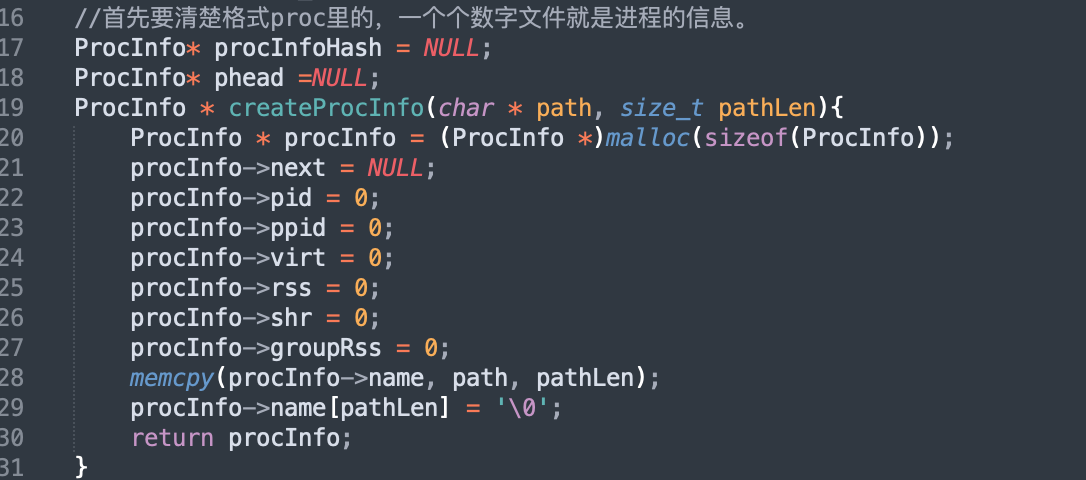
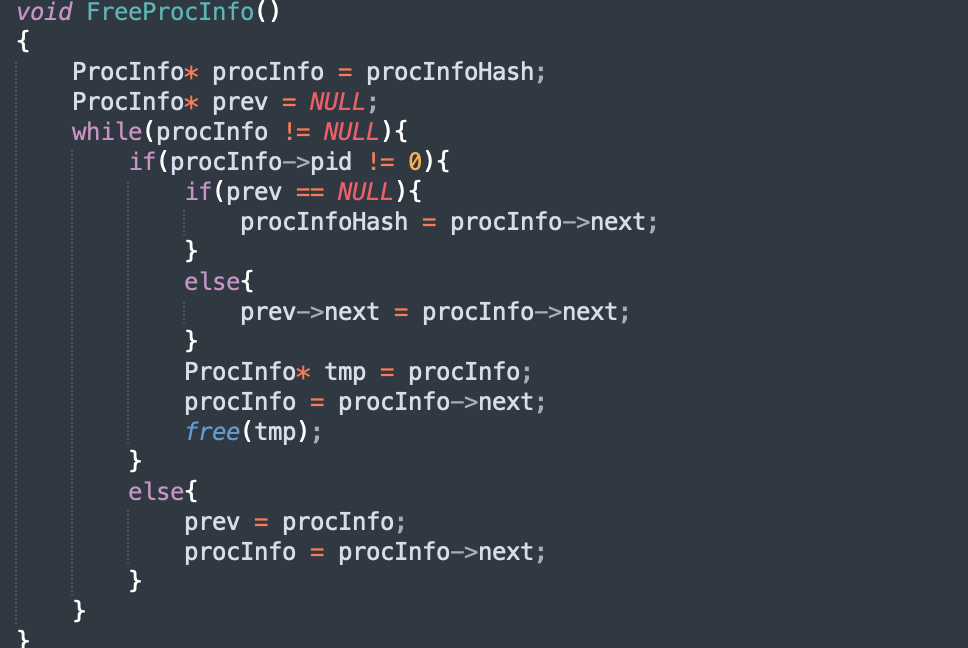
创建一个头结点



Free这里应该更改为pid!=0的时候进行释放内存



不是空状态的时候进行全部清空

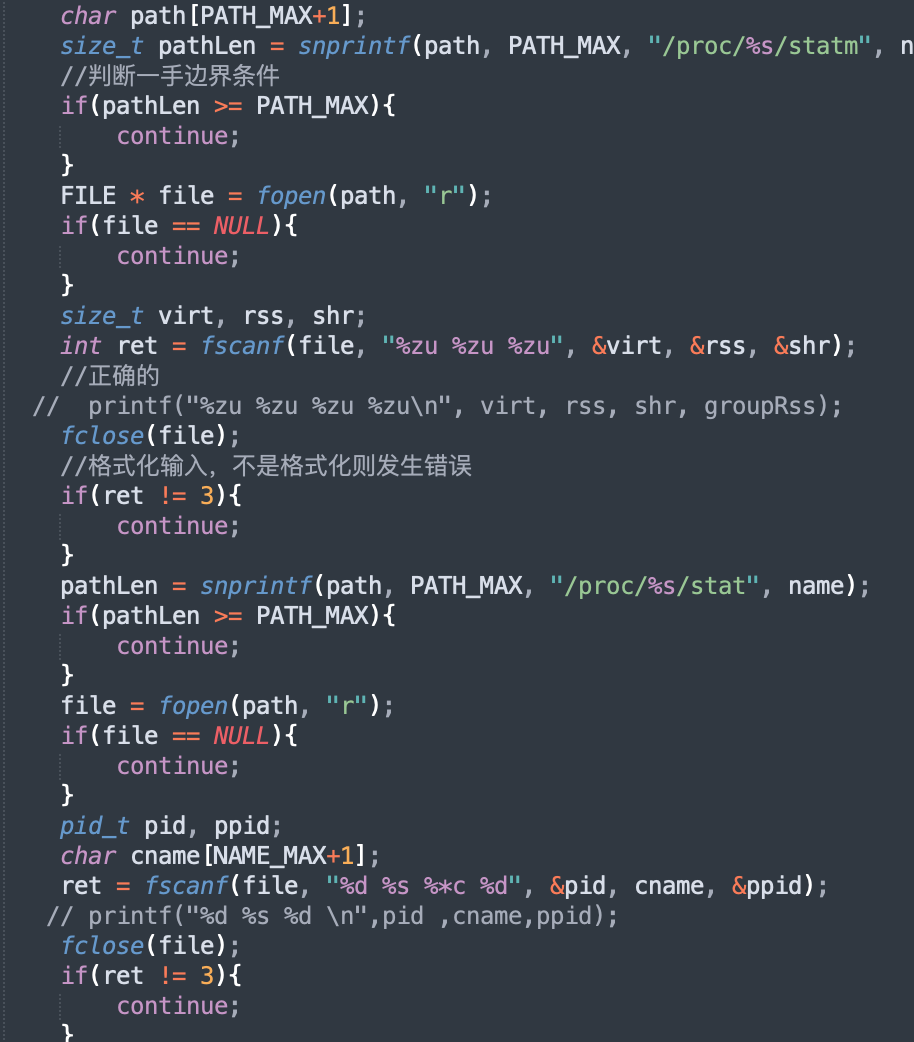


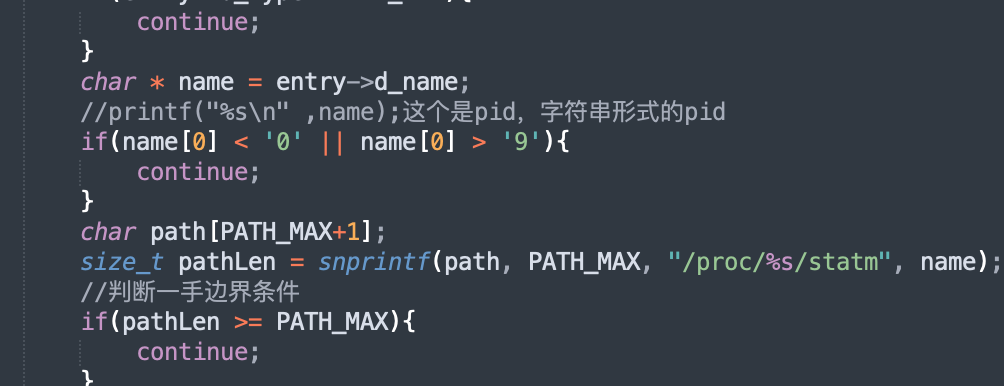
用第一个字母进行判断是否为进程文件，

然后读取stat和statm中两个文件中的信息，

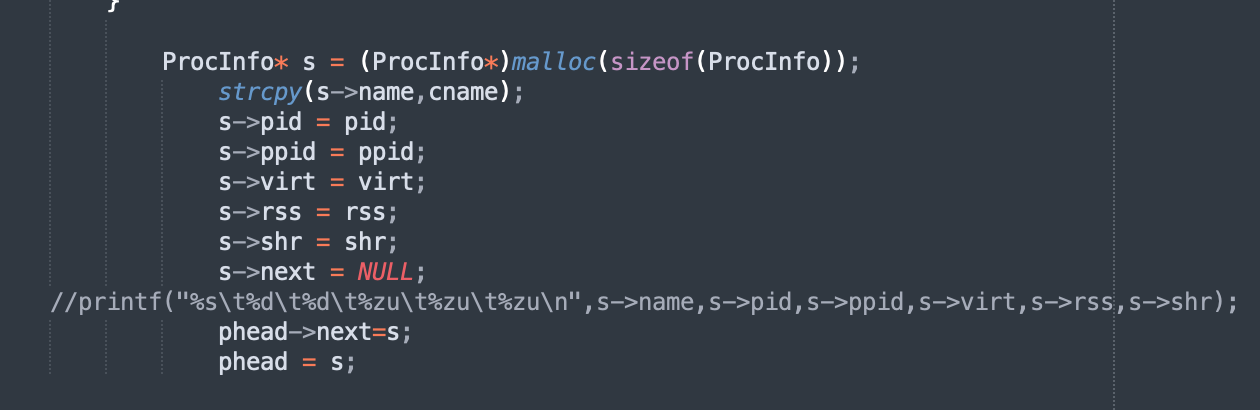
我是在这一个

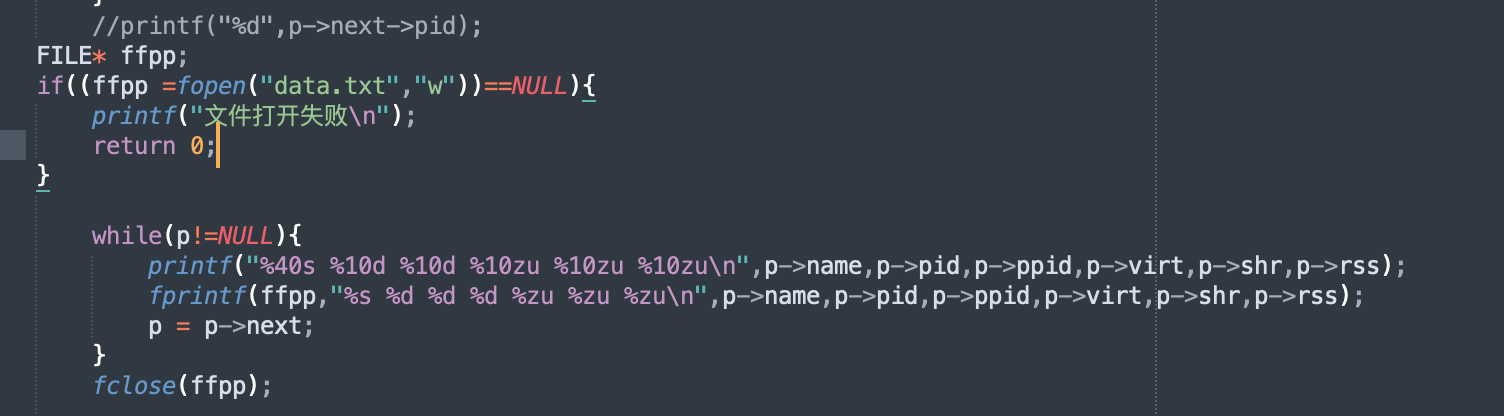
ProcInfo\*syncProcInfo()这一个函数中读取了全部的信息。



用的是fscanf的格式化输入

进行链表建立



进行格式化的输出，对齐。

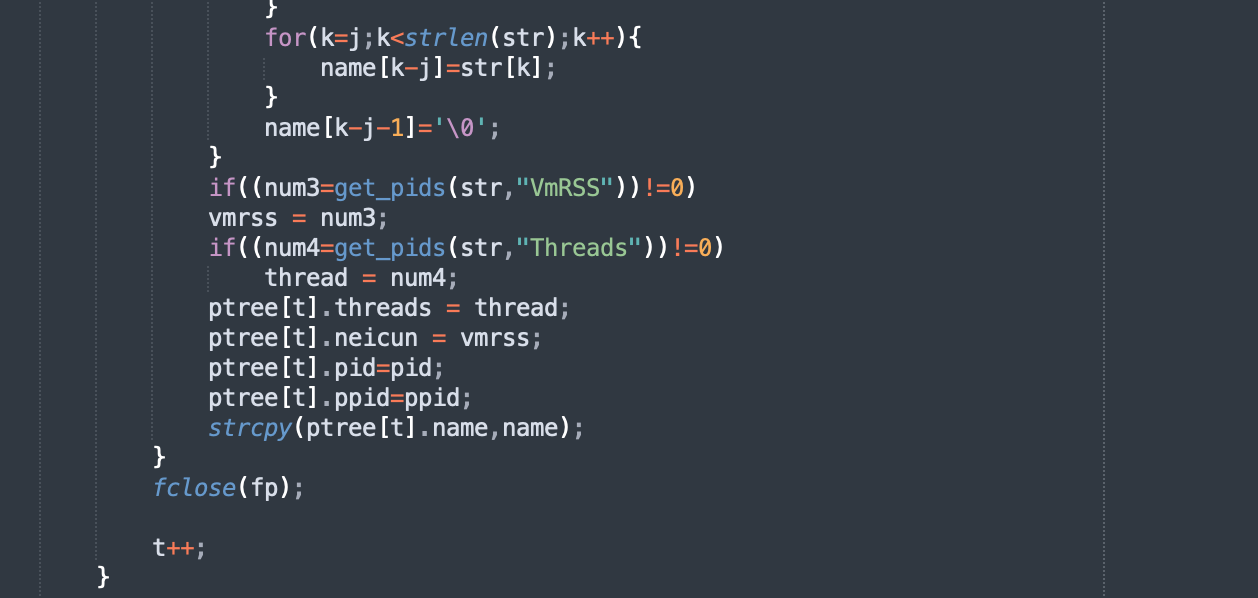
表格

描述已自动生成

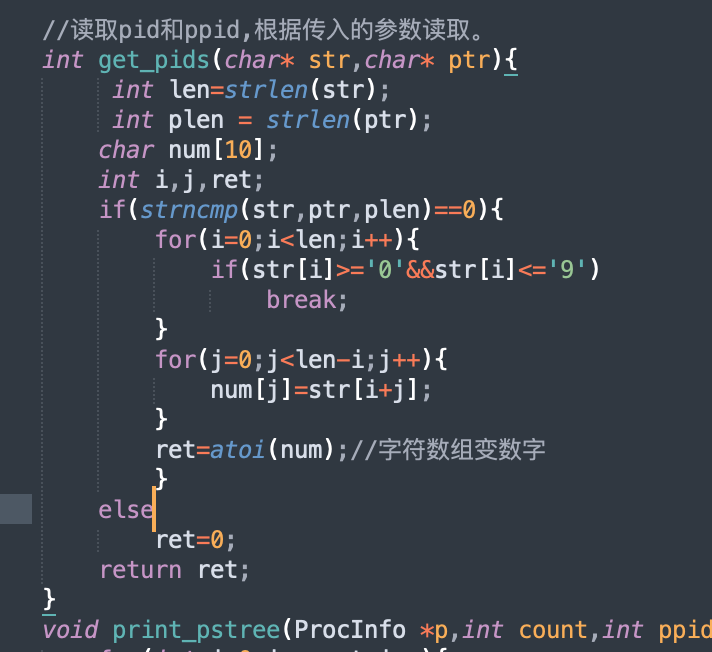
1.2打印进程树的过程

用的是递归的方式，在一颗树下遍历到叶子结点。

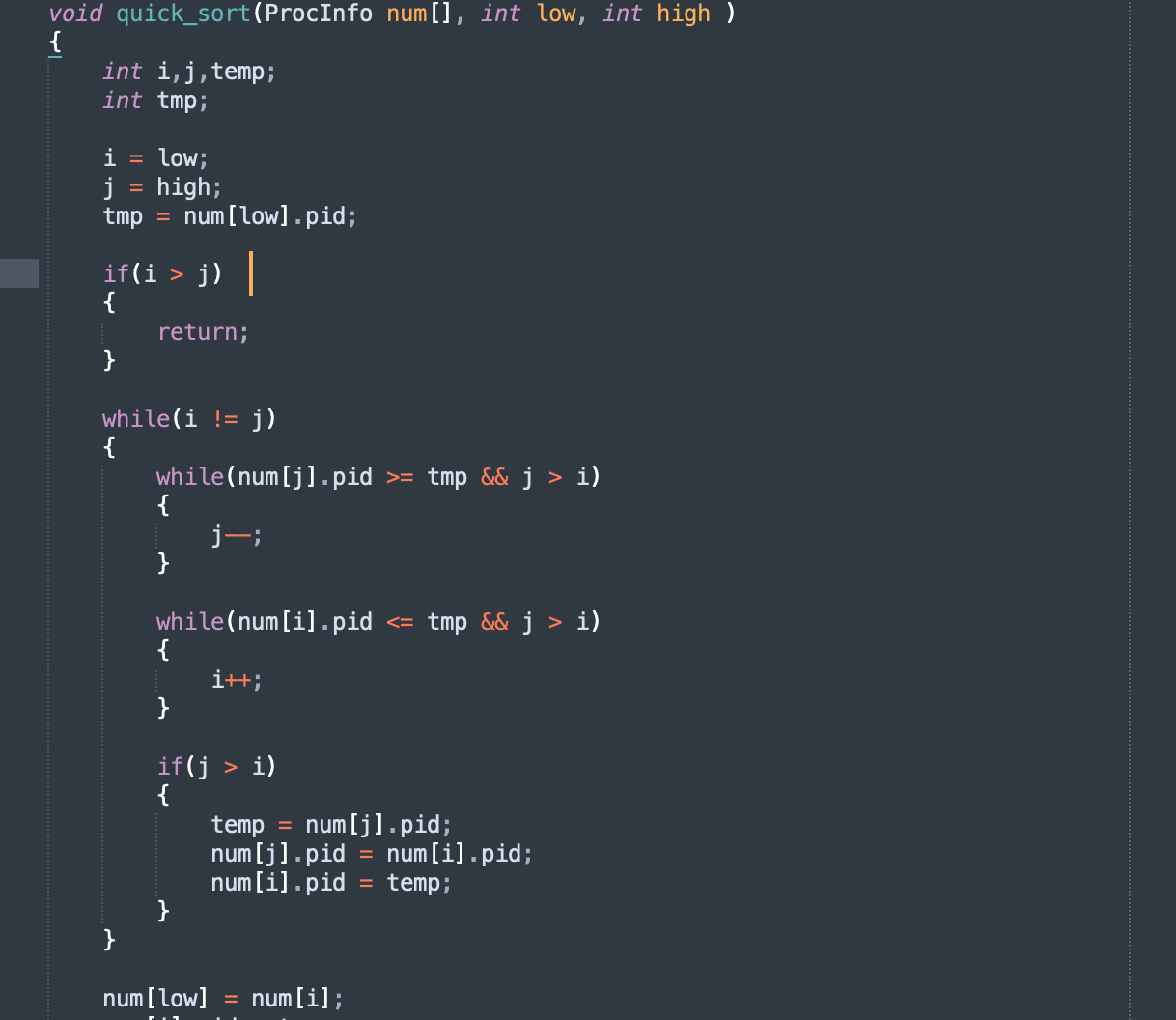
结构体数组下的遍历，根据ppid进行遍历。



输入两个参数，获取目标信息



根据pid进行快速排序。



2.3我的想法

（确实是非常的麻烦，但是目前水平有限，听了老师讲的目前也只能想到这里。。）

我采用的是用txt文档记录上一次刷新时的进程的记录。

默认标志位是0

输入不同的标志位

1，2，3

1是更新，同时更新data.txt

2是部分更新，加上与现在获得的信息进行比对，根据更新的信息，进行与data.txt比对，

如果是里面没有的，就加进去，如果是有多就不用更新，每一个有个flag位，如果是没有重合的就为0，说明是旧的死亡的进程，删去，再输出data.txt文件。

3是全面更新，取时间t与1，奇数时间刷新，偶数时间输出data.txt