# 23 Shell综合案例二

## 案例一：获取系统信息

### 案例分析

本案例目的是模拟Windows的计算机属性对话框显示的系统信息。

1. 用户信息

主机名：`hostname`

用户名：`id -un`

用户组名：`id -gn`

2. 操作系统信息

操作系统版本：`cat /etc/redhat-release`

内核版本和操作系统架构：`hostnamectl`

3. CPU信息

`cat /proc/cpuinfo`

4. 内存信息

`free -h`

5. 根目录空间信息

`df / -m`

6. 网络信息

`ip -a`

### 案例代码

Bash | 复制代码

```
[root@shell ~]# vi pro4_sysinfo.sh
#! /bin/bash
# Centos7 系统信息
#1.用户信息
hostname=$(hostname)
username=$(id -un)
usergroup=$(id -gn)
echo "主机名: ${hostname}"
echo "当前用户: ${username}"
echo "当前用户组名: ${usergroup}"

#2.操作系统
sys_version=$(cat /etc/redhat-release)
kernel=$(hostnamectl | grep 'Kernel' | awk -F: '{print $2}')
architecture=$(hostnamectl | grep 'Arch' | awk -F: '{print $2}')
echo "系统版本: ${sys_version}"
echo "内核版本: ${kernel}"
echo "系统架构: ${architecture}"

#3.CPU
#CPU型号
cpu_name=$(cat /proc/cpuinfo | awk -F: '/model name/ {print $2}'|uniq)
#逻辑CPU个数
cpu_logical=$(cat /proc/cpuinfo | grep "processor" | wc -l)
#物理CPU个数
cpu_physical=$(cat /proc/cpuinfo| grep "physical id" | sort  | uniq | wc -l)
#每颗物理CPU的核数
cpu_cores=$(cat /proc/cpuinfo | grep 'cores'  | wc -l)
echo "CPU型号:  ${cpu_name}"
echo "逻辑CPU个数: ${cpu_logical}"
echo "物理CPU个数: ${cpu_physical}"
echo "每颗物理CPU的核数: ${cpu_cores}"

#4. 内存
mem_total=$(free -h | awk 'NR==2{print $2}')
mem_free=$(free -h | awk 'NR==2{print $4}')
echo "总内存: ${mem_total}"
echo "剩余内存: ${mem_free}"

#5. SCSI设备
echo "服务器已挂载SCSI设备: "
lsscsi

#6. 根目录磁盘剩余空间
```

```
total_space=$(df / -m |grep /|awk '{print $2}')
free_space_m=$(df / -m |grep /|awk '{print $4}')
total_space_g=$(echo "scale=2;${total_space}/1024.0" | bc )
free_space_g=$(echo "scale=2;${free_space_m}/1024.0" | bc )
echo "根目录磁盘总空间${total_space_g}G"
echo "根目录磁盘空间剩余${free_space_g}G"

#7. 网络
links=$(ip a | sed -n '/^[0-9]/p'| sed -n '/\blo\b/!p'  |cut -d':' -f2 |sed 's/ //g')
echo "本机网卡列表为：${links}"
for i in ${links}
do
    ip=$(ip a show ${i} | grep 'inet\b' | cut -d' ' -f6| cut -d '/' -f1)
    mac=$(ip a show ${i}  | grep link/ | awk '{print $2}'| awk -F: '{print $1$2"-"$3$4"-"$5$6 }')
    echo -e "${i}: IP地址为${ip} MAC地址为${mac}"
done
[root@shell ~]# . pro4_sysinfo.sh
主机名：shell
当前用户：root
当前用户组名：root
系统版本：CentOS Linux release 7.9.2009 (Core)
内核版本：  Linux 3.10.0-1160.el7.x86_64
系统架构：  x86-64
CPU型号：   Intel(R) Core(TM) i5-9400 CPU @ 2.90GHz
逻辑CPU个数：1
物理CPU个数：1
每颗物理CPU的核数：1
总内存：1.8G
剩余内存：639M
服务器已挂载SCSI设备：
[0:0:0:0]    disk    VMware,  VMware Virtual S 1.0   /dev/sda
[1:0:0:0]    cd/dvd  NECVMWar VMware IDE CDR00 1.00  /dev/sr0
根目录磁盘总空间16.98G
根目录磁盘空间剩余14.36G
本机网卡列表为：ens33
ens33: IP地址为192.168.149.3 MAC地址为000c-291b-9e1b
```

# 案例二：MySQL信息统计

## 案例分析

在日常运维工作中，MySQL作为主流的关系数据库，使用场景非常广泛，监控MySQL状态信息是重要的运维工作。

```shell
# yum 安装mariadb-server mariadb
[root@Shell  ~]# yum -y install mariadb-server mariadb
# 启动mariadb
[root@Shell  ~]# systemctl start mariadb
# 使用空密码登入mysql控制台可以执行增删改查操作
[root@Shell ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> \q
Bye
# mysqladmin命令可以管理mysql，看mysql状态，也可以改密码
[root@Shell ~]# mysqladmin status
Uptime: 126  Threads: 1  Questions: 6  Slow queries: 0  Opens: 0  Flush tables: 2  Open tables: 26  Queries per second avg: 0.047
# 利用mysql -e执行SQL语句查询MySQL状态变量
[root@Shell ~]# mysql -e "SHOW GLOBAL STATUS LIKE 'uptime'"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| Uptime        | 9034  |
+---------------+-------+
[root@Shell ~]# mysql -e "SHOW GLOBAL STATUS LIKE 'uptime'"  | awk '/Uptime/{print $0}'
Uptime  9128
[root@Shell ~]# mysql -e "SHOW GLOBAL STATUS LIKE 'uptime'"  | awk '/Uptime/{print $2}'
9132
```

MySQL常用性能状态变量如表所示。

| 状态 | 含义 |
| --- | --- |
| com_commit | 数据库执行的提交指令个数 |
| com_delete | 数据库执行的删除指令个数 |
| com_insert | 数据库执行的插入指令个数 |

| | |
|---|---|
| com_rollback | 数据库执行回滚指令的个数 |
| com_select | 数据库执行的查询指令个数 |
| com_update | 数据库执行的更新指令个数 |
| slow_queries | 数据库慢查询语句的个数 |
| max_connections | 数据库的最大并发连接数 |
| Questions | 数据库执行的指令数量 |
| uptime | 数据库的运行时间 |

数据库常见的性能指标为：

- QPS：每秒执行指令数=数据库执行的指令数/数据库的运行时间
- TPS：每秒执行事务数=数据库提交的指令数+数据库回滚的指令数/数据库的运行时间

## 案例代码

Bash | 复制代码

```
[root@shell ~]# vi pro5_mysql.sh
#!/bin/bash

#定义数据库相关变量.
# MYSQL_USER=root
# MYSQL_PASS=root
# MYSQL_PORT=3306
# MYSQL_HOST=localhost
# MYSQL_ADMIN="mysqladmin -u$MYSQL_USER -p$MYSQL_PASS -P$MYSQL_PORT -h$MYSQL_HOST"
# MYSQL_COMM="mysql -u$MYSQL_USER -p$MYSQL_PASS -P$MYSQL_PORT -h$MYSQL_HOST -e"

MYSQL_ADMIN="mysqladmin "
MYSQL_COMM="mysql -e"

#定义变量:显示信息的颜色属性.
SUCCESS="echo -en \\033[1;32m"    #绿色
FAILURE="echo -en \\033[1;31m"    #红色
WARNING="echo -en \\033[1;33m"    #黄色
NORMAL="echo -en \\033[0;39m"     #黑色
#注意颜色设置是持久性的，如果需要改变颜色必须再次设置颜色

#检查数据库服务器状态.
$MYSQL_ADMIN ping &> /dev/null
if [ $? -ne 0 ];then
    $FAILURE
    echo "无法连接数据库服务器"
    $NORMAL
    exit
else
    echo -n "数据库状态: "
    $SUCCESS
    echo "[OK]"
    $NORMAL
fi

#过滤数据库启动时间
RUN_TIME=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'uptime'" | awk '/Uptime/{print $2}')
echo -n "数据库已运行时间(秒): "
$SUCCESS
echo $RUN_TIME
$NORMAL

```

```
#过滤数据库列表
DB_LIST=$($MYSQL_COMM "SHOW DATABASES")
DB_COUNT=$($MYSQL_COMM "SHOW DATABASES" | awk 'NR>=2&&/^[^+]/{db_count++} END{print db_count}')
echo -n "该数据库有$DB_COUNT个数据库,分别为:"
$SUCCESS
echo $DB_LIST
$NORMAL

#查询MySQL最大并发连接数
MAX_CON=$($MYSQL_COMM "SHOW VARIABLES LIKE 'max_connections'" | awk '/max/{print $2}')
echo -n "MySQL最大并发连接数: "
$SUCCESS
echo $MAX_CON
$NORMAL

#查看SELECT指令被执行的次数
NUM_SELECT=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_select'" | awk '/Com_select/{print $2}')
echo -n "SELECT被执行的次数: "
$SUCCESS
echo $NUM_SELECT
$NORMAL

#查看UPDATE指令被执行的次数
NUM_UPDATE=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_update'" | awk '/Com_update/{print $2}')
echo -n "UPDATE被执行的次数: "
$SUCCESS
echo $NUM_UPDATE
$NORMAL

#查看DELETE指令被执行的次数
NUM_DELETE=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_delete'" | awk '/Com_delete/{print $2}')
echo -n "DELETE被执行的次数: "
$SUCCESS
echo $NUM_DELETE
$NORMAL

#查看INSERT指令被执行的次数
NUM_INSERT=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_insert'" | awk '/Com_insert/{print $2}')
echo -n "INSERT被执行的次数: "
$SUCCESS
echo $NUM_INSERT
$NORMAL
```

```
85
86
87  #查看COMMIT指令被执行的次数
    NUM_COMMIT=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_commit'" | awk '/Com_commit/{print $2}')
88  echo -n "COMMIT被执行的次数: "
89  $SUCCESS
90  echo $NUM_COMMIT
91  $NORMAL
92
93
94  #查看ROLLBACK指令被执行的次数
    NUM_ROLLBACK=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'com_rollback'" | awk '/Com_rollback/{print $2}')
95  echo -n "ROLLBACK被执行的次数: "
96  $SUCCESS
97  echo $NUM_ROLLBACK
98  $NORMAL
99
100
101 #查看服务器执行的指令数量
    NUM_QUESTION=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'Questions'" | awk '/Questions/{print $2}')
102 echo -n "Questions服务器执行的指令数量: "
103 $SUCCESS
104 echo $NUM_QUESTION
105 $NORMAL
106
107 NUM_SLOW_QUERY=$($MYSQL_COMM "SHOW GLOBAL STATUS LIKE 'slow_queries'" | awk '/Slow_queries/{print $2}')
108 echo -n "SLOW Query慢查询数量: "
109 $SUCCESS
110 echo $NUM_SLOW_QUERY
111 $NORMAL
112
113
114 #QPS即每秒执行的指令数
115 echo -n "数据库QPS: "
116 $SUCCESS
117 awk 'BEGIN{print '"$NUM_QUESTION/$RUN_TIME"'}'
118 $NORMAL
119 #TPS即每秒执行的事务数
120 echo -n "数据库TPS: "
121 $SUCCESS
122 awk 'BEGIN{print '"($NUM_COMMIT+$NUM_ROLLBACK)/$RUN_TIME"'}'
123 $NORMAL
124 [root@shell ~]# . pro5_mysql.sh
125 数据库状态: [OK]
126 数据库已运行时间(秒): 55
    该数据库有4个数据库,分别为:Database information_schema mysql performance_schema test
127 MySQL最大并发连接数: 151
```

```
SELECT被执行的次数: 5
UPDATE被执行的次数: 0
DELETE被执行的次数: 0
INSERT被执行的次数: 0
COMMIT被执行的次数: 0
ROLLBACK被执行的次数: 0
Questions服务器执行的指令数量: 34
SLOW Query慢查询数量: 0
数据库QPS: 0.618182
数据库TPS: 0
```

## 小结

- 获取系统信息
- MySQL信息统计

## 课程目标

- 知识目标：了解shell脚本的分析设计方法。
- 技能目标：能够根据需求设计编写shell脚本。

## 课外拓展

- 了解更多实用Shell脚本的编写思路。

## 参考资料


- 《Linux Shell核心编程指南》，丁明一，电子工业出版社