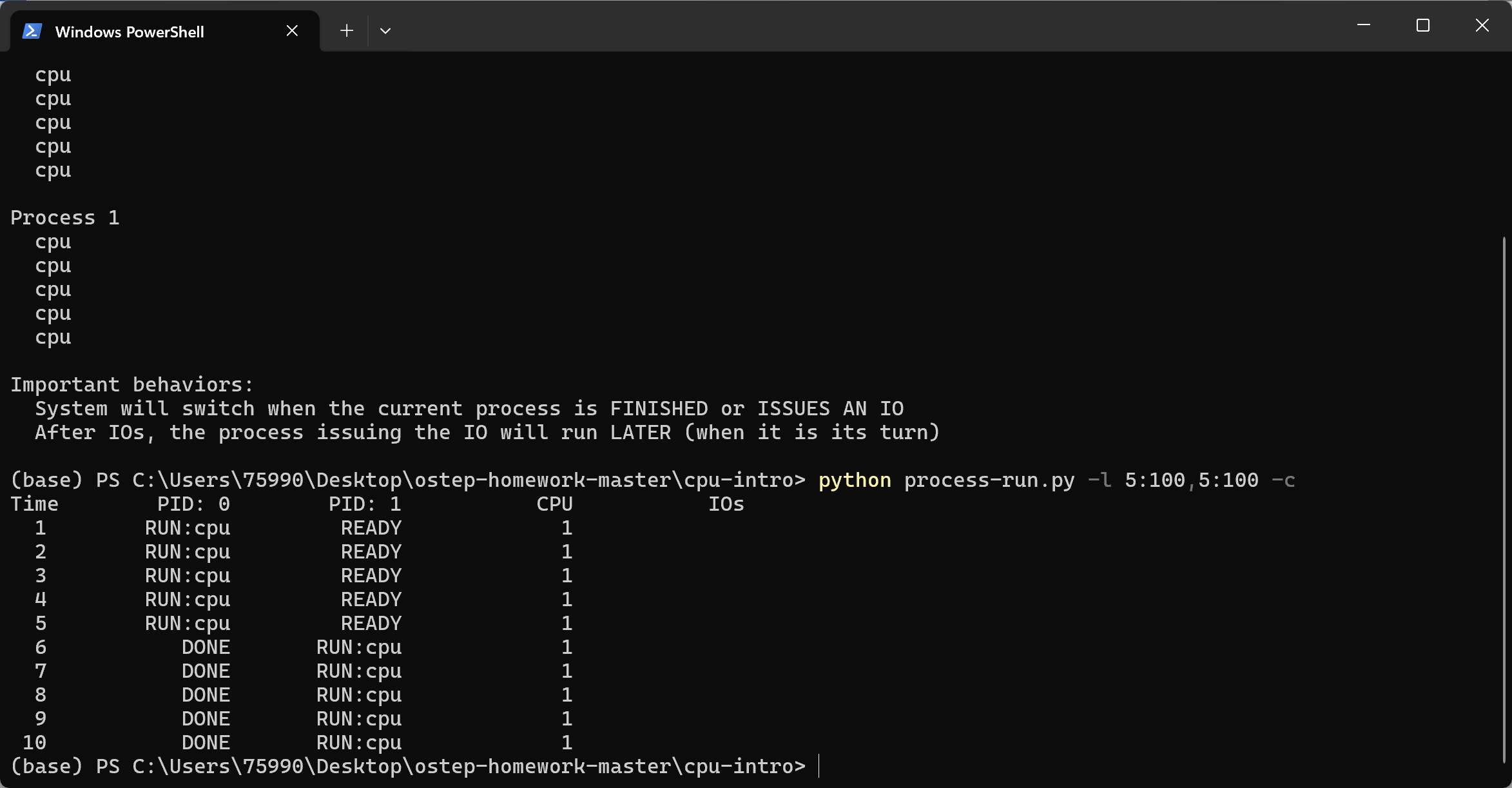
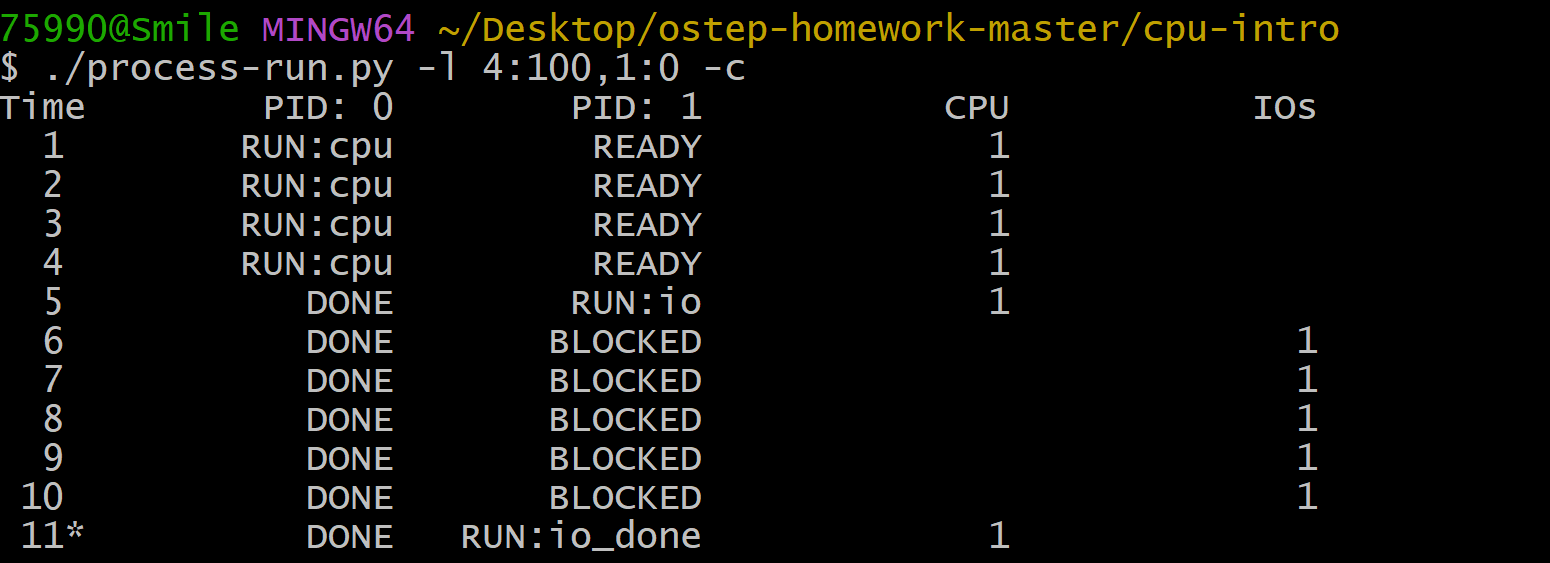
# Chp 4

1. CPU的利用率为100%，因为没有进程执行IO操作。

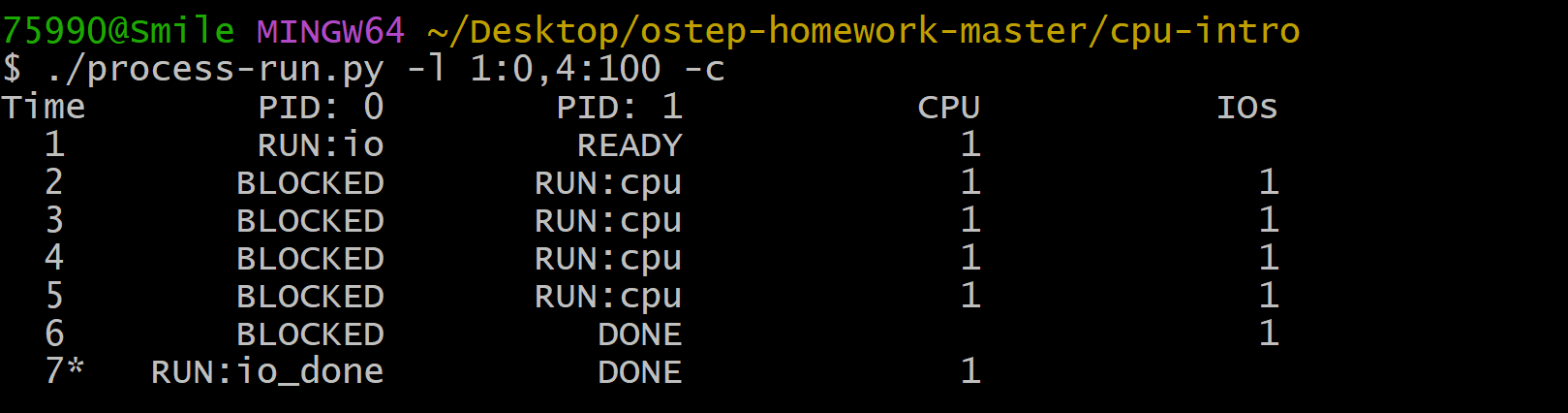


1. 执行4条执行需要4个时间单元，一次IO操作的发起和结束一共需要两个时间单元，IO操作需要等待5个时间单元。

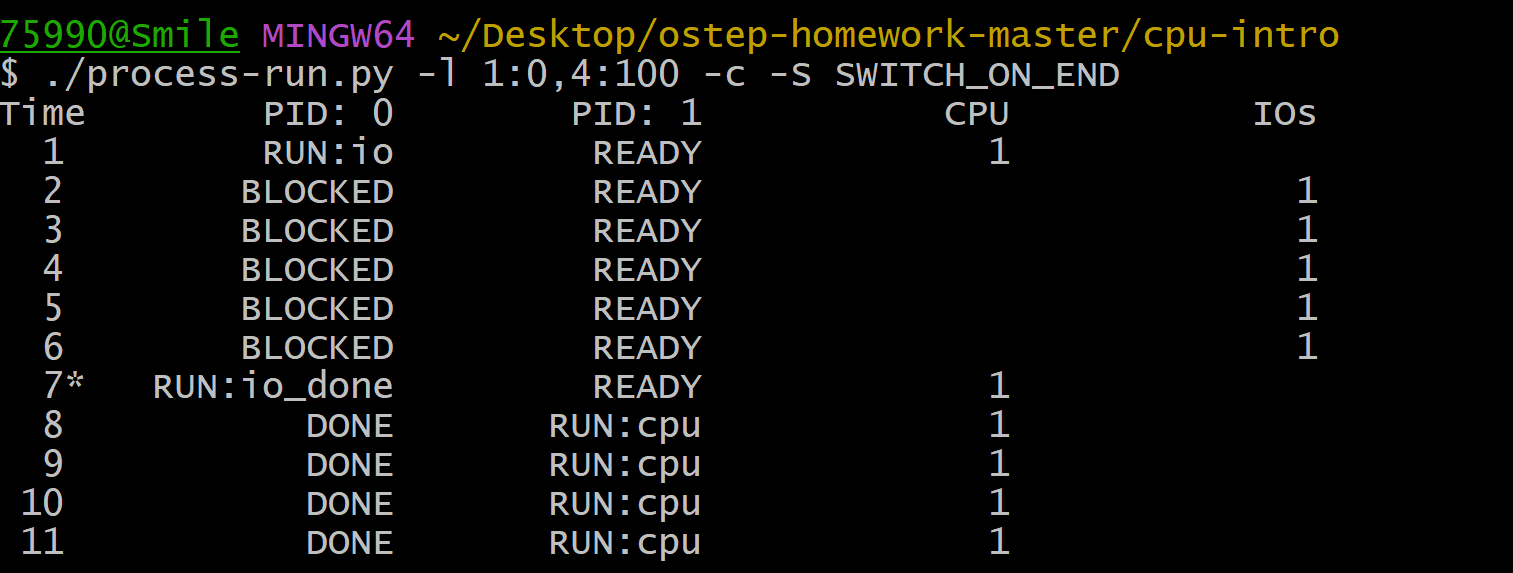
故一共需要11个时间单元的时间。



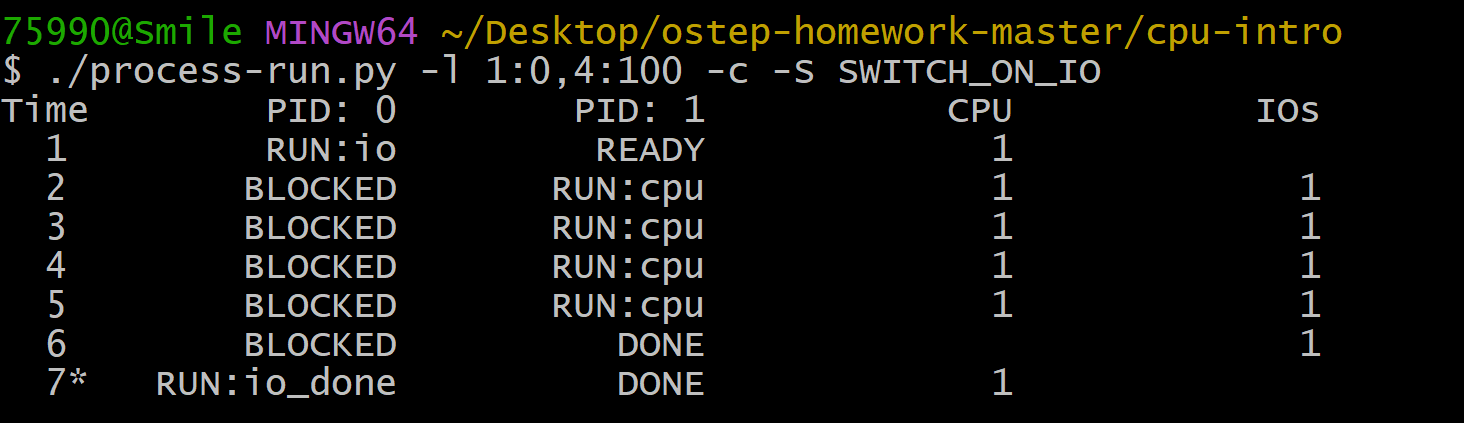
1. 交换顺序后，一共需要7个时间单元。因为进程0在等待IO时，进程1可以上CPU执行。



1. CPU花了11个时间单元才执行完。

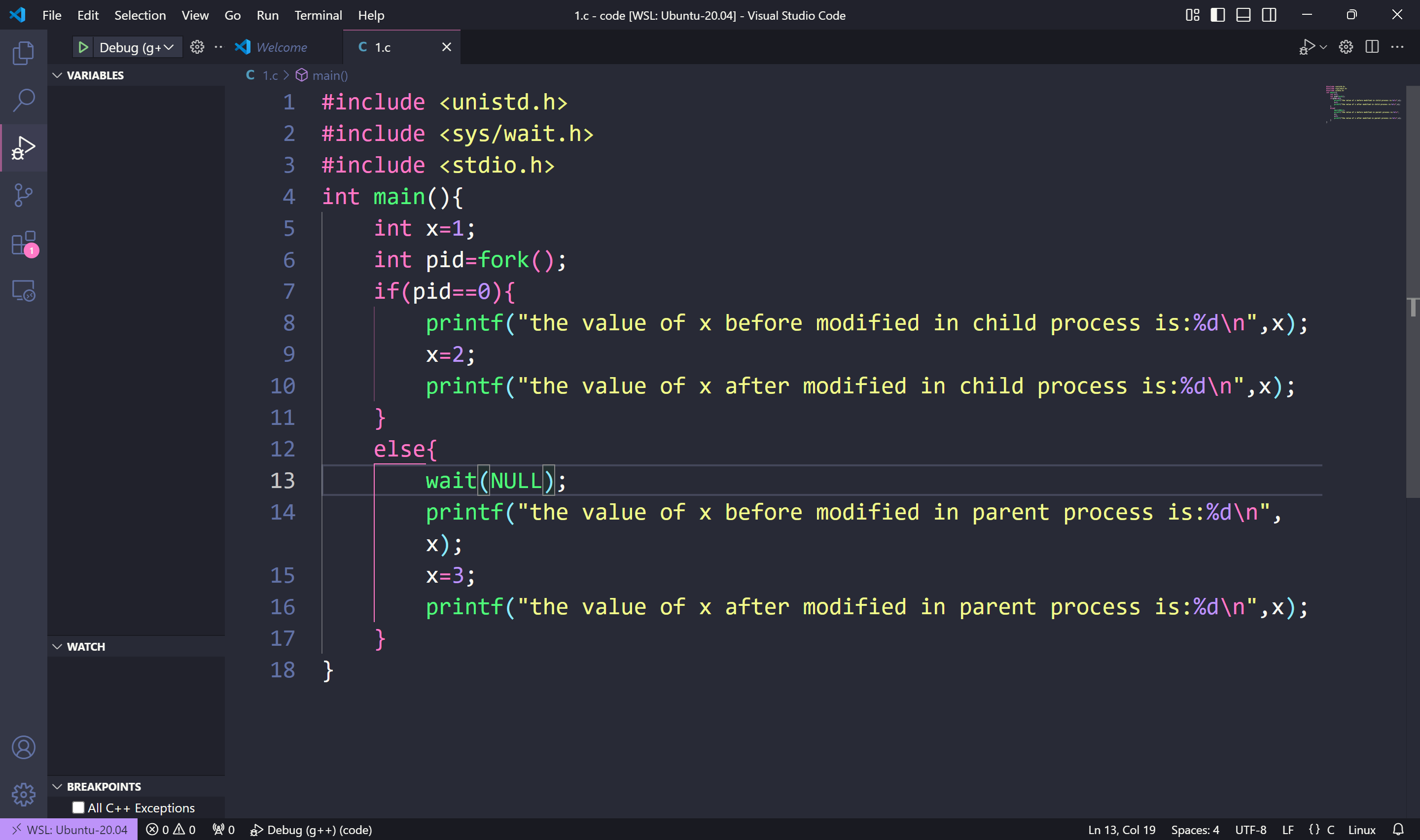


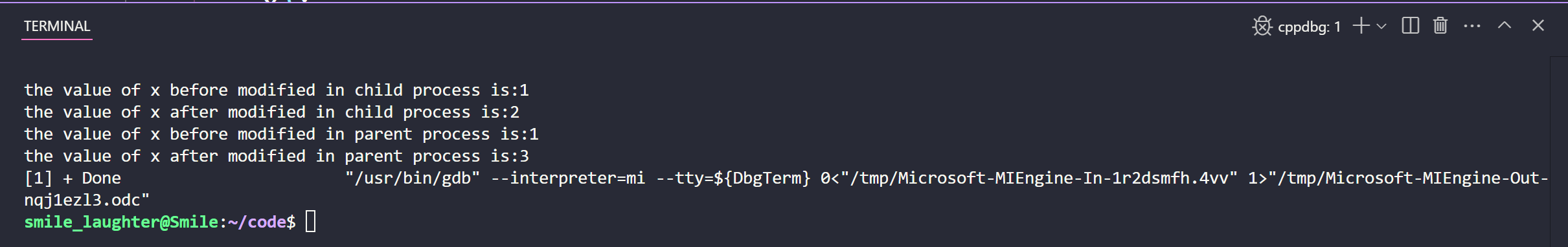
1. CPU花了7个时间单元才执行完。情况和第3问是一样的。



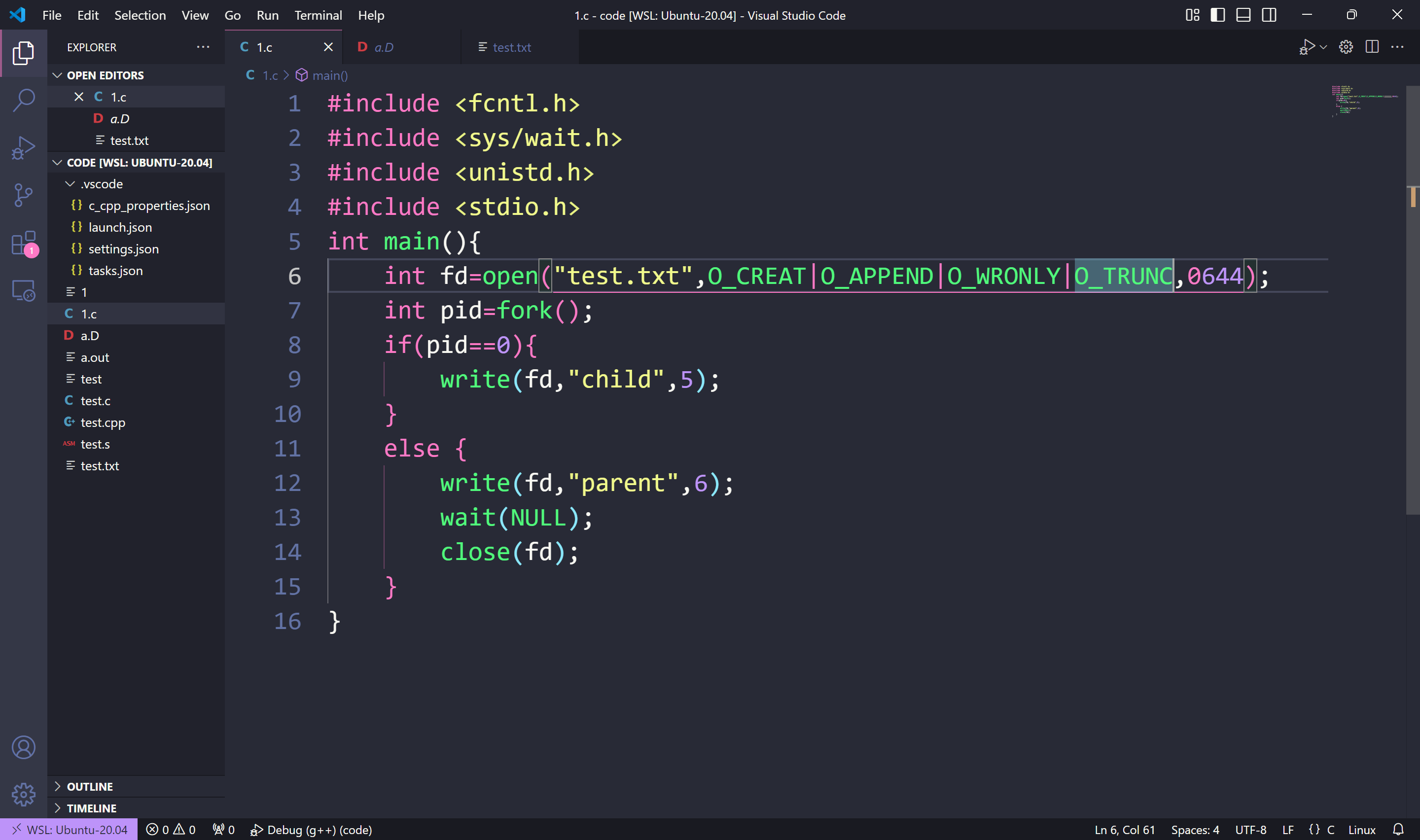
# Chp5

1.



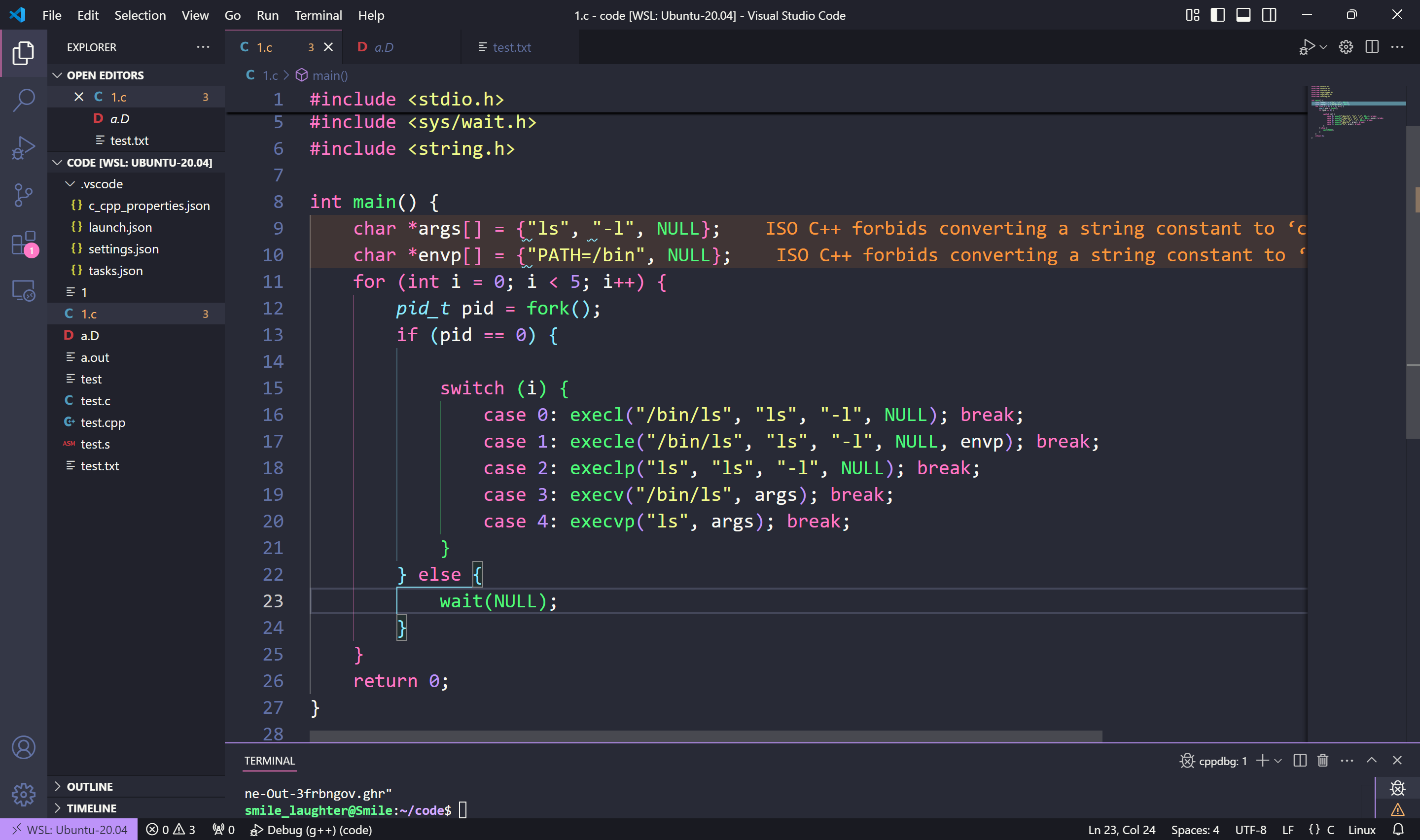


子进程中变量的值和主进程设置的值一样，子进程和父进程可以分别修改变量的值，而不会发生冲突。

2. 

子进程和父进程都可以访问主进程创建的文件描述符，同时写入时，test.txt中可能的结果有两种：

1. parentchild
2. childparent

4. 

原因：

1. 参数传递方式不同：可变参数列表与参数数组
2. 路径解析方式不同：提供绝对路径，利用环境变量解析路径
3. 是否可以支持自定义环境变量

# Chp 7

1.

|  |  |  |
| --- | --- | --- |
| SJF | 响应时间 | 周转时间 |
| 作业0 | 0 | 200 |
| 作业1 | 200 | 400 |
| 作业2 | 400 | 600 |
| 平均 | 200 | 400 |

|  |  |  |
| --- | --- | --- |
| FIFO | 响应时间 | 周转时间 |
| 作业0 | 0 | 200 |
| 作业1 | 200 | 400 |
| 作业2 | 400 | 600 |
| 平均 | 200 | 400 |

2.

|  |  |  |
| --- | --- | --- |
| SJF | 响应时间 | 周转时间 |
| 作业0 | 0 | 100 |
| 作业1 | 100 | 300 |
| 作业2 | 300 | 600 |
| 平均 | 400/3 | 1000/3 |

|  |  |  |
| --- | --- | --- |
| FIFO | 响应时间 | 周转时间 |
| 作业0 | 0 | 100 |
| 作业1 | 100 | 300 |
| 作业2 | 300 | 600 |
| 平均 | 400/3 | 1000/3 |

3.

|  |  |  |
| --- | --- | --- |
| RR | 响应时间 | 周转时间 |
| 作业0 | 0 | 598 |
| 作业1 | 1 | 599 |
| 作业2 | 2 | 600 |
| 平均 | 1 | 599 |

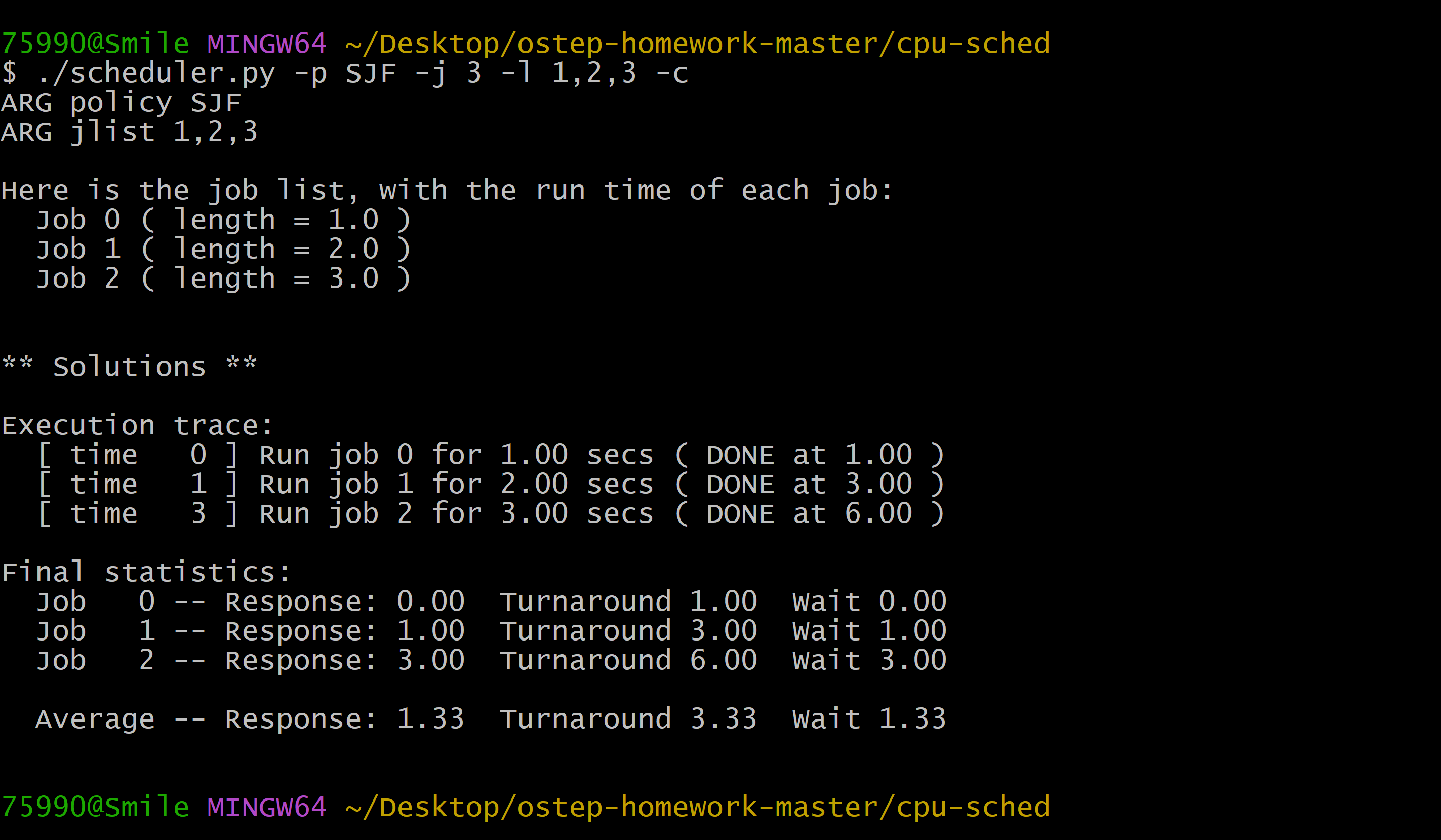
4.作业长度按作业到达顺序非严格递增

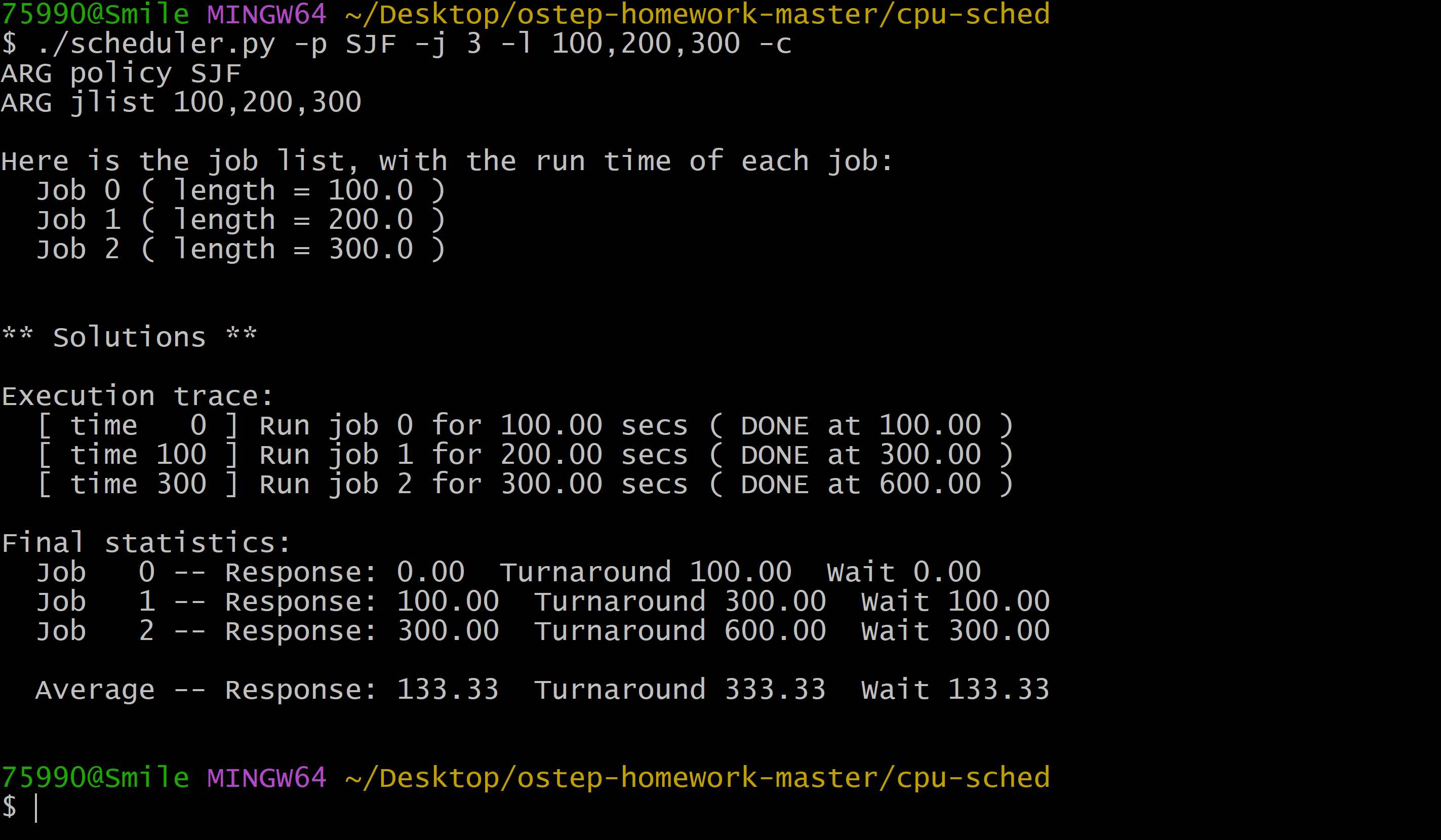
5.

（1）所有作业长度相同，并且 RR 的时间片足够大时

（2）每个工作的时长相同且等于时间切片时

1. 工作长度最短的任务响应时间不变，其它工作的响应时间变长。





1. 假设工作时间远大于时间片长度：

随着量子长度的增加，RR 的响应时间会变大

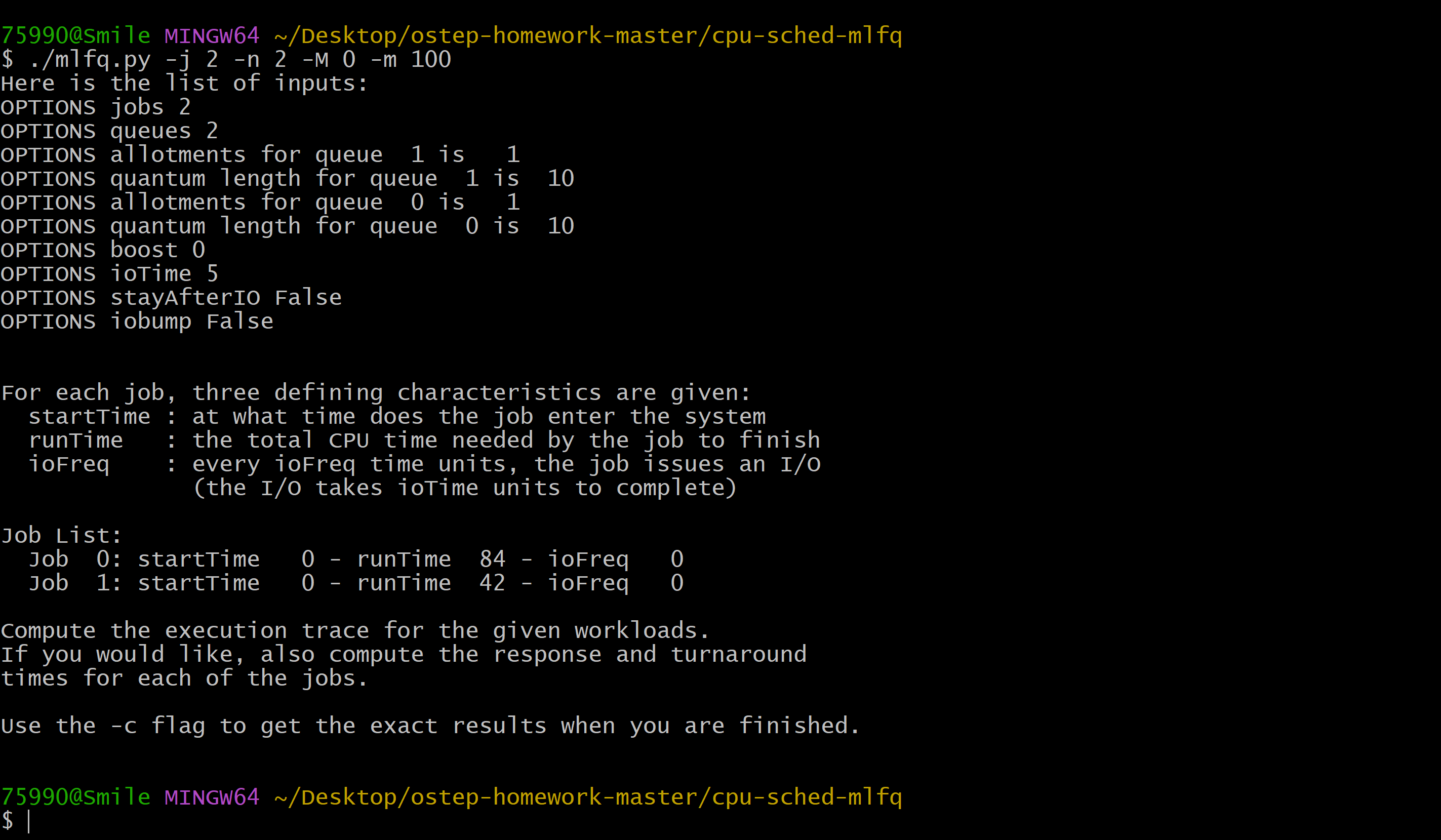
假设量子长度为t，则平均响应时间为

最坏情况的响应时间即是最后一个工作的响应时间，为(N-1)\*t

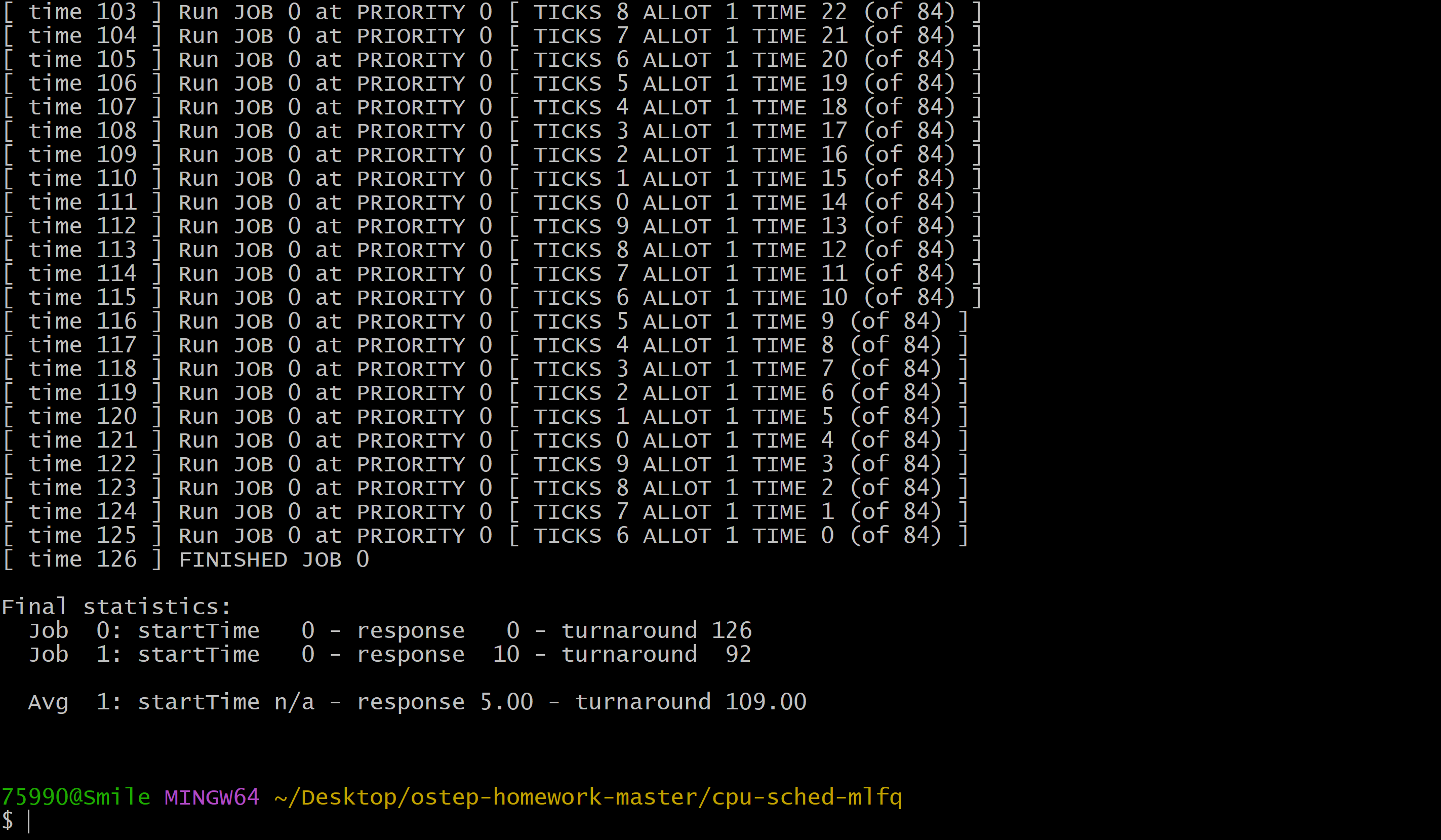
# Chp 8

1.

./mlfq.py -j 2 -n 2 -M 0 -m 100



|  |  |
| --- | --- |
| time 0-9 | Run JOB 0 at PRIORITY 1 |
| time 10-19 | Run JOB 1 at PRIORITY 1 |
| time 20-29 | Run JOB 0 at PRIORITY 0 |
| time 30-39 | Run JOB 1 at PRIORITY 0 |
| time 40-49 | Run JOB 0 at PRIORITY 0 |
| time 50-59 | Run JOB 1 at PRIORITY 0 |
| time 60-69 | Run JOB 0 at PRIORITY 0 |
| time 70-79 | Run JOB 1 at PRIORITY 0 |
| time 80-89 | Run JOB 0 at PRIORITY 0 |
| time 90-91 | Run JOB 1 at PRIORITY 0 |
| time 92-101 | Run JOB 0 at PRIORITY 0 |
| time 102-111 | Run JOB 0 at PRIORITY 0 |
| time 112-121 | Run JOB 0 at PRIORITY 0 |
| time 122-125 | Run JOB 0 at PRIORITY 0 |



3. 当工作在同一队列时，进行轮转调度工作，因此只需要将mlfq调度的队列数设置为1，所以答案为：./mlfq.py -n 1

5. 假设每隔T时间，系统使用-B标志将所有进程（包括这个被饥饿的）推回最高优先级队列。那么，这个进程每T时间可以运行10ms。那么它获得的CPU时间比例是10ms / T。根据题目要求，这个比例至少要是5%，也就是：

10ms / T ≥ 5% → T ≤ 10ms / 0.05 → T ≤ 200ms。

所以答案是每200ms将工作推回到最高优先级级别。

# Chp 9

1.

Seed=1:



Seed=2:



Seed=3:



2.



这里工作0与工作1的彩票数量差距过大了，导致在工作1完成之前工作0占用CPU几乎是不可能的。从模拟结果上看，在工作1完成之前，工作0没有运行。

在这种情况下，持有份额小的工作响应时间与周转时间非常长，且基本上占用不了CPU。极有可能会“饿死”。

3. 

提前完成的工作所用时间平均比后完成的快了8左右，对比工作长度只能保证基本接近公平，假设以提前完成时间比工作长度为指标，不公平度约为7.9%。