Shell 脚本示例分析

内容提要

1. 学会阅读分析Shell脚本
2. 进一步熟悉各种命令行工具的使用
3. 进一步掌握正则表达式的使用
4. 进一步熟悉Shell编程的语法
5. 学会编写shell脚本解决实际问题

sysinfo 脚本分析

下面的脚本来自 http://wiki.splitbrain.org/sys2wiki.sh [http://www.proxyserve.net/index.php?q=aHR0cDovL3dpa2kuc3BsaXRicmFpbi5vcmcvc3lzMndpa2kuc2g%3D]，此脚本收集系统信息，并格式化为 DokuWiki [http://www.proxyserve.net/index.php?q=aHR0cDovL3dpa2kuc3BsaXRicmFpbi5vcmcvd2lraTpkb2t1d2lraQ%3D%3D] 格式的输出。

```
#!/bin/bash
# FROM http://wiki.splitbrain.org/sys2wiki.sh

# 为了使脚本正确运行，将语系设置为通用 C
LANG=C

# 指定不同输出部分以 H2 输出
HL='====='

# 将从 /proc/cpuinfo 获得的 CPU 型号信息存于变量 CPU 中
CPU=`cat /proc/cpuinfo  | grep 'model name' | awk -F\: '{print $2}'|uniq|sed -e 's/ //'`
# 将从 /proc/cpuinfo 获得的 CPU 频率信息存于变量 MHz 中
MHz=`cat /proc/cpuinfo | grep 'cpu MHz' | awk -F\: '{print $2}'|uniq|sed -e 's/ //'`
# 将从 /proc/cpuinfo 获得的 CPU 个数信息存于变量 CPUCOUNT 中
CPUCOUNT=`cat /proc/cpuinfo|grep "physical id"|uniq|wc -l`
# 将从 /proc/cpuinfo 获得的 CPU 核数信息存于变量 CPUKENELCOUNT 中
CPUKENELCOUNT=`cat /proc/cpuinfo|grep "processor"|uniq|wc -l`
# 将从 /proc/meminfo 获得的物理内存数量存于变量 RAM 中
RAM=`cat /proc/meminfo | grep MemTotal | awk -F\: '{print $2}' | awk -F\  '{print $1 " " $2}'`
# 将从 /proc/meminfo 获得的虚拟内存数量存于变量 SWAP 中
SWAP=`cat /proc/meminfo | grep SwapTotal | awk -F\: '{print $2}' | awk -F\  '{print $1 " " $2}'`
# 将操作系统类型信息存于变量 SYSTEM 中
SYSTEM=`uname -sr`
# 将计算机名存于变量 HOSTNAME 中
HOSTNAME=`(hostname -f || hostname) 2>/dev/null`

# 将操作系统发型版信息存于变量 OS 中
if [ -e /etc/debian_version ]; then
  OS="Debian `cat /etc/debian_version`"
elif [ -e /etc/redhat-release ]; then
  OS=`cat /etc/redhat-release`
elif [ -e /etc/SuSE-release ]; then
  OS=`cat /etc/SuSE-release |head -n1`
elif [ -e /etc/gentoo-release ]; then
  OS=`< /etc/gentoo-release`
else
  OS='unknown'
fi

# 打印常规信息
echo "

# 打印常规信息标题
$HL General $HL

# 以表格形式打印上面收集的信息变量
^ Hostname | $HOSTNAME |
^ CPU      | $CPU      |
^ MHz      | $MHz      |
^ # CPU    | $CPUCOUNT |
^ # CPUKENEL | $CPUKENELCOUNT |
^ RAM      | $RAM      |
^ Swap     | $SWAP     |
^ System   | $SYSTEM   |
^ OS       | $OS       |
"

# 打印网络信息标题
echo -e "$HL Network $HL\n"
```

```
# 以 for 循环用表格形式打印各个网络接口及 IP
for DEV in `/sbin/ifconfig -a |grep '^\w'|awk '!/lo/{print $1}'`
do
  IP=`/sbin/ifconfig $DEV |awk -F\: '/inet / {print $2}'|awk '{print $1}'`
  echo "` $DEV | $IP |"
done
# 打印空行
echo

# 打印PCI设备信息标题
echo -e "$HL PCI $HL\n"
# 将 lspci 命令输出的每一行前加两个空格，即以CODE方式显示 lspci 命令的输出
lspci |sed 's/^/  /'
# 打印空行
echo

# 打印文件系统信息标题
echo -e "$HL Filesystems $HL\n"
# 以表格方式打印除了 tmpfs 之外所有文件系统的信息
# （包括： Filesystem、Type、Size、Mounted 四项信息）
df -hPT -x tmpfs | awk '{print "| " $1 " | " $2 " | " $3 " | " $7 " |"}'
echo

# 打印IDE设备信息标题
echo -e "$HL IDE devices $HL\n"

# 以 for 循环用表格形式输出每一个 IDE 设备的信息
for DEV in `ls -1d /proc/ide/hd* |sed 's/.*\///'`
do
  MODEL=`cat /proc/ide/$DEV/model`
  # 由于不同的Linux发布系统记录设备大小的文件不同
  # 因此使用下面的 if 嵌套语句进行判断
  if [ -e /proc/ide/$DEV/capacity ]; then
    SIZE=`cat /proc/ide/$DEV/capacity`
    SIZE=`expr $SIZE / 2097152`
  else
    if [ -e /sys/block/$DEV/size ]; then
      SIZE=`cat /sys/block/$DEV/size`
      SIZE=`expr $SIZE / 2097152`
    else
      SIZE='(unknown)'
    fi
  fi

  echo "| /dev/$DEV | $MODEL | $SIZE GB |"
done

# 当 /sys/block/sd* 存在时
if [ "$(ls -1d /sys/block/sd* 2> /dev/null)" ]; then
 # 打印SCSI设备信息标题
  echo -e "$HL SCSI devices $HL\n"
  # for 循环用表格形式输出每一个 SCSI 设备的信息
  for DEV in `ls -1d /sys/block/sd* |sed 's/.*\///'`
  do
    MODEL=`cat /sys/block/$DEV/device/model`
    SIZE=`cat /sys/block/$DEV/size`
    SIZE=`expr $SIZE / 2097152`

    echo "| /dev/$DEV | $MODEL | $SIZE GB |"
  done
  echo
fi
```

上述脚本使用的 sed 和 awk 命令请参见 sed 和 awk。

上述脚本的输出结果在 dokuwiki 上的显示结果，参见如下两个实例：

- sysinfo
- sysinfo

参考

其他 sysinfo 脚本：

- http://www.talug.org/howto/sysinfo.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy50YWx1Zy5vcmcvaG93dG8vc3lzaW5mby5odG1s]
- http://www.novell.com/coolsolutions/tools/16447.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5ub3ZlbGwuY29tL2Nvb2xzb2x1dGlvbnMvdG9vbHMvMTY0NDcuaHRtbA%3D%3D]
- http://www.sun.com/bigadmin/jsp/descFile.jsp?url=descAll/system_info_sysinfo [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5zdW4uY29tL2JpZ2FkbWluL2pzcC9kZXNjRmlsZS5qc3A%2FdXJsPWRlc2NBbGwvc3lzdGVtX2luZ%3D%3D]
- http://phpsysinfo.sourceforge.net/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3BocHN5c2luZm8uc291cmNlZm9yZ2UubmV0Lw%3D%3D]

init 脚本分析

系统的 init 脚本都是 Shell 脚本，这些脚本存在 /etc/init.d 目录下。这些脚本具有相同的结构，下面以 CentOS 系统下的 /etc/init.d /yum-updatesd 为例进行分析。

```
#!/bin/bash
# 依Shell 脚本的编码规范书写脚本说明
# yum           This shell script enables the yum-updates daemon
#
# Author:       Jeremy Katz <katzj@redhat.com>
#
# chkconfig:    345 97 03
#
# description:  This is a daemon which periodically checks for updates \
#               and can send notifications via mail, dbus or syslog.
# processname:  yum-updatesd
# config: /etc/yum/yum-updatesd.conf
# pidfile: /var/run/yum-updatesd.pid
#

# 在当前 Shell 中运行函数库文件 /etc/rc.d/init.d/functions
. /etc/rc.d/init.d/functions

# 设置脚本返回值变量
RETVAL=0

# 定义 start 函数
start() {
        echo -n $"Starting yum-updatesd: "
        # 用 /etc/rc.d/init.d/functions 中的函数 daemon 调用守护进程 yum-updatesd
        daemon yum-updatesd
        # 将函数 daemon 的返回值赋予变量 RETVAL
        RETVAL=$?
        echo
        # 如果函数 daemon 执行成功，生成锁定文件 /var/lock/subsys/yum-updatesd
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/yum-updatesd
}

# 定义 stop 函数
stop() {
        echo -n $"Stopping yum-updatesd: "
        # 用 /etc/rc.d/init.d/functions 中的函数 killproc 杀死守护进程 yum-updatesd
        killproc yum-updatesd
        echo
        # 如果函数 killproc 执行成功，删除锁定文件 /var/lock/subsys/yum-updatesd
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/yum-updatesd
}

# 定义 restart 函数
restart() {
        stop
        start
}

# 根据调用本脚本的第一个位置参数的值执行不同的操作
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|force-reload|reload)
        restart
        ;;
  condrestart)
        # 若服务已经启动（锁定文件 /var/lock/subsys/yum-updatesd存在）则重新启动
        [ -f /var/lock/subsys/yum-updatesd ] && restart
        ;;
  status)
        # 用 /etc/rc.d/init.d/functions 中的函数 status 查看守护进程 yum-updatesd 的状态
        status yum-updatesd
        # 将函数 status 的返回值赋予变量 RETVAL
        RETVAL=$?
        ;;
  *)
        # 对于$1的其他值显示用法
        echo $"Usage: $0 {start|stop|status|restart|reload|force-reload|condrestart}"
        # 退出本脚本，并以 1 为该脚本的返回值
        exit 1
esac

# 退出本脚本，并以变量 RETVAL 的值为该脚本的返回值
exit $RETVAL
```

参考

- A quick guide to writing scripts using the bash shell [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5wYW5peC5jb20vfmVsZmxvcmQvdW5peC9iYXNoLXR1dGUuaHRtbA%3D%3D]
- Bash Guide for Beginners [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy50bGRwLm9yZy9MRFAvQmFzaC1CZWdpbm5lcnMtR3VpZGUvaHRtbC9pbmRleC5odG1s]
  - Bash Guide for Beginners 国内镜像 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5nZWNrLmNuL3RsZHAvTERQL0Jhc2gtQmVnaW5uZXJzLUd1aWRlL2h0bWwvaW5kZXguaHRtb%3D%3D]
  - Bash新手指南 [http://www.proxyserve.net/index.php?q=aHR0cDovL3hpYW93YW5nLm5ldC9iZ2ItY24vaW5kZXguaHRtbA%3D%3D]
- http://www.linuxcommand.org/writing_shell_scripts.php [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5saW51eGNvbW1hbmQub3JnL3dyaXRpbmdfc2hlbGxfc2NyaXB0cy5waHA%3D]
- Advanced Bash-Scripting Guide [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy50bGRwLm9yZy9MRFAvYWJzL2h0bWw]
  - Advanced Bash-Scripting Guide 国内镜像 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5nZWNrLmNuL3RsZHAvTERQL2Ficy9odG1sL2luZGV4Lmh0bWw%3D]
  - Advanced Bash-Scripting Guide 3.9.1 （包括中译本） [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5saW51eHNpci5vcmcvbWFpbi8%2FcT1ub2RlLzE0MA%3D%3D] -- 下载版
  - http://www.linuxpk.com/doc/abs/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5saW51eHBrLmNvbS9kb2MvYWJzLw%3D%3D] -- 在线版
- http://www.ibm.com/developerworks/cn/linux/shell/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5pYm0uY29tL2RldmVsb3BlcndvcmtzL2NuL2xpbnV4L3NoZWxsLw%3D%3D]
- http://www.ccidnet.com/images/tech/linux/zhuanti/shell/index.htm [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jY2lkbmV0LmNvbS9pbWFnZXMvdGVjaC9saW51eC96aHVhbnRpL3NoZWxsL2luZGV4Lmh0bQ%3D%3D]
- http://net.pku.edu.cn/~yhf/shellProgramIntro.htm [http://www.proxyserve.net/index.php?q=aHR0cDovL25ldC5wa3UuZWR1LmNuL355aGYvc2hlbGxQcm9ncmFtSW50cm8uaHRt]
- 一个bash script的简单例子(if case的使用) [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5md29sZi5jb20vYmxvZy9wb3N0LzE3MQ%3D%3D]
- BASH with Debugger — http://bashdb.sourceforge.net/ [http://www.proxyserve.net/index.php?q=aHR0cDovL2Jhc2hkYi5zb3VyY2Vmb3JnZS5uZXQv]
- NanoBlogger (a small weblog engine written in Bash for the command line) — http://nanoblogger.sourceforge.net/ [http://www.proxyserve.net/index.php?q=aHR0cDovL25hbm9ibG9nZ2VyLnNvdXJjZWZvcmdlLm5ldC8%3D]
- http://www.oreilly.com/catalog/9780596526788/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5vcmVpbGx5LmNvbS9jYXRhbG9nLzk3ODA1OTY1MjY3ODgv]
- http://bashscripts.org/ [http://www.proxyserve.net/index.php?q=aHR0cDovL2Jhc2hzY3JpcHRzLm9yZy8%3D]
- 显示源文件
- 登录