

# Shell Script : Find Linux Memory Usage Of Application/Program

Memstat.sh is a shell script that calculates linux memory usage for each program / application. Script outputs shared and private memory for each program in linux. Since memory calculation is bit complex, this shell script tries best to find more accurate results. Script use 2 files ie /proc/<PID>/status (to get name of process) and /proc/<PID>/smaps for memory statistic of process. Then script will convert all data into Kb, Mb, Gb. Also make sure you install bc command.

Source : <http://linuxide.com/shell-script-2/linux-memory-usage-program>

## Memstat.sh Shell Script

```
#!/bin/bash
# Make sure only root can run our script

if [ "$(id -u)" != "0" ]; then
echo "This script must be run as root" 1>&2
exit 1
fi

### Functions
#This function will count memory statistic for passed PID
get_process_mem ()
{
PID=$1
#we need to check if 2 files exist
if [ -f /proc/$PID/status ];
then
if [ -f /proc/$PID/smaps ];
then
#here we count memory usage, Pss, Private and Shared = Pss-Private
Pss=`cat /proc/$PID/smaps | grep -e "^Pss:" | awk '{print $2}' | paste -sd+ | bc `
Private=`cat /proc/$PID/smaps | grep -e "^Private" | awk '{print $2}' | paste -sd+ | bc `
#we need to be sure that we count Pss and Private memory, to avoid errors
if [ `x"$Pss" != "x" -o x"$Private" != "x" `];
```

then

```
let Shared=${Pss}-${Private}
Name=`cat /proc/$PID/status | grep -e "^Name:" |cut -d':' -f2`
#we keep all results in bytes
let Shared=${Shared}*1024
let Private=${Private}*1024
let Sum=${Shared}+${Private}
```

```
echo -e "$Private + $Shared = $Sum \t $Name"
fi
fi
fi
}
```

#this function make conversion from bytes to Kb or Mb or Gb

convert()

```
{
value=$1
power=0
#if value 0, we make it like 0.00
if [ "$value" = "0" ];
then
value="0.00"
fi
```

#We make conversion till value bigger than 1024, and if yes we divide by 1024

```
while [ $(echo "${value} > 1024" |bc) -eq 1 ]
do
value=$(echo "scale=2;${value}/1024" |bc)
let power=power+1
done
```

#this part get b, kb, mb or gb according to number of divisions

```
case $power in
0) reg=b;;
1) reg=kb;;
2) reg=mb;;
```

```
3) reg=gb;;
esac
```

```
echo -n "${value} ${reg} "
}
```

```
#to ensure that temp files not exist
[[ -f /tmp/res ]] && rm -f /tmp/res
[[ -f /tmp/res2 ]] && rm -f /tmp/res2
[[ -f /tmp/res3 ]] && rm -f /tmp/res3
```

```
#if argument passed script will show statistic only for that pid, of not - we list all processes in /proc/ #and get statistic for all of them, all result we store in file /tmp/res
```

```
if [ $# -eq 0 ]
then
pids=`ls /proc | grep -e [0-9] | grep -v [A-Za-z]`
for i in $pids
do
get_process_mem $i >> /tmp/res
done
else
get_process_mem $1>> /tmp/res
fi
```

```
#This will sort result by memory usage
cat /tmp/res | sort -gr -k 5 > /tmp/res2
```

```
#this part will get uniq names from process list, and we will add all lines with same process list
```

```
#we will count number of processes with same name, so if more that 1 process where will be
```

```
# process(2) in output
```

```
for Name in `cat /tmp/res2 | awk '{print $6}' | sort | uniq`
```

```
do
```

```
count=`cat /tmp/res2 | awk -v src=$Name '{if ($6==src) {print $6}}|wc -l| awk '{print $1}'`
```

```
if [ $count = "1" ];
```

```
then
```

```
count=""
```

```

else
count="(${count})"
fi

VmSizeKB=`cat /tmp/res2 | awk -v src=$Name '{if ($6==src) {print $1}}' | paste -sd+ | bc`
VmRssKB=`cat /tmp/res2 | awk -v src=$Name '{if ($6==src) {print $3}}' | paste -sd+ | bc`
total=`cat /tmp/res2 | awk '{print $5}' | paste -sd+ | bc`
Sum=`echo "${VmRssKB}+${VmSizeKB}"|bc`
#all result stored in /tmp/res3 file
echo -e "$VmSizeKB + $VmRssKB = $Sum \t ${Name}${count}" >>/tmp/res3
done

#this make sort once more.
cat /tmp/res3 | sort -gr -k 5 | uniq > /tmp/res

#now we print result , first header
echo -e "Private \t + \t Shared \t = \t RAM used \t Program"
#after we read line by line of temp file
while read line
do
echo $line | while read a b c d e f
do
#we print all processes if Ram used if not 0
if [ $e != "0" ]; then
#here we use function that make conversion
echo -en "convert $a` \t $b \t `convert $c` \t $d \t `convert $e` \t $f"
echo ""
fi
done
done < /tmp/res

#this part print footer, with counted Ram usage
echo "-----"
echo -e "\t\t\t\t\t\t\t `convert $total`"
echo "=====

# we clean temporary file

```

```
[[ -f /tmp/res ]] && rm -f /tmp/res  
[[ -f /tmp/res2 ]] && rm -f /tmp/res2  
[[ -f /tmp/res3 ]] && rm -f /tmp/res3
```

## Memstat.sh Shell Script Output

```
[root@centos-cluster-node1 ~]# ./memstat.sh  
Private + Shared = RAM used Program  
36.26 mb + 268.00 kb = 36.52 mb python  
20.49 mb + 238.00 kb = 20.72 mb iscsiuiio  
4.78 mb + 451.00 kb = 5.22 mb rgmanager(2)  
3.62 mb + 283.00 kb = 3.90 mb NetworkManager  
2.53 mb + 1.36 mb = 3.89 mb sshd(3)  
2.30 mb + 355.00 kb = 2.64 mb multipathd  
2.25 mb + 176.00 kb = 2.42 mb hald  
1.69 mb + 298.00 kb = 1.98 mb iscsid(2)  
1.45 mb + 432.00 kb = 1.87 mb dhclient(3)  
1.62 mb + 161.00 kb = 1.77 mb cupsd  
704.00 kb + 819.00 kb = 1.48 mb udevd(3)  
856.00 kb + 554.00 kb = 1.37 mb bash(2)  
1.00 mb + 314.00 kb = 1.31 mb qmgr  
984.00 kb + 314.00 kb = 1.26 mb pickup  
976.00 kb + 316.00 kb = 1.26 mb master  
1.07 mb + 21.00 kb = 1.09 mb rsyslogd  
804.00 kb + 240.00 kb = 1.01 mb modem-manager  
904.00 kb + 40.00 kb = 944.00 kb pcscd  
804.00 kb + 33.00 kb = 837.00 kb ricci  
788.00 kb + 38.00 kb = 826.00 kb dbus-daemon  
660.00 kb + 32.00 kb = 692.00 kb crond  
536.00 kb + 69.00 kb = 605.00 kb rpc.statd  
528.00 kb + 46.00 kb = 574.00 kb init  
216.00 kb + 357.00 kb = 573.00 kb saslauthd(5)  
544.00 kb + 21.00 kb = 565.00 kb wpa_supplicant  
484.00 kb + 72.00 kb = 556.00 kb mingetty(6)
```

```
316.00 kb + 58.00 kb = 374.00 kb rpcbind
116.00 kb + 237.00 kb = 353.00 kb memstat.sh
328.00 kb + 13.00 kb = 341.00 kb auditd
248.00 kb + 84.00 kb = 332.00 kb hald-runner
312.00 kb + 8.00 kb = 320.00 kb oddjobd
216.00 kb + 81.00 kb = 297.00 kb hald-addon-stor
196.00 kb + 88.00 kb = 284.00 kb hald-addon-inpu
272.00 kb + 4.00 kb = 276.00 kb rpc.idmapd
176.00 kb + 52.00 kb = 228.00 kb hald-addon-acpi
```

```
-----
98.57 mb
```

```
=====
[root@centos-cluster-node1 ~]#
```