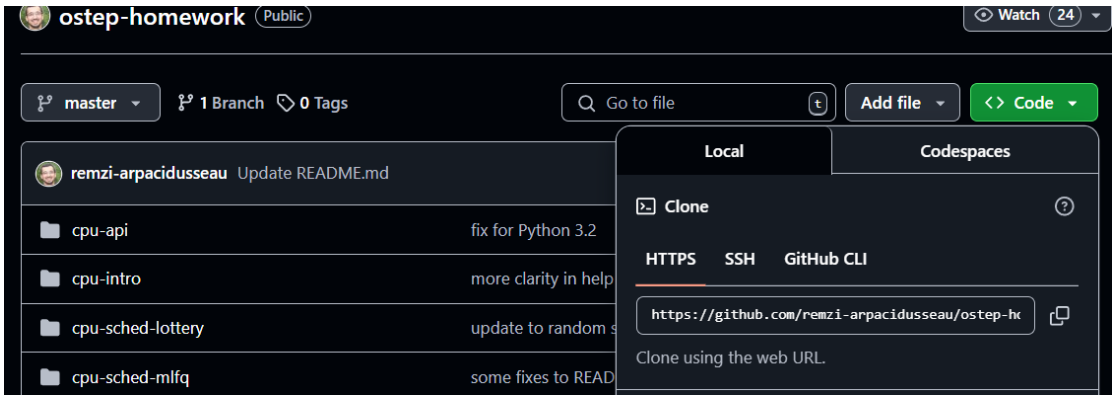
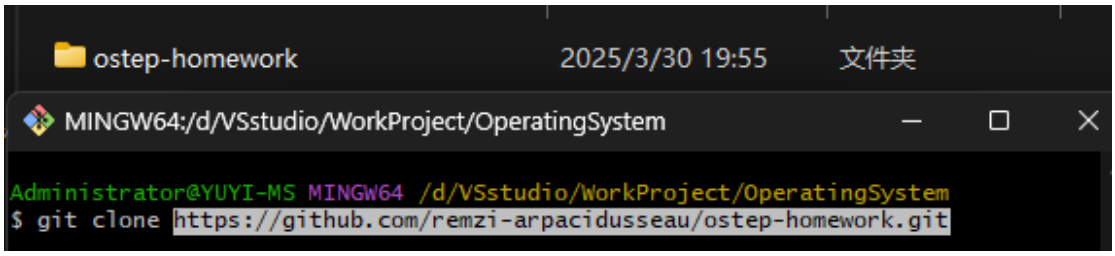
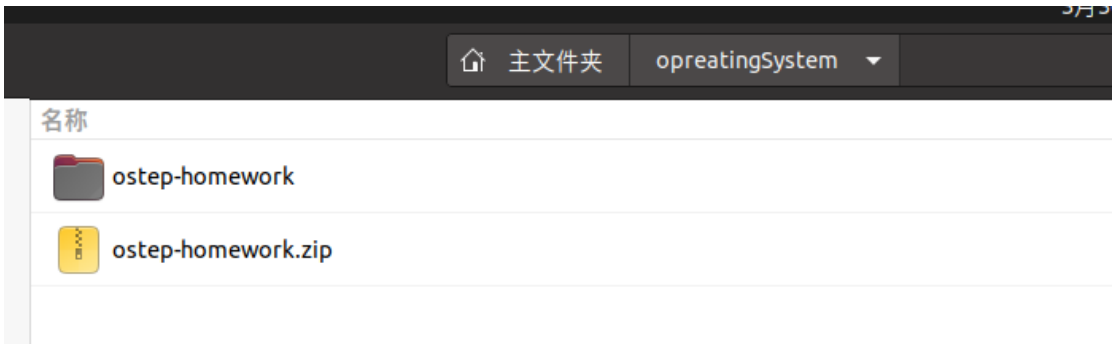
计科 2302 王任杰 202308010212

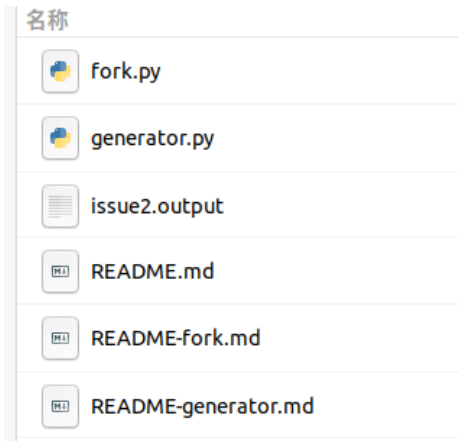
准备工作





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

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 5:100,5:100
Produce a trace of what would happen when you run these processes:
Process 0
  cpu
  cpu
  cpu
  cpu
  cpu

Process 1
  cpu
  cpu
  cpu
  cpu
  cpu

Important behaviors:
  System will switch when the current process is FINISHED or ISSUES AN IO
  After IOs, the process issuing the IO will run LATER (when it is its turn)
```

cpu 利用率为 100%，图中都只使用 cpu
使用-c 查看如下

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 5:100,5:100 -c
Time        PID: 0        PID: 1           CPU           IOs
  1        RUN:cpu         READY             1
  2        RUN:cpu         READY             1
  3        RUN:cpu         READY             1
  4        RUN:cpu         READY             1
  5        RUN:cpu         READY             1
  6           DONE       RUN:cpu             1
  7           DONE       RUN:cpu             1
  8           DONE       RUN:cpu             1
  9           DONE       RUN:cpu             1
 10           DONE       RUN:cpu             1
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$
```

4.2

```
 10           DONE       RUN:cpu             1
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 4:100,1:0
Produce a trace of what would happen when you run these processes:
Process 0
  cpu
  cpu
  cpu
  cpu

Process 1
  io
  io_done

Important behaviors:
  System will switch when the current process is FINISHED or ISSUES AN IO
  After IOs, the process issuing the IO will run LATER (when it is its turn)
```

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

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 4:100,1:0 -c
Time        PID: 0        PID: 1           CPU           IOs
  1        RUN:cpu         READY             1
  2        RUN:cpu         READY             1
  3        RUN:cpu         READY             1
  4        RUN:cpu         READY             1
  5           DONE        RUN:io             1
  6           DONE       BLOCKED                           1
  7           DONE       BLOCKED                           1
  8           DONE       BLOCKED                           1
  9           DONE       BLOCKED                           1
 10           DONE       BLOCKED                           1
 11*          DONE   RUN:io_done             1
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$
```

4.3

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 1:0,4:100
Produce a trace of what would happen when you run these processes:
Process 0
  io
  io_done

Process 1
  cpu
  cpu
  cpu
  cpu

Important behaviors:
  System will switch when the current process is FINISHED or ISSUES AN IO
  After IOs, the process issuing the IO will run LATER (when it is its turn)
```

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

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 1:0,4:100 -c
Time        PID: 0        PID: 1           CPU           IOs
  1         RUN:io         READY             1
  2        BLOCKED       RUN:cpu             1             1
  3        BLOCKED       RUN:cpu             1             1
  4        BLOCKED       RUN:cpu             1             1
  5        BLOCKED       RUN:cpu             1             1
  6        BLOCKED          DONE                           1
  7*   RUN:io_done          DONE             1
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$
```

4.4

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 1:0,4:100 -c -S SWITCH_ON_END
Time        PID: 0        PID: 1           CPU           IOs
  1         RUN:io         READY             1
  2        BLOCKED         READY                           1
  3        BLOCKED         READY                           1
  4        BLOCKED         READY                           1
  5        BLOCKED         READY                           1
  6        BLOCKED         READY                           1
  7*   RUN:io_done         READY             1
  8           DONE       RUN:cpu             1
  9           DONE       RUN:cpu             1
 10           DONE       RUN:cpu             1
 11           DONE       RUN:cpu             1
```

此时 cpu 利用率很低

4.5

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$ ./process-run.p
y -l 1:0,4:100 c-S- SWITCH_ON_IO
Produce a trace of what would happen when you run these processes:
Process 0
  io
  io_done

Process 1
  cpu
  cpu
  cpu
  cpu

Important behaviors:
  System will switch when the current process is FINISHED or ISSUES AN IO
  After IOs, the process issuing the IO will run LATER (when it is its turn)

wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-intro$
```

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

```
wang@wang-VirtualBox: ~/opreatingSystem/ostep-homewor...
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ gcc -o test5.1 te
st5.1.c
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ gcc -o test5.2 te
st5.2.c
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ gcc -o test5.4 te
st5.4.c
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$
```

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ ./test5.1
father's x=10
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ child's x=-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

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ ./test5.2
father's 3
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ child's 3
```

都可以访问 open（）返回的文件描述符

5.4

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-api$ ./test5.4
This is child
总用量 128
-rwxrwxrwx 1 wang wang 12111 3月  30 19:55 fork.py
-rwxrwxrwx 1 wang wang 19872 3月  30 19:55 generator.py
-rwx------ 1 wang wang    23 3月  30 20:42 issue2.output
-rwxrwxrwx 1 wang wang  4954 3月  30 19:55 README-fork.md
-rwxrwxrwx 1 wang wang  4996 3月  30 19:55 README-generator.md
-rwxrwxrwx 1 wang wang   448 3月  30 19:55 README.md
-rwxrwxr-x 1 wang wang 16784 3月  30 20:40 test5.1
-rw-rw-r-- 1 wang wang   308 3月  30 20:36 test5.1.c
-rwxrwxr-x 1 wang wang 16968 3月  30 20:41 test5.2
-rw-rw-r-- 1 wang wang   570 3月  30 20:38 test5.2.c
-rwxrwxr-x 1 wang wang 17096 3月  30 20:41 test5.4
-rw-rw-r-- 1 wang wang   552 3月  30 20:37 test5.4.c
This is father
total 128
-rwxrwxrwx 1 wang wang  4954 Mar 30 19:55 README-fork.md
-rwxrwxrwx 1 wang wang  4996 Mar 30 19:55 README-generator.md
-rwxrwxrwx 1 wang wang   448 Mar 30 19:55 README.md
-rwxrwxrwx 1 wang wang 12111 Mar 30 19:55 fork.py
-rwxrwxrwx 1 wang wang 19872 Mar 30 19:55 generator.py
-rwx------ 1 wang wang    23 Mar 30 20:42 issue2.output
-rwxrwxr-x 1 wang wang 16784 Mar 30 20:40 test5.1
-rw-rw-r-- 1 wang wang   308 Mar 30 20:36 test5.1.c
-rwxrwxr-x 1 wang wang 16968 Mar 30 20:41 test5.2
-rw-rw-r-- 1 wang wang   570 Mar 30 20:38 test5.2.c
-rwxrwxr-x 1 wang wang 17096 Mar 30 20:41 test5.4
-rw-rw-r-- 1 wang wang   552 Mar 30 20:37 test5.4.c
```

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

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched$ ./scheduler.py
-p FIFO -j 3 -l 200,200,200 -c
ARG policy FIFO
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
  Job 0 ( length = 200.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 200.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
  Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

  Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

Sjf

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched$ ./scheduler.py
-p SJF -j 3 -l 200,200,200 -c
ARG policy SJF
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
  Job 0 ( length = 200.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 200.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
  [ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
  [ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
  Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
  Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

  Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

7.2

Fifo

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched$ ./scheduler.py
-p FIFO -j 3 -l 100,200,300 -c
ARG policy FIFO
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

Sjf

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched$ ./scheduler.py
-p SJF -j 3 -l 100,200,300 -c
ARG policy SJF
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job 0 for 100.00 secs ( DONE at 100.00 )
  [ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
  [ time 300 ] Run job 2 for 300.00 secs ( DONE at 600.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 100.00  Wait 0.00
  Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
  Job   2 -- Response: 300.00  Turnaround 600.00  Wait 300.00

  Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

7.3

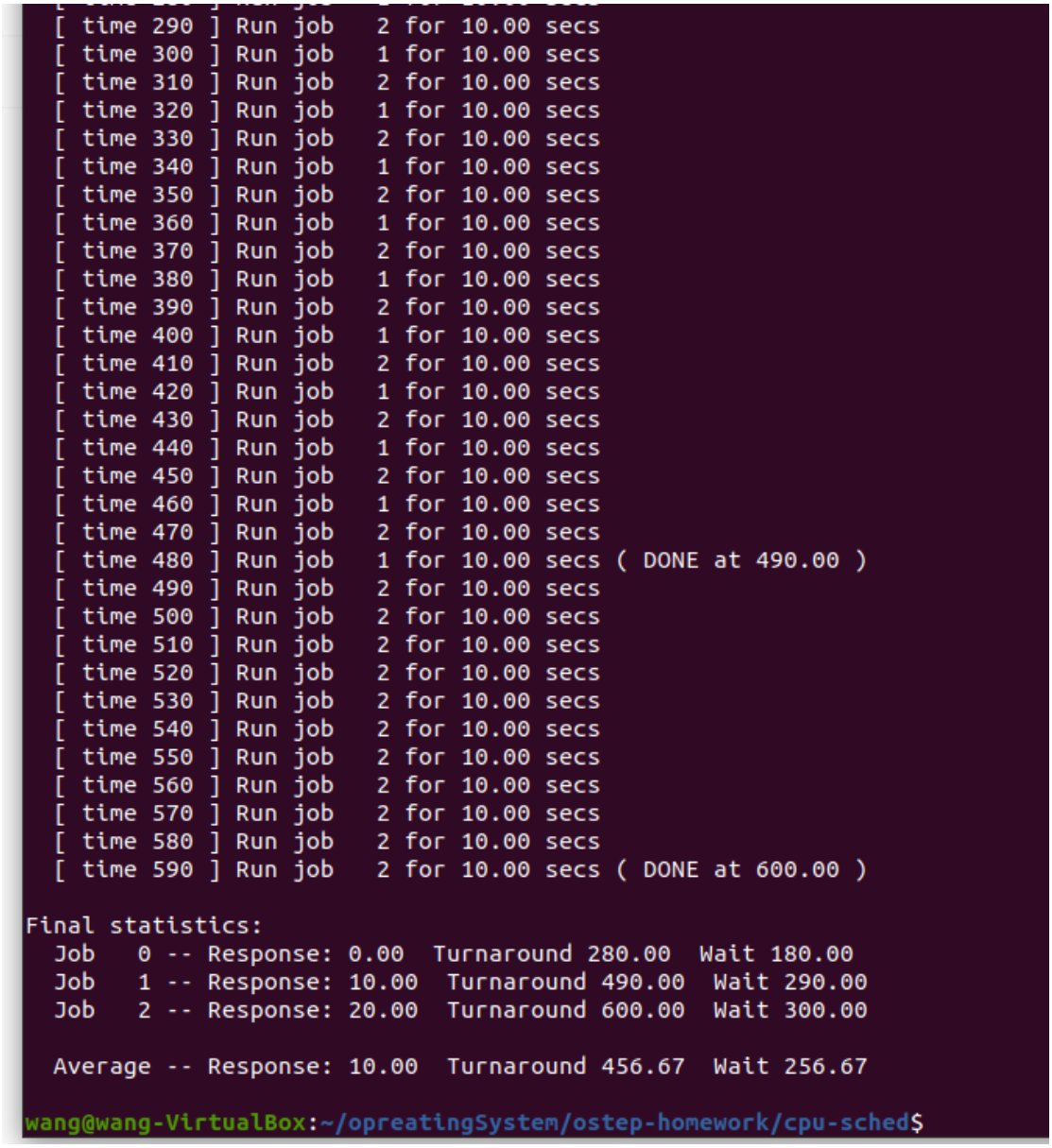
使用 RR 调度

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched$ ./scheduler.py
-p RR -j 3 -l 100,200,300 -q 10 -c
ARG policy RR
ARG jlist 100,200,300

Here is the job list, with the run time of each job:
  Job 0 ( length = 100.0 )
  Job 1 ( length = 200.0 )
  Job 2 ( length = 300.0 )


** Solutions **

Execution trace:
  [ time   0 ] Run job   0 for 10.00 secs
  [ time  10 ] Run job   1 for 10.00 secs
  [ time  20 ] Run job   2 for 10.00 secs
  [ time  30 ] Run job   0 for 10.00 secs
  [ time  40 ] Run job   1 for 10.00 secs
  [ time  50 ] Run job   2 for 10.00 secs
  [ time  60 ] Run job   0 for 10.00 secs
  [ time  70 ] Run job   1 for 10.00 secs
  [ time  80 ] Run job   2 for 10.00 secs
  [ time  90 ] Run job   0 for 10.00 secs
  [ time 100 ] Run job   1 for 10.00 secs
  [ time 110 ] Run job   2 for 10.00 secs
  [ time 120 ] Run job   0 for 10.00 secs
  [ time 130 ] Run job   1 for 10.00 secs
  [ time 140 ] Run job   2 for 10.00 secs
  [ time 150 ] Run job   0 for 10.00 secs
  [ time 160 ] Run job   1 for 10.00 secs
  [ time 170 ] Run job   2 for 10.00 secs
  [ time 180 ] Run job   0 for 10.00 secs
  [ time 190 ] Run job   1 for 10.00 secs
  [ time 200 ] Run job   2 for 10.00 secs
  [ time 210 ] Run job   0 for 10.00 secs
  [ time 220 ] Run job   1 for 10.00 secs
  [ time 230 ] Run job   2 for 10.00 secs
  [ time 240 ] Run job   0 for 10.00 secs
  [ time 250 ] Run job   1 for 10.00 secs
  [ time 260 ] Run job   2 for 10.00 secs
  [ time 270 ] Run job   0 for 10.00 secs ( DONE at 280.00 )
  [ time 280 ] Run job   1 for 10.00 secs
```



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

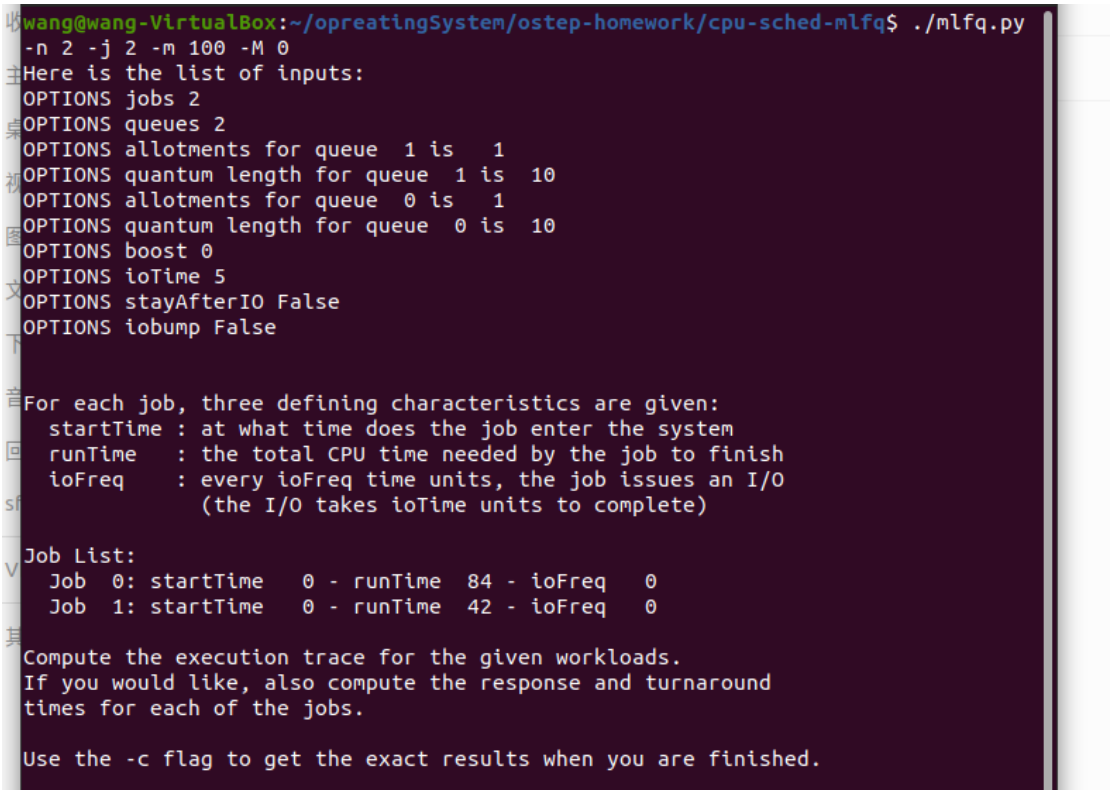
7.6

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

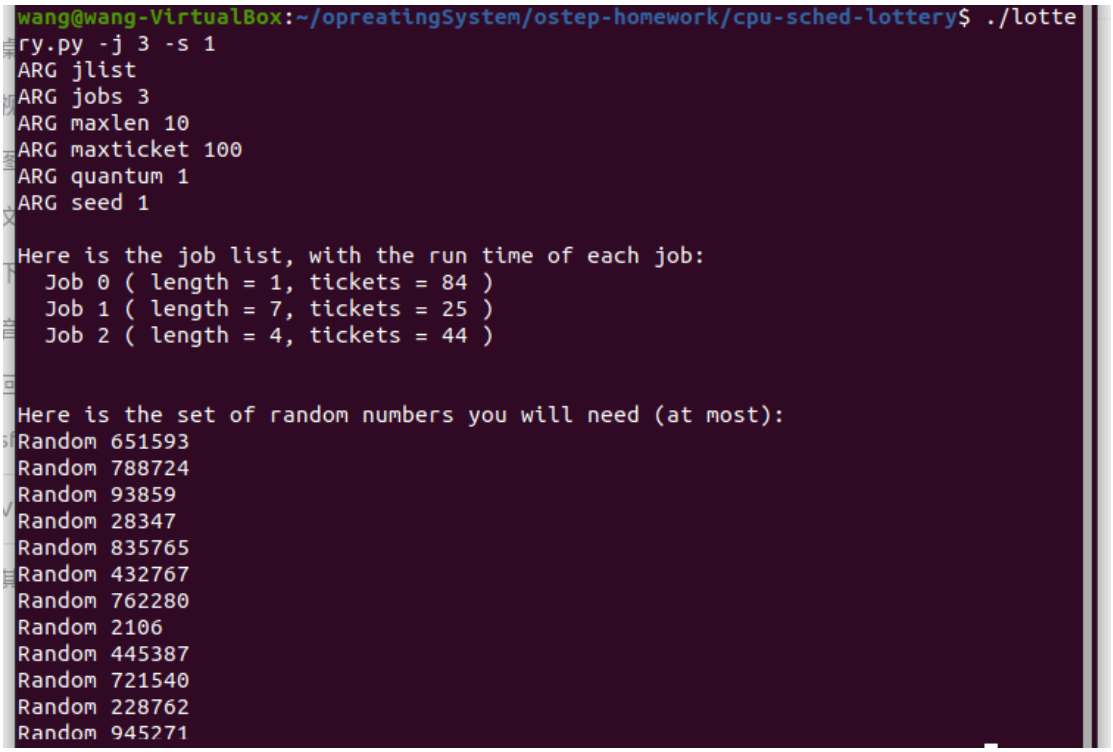
将 mlfq 调度的队列数设置为 1，重新加入最高优先级队列的时间 S 意义不大，不需考虑。

8.5

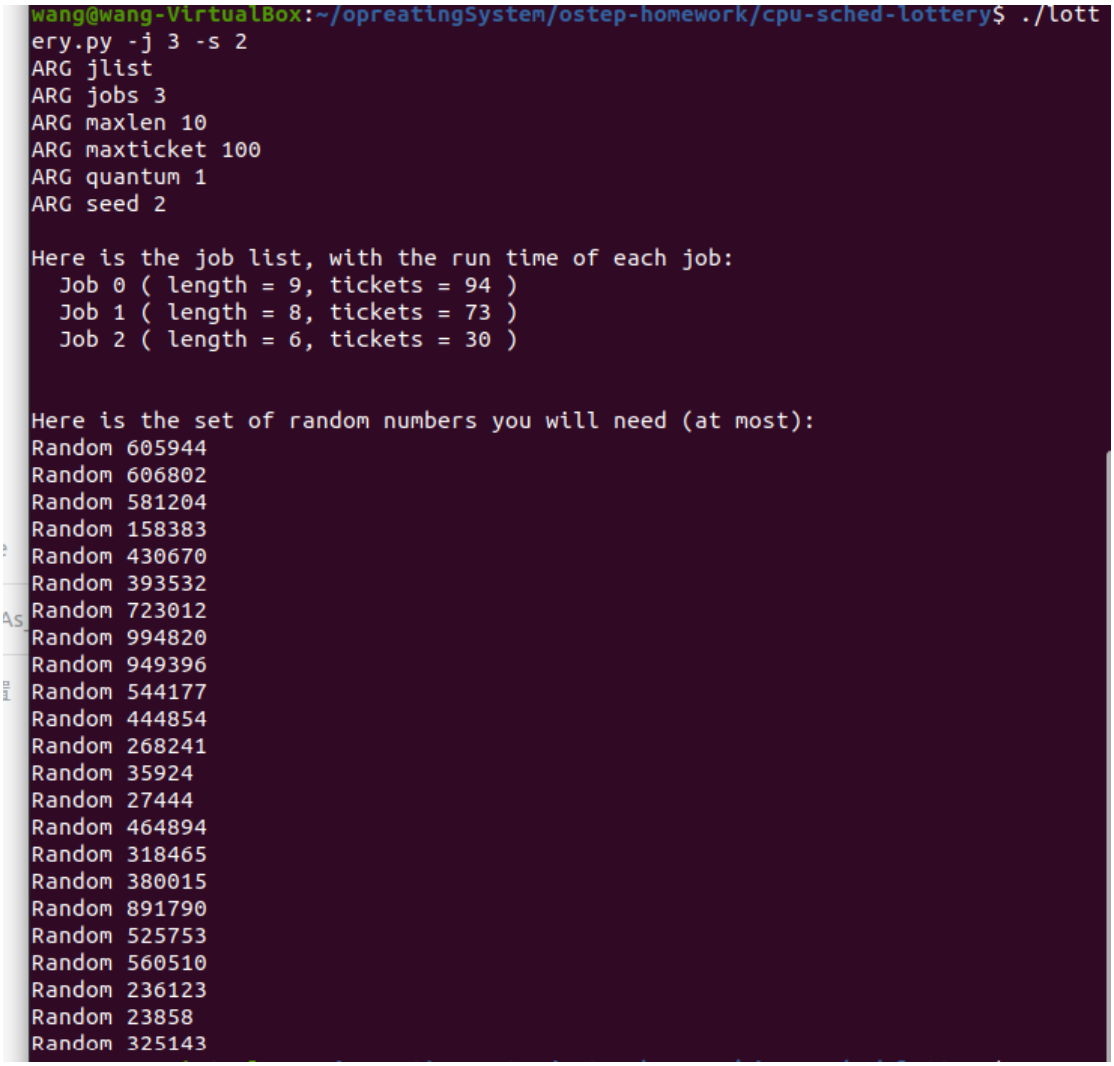
至少为 200ms

9.1

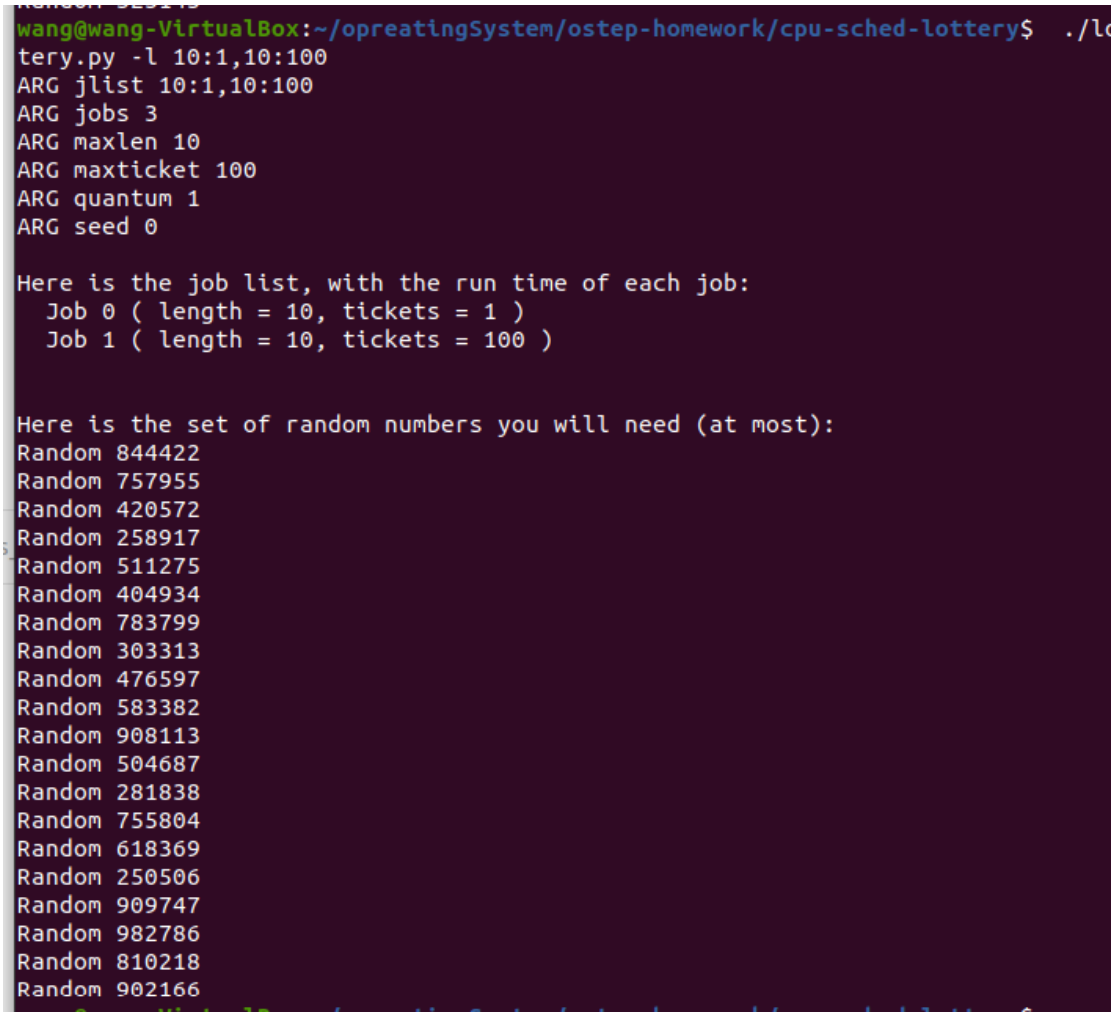
Speed 为 1


wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched-lottery$ ./lotte
ry.py -j 3 -s 1
ARG jlist
ARG jobs 3
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 1
Here is the job list, with the run time of each job:
Job 0 (length = 1, tickets = 84)
Job 1 (length = 7, tickets = 25)
Job 2 (length = 4, tickets = 44)
Here is the set of random numbers you will need (at most):
Random 651593
Random 788724
Random 93859
Random 28347
Random 835765
Random 432767
Random 762280
Random 2106
Random 445387
Random 721540
Random 228762
Random 945271

Speed 为 2


wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched-lottery$ ./lott
ery.py -j 3 -s 2
ARG jlist
ARG jobs 3
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 2
Here is the job list, with the run time of each job:
Job 0 (length = 9, tickets = 94)
Job 1 (length = 8, tickets = 73)
Job 2 (length = 6, tickets = 30)
Here is the set of random numbers you will need (at most):
Random 605944
Random 606802
Random 581204
Random 158383
Random 430670
Random 393532
Random 723012
Random 994820
Random 949396
Random 544177
Random 444854
Random 268241
Random 35924
Random 27444
Random 464894
Random 318465
Random 380015
Random 891790
Random 525753
Random 560510
Random 236123
Random 23858
Random 325143

9.2



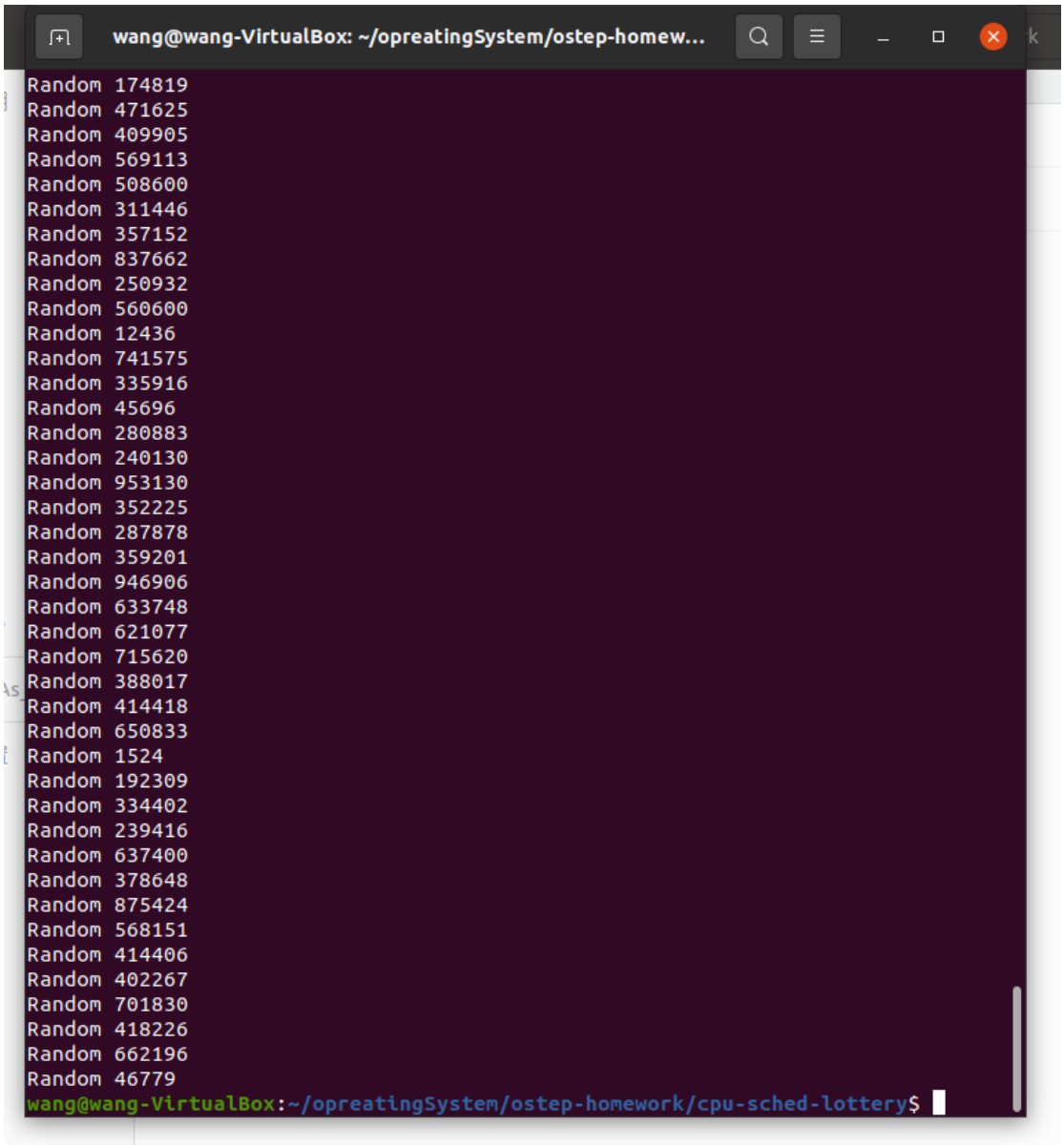
只有一张彩票被调度的可能性非常小，可能会饿死。
在工作 1 完成之前，工作 0 可能会运行。
这种行为可能会导致平均周转和响应时间变得很差

9.3

```
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched-lottery$ ./lott
ery.py -l 100:100,100:100
ARG jlist 100:100,100:100
ARG jobs 3
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 0

Here is the job list, with the run time of each job:
  Job 0 ( length = 100, tickets = 100 )
  Job 1 ( length = 100, tickets = 100 )


Here is the set of random numbers you will need (at most):
Random 844422
Random 757955
Random 420572
Random 258917
Random 511275
Random 404934
Random 783799
Random 303313
Random 476597
Random 583382
Random 908113
Random 504687
Random 281838
Random 755804
Random 618369
Random 250506
Random 909747
Random 982786
Random 810218
Random 902166
Random 310147
```


wang@wang-VirtualBox: ~/opreatingSystem/ostep-homew...
Random 174819
Random 471625
Random 409905
Random 569113
Random 508600
Random 311446
Random 357152
Random 837662
Random 250932
Random 560600
Random 12436
Random 741575
Random 335916
Random 45696
Random 280883
Random 240130
Random 953130
Random 352225
Random 287878
Random 359201
Random 946906
Random 633748
Random 621077
Random 715620
Random 388017
Random 414418
Random 650833
Random 1524
Random 192309
Random 334402
Random 239416
Random 637400
Random 378648
Random 875424
Random 568151
Random 414406
Random 402267
Random 701830
Random 418226
Random 662196
Random 46779
wang@wang-VirtualBox:~/opreatingSystem/ostep-homework/cpu-sched-lottery$