304丁海桐_作业2

> 9-1．计算 3 个工作在随机种子为 1、2 和 3 时的模拟解。

SEED = 1时：

```
$ ./lottery.py -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
```

模拟解：

SEED-1	JOB-0	JOB-1	JOB-2	Random	SUM	MOD
彩票个数	84	25	44			
运行长度	1	7	4			
彩票归属	*0-83*	*84-108*	*109-152*			
CLK-1	0	0	*1	651593	153	119
CLK-2	*1 (done)	0	1	788724	153	9
彩票归属		*0-24*	*25-68*			
CLK-3		1	*1	93859	69	19
CLK-4		1	*2	28347	69	57
CLK-5		1	*3	835765	69	37
CLK-6		1	*4 (done)	432767	69	68
彩票归属		*0-24*				
CLK-7		*2		762280	25	5
CLK-8		*3		2106	25	6
CLK-9		*4		445387	25	12
CLK-10		*5		721540	25	15
CLK-11		*6		228762	25	12
CLK-12		*7 (done)		945271	25	21

SEED=2时：

```
$ ./lottery.py -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
```

模拟解：

SEED-2	JOB-0	JOB-1	JOB-2	Random	SUM	MOD
彩票个数	94	73	30			
运行长度	9	8	6			
彩票归属	*0-93*	*94-166*	*167-196*			
CLK-1	0	0	*1	605944	197	169
CLK-2	*1	0	1	606802	197	42
CLK-3	*2	0	1	581204	197	54
CLK-4	2	0	*2	158383	197	192
CLK-5	*3	0	2	430670	197	28
CLK-6	3	*1	2	393532	197	123
CLK-7	*4	1	2	723012	197	22
CLK-8	4	1	*3	994820	197	167
CLK-9	*5	1	3	949396	197	53
CLK-10	*6	1	3	544177	197	63
CLK-11	*7	1	3	444854	197	28
CLK-12	7	*2	3	268241	197	124
CLK-13	*8	2	3	35924	197	70
CLK-14	*9 (done)	2	3	27444	197	61
彩票归属		*0-72*	*73-102*			
CLK-15		*3	4	464894	103	55
CLK-16		3	*4	318465	103	92
CLK-17		*4	4	380015	103	48
CLK-18		*5	4	891790	103	16
CLK-19		*6	4	525753	103	41
CLK-20		6	*5	560510	103	87
CLK-21		*7	5	236123	103	47
CLK-22		*8 (done)	5	23858	103	65
彩票归属			*0-29*			
CLK-23			*6 (done)	325143	30	3

SEED = 3时：

```
$ ./lottery.py -s 3
ARG jlist
ARG jobs 3
ARG maxlen 10
ARG maxticket 100
ARG quantum 1
ARG seed 3

Here is the job list, with the run time of each job:
  Job 0 ( length = 2, tickets = 54 )
  Job 1 ( length = 3, tickets = 60 )
  Job 2 ( length = 6, tickets = 6 )


Here is the set of random numbers you will need (at most):
Random 13168
Random 837469
Random 259354
Random 234331
Random 995645
Random 470263
Random 836462
Random 476353
Random 639068
Random 150616
Random 634861
```

模拟解：

SEED-3	JOB-0	JOB-1	JOB-2	Random	SUM	MOD
彩票个数	54	60	6			
运行长度	2	3	6			
彩票归属	*0-53*	*54-113*	*114-119*			
CLK-1	0	*1	0	13168	120	88
CLK-2	0	*2	0	837469	120	109
CLK-3	*1	2	0	259354	120	34
CLK-4	1	*3	0	234331	120	91
彩票归属	*0-53*		*54-59*			
CLK-5	*2		0	995645	60	5
彩票归属			*0-5*			
CLK-6			*1	470263	6	1
CLK-7			*2	836462	6	2
CLK-8			*3	476353	6	1
CLK-9			*4	639068	6	2
CLK-10			*5	150616	6	4
CLK-11			*6	634861	6	1

> 9-2．现在运行两个具体的工作：每个长度为 10，但是一个（工作 0）只有一张彩票，另 一个（工作 1）有 100 张（-l 10 : 1,10 : 100）。彩票数量如此不平衡时会发生什么？在工作 1 完成之前，工作 0 是否会运行？多久？ 一般来说，这种彩票不平衡对彩票调度的行为有什么影响？

```
$ ./lottery.py -l 10:1,10:100
ARG jlist 10:1,10:100
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 = 10, tickets = 1 )
  Job 1 ( length = 10, 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
```

	JOB-0	JOB-1	Random	SUM	MOD
彩票个数	100	1			
运行长度	10	10			
彩票归属	***0-99***	***100***			
CLK-1	*1	0	844422	101	62
CLK-2	*2	0	757955	101	51
CLK-3	*3	0	420572	101	8
CLK-4	*4	0	258917	101	54
CLK-5	*5	0	511275	101	13
CLK-6	*6	0	404934	101	25
CLK-7	*7	0	783799	101	39
CLK-8	*8	0	303313	101	10
CLK-9	*9	0	476597	101	79
CLK-10	*10（done）	0	583382	101	6
彩票归属		***0***			
CLK-11		*1	908113	1	0
CLK-12		*2	504687	1	0
CLK-13		*3	281838	1	0
CLK-14		*4	755804	1	0
CLK-15		*5	618369	1	0
CLK-16		*6	250506	1	0
CLK-17		*7	909747	1	0
CLK-18		*8	982786	1	0
CLK-19		*9	810218	1	0
CLK-20		*10（done）	902166	1	0

- 彩票数量如此不平衡，会导致CPU先把彩票量多的工作做完才去做彩票量少的工作。
- 在工作0完成之前工作不运行。
- 会导致彩票量少的工作的响应时间和周转时间都很长。

> 9-3．如果运行两个长度为 100 的工作，都有 100 张彩票（-l100 : 100,100 : 100），调度 程序有多不公平？运行一些不同的随机种子来确定（概率上的）答案。不公平性取决于一 项工作比另一项工作早完成多少

SEED	JOB-0	JOB-1	SUB
100	200	192	8
101	200	191	9
101	200	190	10
101	200	192	8
101	200	190	10
101	200	191	9
101	200	190	10
AVG	200	190.86	9.14

先完成的比后完成的平均时间上快了9.14，这种算法相对公平。

> 15-1．用种子 1、2 和 3 运行，并计算进程生成的每个虚拟地址是处于界限内还是界限外？ 如果在界限内，请计算地址转换。

SEED-1：

```
$ ./relocation.py -s 1

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x0000363c (decimal 13884)
  Limit  : 290

Virtual Address Trace
  VA  0: 0x0000030e (decimal:  782) --> PA or segmentation violation?
  VA  1: 0x00000105 (decimal:  261) --> PA or segmentation violation?
  VA  2: 0x000001fb (decimal:  507) --> PA or segmentation violation?
  VA  3: 0x000001cc (decimal:  460) --> PA or segmentation violation?
  VA  4: 0x0000029b (decimal:  667) --> PA or segmentation violation?
```

ANSWER：

```
  VA  0: 0x0000030e (decimal:  782) --> SEGMENTATION VIOLATION
  VA  1: 0x00000105 (decimal:  261) --> VALID: 0x00003741 (decimal: 14145)
  VA  2: 0x000001fb (decimal:  507) --> SEGMENTATION VIOLATION
  VA  3: 0x000001cc (decimal:  460) --> SEGMENTATION VIOLATION
  VA  4: 0x0000029b (decimal:  667) --> SEGMENTATION VIOLATION
```

SEED-2：

```
$ ./relocation.py -s 2

ARG seed 2
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x00003ca9 (decimal 15529)
  Limit  : 500

Virtual Address Trace
  VA  0: 0x00000039 (decimal:   57) --> PA or segmentation violation?
  VA  1: 0x00000056 (decimal:   86) --> PA or segmentation violation?
  VA  2: 0x00000357 (decimal:  855) --> PA or segmentation violation?
  VA  3: 0x000002f1 (decimal:  753) --> PA or segmentation violation?
  VA  4: 0x000002ad (decimal:  685) --> PA or segmentation violation?
```

```
  VA  0: 0x00000039 (decimal:   57) --> VALID: 0x00003ce2 (decimal: 15586)
  VA  1: 0x00000056 (decimal:   86) --> VALID: 0x00003cff (decimal: 15615)
  VA  2: 0x00000357 (decimal:  855) --> SEGMENTATION VIOLATION
  VA  3: 0x000002f1 (decimal:  753) --> SEGMENTATION VIOLATION
  VA  4: 0x000002ad (decimal:  685) --> SEGMENTATION VIOLATION
```

SEED-3：

```
$ ./relocation.py -s 3

ARG seed 3
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x000022d4 (decimal 8916)
  Limit  : 316

Virtual Address Trace
  VA  0: 0x0000017a (decimal:  378) --> PA or segmentation violation?
  VA  1: 0x0000026a (decimal:  618) --> PA or segmentation violation?
  VA  2: 0x00000280 (decimal:  640) --> PA or segmentation violation?
  VA  3: 0x00000043 (decimal:   67) --> PA or segmentation violation?
  VA  4: 0x0000000d (decimal:   13) --> PA or segmentation violation?
```

```
  VA  0: 0x0000017a (decimal:  378) --> SEGMENTATION VIOLATION
  VA  1: 0x0000026a (decimal:  618) --> SEGMENTATION VIOLATION
  VA  2: 0x00000280 (decimal:  640) --> SEGMENTATION VIOLATION
  VA  3: 0x00000043 (decimal:   67) --> VALID: 0x00002317 (decimal: 8983)
  VA  4: 0x0000000d (decimal:   13) --> VALID: 0x000022e1 (decimal: 8929)
```

> 15-2．使用以下标志运行：-s 0 -n 10。为了确保所有生成的虚拟地址都处于边界内，要将 -l（界限寄存器）设置为什么值？

```
$ ./relocation.py -s 0 -n 10

ARG seed 0
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x00003082 (decimal 12418)
  Limit  : 472

Virtual Address Trace
  VA  0: 0x000001ae (decimal:  430) --> PA or segmentation violation?
  VA  1: 0x00000109 (decimal:  265) --> PA or segmentation violation?
  VA  2: 0x0000020b (decimal:  523) --> PA or segmentation violation?
  VA  3: 0x0000019e (decimal:  414) --> PA or segmentation violation?
  VA  4: 0x00000322 (decimal:  802) --> PA or segmentation violation?
  VA  5: 0x00000136 (decimal:  310) --> PA or segmentation violation?
  VA  6: 0x000001e8 (decimal:  488) --> PA or segmentation violation?
  VA  7: 0x00000255 (decimal:  597) --> PA or segmentation violation?
```

```
  VA  8: 0x000003a1 (decimal:  929) --> PA or segmentation violation?
  VA  9: 0x00000204 (decimal:  516) --> PA or segmentation violation?
```

VA中最大的是VA 8: 0x000003a1 (decimal: 929)，让其在边界内，所以接线寄存器设置为0x000003a2（decimal: 930）。

> 15-3．使用以下标志运行：-s 1 -n 10 -l 100。可以设置基址的最大值是多少，以便地址空间 仍然完全放在物理内存中?

```
$ ./relocation.py -s 1 -n 10 -l 100

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

  Base   : 0x00000899 (decimal 2201)
  Limit  : 100

Virtual Address Trace
  VA  0: 0x00000363 (decimal:  867) --> PA or segmentation violation?
  VA  1: 0x0000030e (decimal:  782) --> PA or segmentation violation?
  VA  2: 0x00000105 (decimal:  261) --> PA or segmentation violation?
  VA  3: 0x000001fb (decimal:  507) --> PA or segmentation violation?
  VA  4: 0x000001cc (decimal:  460) --> PA or segmentation violation?
  VA  5: 0x0000029b (decimal:  667) --> PA or segmentation violation?
  VA  6: 0x00000327 (decimal:  807) --> PA or segmentation violation?
  VA  7: 0x00000060 (decimal:   96) --> PA or segmentation violation?
  VA  8: 0x0000001d (decimal:   29) --> PA or segmentation violation?
  VA  9: 0x00000357 (decimal:  855) --> PA or segmentation violation?
```

base = 16k - 100 = 16284 = 0x3F9C