一．MEN

需求：在树莓派上，使用Shell编程，输出以下内存信息：

时间 总量 剩余量 当前占用（%） 占用百分比动态平均值

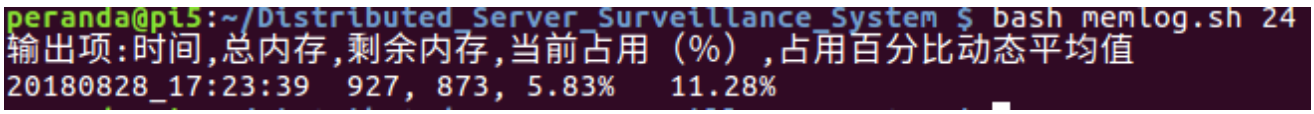
代码：

```
#!/bin/bash
echo "输出项:时间,总内存,剩余内存,当前占用（%）,占用百分比动态平均值 "
echo -n "`date +%Y%m%d_%H:%M:%S` "
ans=`free -m | grep "^Mem:" | awk -v s="%" '{printf " %d, %d, %.2f%s", $2,  $2-$3,
 $3/$2*100,s }' `
echo -n "$ans  "
ans=${1},${ans}
echo $ans | awk -v s="%" 'BEGIN{FS=","} {printf " %.2f%s\n", $1 * 0.3 + 0.7 * $4 ,s}'
```

总结:

1. 学会了awk的Ｆ Ｓ的用法。
2. 这代不是最初的，最初的很长，学到了不少东西

结果：



二．用户统计

需求：时间 用户总数（非系统用户）近期活跃用户（3个） 具有ROOT权限用户 当前在线用户_登录IP_TTY

代码：

```
#!/bin/bash
```

```
echo "time  users  user(3)  user(root)  xxx"
time_key=`date +%Y%m%d-%H:%M:%S `
echo -n "[$time_key]  "
ans=`last | grep [a-zA-Z0-9] | grep -v "wtmp" | grep -v  "boot" | wc -l`
echo -n "[$ans] "
key=`last | grep [a-zA-Z0-9] | grep -v "wtmp" | grep -v  "boot" | cut -d " "  -f  1 |
sort | uniq -c |  sort -n -r | xargs `
key1=`echo ${key} | cut -d " " -f 2`
key2=`echo ${key} | cut -d " " -f 4`
key3=`echo ${key} | cut -d " " -f 6`
echo -n "[$key1,$key2,$key3] "

echo -n "[`cat /etc/group | grep "sudo" | cut -d ":" -f 4 `] "

echo -n "["
w -h | awk '{printf"%s_%s_%s\n", $1,$3,$2}' |awk '{print $1}' | xargs | tr " " ","
echo  "]"
```

总结：

1．精简了三个活跃用户的代码

2．对在线用户_IP_tty用ＡＷＫ实现了，并且比较早期暴力做法更为精简

结果：

```
peranda@pts:~/Distributed_Server_Surveillance_System $ bash Users.sh
time  users  user(3)  user(root)  xxx
[20180828-21:10:14]  [174] [peranda,james,lkrkrl] [pi] [peranda_127.0.0.1_pts/0
]
```

三．DISK

磁盘总量

磁盘剩余量

占用比例

各分区占用比例

各分区总量

分区剩余情况

例子 ： 2018-01-12_16:48:23 标识（0为整个磁盘，1为分区） 磁盘还是分区（disk| /boot , /） 磁盘/分区总量 磁盘/分区剩余量 占用比

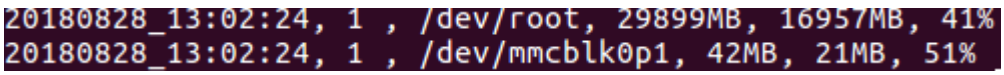
代码：

```
#!/bin/bash

#0分区  1磁盘
ans=`date +%Y%m%d_%H:%M:%S`

df -T -m -x "tmpfs" | grep "disk"  | tail -n +2  | awk -v anstime=$ans '{ printf("%s, 0 ,%s, %dMB, %dMB, %s \n", anstime, $1, $3, $5, $6) }'
df -T -m -x "tmpfs" | grep -v "tmpfs"  | tail -n +2  | awk -v anstime=$ans '{ printf("%s, 1 , %s, %dMB, %dMB, %s \n", anstime, $1, $3, $5, $6)}'
```

总结：

1.学会了在awk里插入一个变量，并且由脚本外部输入的变量无法传入w a k 里

结果：



四．CPU信息获取

需求：时间 负载1（1分钟） 负载2（5分钟） 负载3（15分钟） 占用率 （时间间隔0.5） 当前温度 警告（normal，note（50-70），warning（70~））

代码：

```
#!/bin/bash

echo -n "`date +%Y%m%d_%H:%M:%S` " #时间
echo -n "`cat /proc/loadavg | cut -d " " -f 1-3` " #cpu平均负载

###  cpu占用率(间隔0.5s)
time1=`cat /proc/stat | head -n 1 | awk '{print $5; print $2 + $3 + $4 + $5 + $6 + $7
+ $8 + $9 + $10 + $11 }' `
sleep 0.5s
time2=`cat /proc/stat | head -n 1 | awk '{print $5; print $2 + $3 + $4 + $5 + $6 + $7
+ $8 + $9 + $10 + $11 }' `
key1=$[`echo $time2 | cut -d " " -f 1` - `echo $time1 | cut -d " " -f 1`]
key2=$[`echo $time2 | cut -d " " -f 2` - `echo $time1 | cut -d " " -f 2`]
echo -n `echo "scale=2; (1 - ${key1}/${key2}) * 100" | bc `
echo -n "% "
###

### cpu温度
cpu_key=`cat /sys/devices/virtual/thermal/thermal_zone0/temp`
cpu_size=`echo "ibase=10;($cpu_key/1000)" | bc`
echo -n "${cpu_size}°C "
###判断cpu温度级别
echo ${cpu_size} | awk 'END{if($1 < 50) {print "normal"} else if ($i < 70) {"note"}
else {"warning"}}'
```

总结：

1. 对bc这个命令有了更深的认识，它里面的scale=2;代表了浮点型留两位小数计算,ibase=10;是输出十进制的数
2. 对awkBEGIN,END有了一个新的认识

结果：

```
peranda@pi5:~/Distributed_Server_Surveillance_System $ bash cpulog.sh
20180828_18:39:48 0.00 0.00 0.00 1.00% 54°C note
```

五．系统运行概况

需求：时间 主机名 OS版本 内核版本 运行时间 平均负载 磁盘总量 磁盘已用% 内存大小 内存已用% CPU温度 磁盘报警级别 内存报警级别 CPU报警级别

代码：

```
#!/bin/bash

echo "time    hostname    OS    xxxx    runtime    平均负载    磁盘总量    磁盘所用百分比    内存大
小    内存已用百分比    ＣＰＵ温度    磁盘警报级别    内存警报级别    ＣＰＵ警报级别"
```

```
echo -n "`date +%Y%m%d_%H:%M:%S`  " #时间
echo -n "`hostname` " # 主机名字
echo -n "`uname -o` " # os版本
echo -n "`cat /proc/version | cut -d " " -f 3 `  " #内核版本
echo -n "`uptime -p | tr " " "_"` " #运行时间
echo -n " `cat /proc/loadavg | cut -d " " -f 1-3` " #平均负载

###########磁盘总量（所有的量）
ans=`grep MemTotal /proc/meminfo | xargs | cut -d " " -f 2` # MemTotal里面找了磁盘内存总量
echo -n "${ans}MB " #输出磁盘内存总量
ans1=`grep MemFree /proc/meminfo | xargs | cut -d " " -f 2` # MemFree 里面找了空闲的内存量
echo -n "${ans1}MB " #输出磁盘空闲内存量
ans_add=`echo "ibase=10; (${ans}-${ans1})*10/${ans}*10" | bc` #计算内存已用的百分比
echo -n "${ans_add}% " #输出磁盘内存已用百分比
################################

################################
df -m | grep "^/dev" | grep -v "/boot" | awk '{printf "%sMB %sMB %s ",$2,$3,$5}'
#输出系统内存大小  系统内存空闲  内存已用百分比

#################计算ＣＰＵ的温度
cpu_key=`cat /sys/devices/virtual/thermal/thermal_zone0/temp`
cpu_size=`echo "ibase=10;($cpu_key/1000)" | bc`
echo -n "${cpu_size}°C "
#############################

######################
df_size=`df -m | grep "^/dev" | grep -v "/boot" | xargs | cut -d " " -f 5 |  cut -d "%" -f 1`

#echo $ans_add
#echo $df_size
#echo $cpu_size

if [[ ${ans_add} < 80 ]]; then
    echo -n "normal ";
elif [[ ${ans_add} < 90 ]]; then
    echo -n "note ";
else
    echo -n "waring ";
fi

if [[ ${df_size} < 50 ]]; then
    echo -n "normal ";
elif [[ ${df_size} < 70 ]]; then
    echo -n "note ";
else
    echo -n "waring ";
fi

if [[ ${cpu_size} < 70 ]]; then
```

```
    echo  "normal ";
elif [[ ${cpu_size} < 80 ]]; then
    echo  "note ";
else
    echo  "waring ";
fi
```

总结：

1．首先这是更”深刻“的知道了linux的”万可文件化“是什么意思了，一些物理的参数居然都能动态的保存记录在相应的文件里

2．对最近的一些命令有了一次统筹和运用，对一些细枝末节操作有了更多的理解

3．应该还能继续优化（如果想得到）

结果：

```
time    hostname    OS    xxxx    runtime  平均负载    磁盘总量    磁盘所用百分比
  内存大小    内存已用百分比   CPU温度      磁盘警报级别    内存警报级别    C
PU警报级别
20180828_13:09:18  pi5 GNU/Linux 4.9.41-v7+  up_3_hours,_9_minutes  0.09 0.07 0.
01 949572MB 730956MB 20% 29899MB 11696MB 41% 49°C normal normal normal
```