**一、实验思路**

**1、driver.c**

该文件主要是实现主函数中参数的传递。将传递进来的文件打开，读取，然后写入定义好的链表中。读取时用的函数是fget（）。同时在主函数中调用执行函数shedule（counter）。counter是指进程的数目。

**2、cpu.h\Shedule.h\task.h**

cpu.h文件主要是实现输出，也就是模拟cpu。run（）函数的实现。

Schedule.h文件中主要实现了链表的插入和删除函数，另外计算平均响应时间，等待时间，周转时间的函数compute\_time()也在此实现。

task.h中定义了每个节点和链表，每个节点由四个元素name,tid,burst,priority组成以及虚函数schedule（int counter）.该函数在具体的周转算法中实现。

**3、FCFS.c(以此为例)**

首先是pickNextTask（），该函数是用来选择下一个任务的，在FCFS算法中，只需要根据tid（进入链表的顺序）的大小顺序进行选择。

shedule（int counnter）该函数是调度算法的实现，在该算法中，将选择好的任务进行输出，然后依次将周转、等待、响应时间保存。进行下一条任务的输出就可以。

***SJF算法***就是遍历整个链表，依次选出下执行时间最短的进程，然后运行并删除。

***priority算法***就是遍历整个链表，依次选出优先级最高的进程，然后运行并删除。

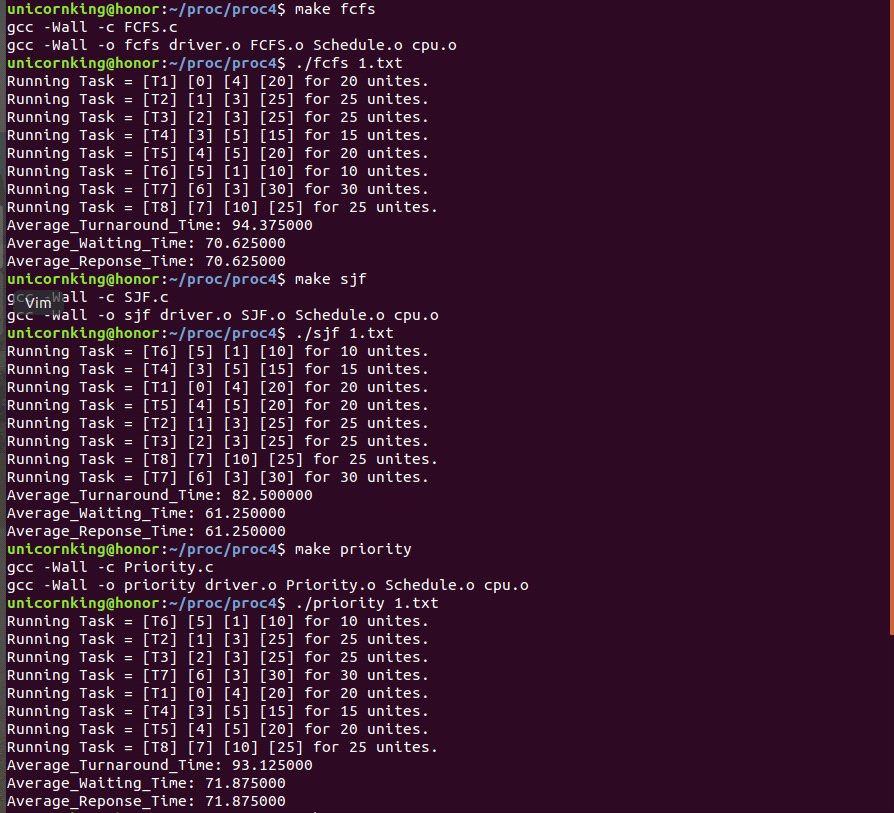
***RR算法***就是运行遍历整个链表。判断执行时间是否大于QUANTUM，若不大于就执行burst时间，若大于，执行完QUANTUM时间后，将其burst减去QUANTUM。

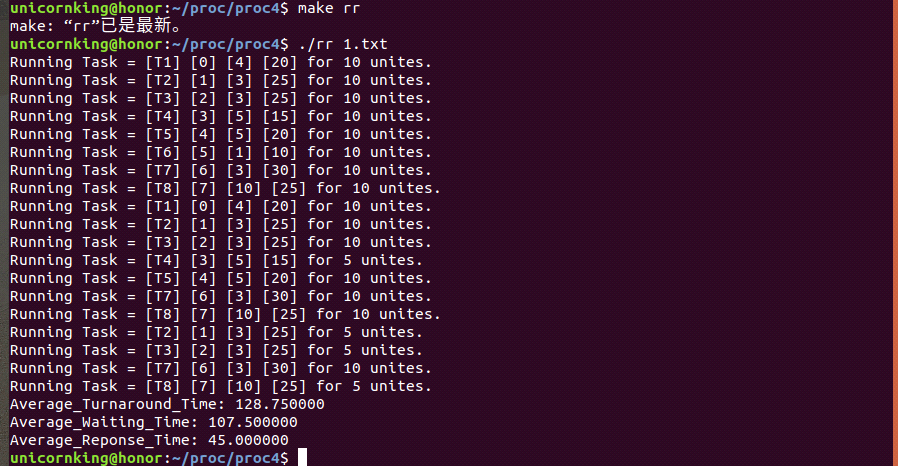
***RR\_Priority算法***的就是将相同优先级的进程号存储在一起，在挑选进程的时候，将最大优先级的进程进行RR算法的执行。若其bursttime变为0，就将其进程号删除。

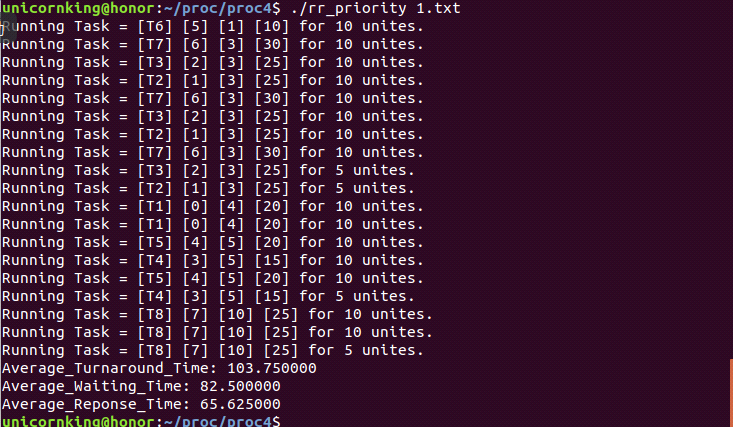
**4、makefile**

将头文件和实现函数进行编译成.o文件，然后再把算法的实现函数进行编译成.o文件，最后将所有.o文件进行编译就可以

**二、实现效果**







输出结构为[name][tid:进程号][priority][burst]

三、实验讨论

1、运行./fcfs schedule.txt时出现了段错误，后来通过检查发现在读取文件时出错，指针的释放有问题，改正之后，发现用scanf（）函数读取的文档出现了错误，后来改成了fgets函数读取。

2、具体的函数的实现比较简单，但是最后出错比较多的还是全局变量的引用有问题。另外就是.h文件中如果直接实现函数，如果该头文件在两个.c文件中引用，在make时，会显示函数多次定义。