

作业 4

贾城昊

2021K8009929010

5.1 现有5个作业要在一台计算机上依次执行，它们的运行时间分别是9, 3, 5, 11和X。请问：1) 该以何种顺序运行这5个作业，从而可以获得最短的平均响应时间？2) 如果要获得最短的平均周转时间，该以何种顺序运行这5个作业

(1) 需要根据 X 进行分类讨论：

a. 如果 $X \leq 3$ ，顺序应该是 X、3、5、9、11。平均响应时间为

$$\frac{[0 + X + (X+3) + (X+3+5) + (X+3+5+9)]}{5} = 0.8X + 5.6$$

- 如果 $X = 1$ ，平均响应时间为 6.4
- 如果 $X = 2$ ，平均响应时间为 7.2
- 如果 $X = 3$ ，平均响应时间为 8.0

b. 如果 $3 < X \leq 5$ ，顺序应该是 3、X、5、9、11。平均响应时间为

$$\frac{[0 + 3 + (3+X) + (3+X+5) + (3+X+5+9)]}{5} = 0.6X + 6.2$$

- 如果 $X = 4$ ，平均响应时间为 8.6
- 如果 $X = 5$ ，平均响应时间为 9.2

c. 如果 $5 < X \leq 9$ ，顺序应该是 3、5、X、9、11。平均响应时间为

$$\frac{[0 + 3 + (3+5) + (3+5+X) + (3+5+X+9)]}{5} = 0.4X + 7.2$$

- 如果 $X = 6$ ，则平均响应时间为 9.6
- 如果 $X = 7$ ，则平均响应时间为 10.0
- 如果 $X = 8$ ，则平均响应时间为 10.4
- 如果 $X = 9$ ，则平均响应时间为 10.8

d. 如果 $9 < X \leq 11$, 则依次为 3、5、9、X、11。平均响应时间为

$$\frac{[0+3+(3+5)+(3+5+9)+(3+5+9+X)]}{5} = 0.2X + 9.0$$

- 如果 $X = 10$, 平均响应时间为 11.0
- 如果 $X = 11$, 平均响应时间为 11.2

e. 如果 $X \geq 11$, 则依次为 3、5、9、11、X。平均响应时间为平均响应时间

$$\text{为} \frac{[0+3+(3+5)+(3+5+9)+(3+5+9+11)]}{5} = 11.2$$

(2) 同样需要根据 X 进行分类讨论:

a. 如果 $X \leq 3$, 顺序应该是 X、3、5、9、11。平均周转时间为

$$\frac{[X+(X+3)+(X+3+5)+(X+3+5+9)+(X+3+5+9+11)]}{5} = X + 11.2$$

- 如果 $X = 1$, 平均周转时间为 12.2
- 如果 $X = 2$, 平均周转时间为 13.2
- 如果 $X = 3$, 平均周转时间为 14.2

b. 如果 $3 < X \leq 5$, 顺序应该是 3、X、5、9、11。平均响应时间为

$$\frac{[3+(3+X)+(3+X+5)+(3+X+5+9)+(3+X+5+9+11)]}{5} = 0.8X + 11.8$$

- 如果 $X = 4$, 平均周转时间为 15.0
- 如果 $X = 5$, 平均周转时间为 15.8

c. 如果 $5 < X \leq 9$, 顺序应该是 3、5、X、9、11。平均响应时间为

$$\frac{[3+(3+5)+(3+5+X)+(3+5+X+9)+(3+5+X+9+11)]}{5} = 0.6X + 12.8$$

- 如果 $X = 6$, 则平均周转时间为 16.4
- 如果 $X = 7$, 则平均周转时间为 17.0
- 如果 $X = 8$, 则平均周转时间为 17.6
- 如果 $X = 9$, 则平均周转时间为 18.2

- d. 如果 $9 < X \leq 11$, 则依次为 3、5、9、X、11。平均响应时间为
- $$\frac{[3+(3+5)+(3+5+9)+(3+5+9+X)+(3+5+9+X+11)]}{5} = 0.4X + 14.6$$
- 如果 $X = 10$, 平均周转时间为 18.6
 - 如果 $X = 11$, 平均周转时间为 19.0
- e. 如果 $X \geq 11$, 则依次为 3、5、9、11、X。平均响应时间为
- $$\frac{[3+(3+5)+(3+5+9)+(3+5+9+11)+(3+5+9+11+X)]}{5} = 0.2X + 16.8$$
- 如果 $X = 12$, 平均周转时间为 19.2

5.2 现有5个作业（作业A、B、C、D、E）要在一台计算机上执行。假设它们在同一时间被提交，同时它们的运行时间分别是10、8、4、12和15分钟。当使用以下CPU调度算法运行这5个作业时，请计算平均等待时间。

- (1) Round robin算法（使用该算法时，每个作业分到的CPU时间片相等）
- (2) 优先级调度算法（作业A-E的优先级分别是：2,5,1,3,4，其中5是最高优先级，1是最低优先级）
- (3) First-come, first-served算法（假设作业的达到顺序是A, B, C, D, E）
- (4) Shortest job first算法

注意：假设作业切换可以瞬时完成，即开销为0。

(1)假设时间片为 4min，则这五个任务的甘特图如下：

A	B	C	D	E	A	B	D	E	A	D	E	E	
0	4	8	12	16	20	24	28	32	36	38	42	46	49

所以 A 的等待时间为 $(20-4) + (36-24) = 28$ 分钟

B 的等待时间为 $(4-0) + (24-8) = 20$ 分钟

C 的等待时间为 $8-0 = 8$ 分钟

D 的等待时间为 $(12-0) + (28-16) + (38-32) = 30$ 分钟

E 的等待时间为 $(16-0) + (32-20) + (42-36) = 34$ 分钟

平均等待时间为 $\frac{28 + 20 + 8 + 30 + 34}{5} = 24$ 分钟

(2) 根据 A B C D E 的优先级, 可得其运行顺序是 B、E、D、A、C。所以平均等待时间为 $\frac{[0+8+(8+15) + (8+15+12)+(8+15+12+10)]}{5} = 22.2$ 分钟

(3) 根据 FCFS 算法, 可知运行顺序为 A、B、C、D、E, 所以平均等待时间为 $\frac{[0+10+(10+8)+(10+8+4)+ (10+8+4+12)]}{5} = 16.8$ 分钟

(4) 根据 SJF 算法, 可知运行顺序为 C、B、A、D、E, 所以平均等待时间为 $\frac{[0+4+(4+8)+(4+8+10)+(4+8+10+12)]}{5} = 14.4$ 分钟

5.3 A real-time system needs to handle two voice calls that each run every 5 msec and consume 1 msec of CPU time per burst, plus one video at 24 frames/sec, with each frame requiring 20 msec of CPU time. Is this system schedulable?

解:

$$\text{可知} \sum \frac{C_i}{T_i} = \frac{1}{5} + \frac{1}{5} + \frac{24 \times 20}{1000} = \frac{22}{25} < 1$$

所以这个系统是可以调度的。

5.4 作为容器技术的重要基础，cgroups 为 Linux 提供了内存、CPU 等资源的分配与限制功能。Cgroups 的使用手册可在 Linux 系统中通过`man cgroups`指令查看，或访问官方网页[1]。请重点关注 cgroups 中 cpu 子系统的文档[2]及其基本用法，回答以下问题，并附上必要代码和截图：

- (1) 回顾上一次作业中的绑核操作，写一个简单的程序，使其绑定 1 号 CPU，且 CPU 占用率达到 100%
- (2) 应用 cgroups 功能，将 (1) 中程序的 CPU 占用率限制在 30%以下
- (3) 重新启动共计 2 个 (1) 中的程序，观察它们各自的 CPU 占用率
- (4) 应用 cgroups 功能，将 (3) 中程序的 CPU 占用率调整为 2:1，并验证你的实现效果

注 1: 强烈建议同学优先尝试阅读官方手册，学习从说明文档获取关键信息的能力；若确实存在困难，可查询中文资料并注明参考出处

注 2: 可以通过 Linux 的`top`命令查看系统内各进程的 CPU 等资源占用率

[1] <https://man7.org/linux/man-pages/man7/cgroups.7.html>

[2] <https://www.kernel.org/doc/Documentation/scheduler/sched-bwc.txt>

(1) 运行 hw5 (使得 CPU 占用率为 100%)

1. C 程序如下:

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <sched.h>
#include <unistd.h>

int main() {
    cpu_set_t mask;
    CPU_ZERO(&mask);
    CPU_SET(1, &mask); // 绑定到 CPU 1

    if (sched_setaffinity(0, sizeof(mask), &mask) == -1) {
        perror("sched_setaffinity");
    }

    while (1) {
        int a = 1;
        int b = 1;
        int c = a + b;
    }

    return 0;
}
```

2. 运行结果

上述程序运行后的 CPU 占用率如下所示:

```
top - 16:50:47 up 2 min, 1 user, load average: 1.03, 0.33, 0.12
Tasks: 251 total, 2 running, 249 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8.6 us, 0.0 sy, 0.0 ni, 91.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 4045564 total, 1928488 free, 1067780 used, 1049296 buff/cache
KiB Swap: 2095100 total, 2095100 free, 0 used. 2644876 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2545	sai	20	0	4216	652	584	R	100.0	0.0	0:25.24	hw5
1822	sai	20	0	1480396	255604	93252	S	0.7	6.3	0:13.69	compiz
1174	root	20	0	523104	147972	58276	S	0.3	3.7	0:02.81	Xorg
1247	sai	20	0	180604	2812	2464	S	0.3	0.1	0:00.10	VBoxClient
1	root	20	0	119740	5848	3916	S	0.0	0.1	0:00.68	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
6	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kworker/u1+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.03	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/1
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:+

可见该程序的 CPU 占用率为 100%

(2) 应用 cgroups 功能限制 hw5 的 CPU 占用率:

在linux终端输入如下指令:

```
$mkdir /sys/fs/cgroup/cpu/test
```

```
$ cd /sys/fs/cgroup/cpu/ test
```

```
$ sudo -s
```

```
$ sudo echo 100000 > cpu.cfs_period_us
```

```
$ sudo echo 30000 > cpu.cfs_quota_us
```

```
$ echo 777 > tasks
```

(该hw5进程的pid为777)

查看CPU占用情况, 如下所示:

```
top - 21:27:45 up 4:39, 1 user, load average: 0.22, 0.22, 0.29
Tasks: 248 total, 2 running, 246 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.9 us, 0.1 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 4045564 total, 1681424 free, 1246040 used, 1118100 buff/cache
KiB Swap: 2095100 total, 2095100 free, 0 used. 2458748 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
777	root	20	0	4216	620	552	R	29.9	0.0	0:28.17	hw5
1822	sai	20	0	1480532	257044	93876	S	2.3	6.4	22:56.12	compiz
1174	root	20	0	558304	180696	59200	S	0.7	4.5	2:40.75	Xorg
555	root	20	0	0	0	0	S	0.3	0.0	0:00.01	kworker/u1+
714	root	20	0	0	0	0	S	0.3	0.0	0:00.06	kworker/3:2
1247	sai	20	0	180604	2812	2464	S	0.3	0.1	0:10.21	VBoxClient
1	root	20	0	119740	5848	3916	S	0.0	0.1	0:00.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.85	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.07	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.11	migration/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.06	ksoftirqd/1

可见此时hw5程序的CPU占用率变为了29.9%

(3) 启动两个 hw5 程序

启动共计2个 (1) 中的程序，并在CPU终端输入

```
$ echo 841 > tasks
```

(echo 777 > tasks之前已经输入过了)

查看CPU占用率如下：

sai@Computer: ~

op - 21:33:41 up 4:45, 1 user, load average: 1.12, 0.97, 0.61
 asks: 252 total, 3 running, 249 sleeping, 0 stopped, 0 zombie
 Cpu(s): 3.1 us, 0.1 sy, 0.0 ni, 96.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
 iB Mem : 4045564 total, 1671252 free, 1255796 used, 1118516 buff/cache
 iB Swap: 2095100 total, 2095100 free, 0 used. 2448876 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
777	root	20	0	4216	620	552	R	15.0	0.0	2:11.95	hw5
841	sai	20	0	4216	648	580	R	15.0	0.0	2:40.58	hw5
1822	sai	20	0	1480532	257044	93876	S	2.7	6.4	23:07.75	compiz
1174	root	20	0	558304	180696	59200	S	0.3	4.5	2:43.16	Xorg
1247	sai	20	0	180604	2812	2464	S	0.3	0.1	0:10.61	VBoxClient
2193	sai	20	0	635428	55000	40996	S	0.3	1.4	0:04.46	gnome-term+
1	root	20	0	119740	5848	3916	S	0.0	0.1	0:00.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.87	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.07	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.11	migration/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.06	ksoftirqd/1

可见两个hw5程序的CPU占用率为15%，加起来才为30%

(4) 应用 cgroups 功能控制两个程序 CPU 占用率比例

再次运行一个hw5程序（之前进程号为841的hw5程序终止运行了），并在linux终端继续输入如下指令：

```
$mkdir /sys/fs/cgroup/cpu/test2
$ cd /sys/fs/cgroup/cpu/ test2
$ sudo -s
$ sudo echo 100000 > cpu.cfs_period_us
$ sudo echo 15000 > cpu.cfs_quota_us
$ echo 905 > tasks
```

然后查看CPU占用率，结果如下

```

sai@Computer: ~
op - 21:36:23 up 4:48, 1 user, load average: 0.81, 0.81, 0.60
asks: 252 total, 3 running, 249 sleeping, 0 stopped, 0 zombie
Cpu(s): 4.4 us, 0.0 sy, 0.0 ni, 95.4 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
iB Mem : 4045564 total, 1678944 free, 1248120 used, 1118500 buff/cache
iB Swap: 2095100 total, 2095100 free, 0 used. 2456628 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
777	root	20	0	4216	620	552	R	29.7	0.0	2:55.44	hw5
905	root	20	0	4216	616	552	R	15.3	0.0	0:12.06	hw5
1822	sai	20	0	1480532	257044	93876	S	2.7	6.4	23:19.38	compiz
202	root	20	0	0	0	0	S	0.3	0.0	0:00.10	jbd2/sda1-8
1174	root	20	0	558304	180696	59200	S	0.3	4.5	2:45.67	Xorg
2412	sai	20	0	36.410g	69320	52312	S	0.3	1.7	0:00.68	code
1	root	20	0	119740	5848	3916	S	0.0	0.1	0:00.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.88	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.07	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.11	migration/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.06	ksoftirqd/1

可见两个hw5程序分别占用CPU为30%和15%，占用率为2：1