

- 一、创建资源组
- 二、CPU user
- 三、CPU sys
- 四、CPU wait
- 五、结论

cgroup可以限制CPU的使用率，那么这个使用率是否包括 CPU sys 和 IO wait 呢？我们做个小测试就可以知道。

## 一、创建资源组

准备一个资源组，CPU率限制为一个CPU核的50%：

```
[root@bogon ~]# mkdir /sys/fs/cgroup/cpu/yz/
[root@bogon ~]# chown yz:yz -R /sys/fs/cgroup/cpu/yz/
[root@bogon ~]# cd /sys/fs/cgroup/cpu/yz/
[root@bogon yz]# cat cpu.cfs_period_us
100000
[root@bogon yz]# echo 50000 > cpu.cfs_quota_us
```

## 二、CPU user

以下代码可以持续占用 CPU user。

```
int a = 1;
while (1)
{
    ++a;
}
```

编译执行程序，将PID加入上述资源组。改资源组的CPU（CPU user）使用率由100%降到了不到50%。

## 三、CPU sys

以下代码可以持续占用 CPU sys。

```
#include <time.h>

int main()
{
    clock_t start;
    while (1)
    {
        start = clock();
    }
    return 0;
}
```

编译执行程序，将PID加入上述资源组。改资源组的CPU（CPU sys）使用率由100%降到了不到50%。

## 四、CPU wait

---

可以使用 `dd` 的 `direct` 模式，构建 `iowait` 场景。

```
cgexec -g "cpu:yz/" dd if=/dev/zero of=./big.txt bs=1M oflag=direct
```

可以发现 `iowait` 不受限制。

## 五、结论

---

`CPU user` 和 `CPU sys` 使用率受 `cgroup` 控制，但是 `iowait` 不受限。