

Universidad Autónoma de Nicaragua

UNAN-León

Facultad de ciencias y tecnología

Departamento de computación

Ingeniería telemática

10 de may. de 21



Docente:

MSc. Wilmer Matamoros

Componente:

Gestión de red

Practica 3 Agente SNMP

Elaborado por:

Br. Johnny Jesús Andino González

15-15530-7

Johnnyandino.15@est.unanleon.edu.ni



Contenido

Agente SNMP	1
ANEXO	1
Enlace de video demostrativo.....	1
Recursos disponibles en GitHub.....	1
MIB UNAN	1
memoryUsage.c	6
cpuUsage.c	11
Makefile.....	19



Figuras

Instalación de paquetes necesarios para la práctica.	1
Creación de directorios necesarios para la practica	2
Creacion y configuracion del fichero donde estara la MIB (MIB completa en anexo)	3
Comando para generar codigo de cpuUsage y memoryUsage	4
Resultado de la ejecución del comando anterior(Excluyendo los .so).....	5
Configuración del fichero makenfile 1 para generar los .so	6
Configuración de makenfile 2	7
Fin de configuración de makenfile	8
Comando para ejecutar los ficheros .c alojados en el directorio src para generar los .so	9
.so generados con el comando anterior, copiado de estos ficheros al directorio lib	10
verificamos que se hayan copiado correctamente	11
Configuración de snmp.conf	12
Configuración de las comunidades en snmpd.conf	13
Prueba de funcionamiento de la mib unan con snmpwalk.....	14
Prueba de funcionamiento usando snmpwalk y snmpgetnext en mib cpuUsage	15
Prueba de funcionamiento de mib memoryUsage con snmpwalk y snmpgetnext	16
Comparación de resultados obtenidos al ejecutar el comando proc/stat y los resultados con la mib cpuUsage	17
Comparación del comando top y mib memoryUsage 1.....	18
Comparación del comando top y mib memoryUsage 2.....	19



Agente SNMP

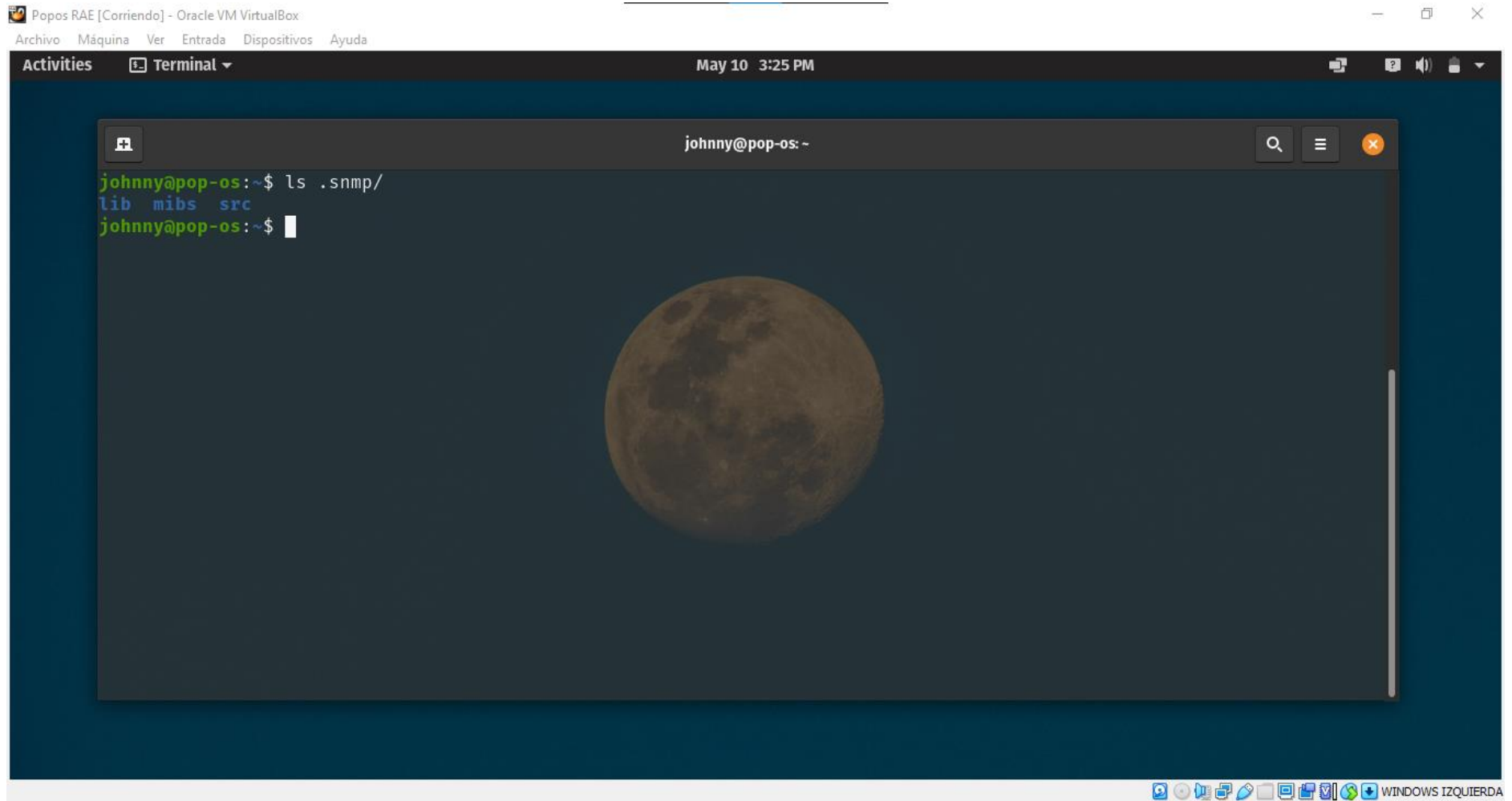
Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:23 PM

```
johnny@pop-os: ~$ sudo apt-get install snmp snmpd snmp-mibs-downloader libsnmp-dev libsnmp-perl make gcc
[sudo] password for johnny:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:10.2.0-1ubuntu1).
libsnmp-dev is already the newest version (5.8+dfsg-5ubuntu1).
make is already the newest version (4.3-4ubuntu1).
snmp is already the newest version (5.8+dfsg-5ubuntu1).
snmpd is already the newest version (5.8+dfsg-5ubuntu1).
libsnmp-perl is already the newest version (5.8+dfsg-5ubuntu1).
snmp-mibs-downloader is already the newest version (1.5).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
johnny@pop-os: ~$
```

1 Instalación de paquetes necesarios para la práctica.



2 Creación de directorios necesarios para la practica



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:26 PM

```
johnny@pop-os: ~$ ls .snmp/  
lib mibs src  
johnny@pop-os: ~$ ls .snmp/mibs/  
UNAN-MIB.txt  
johnny@pop-os: ~$
```

johnny@pop-os: ~

WINDOWS IZQUIERDA

3 Creacion y configuracion del fichero donde estara la MIB (MIB completa en anexo)




Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:27 PM

```
johnny@pop-os: ~/.snmp/src
johnny@pop-os:~$ cd .snmp/src/
johnny@pop-os:~/.snmp/src$ mib2c -c mib2c.scalar.conf cpuUsage memoryUsage
```



4 Comando para generar código de `cpuUsage` y `memoryUsage`




Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:28 PM

```
johnny@pop-os: ~/.snmp/src
johnny@pop-os:~$ cd .snmp/src/
johnny@pop-os:~/.snmp/src$ ls
cpuUsage.c  cpuUsage.h  cpuUsage.so  makefile  memoryUsage.c  memoryUsage.h  memoryUsage.so
johnny@pop-os:~/.snmp/src$
```



5 Resultado de la ejecución del comando anterior(Excluyendo los .so)



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:30 PM

johnny@pop-os: ~/.snmp/src

```
GNU nano 5.2 makefile
#
# Makefile to generate sysInfoGroup.so
#
# usage:
# "make" : generate library
# "make clean" : remove *.o , *.so
#

CC=gcc
LDFLAGS= `net-snmp-config --cflags`

#PROG= sysInfoGroup.so
#SRCS= sysInfoGroup.c
PROG= cpuUsage.so
SRCS= cpuUsage.c
OBJS= $(SRCS:.c=.o)

PROG2= memoryUsage.so
SRCS2= memoryUsage.c
OBJS2= $(SRCS2:.c=.o)

# shared library flags (assumes gcc)
DLFLAGS=-fPIC -shared -Wno-unused-result

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

WINDOWS IZQUIERDA

6 Configuración del fichero makenfile 1 para generar los .so



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:31 PM

johnny@pop-os: ~/.snmp/src

```
GNU nano 5.2 makefile
PROG2= memoryUsage.so
SRCS2= memoryUsage.c
OBS2= $(SRCS2:.c=.o)

# shared library flags (assumes gcc)
DLFLAGS=-fPIC -shared -Wno-unused-result

all: $(PROG) $(PROG2)

$(PROG): $(OBS)
$(CC) -o $@ $(OBS) $(LDFLAGS) $(DLFLAGS)
rm -f $(OBS)

$(PROG2): $(OBS2)
$(CC) -o $@ $(OBS2) $(LDFLAGS) $(DLFLAGS)
rm -f $(OBS2)

.c.o:
@if test "$(CC)" = "" ; then \
echo "La variable de entorno CC debe establecerse con la ruta del compilador" ; \
exit 1 ; \
fi
$(CC) -o $@ -c $< -g $(LDFLAGS) $(DLFLAGS)
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

WINDOWS IZQUIERDA

7 Configuración de makenfile 2



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:32 PM

johnny@pop-os: ~/.snmp/src

```
GNU nano 5.2 makefile
DLFLAGS=-fPIC -shared -Wno-unused-result

all: $(PROG) $(PROG2)

$(PROG): $(OBJS)
$(CC) -o $@ $(OBJS) $(LDFLAGS) $(DLFLAGS)
rm -f $(OBJS)

$(PROG2): $(OBJS2)
$(CC) -o $@ $(OBJS2) $(LDFLAGS) $(DLFLAGS)
rm -f $(OBJS2)

.c.o:
@if test "$(CC)" = "" ; then \
echo "La variable de entorno CC debe establecerse con la ruta del compilador" ; \
exit 1 ; \
fi
$(CC) -o $@ -c $< -g $(LDFLAGS) $(DLFLAGS)

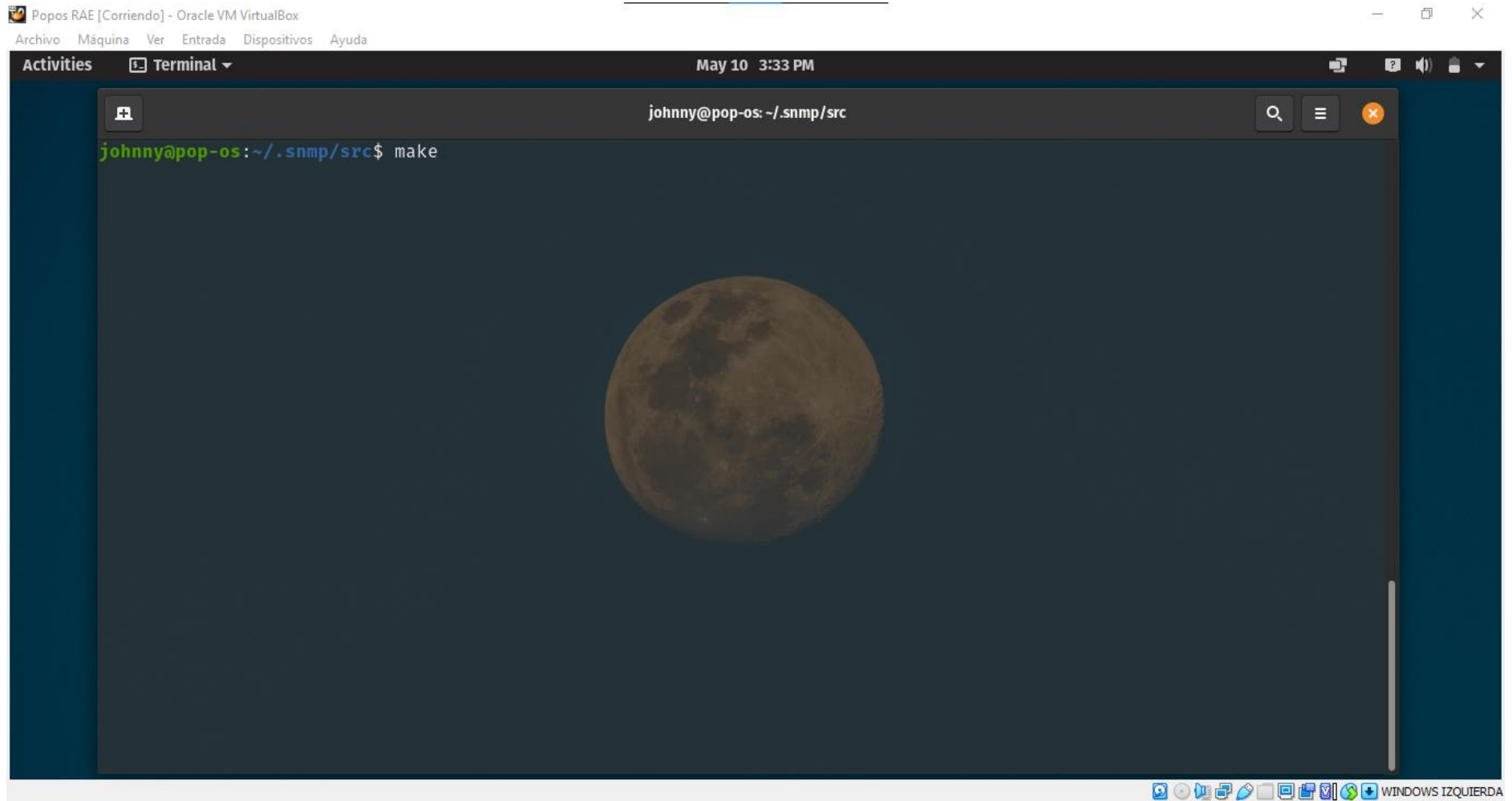
clean:
rm -f $(OBJS) $(PROG)
rm -f $(OBJS2) $(PROG2)
```

Shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo

WINDOWS IZQUIERDA

8 Fin de configuración de makenfile



9 Comando para ejecutar los ficheros .c alojados en el directorio src para generar los .so



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:33 PM

```
johnny@pop-os: ~/.snmp/src$ ls
cpuUsage.c  cpuUsage.h  cpuUsage.so  makefile  memoryUsage.c  memoryUsage.h  memoryUsage.so
johnny@pop-os: ~/.snmp/src$ cp cpuUsage.so memoryUsage.so ../lib/
```

10 .so generados con el comando anterior, copiado de estos ficheros al directorio lib




Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:34 PM

```
johnny@pop-os: ~/.snmp/lib$ ls
cpuUsage.so  memoryUsage.so
johnny@pop-os: ~/.snmp/lib$
```



11 verificamos que se hayan copiado correctamente



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:35 PM

johnny@pop-os: ~

```
GNU nano 5.2 /etc/snmp/snmp.conf
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenale
# loading them by commenting out the following line.
#mibs :

# If you want to globally change where snmp libraries, commands and daemons
# look for MIBS, change the line below. Note you can set this for individual
# tools with the -M option or MIBDIRS environment variable.
#
mibdirs /usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf:/home/johnny/.snmp/mibs/

mibs +UNAN-MIB
```

[Read 12 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo

WINDOWS IZQUIERDA

12 Configuración de snmp.conf



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:37 PM

```
johnny@pop-os: ~  
GNU nano 5.2 /etc/snmp/snmpd.conf Modified  
dlmod cpuUsage /home/johnny/.snmp/lib/cpuUsage.so  
dlmod memoryUsage /home/johnny/.snmp/lib/memoryUsage.so  
  
#Acceso y comunidad  
com2sec public localhost public  
#grupos  
group gpublico v1 publico  
  
#Ramas mibs que se permiten "Ver"  
#mib-2  
view vista1 included .1.3.6.1.2.1  
#unan  
view vista1 included .1.3.6.1.4.1.1111  
  
#Establecer permisos de lectura y escritura  
access gpublico "" v1 noauth exact vista1 none none  
  
# instancias del grupo system  
sysContact JOHNNY  
sysLocation LEON_2021
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

WINDOWS IZQUIERDA

13 Configuración de las comunidades en snmpd.conf



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:38 PM

```
johnny@pop-os: ~$ snmpwalk -v1 -c publico localhost unan
UNAN-MIB::cpuUsageUser.0 = Timeticks: (30017) 0:05:00.17
UNAN-MIB::cpuUsageNice.0 = Timeticks: (205) 0:00:02.05
UNAN-MIB::cpuUsageSystem.0 = Timeticks: (15450) 0:02:34.50
UNAN-MIB::cpuUsageIdle.0 = Timeticks: (427244) 1:11:12.44
UNAN-MIB::cpuUsageIo.0 = Timeticks: (27731) 0:04:37.31
UNAN-MIB::cpuUsageIrq.0 = Timeticks: (0) 0:00:00.00
UNAN-MIB::cpuUsageSoftIrq.0 = Timeticks: (1938) 0:00:19.38
UNAN-MIB::memoryUsageTotalRam.0 = INTEGER: 2911 MiB
UNAN-MIB::memoryUsageFreeRam.0 = INTEGER: 306 MiB
UNAN-MIB::memoryUsageSharedRam.0 = INTEGER: 15 MiB
UNAN-MIB::memoryUsageBufferRam.0 = INTEGER: 71 MiB
UNAN-MIB::memoryUsageTotalSwap.0 = INTEGER: 4093 MiB
UNAN-MIB::memoryUsageFreeSwap.0 = INTEGER: 4093 MiB
End of MIB
johnny@pop-os: ~$
```

14 Prueba de funcionamiento de la mib unan con snmpwalk



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:42 PM

```
johnny@pop-os: ~$ snmpwalk -v1 -c publico localhost cpuUsage
UNAN-MIB::cpuUsageUser.0 = Timeticks: (30852) 0:05:08.52
UNAN-MIB::cpuUsageNice.0 = Timeticks: (205) 0:00:02.05
UNAN-MIB::cpuUsageSystem.0 = Timeticks: (15864) 0:02:38.64
UNAN-MIB::cpuUsageIdle.0 = Timeticks: (458426) 1:16:24.26
UNAN-MIB::cpuUsageIo.0 = Timeticks: (28210) 0:04:42.10
UNAN-MIB::cpuUsageIrq.0 = Timeticks: (0) 0:00:00.00
UNAN-MIB::cpuUsageSoftIrq.0 = Timeticks: (2025) 0:00:20.25
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageUser.0
UNAN-MIB::cpuUsageNice.0 = Timeticks: (205) 0:00:02.05
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageNice.0
UNAN-MIB::cpuUsageSystem.0 = Timeticks: (16018) 0:02:40.18
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageSystem.0
UNAN-MIB::cpuUsageIdle.0 = Timeticks: (465579) 1:17:35.79
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageIdle.0
UNAN-MIB::cpuUsageIo.0 = Timeticks: (28221) 0:04:42.21
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageIrq.0
UNAN-MIB::cpuUsageSoftIrq.0 = Timeticks: (2063) 0:00:20.63
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageIo.0
UNAN-MIB::cpuUsageIrq.0 = Timeticks: (0) 0:00:00.00
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost cpuUsageSoftIrq.0
UNAN-MIB::memoryUsageTotalRam.0 = INTEGER: 2911 MiB
johnny@pop-os: ~$
```

WINDOWS IZQUIERDA

15 Prueba de funcionamiento usando snmpwalk y snmpgetnext en mib cpuUsage



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

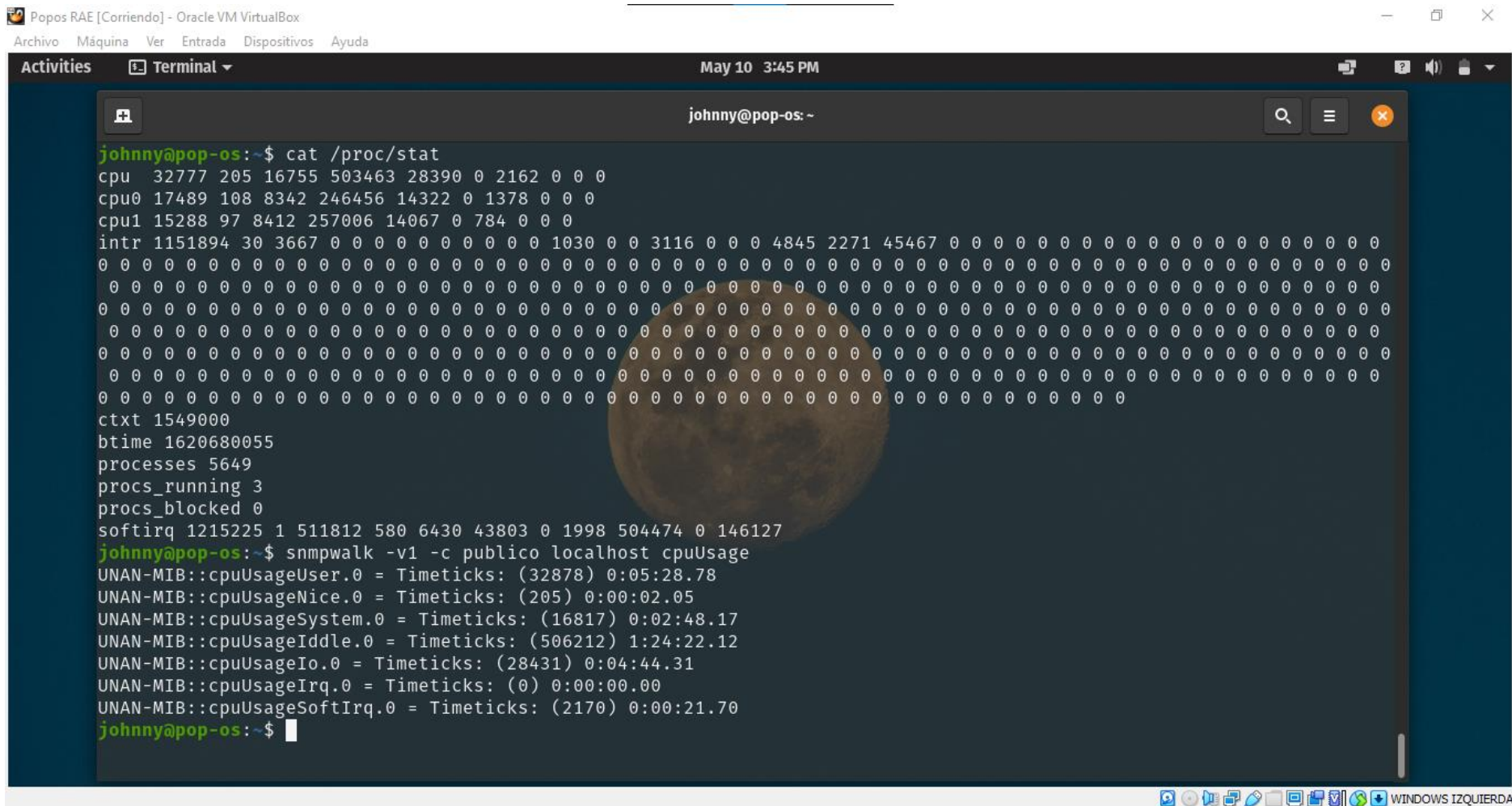
Activities Terminal May 10 3:44 PM

```
johnny@pop-os: ~$ snmpwalk -v1 -c publico localhost memoryUsage
UNAN-MIB::memoryUsageTotalRam.0 = INTEGER: 2911 MiB
UNAN-MIB::memoryUsageFreeRam.0 = INTEGER: 302 MiB
UNAN-MIB::memoryUsageSharedRam.0 = INTEGER: 15 MiB
UNAN-MIB::memoryUsageBufferRam.0 = INTEGER: 73 MiB
UNAN-MIB::memoryUsageTotalSwap.0 = INTEGER: 4093 MiB
UNAN-MIB::memoryUsageFreeSwap.0 = INTEGER: 4093 MiB
End of MIB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.1.0
UNAN-MIB::memoryUsageFreeRam.0 = INTEGER: 302 MiB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.2.0
UNAN-MIB::memoryUsageSharedRam.0 = INTEGER: 15 MiB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.3.0
UNAN-MIB::memoryUsageBufferRam.0 = INTEGER: 73 MiB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.4.0
UNAN-MIB::memoryUsageTotalSwap.0 = INTEGER: 4093 MiB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.5.0
UNAN-MIB::memoryUsageFreeSwap.0 = INTEGER: 4093 MiB
johnny@pop-os: ~$ snmpgetnext -v1 -c publico localhost memoryUsage.6.0
Error in packet.
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: UNAN-MIB::memoryUsageFreeSwap.0

johnny@pop-os: ~$
```

WINDOWS IZQUIERDA

16 Prueba de funcionamiento de mib memoryUsage con snmpwalk y snmpgetnext



17



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:47 PM

johnny@pop-os: ~

```
top - 15:46:42 up 52 min, 1 user, load average: 0.57, 0.54, 0.59
Tasks: 197 total, 1 running, 196 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.2 us, 5.6 sy, 0.0 ni, 88.7 id, 0.0 wa, 0.0 hi, 1.4 si, 0.0 st
MiB Mem : 2912.5 total, 301.6 free, 1335.9 used, 1275.0 buff/cache
MiB Swap: 4095.5 total, 4095.5 free, 0.0 used. 1402.2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3025	johnny	-2	0	4037620	334348	131920	S	5.7	11.2	1:40.44	gnome-shell
5650	johnny	20	0	22488	4072	3432	R	5.7	0.1	0:00.05	top
2718	johnny	20	0	1394896	90360	52684	S	2.9	3.0	0:30.78	Xorg
1	root	20	0	168700	12076	8692	S	0.0	0.4	0:06.93	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude_
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
12	root	20	0	0	0	0	S	0.0	0.0	0:00.58	ksoftirqd/0
13	root	20	0	0	0	0	I	0.0	0.0	0:03.61	rcu_sched
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.07	migration/0
15	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
18	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
19	root	rt	0	0	0	0	S	0.0	0.0	0:00.57	migration/1
20	root	20	0	0	0	0	S	0.0	0.0	0:01.11	ksoftirqd/1
22	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri

johnny@pop-os: ~\$ smpwalk -v1 -c publico localhost memoryUsage

WINDOWS IZQUIERDA

18 Comparación del comando top y mib memoryUsage 1



Popos RAE [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Activities Terminal May 10 3:48 PM

```
johnny@pop-os: ~  
2718 johnny 20 0 1394896 90360 52684 S 2.9 3.0 0:30.78 Xorg  
1 root 20 0 168700 12076 8692 S 0.0 0.4 0:06.93 systemd  
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd  
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp  
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp  
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri  
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq  
10 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_rude_  
11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_trace  
12 root 20 0 0 0 0 S 0.0 0.0 0:00.58 ksoftirqd/0  
13 root 20 0 0 0 0 I 0.0 0.0 0:03.61 rcu_sched  
14 root rt 0 0 0 0 S 0.0 0.0 0:00.07 migration/0  
15 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0  
16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0  
17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1  
18 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/1  
19 root rt 0 0 0 0 S 0.0 0.0 0:00.57 migration/1  
20 root 20 0 0 0 0 S 0.0 0.0 0:01.11 ksoftirqd/1  
22 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/1:0H-events_highpri  
johnny@pop-os:~$ snmpwalk -v1 -c publico localhost memoryUsage  
UNAN-MIB::memoryUsageTotalRam.0 = INTEGER: 2911 MiB  
UNAN-MIB::memoryUsageFreeRam.0 = INTEGER: 300 MiB  
UNAN-MIB::memoryUsageSharedRam.0 = INTEGER: 15 MiB  
UNAN-MIB::memoryUsageBufferRam.0 = INTEGER: 75 MiB  
UNAN-MIB::memoryUsageTotalSwap.0 = INTEGER: 4093 MiB  
UNAN-MIB::memoryUsageFreeSwap.0 = INTEGER: 4093 MiB  
End of MIB  
johnny@pop-os:~$
```

WINDOWS IZQUIERDA

19 Comparación del comando top y mib memoryUsage 2



ANEXO

Enlace de video demostrativo

<https://youtu.be/xixlSjYmRvA>

Recursos disponibles en GitHub

<https://github.com/JonesoO/gestion-de-red>

MIB UNAN

UNAN-MIB DEFINITIONS ::= BEGIN

IMPORTS

DisplayString

FROM RFC1213-MIB

enterprises, TimeTicks, Gauge, INTEGER

FROM RFC1155-SMI

OBJECT-TYPE

FROM RFC-1212;

unan OBJECT IDENTIFIER ::= { enterprises 1111 }

resources OBJECT IDENTIFIER ::= { unan 1 }

cpuUsage OBJECT IDENTIFIER ::= { resources 1 }

memoryUsage OBJECT IDENTIFIER ::= { resources 2 }

cpuUsageUser OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Consumo de cpu de usuario"



::= { cpuUsage 1 }

cpuUsageNice OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Consumo de cpu de usuario de baja prioridad"

::= { cpuUsage 2 }

cpuUsageSystem OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Consumo de cpu del sistema"

::= { cpuUsage 3 }

cpuUsageIdle OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"¿?"

::= { cpuUsage 4 }

cpuUsageIo OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION



"Consumo de cpu en peticiones IO"

::= { cpuUsage 5 }

cpuUsageIrq OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Interrupciones"

::= { cpuUsage 6 }

cpuUsageSoftIrq OBJECT-TYPE

SYNTAX TimeTicks

ACCESS read-only

STATUS mandatory

DESCRIPTION

"¿Soft IRQ?"

::= { cpuUsage 7 }

--- Subsystems group

memoryUsageTotalRam OBJECT-TYPE

SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current

DESCRIPTION

"Cantidad en Mebibyte de RAM total"

::= { memoryUsage 1 }

memoryUsageFreeRam OBJECT-TYPE



SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current

DESCRIPTION

"Cantidad en Mebibyte de RAM libre"

::= { memoryUsage 2 }

memoryUsageSharedRam OBJECT-TYPE

SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current

DESCRIPTION

"Cantidad en Mebibyte de RAM compartida"

::= { memoryUsage 3 }

memoryUsageBufferRam OBJECT-TYPE

SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current

DESCRIPTION

"Cantidad en Mebibyte de RAM Reservada"

::= { memoryUsage 4 }

memoryUsageTotalSwap OBJECT-TYPE

SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current



DESCRIPTION

"Cantidad en Mebibyte de Swap total"

::= { memoryUsage 5 }

memoryUsageFreeSwap OBJECT-TYPE

SYNTAX INTEGER

UNITS "MiB"

ACCESS read-only

STATUS current

DESCRIPTION

"Cantidad en Mebibyte de Swap libre"

::= { memoryUsage 6 }

END



memoryUsage.c

```
/*
 * Note: this file originally auto-generated by mib2c using
 *      $
 */

#include <net-snmp/net-snmp-config.h>
#include <net-snmp/net-snmp-includes.h>
#include <net-snmp/agent/net-snmp-agent-includes.h>
#include "memoryUsage.h"
#include <sys/sysinfo.h>

/** Initializes the memoryUsage module */
void
init_memoryUsage(void)
{
    const oid memoryUsageTotalRam_oid[] = { 1,3,6,1,4,1,1111,1,2,1 };
    const oid memoryUsageFreeRam_oid[] = { 1,3,6,1,4,1,1111,1,2,2 };
    const oid memoryUsageSharedRam_oid[] = { 1,3,6,1,4,1,1111,1,2,3 };
    const oid memoryUsageBufferRam_oid[] = { 1,3,6,1,4,1,1111,1,2,4 };
    const oid memoryUsageTotalSwap_oid[] = { 1,3,6,1,4,1,1111,1,2,5 };
    const oid memoryUsageFreeSwap_oid[] = { 1,3,6,1,4,1,1111,1,2,6 };

    DEBUGMSGTL(("memoryUsage", "Initializing\n"));

    netsnmp_register_scalar(
        netsnmp_create_handler_registration("memoryUsageTotalRam", handle_
memoryUsageTotalRam,
                                           memoryUsageTotalRam_oid, OID_LENGTH(memoryU
UsageTotalRam_oid),
                                           HANDLER_CAN_READONLY
        ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("memoryUsageFreeRam", handle_
memoryUsageFreeRam,
                                           memoryUsageFreeRam_oid, OID_LENGTH(memoryU
sageFreeRam_oid),
                                           HANDLER_CAN_READONLY
        ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("memoryUsageSharedRam", handl
e_memoryUsageSharedRam,
                                           memoryUsageSharedRam_oid, OID_LENGTH(memor
yUsageSharedRam_oid),
                                           HANDLER_CAN_READONLY
        ));
    netsnmp_register_scalar(
```



```
        netsnmp_create_handler_registration("memoryUsageBufferRam", handl
e_memoryUsageBufferRam,
                                           memoryUsageBufferRam_oid, OID_LENGTH(memor
yUsageBufferRam_oid),
                                           HANDLER_CAN_READONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("memoryUsageTotalSwap", handl
e_memoryUsageTotalSwap,
                                           memoryUsageTotalSwap_oid, OID_LENGTH(memor
yUsageTotalSwap_oid),
                                           HANDLER_CAN_READONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("memoryUsageFreeSwap", handl
e_memoryUsageFreeSwap,
                                           memoryUsageFreeSwap_oid, OID_LENGTH(memory
UsageFreeSwap_oid),
                                           HANDLER_CAN_READONLY
    ));
}

int
handle_memoryUsageTotalRam(netsnmp_mib_handler *handler,
                           netsnmp_handler_registration *reginfo,
                           netsnmp_agent_request_info *reqinfo,
                           netsnmp_request_info *requests)
{
    //Declarando la estructura necesaria para almacenar
    //la informacion del sistema.
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            //Obteniendo informacion del sistema y guardandola
            //en la variable especificada.
            sysinfo(&info);
            result = info.totalram / 1.049e+6;

            snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                     //Direccion de memoria.
                                     (u_char*) &result,
                                     //Tamaño de los datos.
                                     sizeof(info.totalram));

            break;

        default:
```



```
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageTot
alRam\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_memoryUsageFreeRam(netsnmp_mib_handler *handler,
                          netsnmp_handler_registration *reginfo,
                          netsnmp_agent_request_info *reqinfo,
                          netsnmp_request_info *requests)
{
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            sysinfo(&info);
            result = info.freeram / 1.049e+6;

            snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                      (u_char*) &result,
                                      sizeof(info.freeram));

            break;

        default:
            snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageFre
eRam\n", reqinfo->mode );
            return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_memoryUsageSharedRam(netsnmp_mib_handler *handler,
                            netsnmp_handler_registration *reginfo,
                            netsnmp_agent_request_info *reqinfo,
                            netsnmp_request_info *requests)
{
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            sysinfo(&info);
```



```
        result = info.sharedram / 1.049e+6; //convirtiendo de Byte a
MibiByte

        snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                   (u_char*) &result,
                                   sizeof(info.sharedram));

        break;

    default:
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageSha
redRam\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_memoryUsageBufferRam(netsnmp_mib_handler *handler,
                            netsnmp_handler_registration *reginfo,
                            netsnmp_agent_request_info *reqinfo,
                            netsnmp_request_info *requests)
{
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            sysinfo(&info);
            result = info.bufferram / 1.049e+6;

            snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                       (u_char*) &result,
                                       sizeof(info.bufferram));

            break;

        default:
            snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageBuf
ferRam\n", reqinfo->mode );
            return SNMP_ERR_GENERR;
        }

        return SNMP_ERR_NOERROR;
    }
}

int
handle_memoryUsageTotalSwap(netsnmp_mib_handler *handler,
                             netsnmp_handler_registration *reginfo,
                             netsnmp_agent_request_info *reqinfo,
                             netsnmp_request_info *requests)
```



```
{
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            sysinfo(&info);
            result = info.totalswap / 1.049e+6;

            snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                     (u_char*) &result,
                                     sizeof(info.totalswap));

            break;

        default:
            snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageTot
alSwap\n", reqinfo->mode );
            return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_memoryUsageFreeSwap(netsnmp_mib_handler *handler,
                           netsnmp_handler_registration *reginfo,
                           netsnmp_agent_request_info *reqinfo,
                           netsnmp_request_info *requests)
{
    struct sysinfo info;
    long result;

    switch(reqinfo->mode) {

        case MODE_GET:
            sysinfo(&info);
            result = info.freeswap / 1.049e+6;

            snmp_set_var_typed_value(requests->requestvb, ASN_INTEGER,
                                     (u_char*) &result,
                                     sizeof(info.freeswap));

            break;

        default:
            /* we should never get here, so this is a really bad error */
            snmp_log(LOG_ERR, "unknown mode (%d) in handle_memoryUsageFre
eSwap\n", reqinfo->mode );
            return SNMP_ERR_GENERR;
    }
}
```




```
    return SNMP_ERR_NOERROR;
}
```

cpuUsage.c

```
/*
 * Note: this file originally auto-generated by mib2c using
 *      $
 */

#include <net-snmp/net-snmp-config.h>
#include <net-snmp/net-snmp-includes.h>
#include <net-snmp/agent/net-snmp-agent-includes.h>
#include "cpuUsage.h"

/** Initializes the cpuUsage module */
void
init_cpuUsage(void)
{
    const oid cpuUsageUser_oid[]    = { 1,3,6,1,4,1,1111,1,1,1 };
    const oid cpuUsageNice_oid[]    = { 1,3,6,1,4,1,1111,1,1,2 };
    const oid cpuUsageSystem_oid[]  = { 1,3,6,1,4,1,1111,1,1,3 };
    const oid cpuUsageIdle_oid[]    = { 1,3,6,1,4,1,1111,1,1,4 };
    const oid cpuUsageIo_oid[]       = { 1,3,6,1,4,1,1111,1,1,5 };
    const oid cpuUsageIrq_oid[]      = { 1,3,6,1,4,1,1111,1,1,6 };
    const oid cpuUsageSoftIrq_oid[]  = { 1,3,6,1,4,1,1111,1,1,7 };

    DEBUGMSGTL(("cpuUsage", "Initializing\n"));

    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageUser", handle_cpuUsageUser,
                                           cpuUsageUser_oid, OID_LENGTH(cpuUsageUser_oid),
                                           HANDLER_CAN_RONLY
        ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageNice", handle_cpuUsageNice,
                                           cpuUsageNice_oid, OID_LENGTH(cpuUsageNice_oid),
                                           HANDLER_CAN_RONLY
```



```
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageSystem", handle_cpuU
sageSystem,
                                           cpuUsageSystem_oid, OID_LENGTH(cpuUsageSys
tem_oid),
                                           HANDLER_CAN_RONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageIdle", handle_cpuUs
ageIdle,
                                           cpuUsageIdle_oid, OID_LENGTH(cpuUsageIdl
e_oid),
                                           HANDLER_CAN_RONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageIo", handle_cpuUsage
Io,
                                           cpuUsageIo_oid, OID_LENGTH(cpuUsageIo_oid)
,
                                           HANDLER_CAN_RONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageIrq", handle_cpuUsag
eIrq,
                                           cpuUsageIrq_oid, OID_LENGTH(cpuUsageIrq_o
id),
                                           HANDLER_CAN_RONLY
    ));
    netsnmp_register_scalar(
        netsnmp_create_handler_registration("cpuUsageSoftIrq", handle_cpu
UsageSoftIrq,
                                           cpuUsageSoftIrq_oid, OID_LENGTH(cpuUsageS
oftIrq_oid),
                                           HANDLER_CAN_RONLY
    ));
}

int
handle_cpuUsageUser(netsnmp_mib_handler *handler,
                   netsnmp_handler_registration *reginfo,
                   netsnmp_agent_request_info *reqinfo,
                   netsnmp_request_info *requests)
{
    //Declarando el puntero para guardar la referencia
    //al fichero que se abra.
    FILE *file;

    //Declarando las variables necesarias para almacenar
```



```
//la informacion que se obtendra.
char cpu[10];
int num, i;

switch(reqinfo->mode) {

    case MODE_GET:
        //Abriendo fichero con permisos de lectura
        //y almacenando la referencia en la variable.
        file = fopen("/proc/stat", "r");

        //Leyendo nombre de cpu.
        //NT: Este valor no se utiliza, pero es necesario leerlo.
        fscanf(file, "%s\t", cpu);

        //Leyendo hasta llegar al valor que queremos.
        //NT: Leera hasta que i sea igual numero especificado,
        // i = 1 | cpuUsageUser.
        for (i = 1; i <= 1; i++)
            fscanf(file, "%d\t", &num);

        //Cerrando el fichero..
        fclose(file);

        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;

    default:
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageUser\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
}

return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageNice(netsnmp_mib_handler *handler,
                    netsnmp_handler_registration *reginfo,
                    netsnmp_agent_request_info *reqinfo,
                    netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;
```



```
switch(reqinfo->mode) {

    case MODE_GET:
        file = fopen("/proc/stat", "r");

        fscanf(file, "%s\t", cpu);

        //Leyendo hasta llegar al valor que queremos.
        //NT: Leera hasta que i sea igual numero especificado,
        // i = 1 | cpuUsageUser.
        // i = 2 | cpuUsageNice.
        for (i = 1; i <= 2; i++)
            fscanf(file, "%d\t", &num);

        fclose(file);

        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;

    default:
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageNice\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
}

return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageSystem(netsnmp_mib_handler *handler,
                      netsnmp_handler_registration *reginfo,
                      netsnmp_agent_request_info *reqinfo,
                      netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;

    switch(reqinfo->mode) {

        case MODE_GET:
            file = fopen("/proc/stat", "r");

            fscanf(file, "%s\t", cpu);
```



```
        //Leyendo hasta llegar al valor que queremos.
        //NT: Leera hasta que i sea igual numero especificado,
        // i = 1 | cpuUsageUser.
        // i = 2 | cpuUsageNice.
        // i = 3 | cpuUsageSystem.
        for (i = 1; i <= 3; i++)
            fscanf(file, "%d\t", &num);

        fclose(file);

        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;

    default:
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageSystem
\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageIdle(netsnmp_mib_handler *handler,
                    netsnmp_handler_registration *reginfo,
                    netsnmp_agent_request_info *reqinfo,
                    netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;

    switch(reqinfo->mode) {

        case MODE_GET:
            file = fopen("/proc/stat", "r");

            fscanf(file, "%s\t", cpu);

            //Leyendo hasta llegar al valor que queremos.
            //NT: Leera hasta que i sea igual numero especificado,
            // i = 1 | cpuUsageUser.
            // i = 2 | cpuUsageNice.
            // i = 3 | cpuUsageSystem.
```



```
        // i = 4 | cpuUsageIddle.
        for (i = 1; i <= 4; i++)
            fscanf(file, "%d\t", &num);

        fclose(file);

        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;
    default:
        /* we should never get here, so this is a really bad error */
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageIddle\
n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageIo(netsnmp_mib_handler *handler,
                  netsnmp_handler_registration *reginfo,
                  netsnmp_agent_request_info *reqinfo,
                  netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;

    switch(reqinfo->mode) {

    case MODE_GET:
        file = fopen("/proc/stat", "r");

        fscanf(file, "%s\t", cpu);

        //Leyendo hasta llegar al valor que queremos.
        //NT: Leera hasta que i sea igual numero especificado,
        // i = 1 | cpuUsageUser.
        // i = 2 | cpuUsageNice.
        // i = 3 | cpuUsageSystem.
        // i = 4 | cpuUsageIddle.
        // i = 5 | cpuUsageIo.
        for (i = 1; i <= 5; i++)
            fscanf(file, "%d\t", &num);
    }
```



```
fclose(file);

//Devolviendo datos de consulta.
snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                          //Direccion de memoria.
                          (u_char*) &num,
                          //Tamaño de los datos.
                          sizeof(num));

break;

default:
    /* we should never get here, so this is a really bad error */
    snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageIo\n",
reqinfo->mode );
    return SNMP_ERR_GENERR;
}

return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageIrq(netsnmp_mib_handler *handler,
                  netsnmp_handler_registration *reginfo,
                  netsnmp_agent_request_info *reqinfo,
                  netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;

    switch(reqinfo->mode) {

        case MODE_GET:
            file = fopen("/proc/stat", "r");

            fscanf(file, "%s\t", cpu);

            //Leyendo hasta llegar al valor que queremos.
            //NT: Leera hasta que i sea igual numero especificado,
            // i = 1 | cpuUsageUser.
            // i = 2 | cpuUsageNice.
            // i = 3 | cpuUsageSystem.
            // i = 4 | cpuUsageIddle.
            // i = 5 | cpuUsageIo.
            // i = 6 | cpuUsageIrq.
            for (i = 1; i <= 6; i++)
                fscanf(file, "%d\t", &num);

            fclose(file);
```



```
        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;

    default:
        /* we should never get here, so this is a really bad error */
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageIrq\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}

int
handle_cpuUsageSoftIrq(netsnmp_mib_handler *handler,
                       netsnmp_handler_registration *reginfo,
                       netsnmp_agent_request_info *reqinfo,
                       netsnmp_request_info *requests)
{
    FILE *file;
    char cpu[10];
    int num, i;

    switch(reqinfo->mode) {

    case MODE_GET:
        file = fopen("/proc/stat", "r");

        fscanf(file, "%s\t", cpu);

        //Leyendo hasta llegar al valor que queremos.
        //NT: Leera hasta que i sea igual numero especificado,
        // i = 1 | cpuUsageUser.
        // i = 2 | cpuUsageNice.
        // i = 3 | cpuUsageSystem.
        // i = 4 | cpuUsageIddle.
        // i = 5 | cpuUsageIo.
        // i = 6 | cpuUsageIrq.
        // i = 7 | cpuUsageSoftIrq.
        for (i = 1; i <= 7; i++)
            fscanf(file, "%d\t", &num);

        fclose(file);
    }
```




```
        //Devolviendo datos de consulta.
        snmp_set_var_typed_value(requests->requestvb, ASN_TIMETICKS,
                                //Direccion de memoria.
                                (u_char*) &num,
                                //Tamaño de los datos.
                                sizeof(num));

        break;

    default:
        /* we should never get here, so this is a really bad error */
        snmp_log(LOG_ERR, "unknown mode (%d) in handle_cpuUsageSoftIr
q\n", reqinfo->mode );
        return SNMP_ERR_GENERR;
    }

    return SNMP_ERR_NOERROR;
}
```

Makefile

```
#
#
# Makefile to generate sysInfoGroup.so
#
# usage:
# "make" : generate library
# "make clean" : remove *.o , *.so
#

CC=gcc
LDFLAGS= `net-snmp-config --cflags`

#PROG= sysInfoGroup.so
#SRCS= sysInfoGroup.c
PROG= cpuUsage.so
SRCS= cpuUsage.c
OBS= $(SRCS:.c=.o)

PROG2= memoryUsage.so
SRCS2= memoryUsage.c
OBS2= $(SRCS2:.c=.o)

# shared library flags (assumes gcc)
DLFLAGS=-fPIC -shared -Wno-unused-result
```



```
all: $(PROG) $(PROG2)

$(PROG): $(OBJS)
    $(CC) -o $@ $(OBJS) $(LDFLAGS) $(DLFLAGS)
    rm -f $(OBJS)

$(PROG2): $(OBJS2)
    $(CC) -o $@ $(OBJS2) $(LDFLAGS) $(DLFLAGS)
    rm -f $(OBJS2)

.c.o:
    @if test "$(CC)" = "" ; then \
        echo "La variable de entorno CC debe establecerse con la ruta del com
pilador" ; \
        exit 1 ; \
    fi
    $(CC) -o $@ -c $< -g $(LDFLAGS) $(DLFLAGS)

clean:
    rm -f $(OBJS) $(PROG)
    rm -f $(OBJS2) $(PROG2)
```