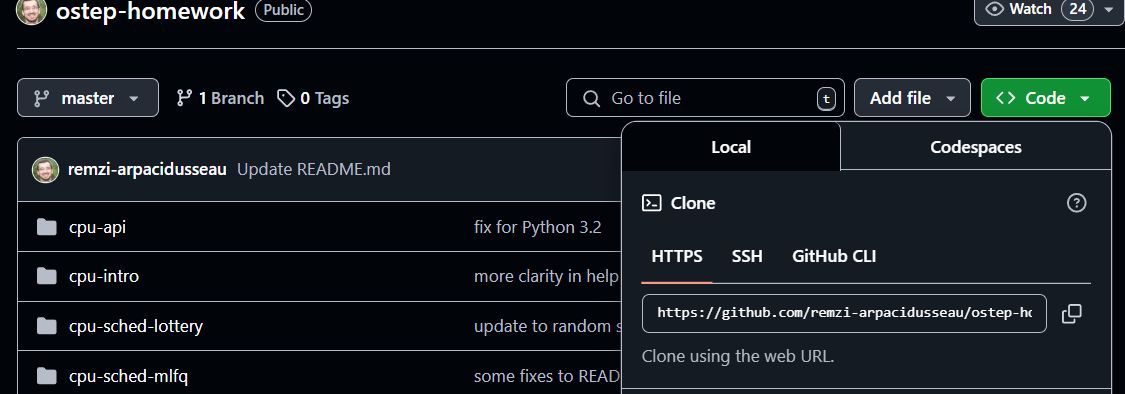
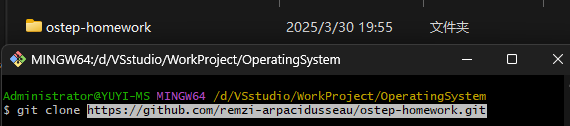
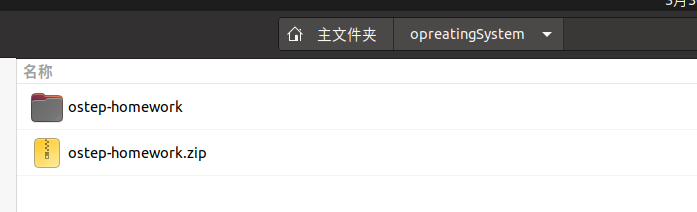
计科2302王任杰 202308010212

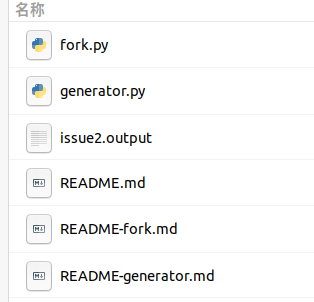
准备工作





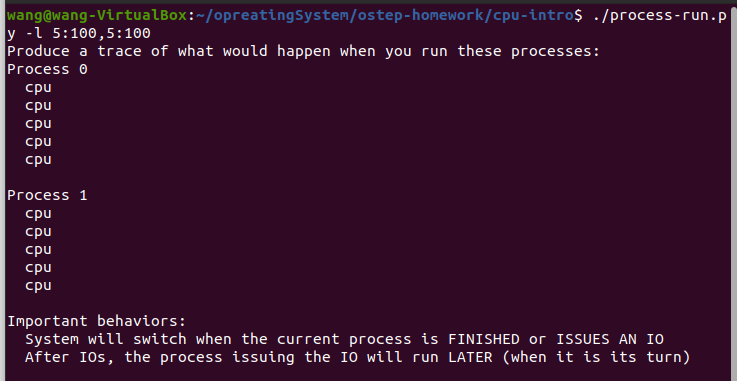
将github上项目通过git下载到本地，压缩后导入虚拟机中





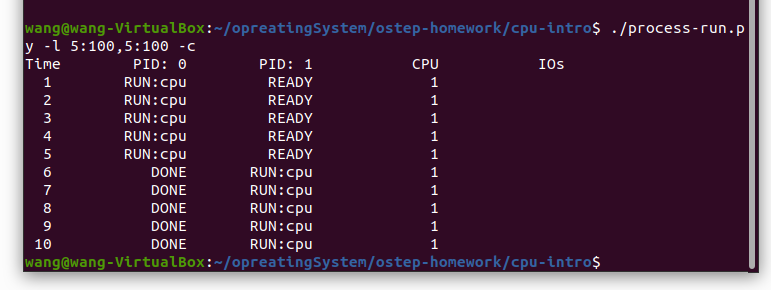
每一部分也有相应的说明，可以翻译后查看理解

4.1

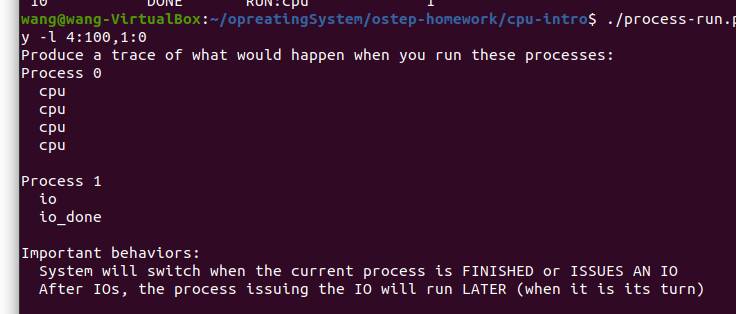


cpu利用率为100%，图中都只使用cpu

使用-c查看如下

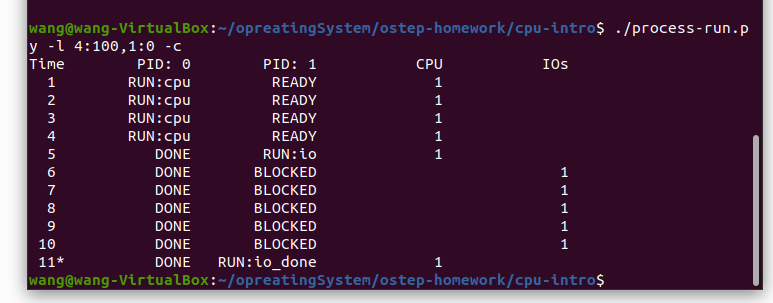


4.2

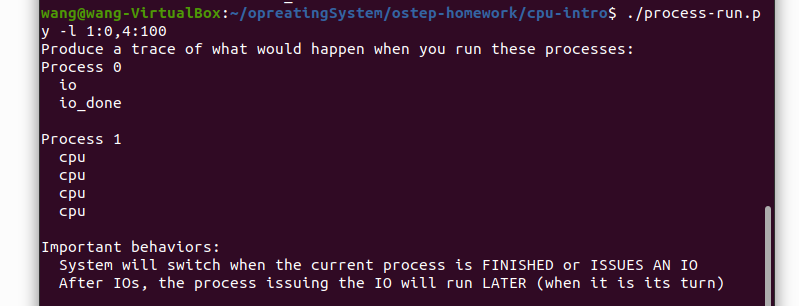


进程完成需要的时间与I/O等待完成需要的时间有关

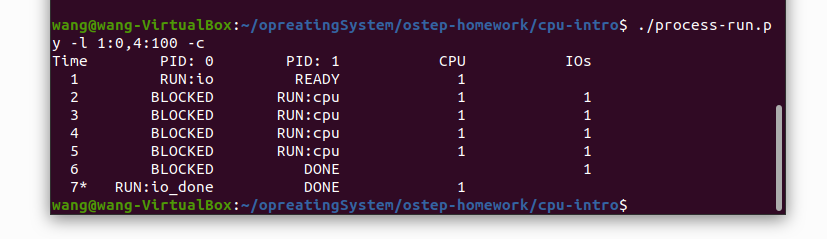
答案应该是11



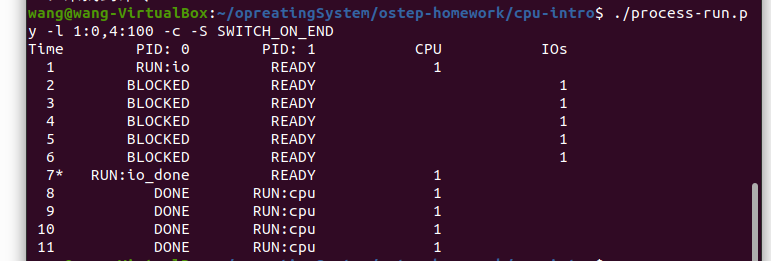
4.3



进程2补齐了进程1空闲的时间，合理地交换顺序十分重要，这样可以大大提高cpu利用率和效率，使用-c查看如下

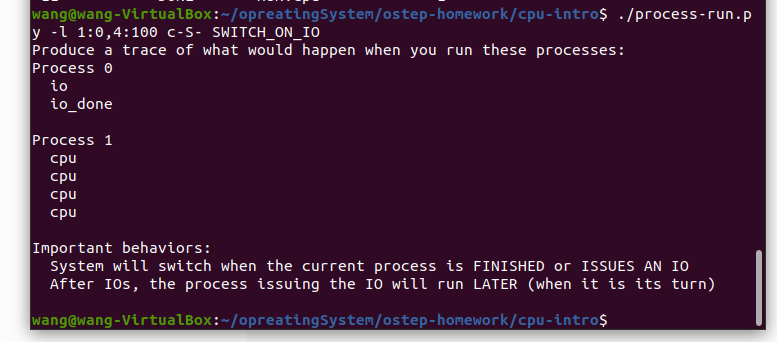


4.4

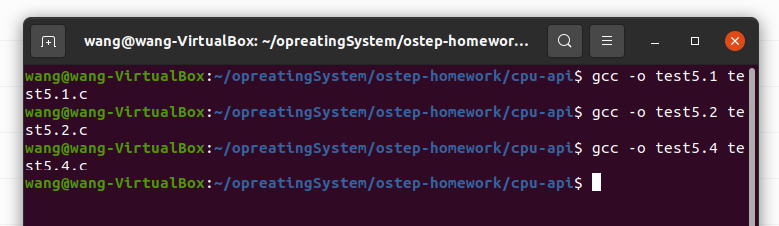


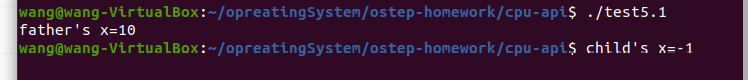
此时cpu利用率很低

4.5

  
系统在i/o操作时将切换其他进程，可以提高cpu利用率

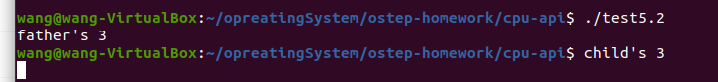
5.1





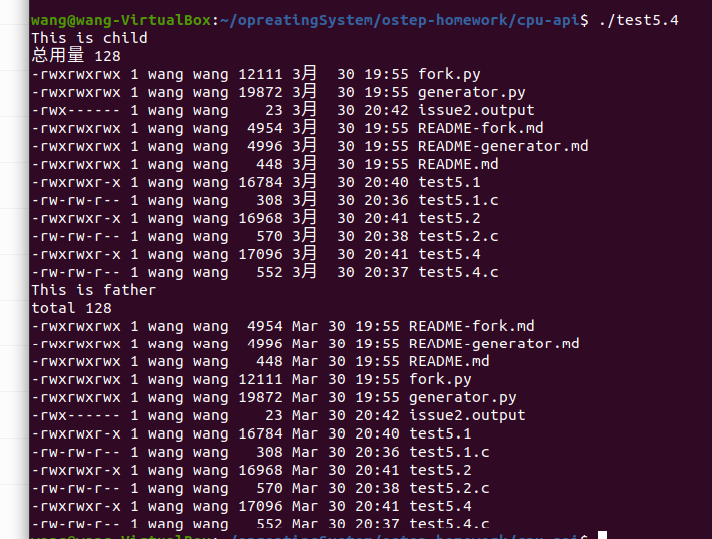
子进程和父进程都改变x的值时，变量互不影响，各自改变

5.2



都可以访问open（）返回的文件描述符

5.4

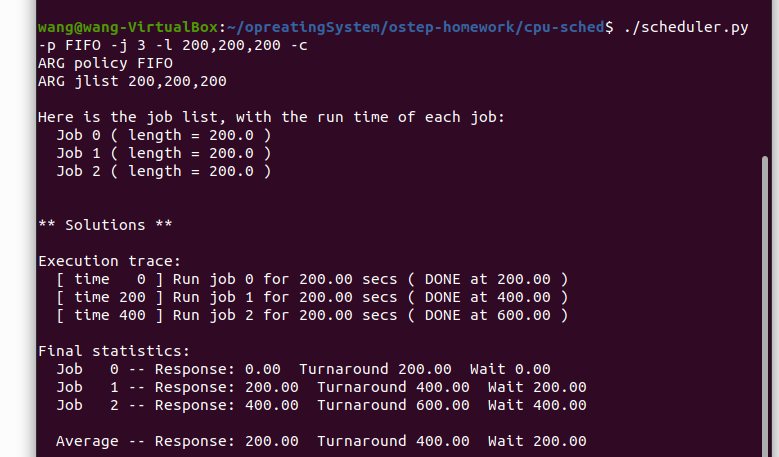


这些函数有这些特点

l : 使用参数列表  
p：使用文件名，并从PATH环境进行寻找可执行文件  
v：应先构造一个指向各参数的指针数组，然后将该数组的地址作为这些函数的参数。  
e：多了envp[]数组，使用新的环境变量代替调用进程的环境变量

7.1

Fifo

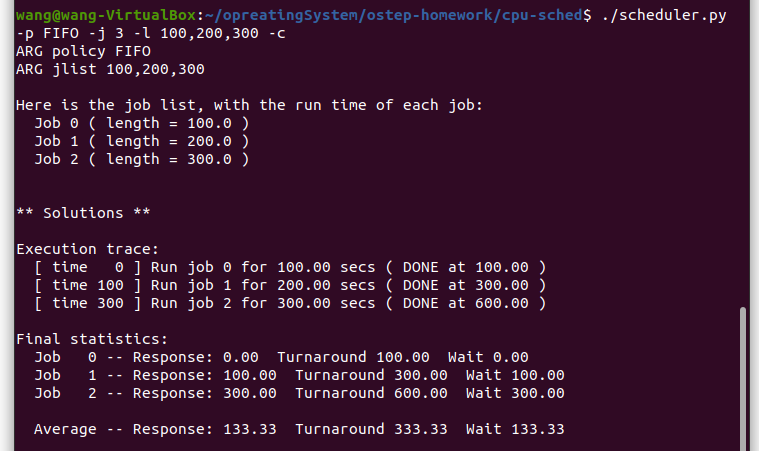


Sjf

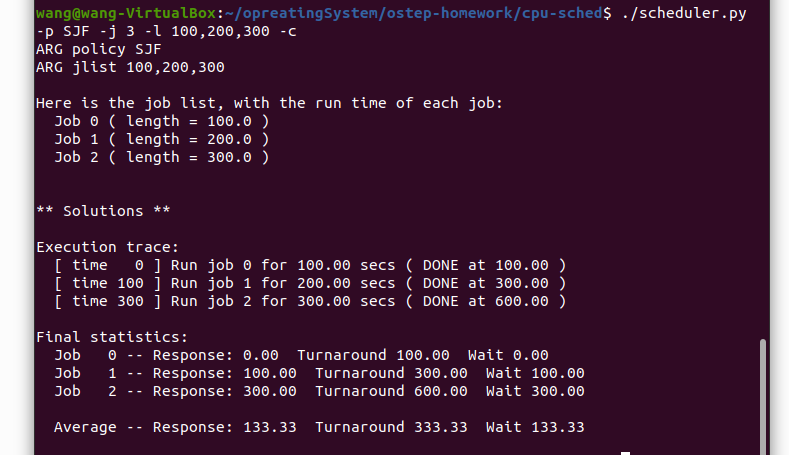


7.2

Fifo

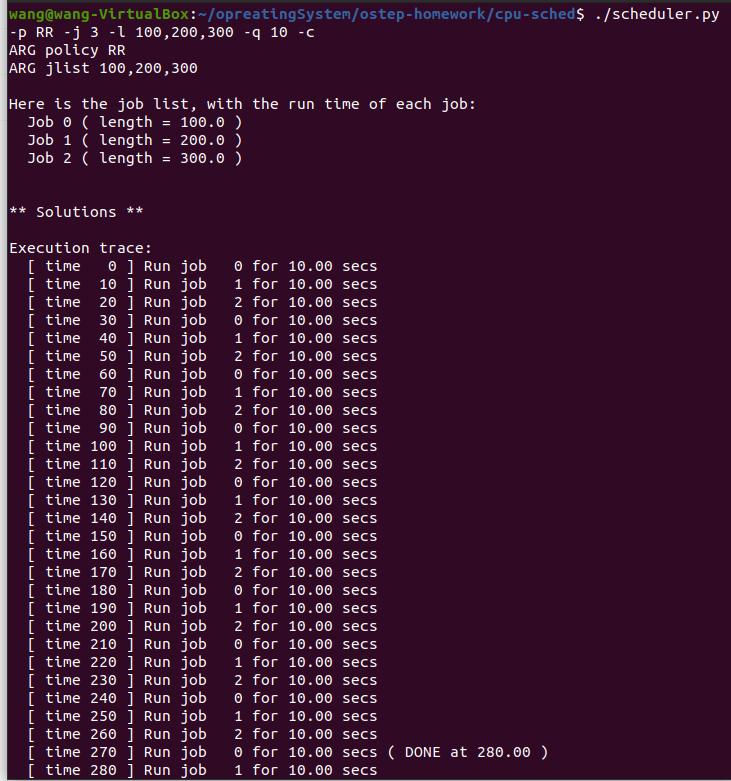


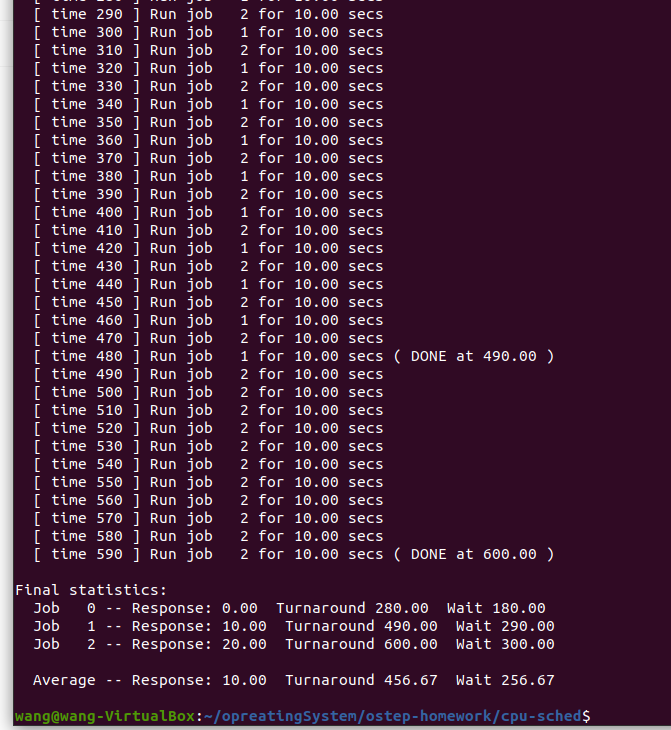
Sjf



7.3

使用RR调度





7.4

任务需要的cpu时间相同、每次都是最短作业先到

7.5

量子长度大于工作负载

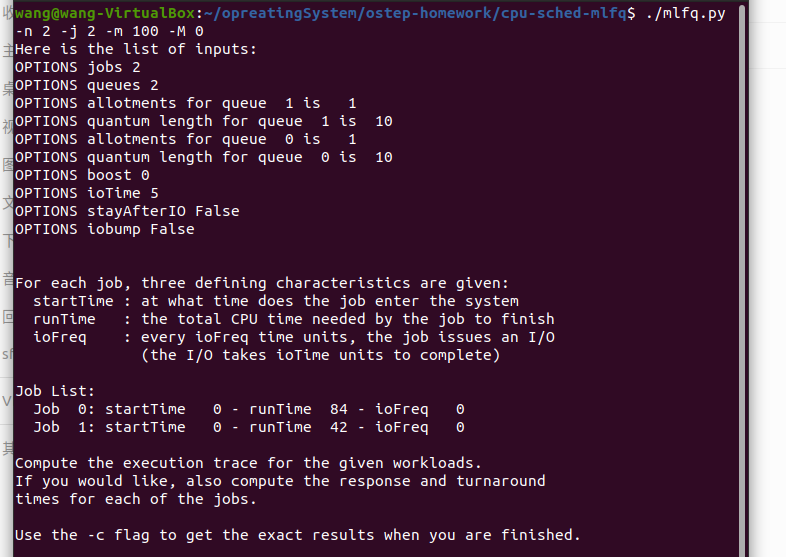
7.6

线性增加

7.7

RR响应时间会线性边长，最坏的响应时间就是量子长度=工时长度。

8.1



平均响应时间为5，平均周转时间为109

8.3

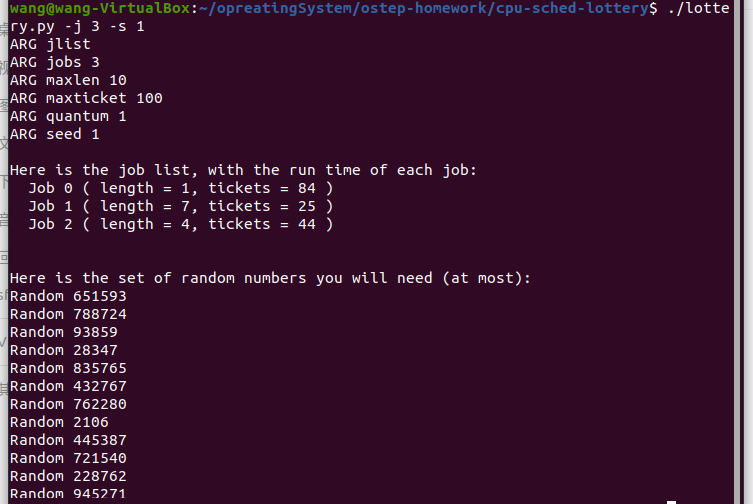
将mlfq调度的队列数设置为1，重新加入最高优先级队列的时间S意义不大，不需考虑。

8.5

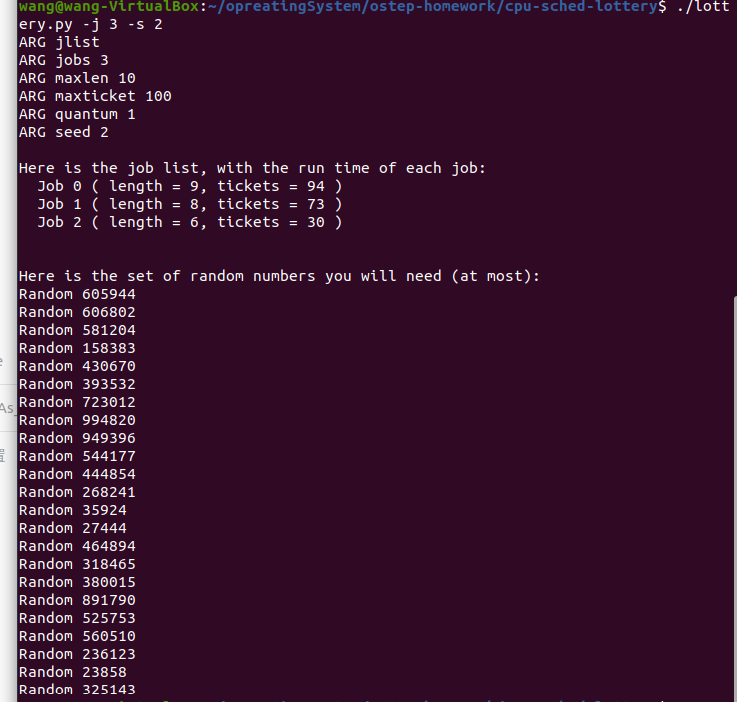
至少为200ms

9.1

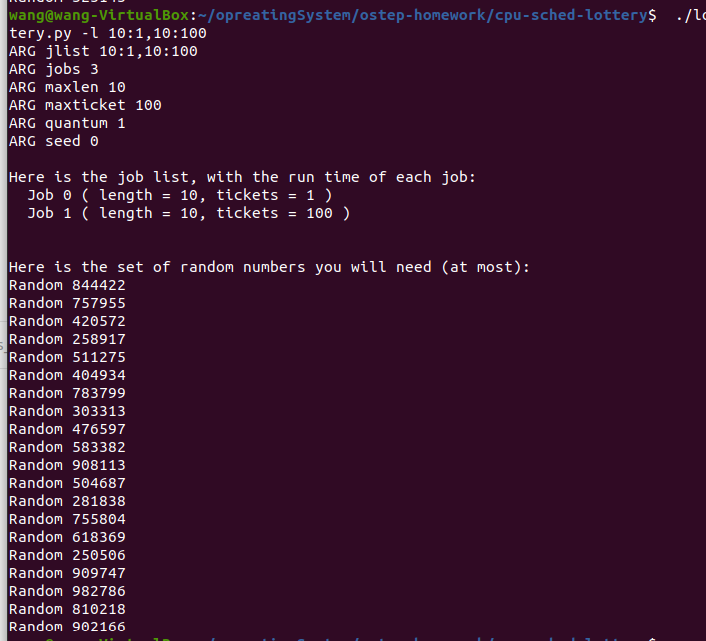
Speed为1



Speed为2



9.2



只有一张彩票被调度的可能性非常小，可能会饿死。

在工作1完成之前，工作0可能会运行。

这种行为可能会导致平均周转和响应时间变得很差

9.3

