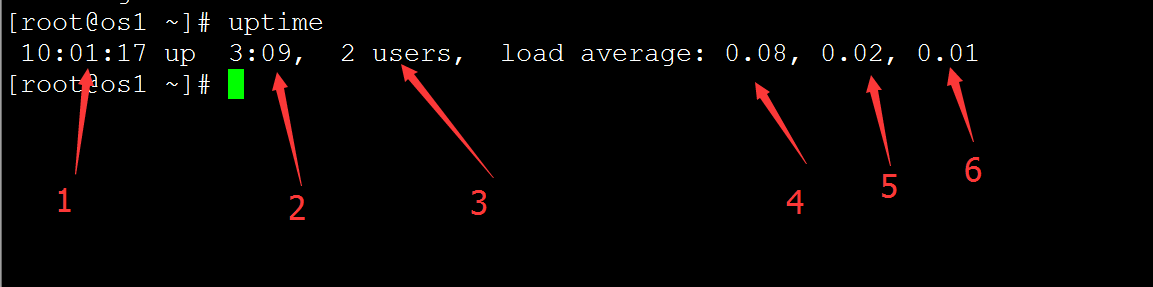
# LINUX性能监控工具

## uptime命令



1,为系统时间

2,系统运行的时间

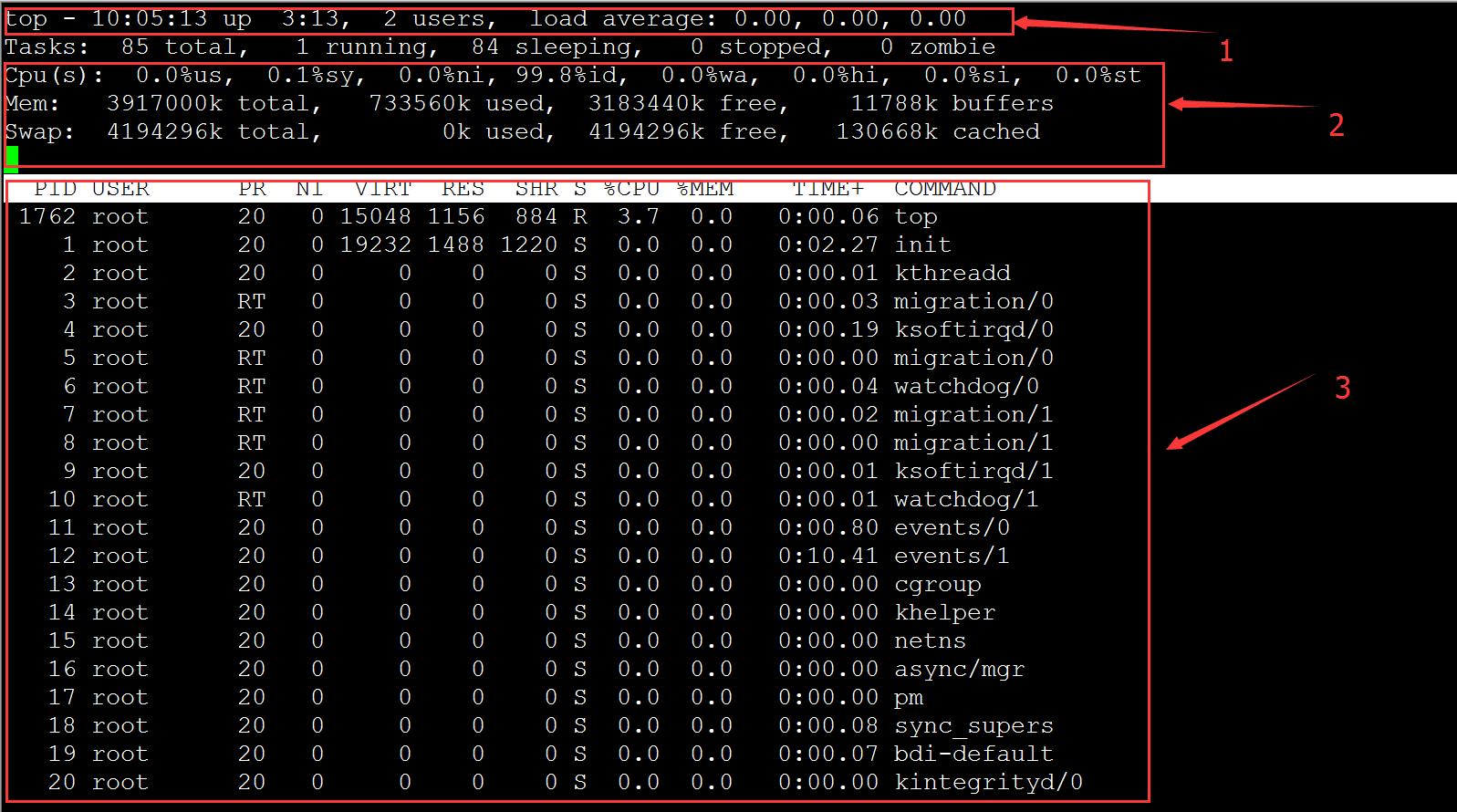
3,几个用户在访问

4, 运行队列中的平均进程数，在1分钟的系统平均负载

5, 运行队列中的平均进程数，在5分钟的系统平均负载

6, 运行队列中的平均进程数，在15分钟的系统平均负载

## top命令



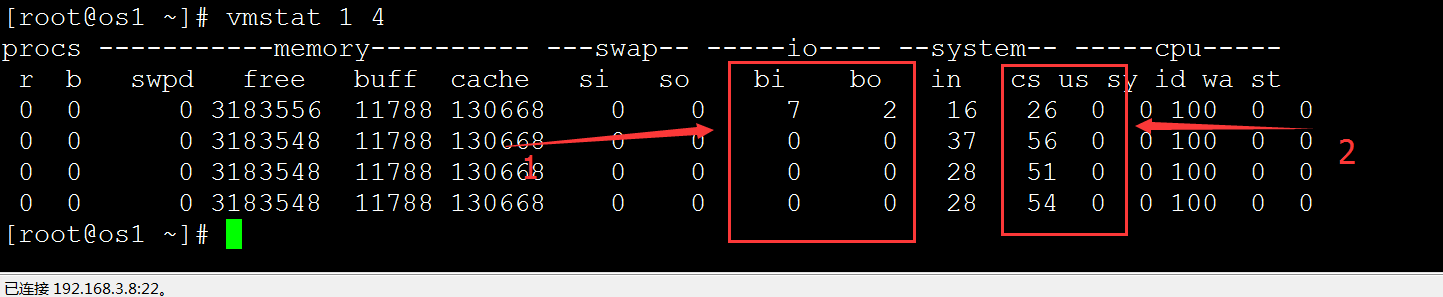
1,与uptime基本一致

2,当前系统的一些资源信息

3,正运行的线程信息

## vmstat命令

* + 可以统计系统的CPU，内存，swap，io等情况



vmstat 1 4 每1秒钟输出一次，一共输出4次 即采样频率与采样次数

1,BI代表Io的输入,bo代表IO的输出

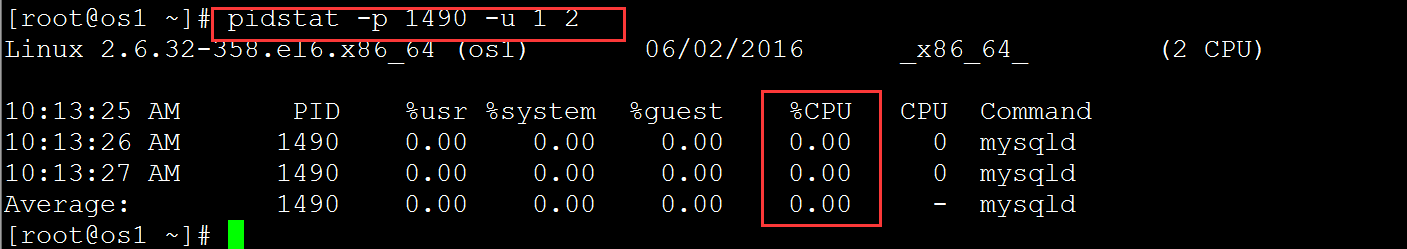
2,cs即系统的上下文切换频率 us 用户的上下文切换频率

## pidstat

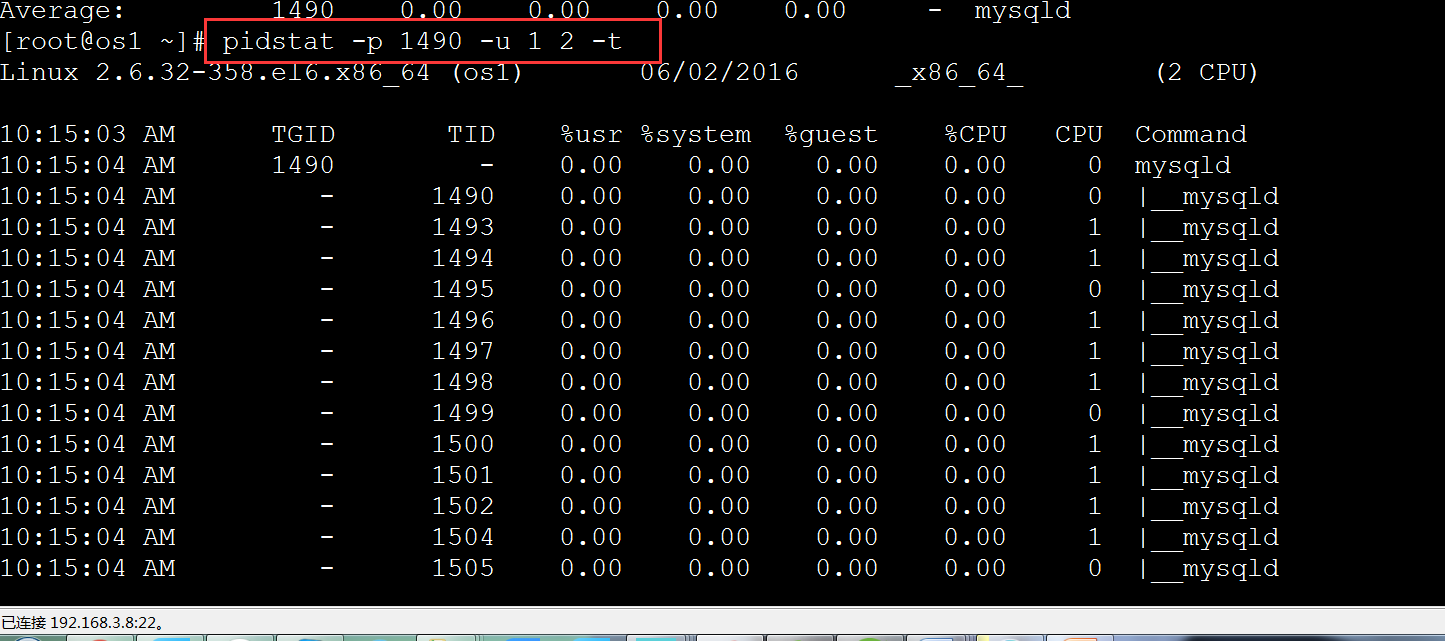
非系统自带，需要安装

如使用yum install sysstat

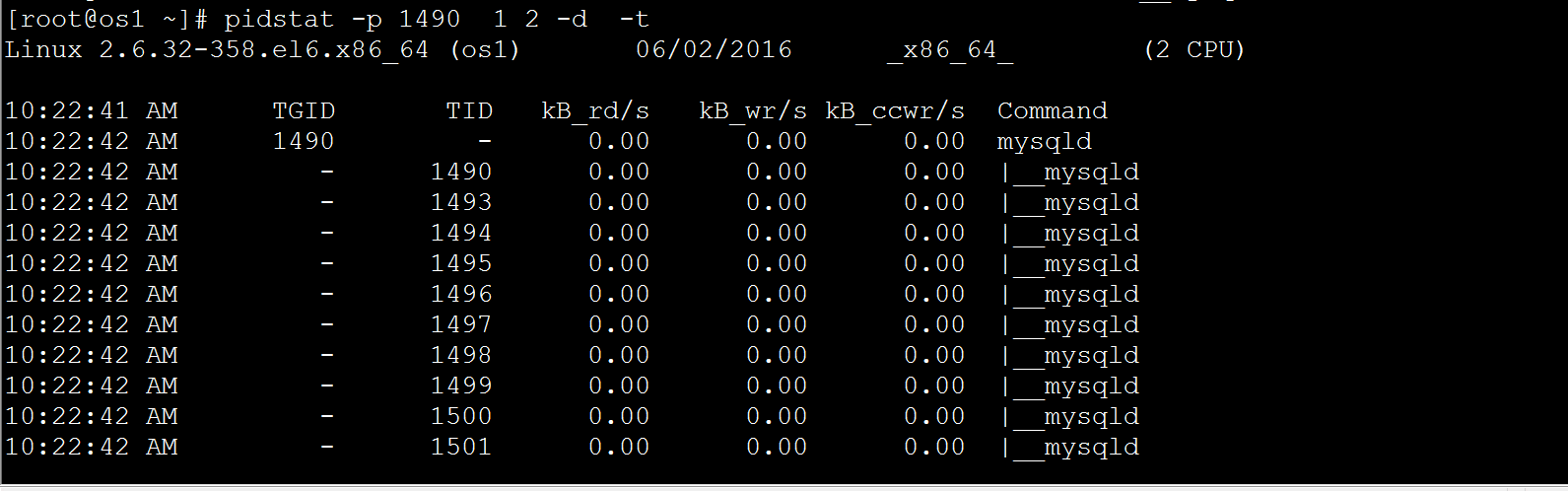
* 监控CPU
* 监控IO
* 监控内存



-p 指定进程 –u 监控CPU 每秒采样 一共2次



-t 显示线程 ,cpu列指在哪个物理CPU上面



-d 每个线程的IO的性能

# JAVA自带的性能监控工具

## jps

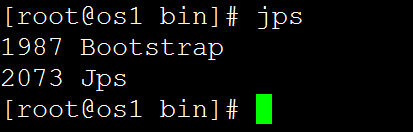
列出java进程，类似于ps命令

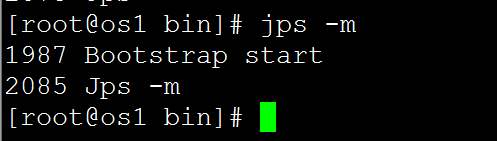
参数-q可以指定jps只输出进程ID ，不输出类的短名称

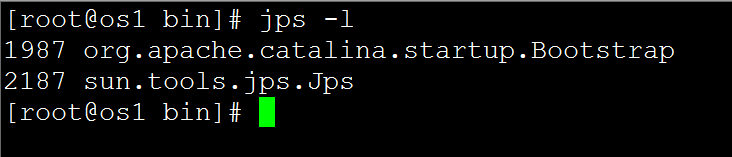
参数-m可以用于输出传递给Java进程（主函数）的参数

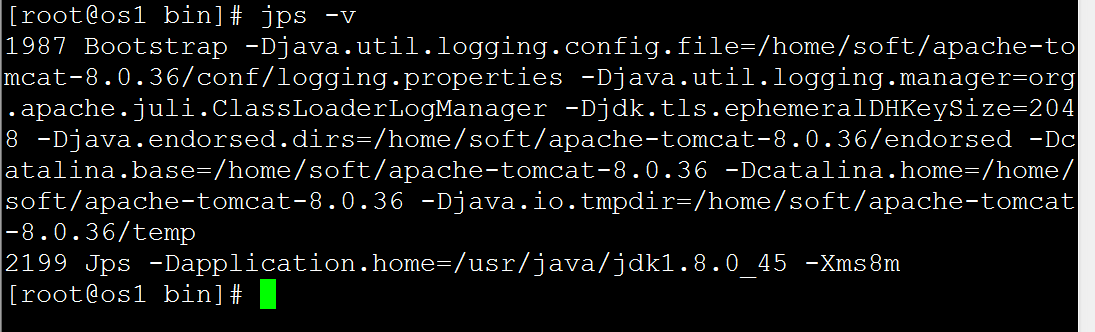
参数-l可以用于输出主函数的完整路径

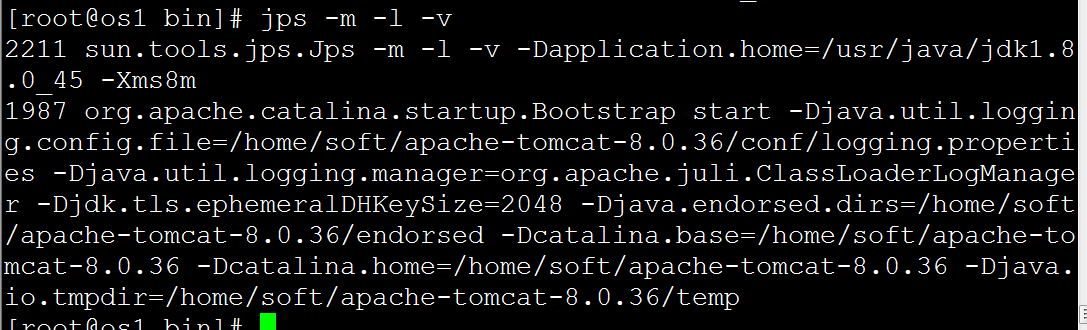
参数-v可以显示传递给JVM的参数











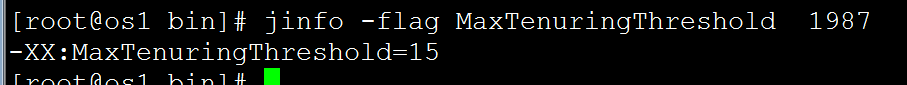
## jinfo

可以用来查看正在运行的Java应用程序的扩展参数，甚至支持在运行时，修改部分参数

-flag <name>：打印指定JVM的参数值

-flag [+|-]<name>：设置指定JVM参数的布尔值

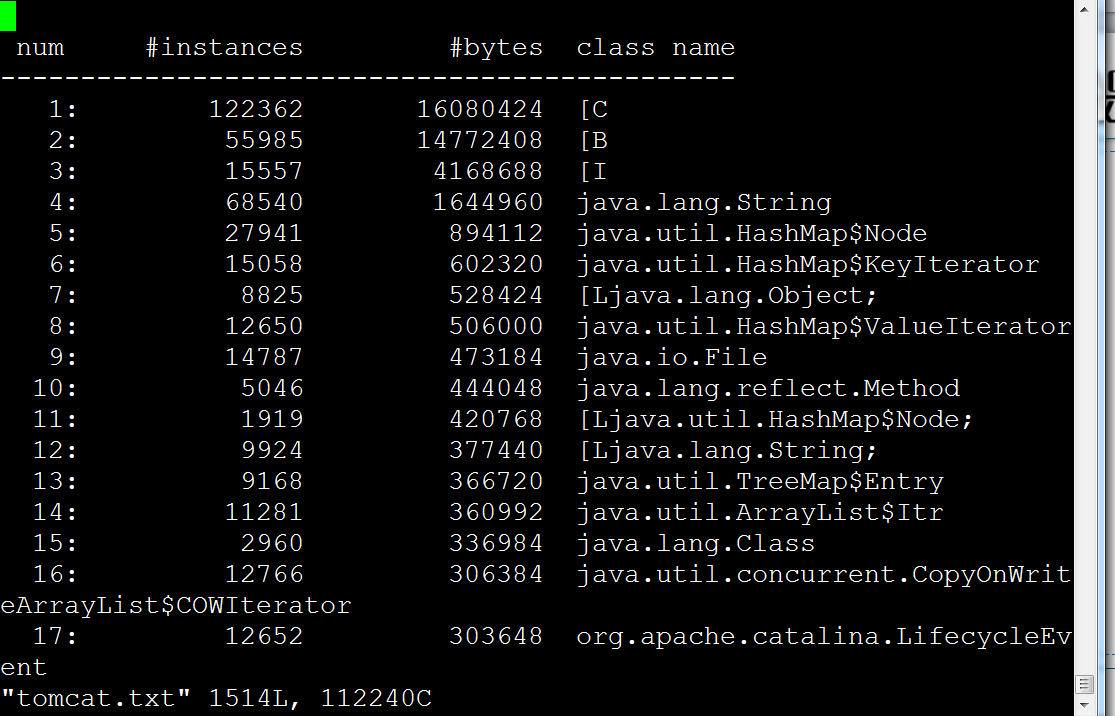
-flag <name>=<value>：设置指定JVM参数的值



## jmap

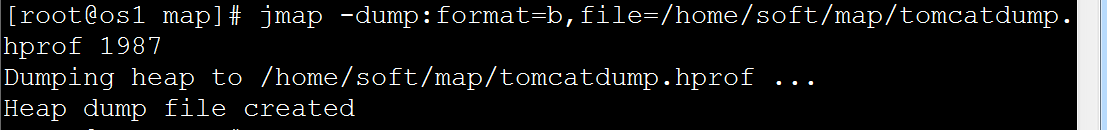
生成Java应用程序的堆快照和对象的统计信息

jmap -histo 2972 >c:\s.txt

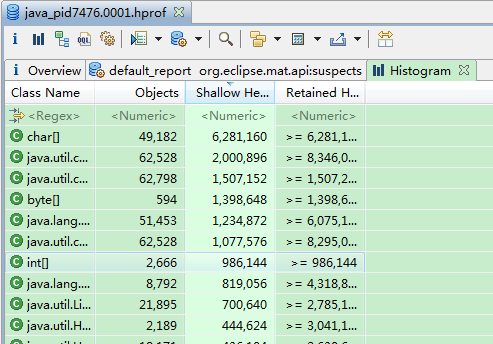


dump堆

jmap -dump:format=b,file=/home/soft/map/tomcatdump.hprof 1987



例如



## jstack

打印线程dump

-l 打印锁信息

-m 打印java和native的帧信息

-F 强制dump，当jstack没有响应时使用