**华东师范大学数据学院上机实践报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**： 操作系统 | **年级**：2019级 | **上机实践成绩**： |
| **指导教师**：翁楚良 | **姓名**：周辛娜 |  |
| **上机实践名称**： 内存管理 | **学号**： | **上机实践日期**：6.15 |
| **上机实践编号**：4 |  |  |

1. **目的**

1.熟悉Minix操作系统的进程管理

2.学习Unix风格的内存管理

1. **内容与设计思想**

修改Minix3.1.2a的进程管理器，改进brk系统调用的实现，使得分配给进程的数据段+栈段空间耗尽时，brk系统调用给该进程分配一个更大的内存空间，并将原来空间中的数据复制至新分配的内存空间，释放原来的内存空间，并通知内核映射新分配的内存段。

1. **使用环境**

VMware Workstation Pro

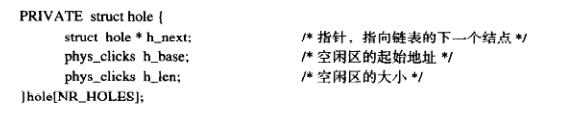
Visual studio code

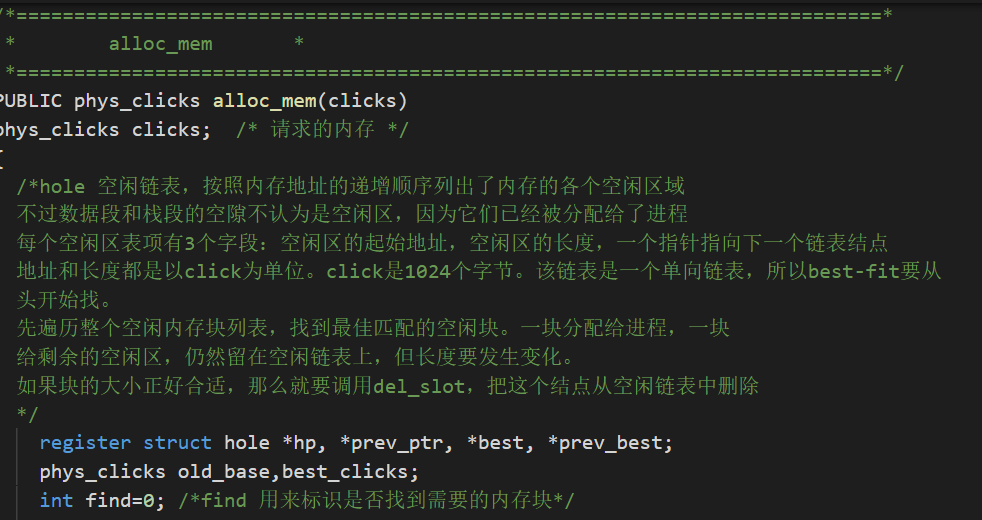
Xftp

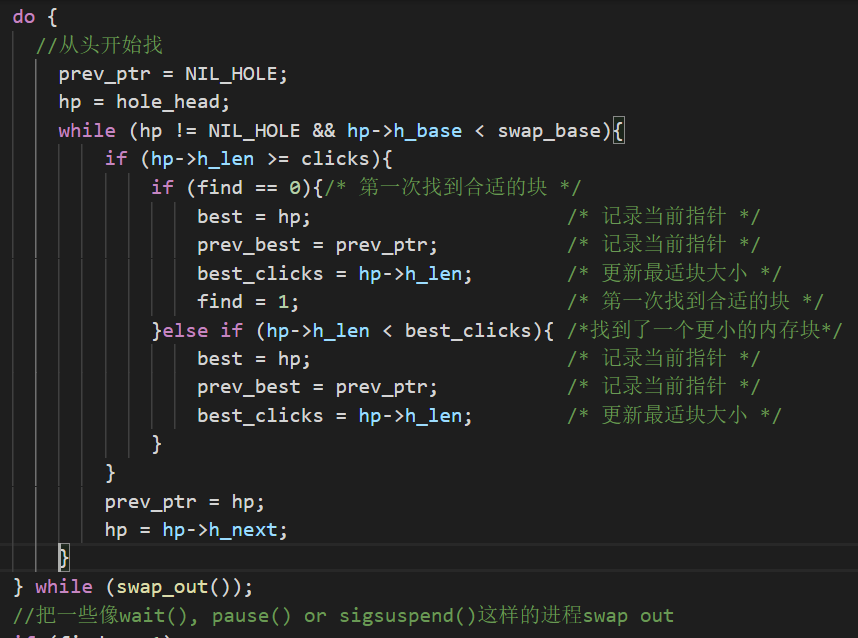
1. **实验过程**

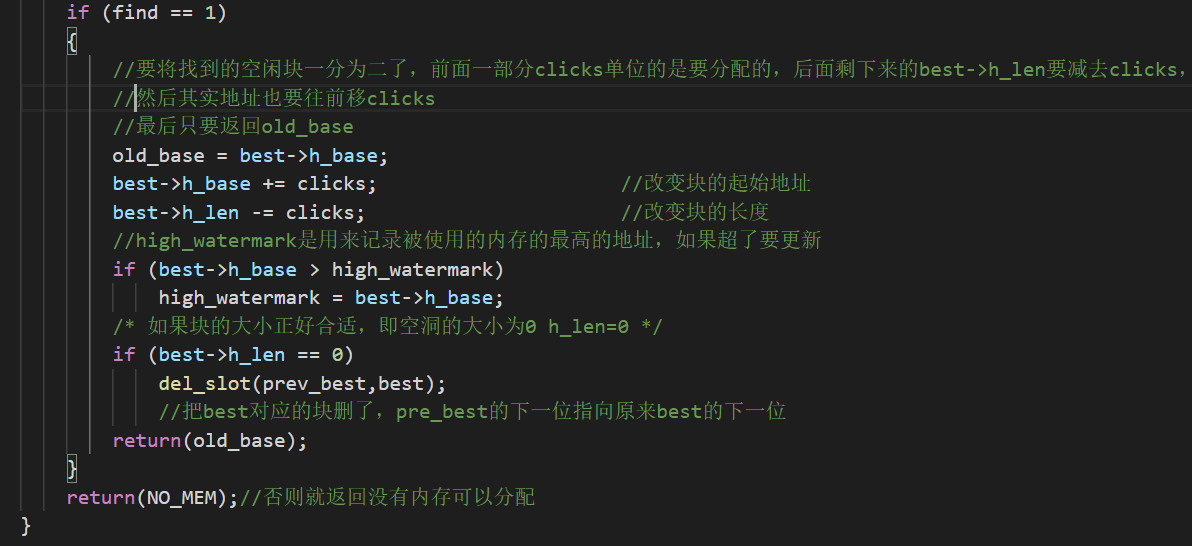
修改内存分配：

1.修改/usr/src/servers/pm/alloc.c中的alloc\_mem函数，把first-fit修改成best-fit，即分配内存之前，先遍历整个空闲内存块列表，找到最佳匹配的空闲块。



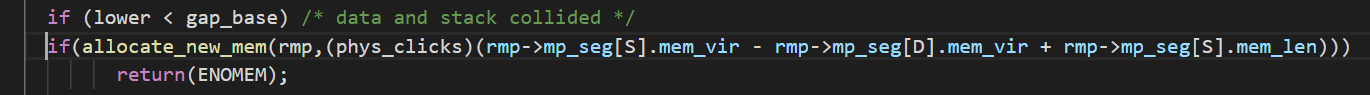






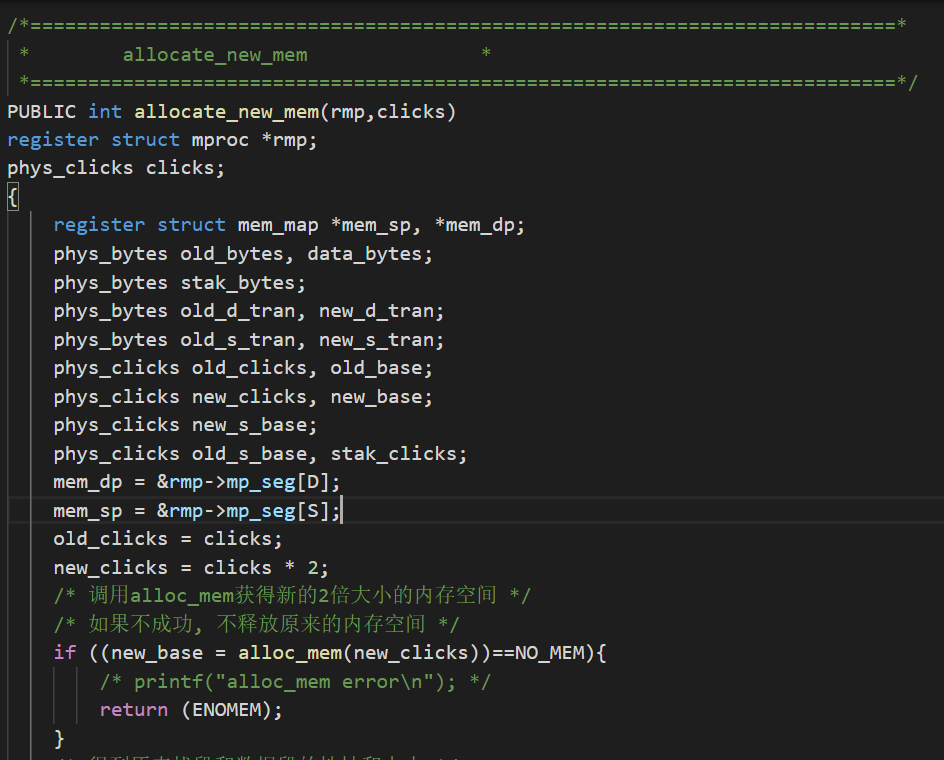
2.修改/usr/src/servers/pm/break.c中的adjust函数，并增加了一个allocate\_new\_mem局部函数在adjust函数中调用。

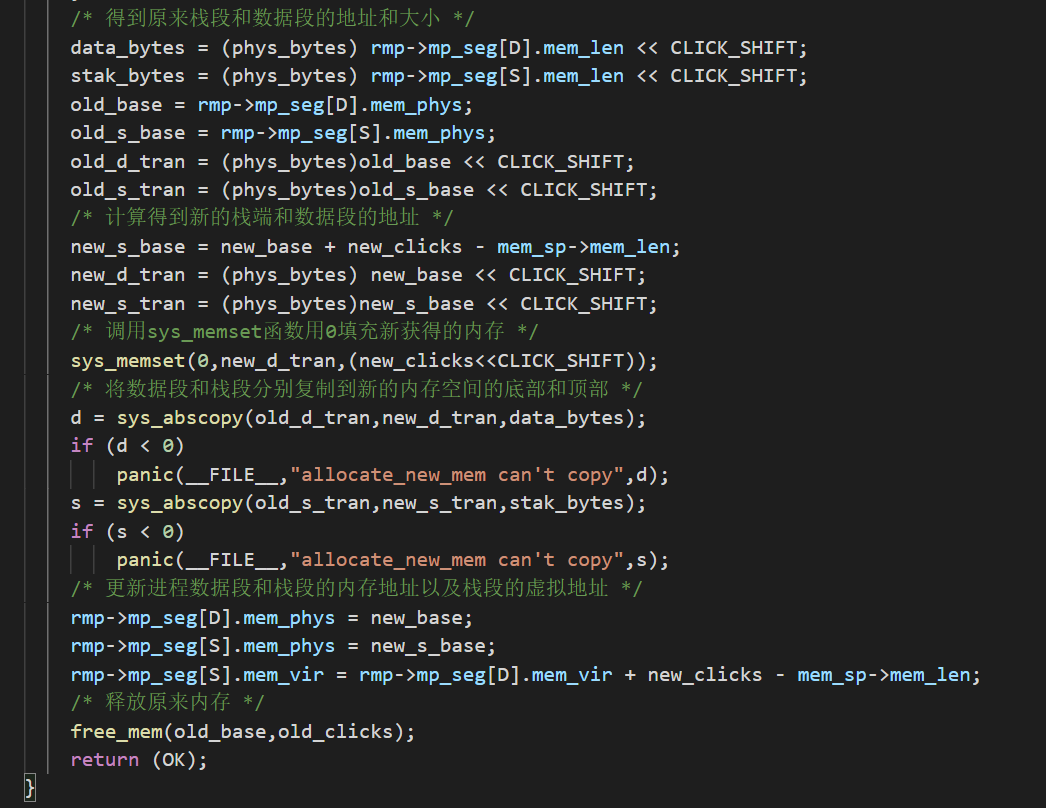
原先的 adjust 函数中当栈段地址和数据段地址有重叠时直接返回 ENOMEM, do\_brk 不能继续为数据段增加空间，修改之后, 当 lower < gap\_base 时, 调用新增的 allocate\_new\_mem 函数申请新的足够够大的空间, 将进程复制到新的内存空间并通知内核程序的数据映像发生了改变。



上面alloc.c是分配内存，现在是内存已经分配好了，就是数据段和栈段怎么分的问题，主要会调整数据段和栈段的空隙，要么增加，要么减少。

allocate\_new\_mem函数首先需要申请新的足够大的内存空间，这里是申请为原来的2倍，然后将程序现有的数据段、堆栈段的内容分别拷贝至新内存区域的底部(bottom)和顶部(top)；通知内核程序的映像发生了变化；返回do\_brk函数。





**2.编译MINIX**

进入 /usr/src/servers 目录, 输入 make image, 编译成功后输入 make install 安装新的PM程序。

进入 /usr/src/tools 目录, 输入 make hdboot, 成功后输入 make install 安装新的内核程序。

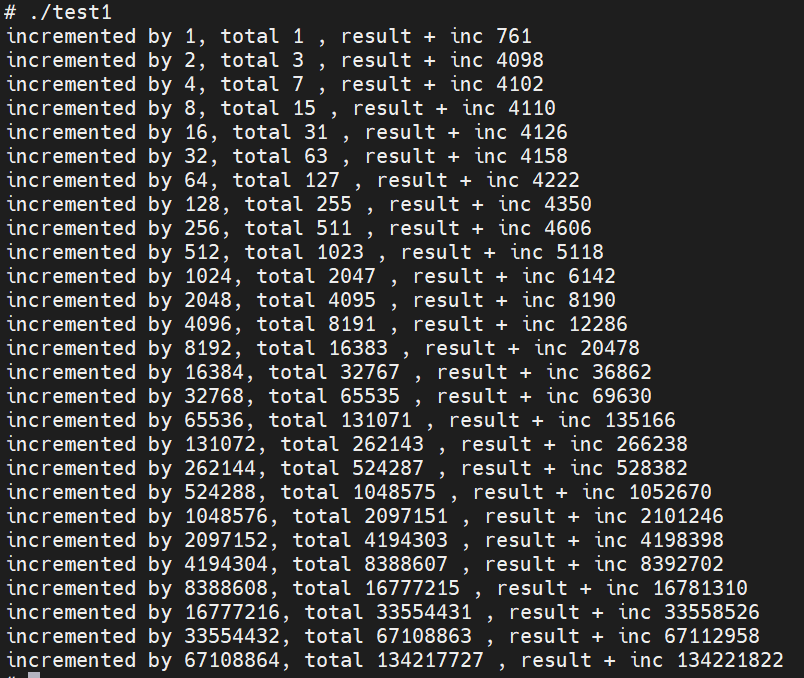
输入 shutdown 命令关机, 进入 boot monitor 界面, 设置启动新内核选项。

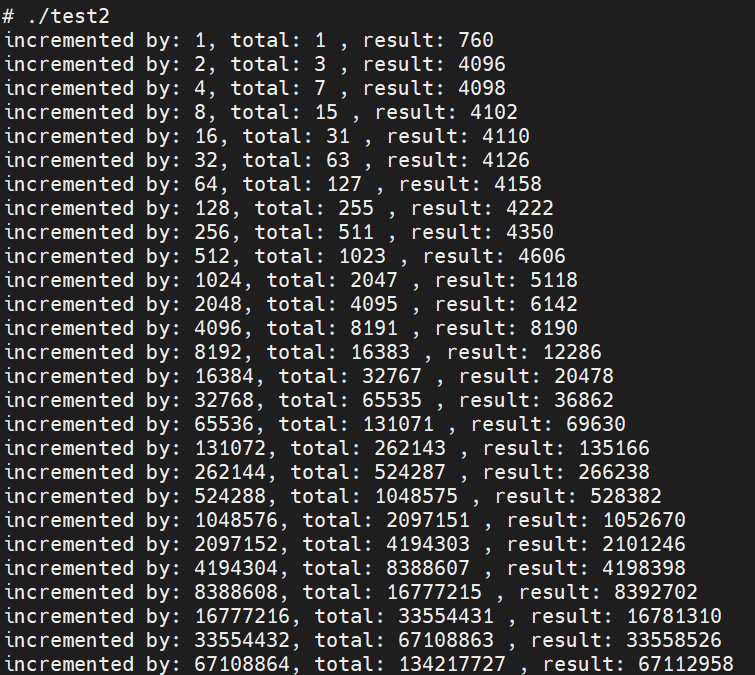
输入 newminix(5,start new kernel){image=/boot/image/3.1.2ar1;boot;} 其中 5 为启动菜单中的选择内核版本的数字键 (可选其他值), 3.1.2ar1 为在 /usr/src/tools 目录输入 make install 之后生成的内核版本号。

输入 save 保存设置。

输入 menu, 然后输入之前设置的数字键启动新内核进行测试。

**实验结果：**





1. **总结**

本次实验也碰到了一些问题：

1. 在minix3.1.2环境配置当中，发现找不到IP，一个重要的原因在于在下载镜像文件时只下载了一个压缩文件，而没有将2个都下载
2. 最终要编译测试文件时，没有反应过来在minix3.1.2中没有gcc、clang，而是cc，但由于后面由于已经是新内核，而编译的话是要在原来的内核，所以只能用已经编译好的test1,test2。