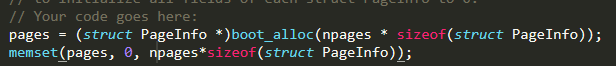
* boot\_alloc()

该函数是一个物理内存分配器，仅在JOS设置它的虚拟内存时使用，page\_alloc()是虚拟内存分配器。根据注释可以知n大于0时分配相应的空间大小，并返回虚拟地址；如果为0则返回下一个空闲页，不实际分配任何空间；小于则输出对应的错误信息。nextfree已经完成初始化，我们只需要将当前的nextfree返回并将nextfree指向后续空闲空间即可，其间需要使用PADDR和KADDR完成两次转换用于检测是否超出了可用空间。代码如下：



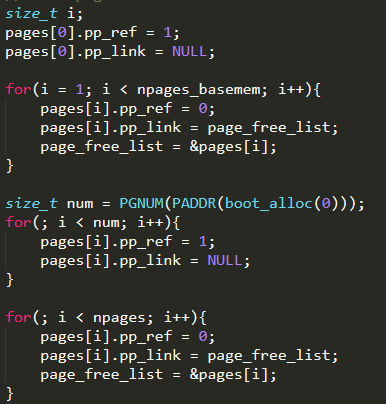
* men\_init()

该函数建立一个两级页表，仅设置UTOP即用户区上方的内核部分的空间。函数做好了大部分初始化，包括获取内存大小、初始化页表目录。我们需要完成的是给所有的页申请空间并初始化，需要申请npages页，代码如下所示：



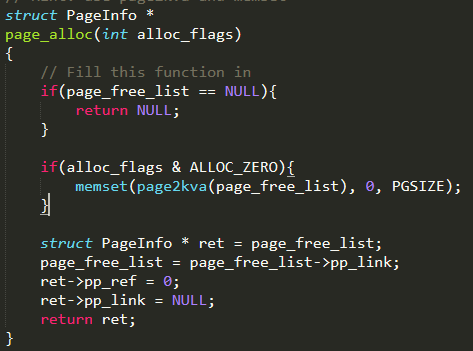
* page\_init()

该函数初始化page结构和空闲页链表，初始化之后所有的页的分配和释放都采用对应的函数完成不再使用boot\_alloc()。第0页应该被标记为使用从而保护实模式IDT和BIOS以防之后可能会用到；之后的PGSIZE-npages\_basemem\*PGSIZE都是空闲的；设备IO空洞IOPHYSMEM-EXTPHYSMEM已经被分配。之后的页部分被内核和页表占用需要自己确认分配。关键是需要我们自己确认哪些是已被分配的部分，注意到boot\_alloc()如果接受到的参数为0则会返回下一个空闲页的虚拟地址，计算出该页的页数我们即可计算出哪些页已经被分配了（其中已经包含了设备IO洞的部分），因此初始化代码如下所示：



* page\_alloc()

该函数需要分配一个页空间，首先检查页空闲链表，如果有空闲页则判断是否需要清零，如果需要则利用page2kva函数获取其物理地址并清零。题目要求分配器不需要修改ref值，因此将其置零将Link变为null返回即可，空闲链表头指向下一位。代码如下所示：



* page\_free()

该函数返回一个空页到空闲链表中，需要注意判断该页是否被页目录条目引用或已经在空闲链表中，如果是则报错，如下所示：

