# 作业5

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：
2. 如果X 3，顺序应该是X、3、5、9、11。平均响应时间为

* 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

1. 如果3 < X 5，顺序应该是3、X、5、9、11。平均响应时间为

* 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

1. 如果5 < X 9，顺序应该是3、5、X、9、11。平均响应时间为

* 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

1. 如果9 < X 11，则依次为3、5、9、X、11。平均响应时间为

* 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

1. 如果X 11，则依次为3、5、9、11、X。平均响应时间为平均响应时间为
2. 同样需要根据X进行分类讨论：
3. 如果X 3，顺序应该是X、3、5、9、11。平均周转时间为

* 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

1. 如果3 < X 5，顺序应该是3、X、5、9、11。平均响应时间为

* 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

1. 如果5 < X 9，顺序应该是3、5、X、9、11。平均响应时间为

* 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

1. 如果9 < X 11，则依次为3、5、9、X、11。平均响应时间为

* 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

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

* 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 分钟

1. 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 分钟
2. 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
3. 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 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：

可知 = < 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%）**
2. 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;

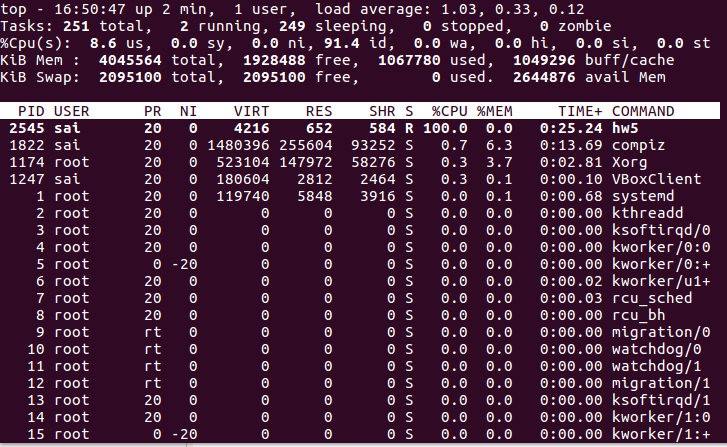
}

return 0;

}

1. 运行结果

上述程序运行后的CPU占用率如下所示：



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

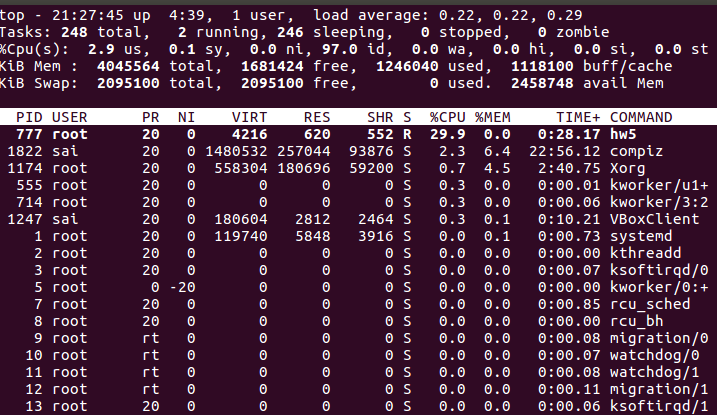
$ sudo echo 100000 > cpu.cfs\_period\_us

$ 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：



可见此时hw5程序的CPU占用率变为了29.9%

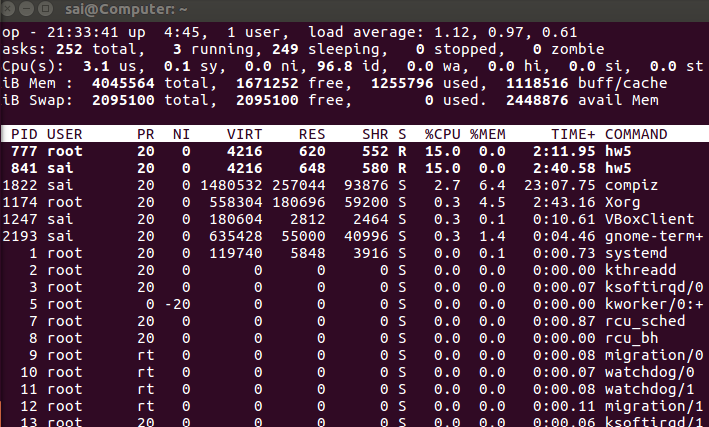
**(3)** **启动两个hw5程序**

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：



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

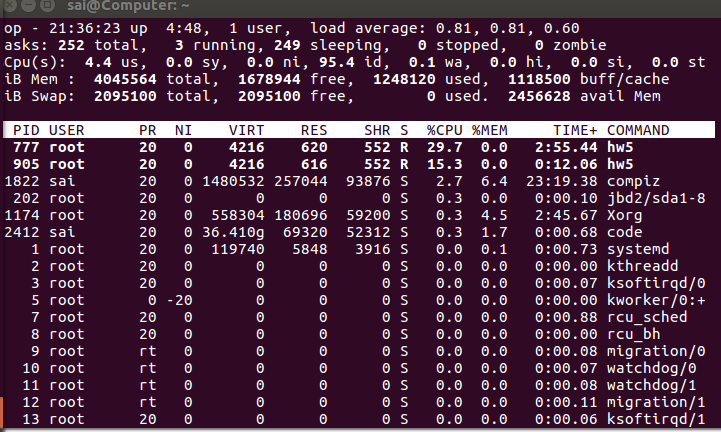
$ sudo -s

$ 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



可见两个hw5程序分别占用CPU为30%和15%，占用率为2：1