

13.14 限制内存和CPU的使用量 ¶

问题 ¶

你想对在Unix系统上面运行的程序设置内存或CPU的使用限制。

解决方案 ¶

`resource` 模块能同时执行这两个任务。例如，要限制CPU时间，可以像下面这样做：

```
import signal
import resource
import os

def time_exceeded(signo, frame):
    print("Time's up!")
    raise SystemExit(1)

def set_max_runtime(seconds):
    # Install the signal handler and set a resource limit
    soft, hard = resource.getrlimit(resource.RLIMIT_CPU)
    resource.setrlimit(resource.RLIMIT_CPU, (seconds, hard))
    signal.signal(signal.SIGXCPU, time_exceeded)

if __name__ == '__main__':
    set_max_runtime(15)
    while True:
        pass
```

程序运行时，`SIGXCPU` 信号在时间过期时被生成，然后执行清理并退出。

要限制内存使用，设置可使用的总内存值即可，如下：

```
import resource

def limit_memory(maxsize):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (maxsize, hard))
```

像这样设置了内存限制后，程序运行到没有多余内存时会抛出 `MemoryError` 异常。

讨论 ¶

在本节例子中，`setrlimit()` 函数被用来设置特定资源上面的软限制和硬限制。软限制是一个值，当超过这个值的时候操作系统通常会发送一个信号来限制或通知该进程。硬限制是用来指定软限制能设定的最大值。通常来讲，这个由系统管理员通过设置系统级参数来决定。尽管硬限制可以改小一点，但是最好不要使用用户进程去修改。

`setrlimit()` 函数还能被用来设置子进程数量、打开文件数以及类似系统资源的限制。更多详情请参考 `resource` 模块的文档。

需要注意的是本节内容只能适用于Unix系统，并且不保证所有系统都能如期工作。比如我们在测试的时候，它能在

Linux上面正常运行，但是在OS X上却不能。