



Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas

Examen Extraordinario

Materia Sistemas Operativos en Tiempo
Real

Profesor: Maza Casas Lamberto

Alumno: Cruz Reyes Antonio

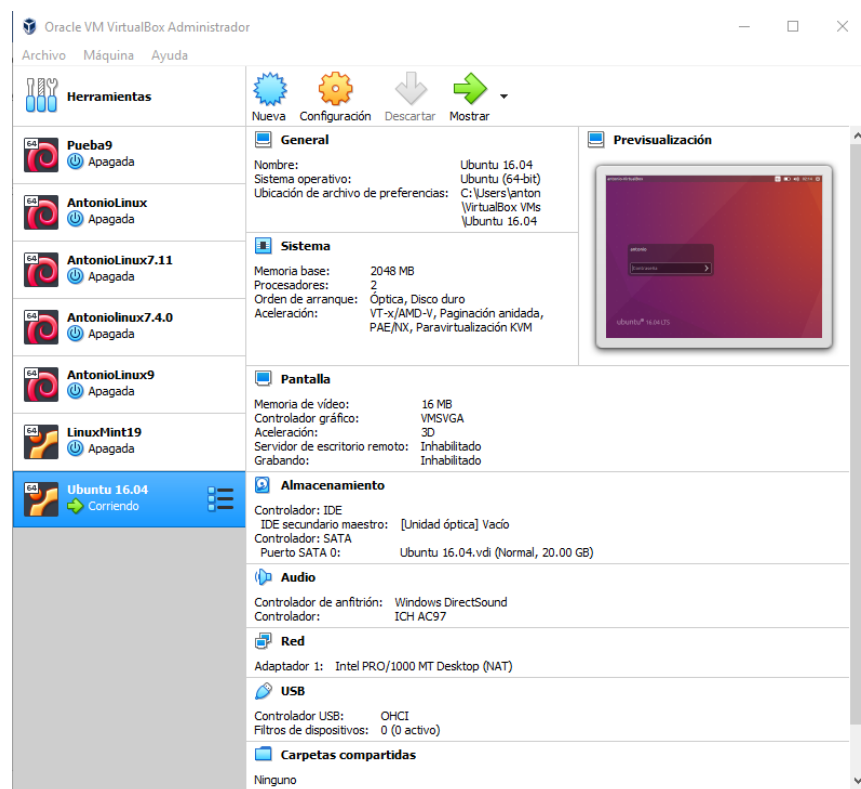
Fecha: 12/06/2019



Se solicitó el realizar la ejecución de un programa el cual estuviese generando carga de memoria en el emulador quemu, a través de un programa en Marte Os el cual esta instalado en una maquina virtual.

Al estarse ejecutando el programa se tiene que comprobar por algún medio que el programa que se esta ejecutando, se esta realizando sin que tenga que estarse imprimiendo algo en la pantalla.

Los primeros pasos fue el tener instalado correctamente un sistema operativo Ubuntu 16.04(Linux) el cual está sobre Virtual Box.

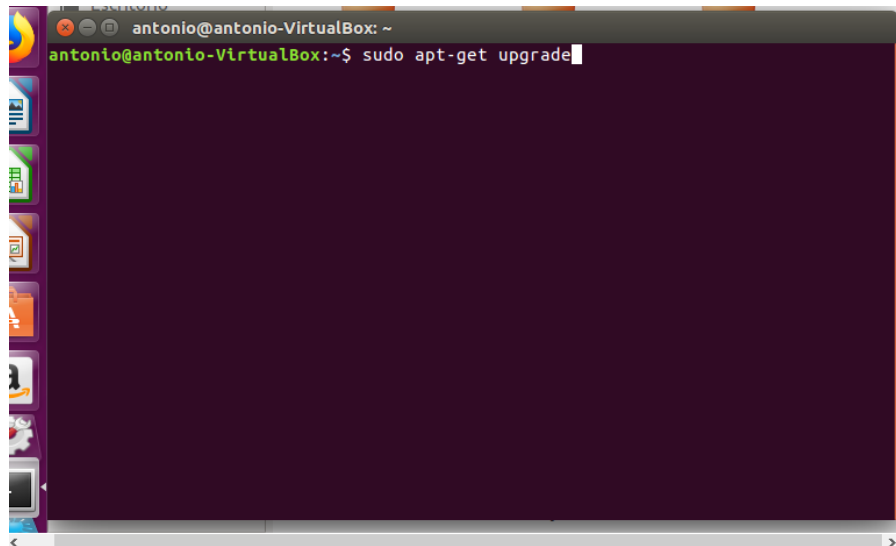


Una vez teniendo la maquina virtual, con el sistema operativo correcto.

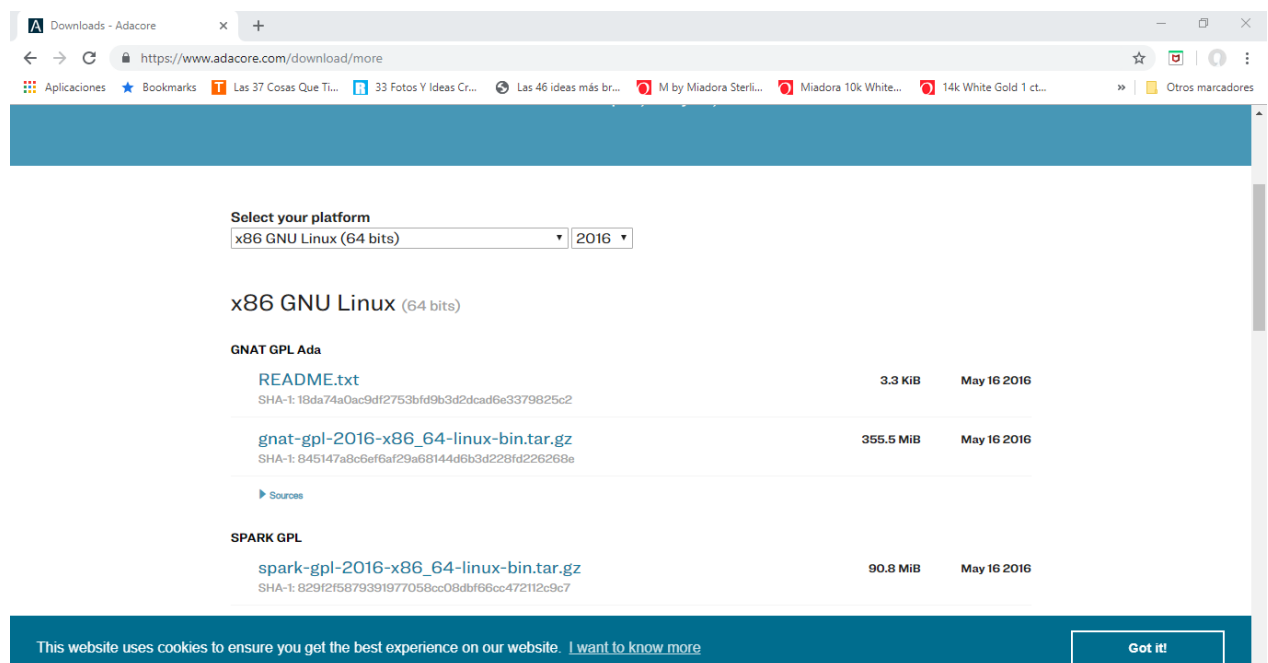


Tenemos que abrir una terminal para poder actualizar los sistemas para poder operar los comandos de la instalación y funcionamiento de GNAT y de Marte OS.

Para ello se implementa el siguiente comando.



Una vez actualizado los programas se tiene que descargar el GNAT de la pagina <https://www.adacore.com/download/more>



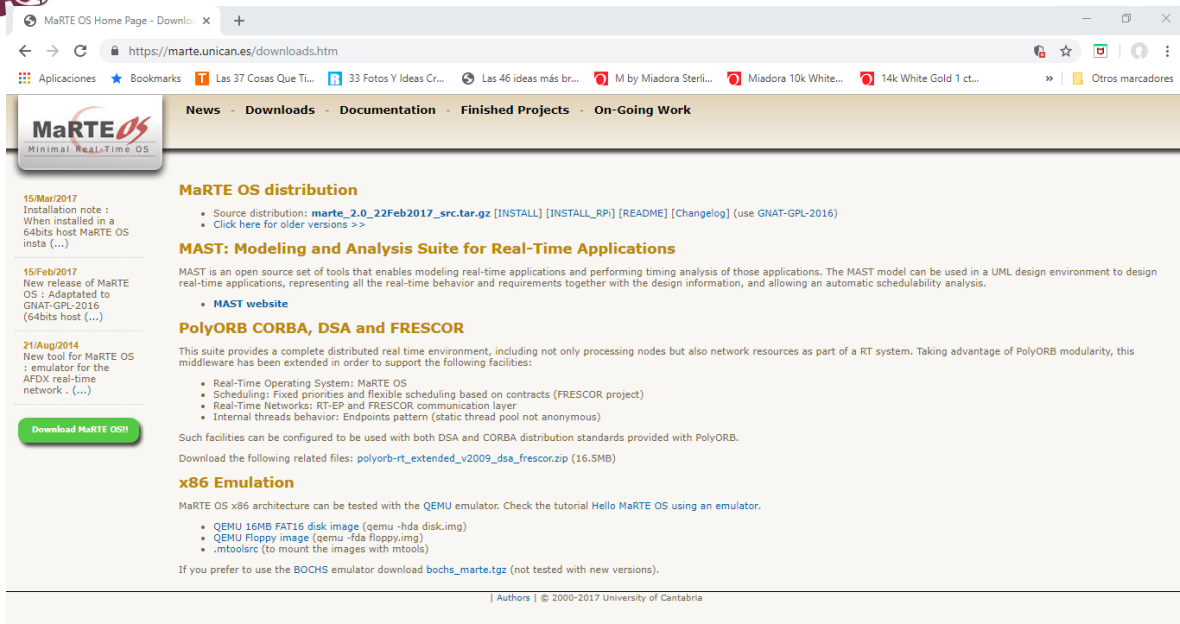


Antes de descomprimirlo se tuvo que haber creado la carpeta myapps y dentro de ella gnat.

```
antonio@antonio-VirtualBox: ~/myapps/gnat
antonio@antonio-VirtualBox:~/myapps/gnat$ ls
bin      gnat-gpl-2016-x86_64-linux-bin  include  lib64      share
etc      gnat-gpl-2016-x86_64-linux-bin.tar.gz  lib      libexec
```

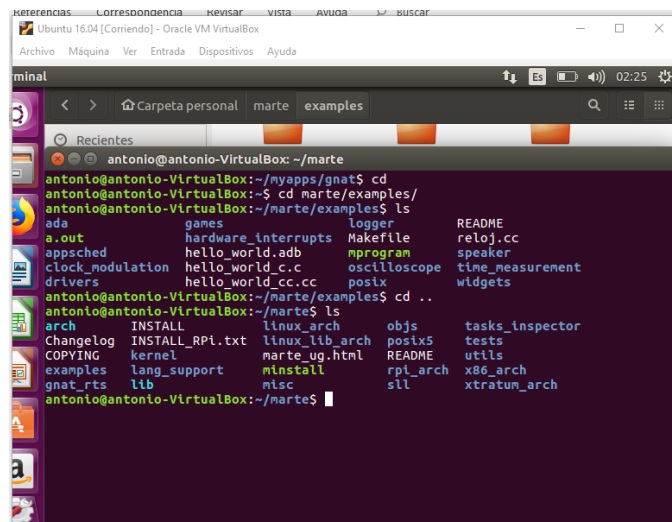
Una vez descomprimido se siguen los pasos de la guía que viene en la carpeta y se debe colocar el PATH en el lugar adecuado.

El siguiente paso es la descarga e instalación de Marte OS el cual la ubicaremos en la siguiente página. <https://marte.unican.es/downloads.htm>



Siguiendo la guía que esta en esa misma página con titulo INSTALL.

Una vez instalado el Marte OS, nos direccionamos a la carpeta de /examples.



En la cual se va a encontrar un ejemplo de Hola mundo en 3 lenguajes diferentes (C+, C++ y Ada).



Una vez metiendo los comandos que se requieren para poder correr el programa. Se va a generar el programa mprogram y el ejecutable a.out el cual van a ser utilizados para poderlos observar en el emulador qemu posteriormente.

```
antonio@antonio-VirtualBox: ~/marTE/examples
antonio@antonio-VirtualBox:~/marTE$ cd
antonio@antonio-VirtualBox:~$ cd marTE/examples/
antonio@antonio-VirtualBox:~/marTE/examples$ ls
ada          games          logger          README
a.out        hardware_interrupts Makefile         reloj.cc
appsched     hello_world.adb  mprogram        speaker
clock_modulation hello_world_c.c  oscilloscope    time_measurement
drivers       hello_world_cc.cc posix            widgets
antonio@antonio-VirtualBox:~/marTE/examples$ cd ..
antonio@antonio-VirtualBox:~/marTE$ ls
arch          INSTALL        linux_arch      objs          tasks_inspector
Changelog     INSTALL_RPi.txt linux_lib_arch  posix5        tests
COPYING       kernel         marTE Ug.html  README        utils
examples      lang_support   mininstall     rpi_arch     x86_arch
gnat_rts      lib            misc           sll           xtratum_arch
antonio@antonio-VirtualBox:~/marTE$ cd examples/
antonio@antonio-VirtualBox:~/marTE/examples$ ls
ada          games          logger          README
a.out        hardware_interrupts Makefile         reloj.cc
appsched     hello_world.adb  mprogram        speaker
clock_modulation hello_world_c.c  oscilloscope    time_measurement
drivers       hello_world_cc.cc posix            widgets
antonio@antonio-VirtualBox:~/marTE/examples$
```

Los comandos utilizados para los programas fueron:

```
mgcc hello_world_c.c
```

```
mgcc hello_world_c.c -o mprogram
```

El siguiente paso fue el comprobar que el emulador qemu ya estuviese instalado.

Gracias a las actualizaciones hechas anteriormente se actualizo el qemu y se pudo comprobar su funcionamiento.

Para ello utilizamos el siguiente comando.

```
qemu-system-i386 -kernel mprogram
```

ó

```
qemu-system-i386 -kernel a.out
```



```
Administrator | Ubuntu 16.04 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

MU
< > Carpeta personal marte examples
Recientes
QEMU
ar Us
z
c
=i
ni
ME
pc
ar
Us
z
c
=i
T.
Copyright (C) Universidad de Cantabria, SPAIN
te TLSF 2.3.2 dynamic memory pool: 131514368 bytes
/Devices initialization...
ar Major Number 1 (stdin) Keyboard ...OK
ac Major Number 2 (stdout) Text/Serial ...OK
a. Major Number 3 (stderr) Text/Serial ...OK
ar
c
Hello, I'm a C program running on MaRTE OS.
dr _exit(0) called; rebooting...
a. Press a key to reboot_
```

Una vez comprobando que el programa qemu y Marte OS estuviesen funcionando correctamente.

Se prosiguió por elaborar un programa el cual estuviera generando un loop o un trabajo en el qemu para poder generar un reporte de memoria utilizada y comprobar que el programa estuviese funcionando sin la necesidad de que se imprimiera algo en la pantalla.

Se opto por realizar un código de un reloj, que estuviese actualizando la hora en lenguaje C++.

```
Referencias Correspondencia Revisar Vista Ayuda Buscar
Ubuntu 16.04 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

tor de textos
reloj.cc (~/.martexamples) - gedit
Abrir Guardar

#include<stdio.h>
#include<unistd.h>
#include<pthread.h>
#include<stdlib.h>

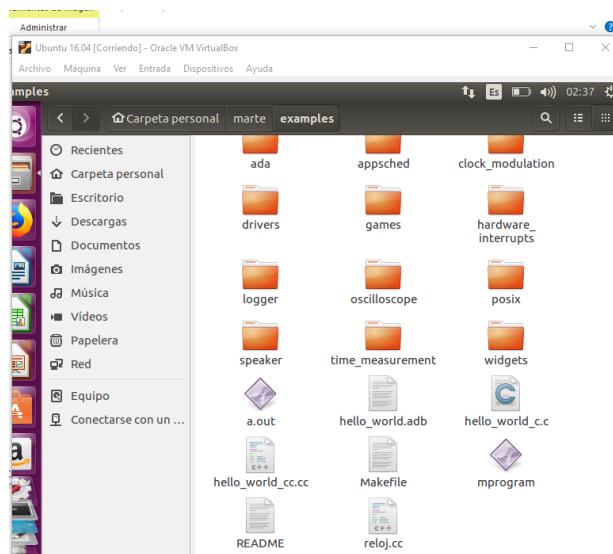
int main ()
{
    int h, m, s;
    for (h=0; h<24; h++)
    {
        for (m=0; m<60; m++)
        {
            for (s=0; s<60; s++)
            {
                printf("%i:%i:%i\n", h, m, s);
                sleep(1000);
            }
        }
    }
}
```

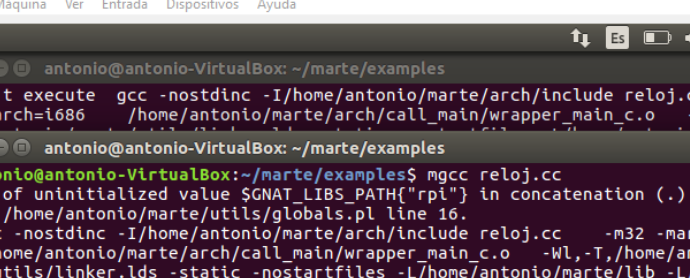


Código:

```
1. #include <stdio.h>
2. #include <unistd.h>
3. #include <pthread.h>
4.
5. int main ()
6. {
7.     int h, m, s, x;
8.     x = 1;
9.
10.    for (h=0;h<60;h++)
11.    {
12.        for (m=0;m<60;m++)
13.        {
14.            for (s=0;s<60;s++)
15.            {
16.                printf ("%i:%i:%i\n", h, m, s);
17.                sleep (x);
18.            }
19.        }
20.    }
21. }
```

Este programa se creo y se coloco en la carpeta de /martex/examples para poder ser compilado, corrido y ejecutado por el gemu.





```
References Corresidencia Revisar Vista Avuda 🔍 Buscar
Ubuntu 16.04 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

minal
antonio@antonio-VirtualBox: ~/martexamples
Can't execute gcc -nostdinc -I/home/antonio/marte/arch/include reloj.cc -m32 -march=i686 /home/antonio/marte/arch/call_main/wrapper_main.c.o -WL,-T,/home/antonio/marte/arch/call_main/wrapper_main.c.o -o mprogram
antonio@antonio-VirtualBox: ~/martexamples
antonio@antonio-VirtualBox:~/martexamples$ mgcc reloj.cc
Use of uninitialized value $GNAT_LIBS_PATH{"rpi"} in concatenation (.) or string
at /home/antonio/marte/utils/globals.pl line 16.
gcc -nostdinc -I/home/antonio/marte/arch/include reloj.cc -m32 -march=i686
/home/antonio/marte/arch/call_main/wrapper_main.c.o -WL,-T,/home/antonio/marte/arch/call_main/wrapper_main.c.o -o mprogram
antonio@antonio-VirtualBox:~/martexamples$ mgcc reloj.cc -o mprogram
Use of uninitialized value $GNAT_LIBS_PATH{"rpi"} in concatenation (.) or string
at /home/antonio/marte/utils/globals.pl line 16.
gcc -nostdinc -I/home/antonio/marte/arch/include reloj.cc -m32 -march=i686
-o mprogram /home/antonio/marte/arch/call_main/wrapper_main.c.o -WL,-T,/home/antonio/marte/arch/call_main/wrapper_main.c.o -o mprogram
antonio@antonio-VirtualBox:~/martexamples$ ls
ada games logger README
a.out hardware_interrupts Makefile reloj.cc
appsched hello_world.adb mprogram speaker
clock_modulation hello_world_c.c oscilloscope time_measurement
drivers hello_world_cc.cc posix widgets
antonio@antonio-VirtualBox:~/martexamples$ gemu-system-i386 -kernel mprogram
antonio@antonio-VirtualBox:~/martexamples$
```

```
qemu-system-i386 -kernel a.out
```



Una vez más se visualizo el programa corriendo en el qemu.

Aunque al imprimir los resultados se imprime tanto la hora como la palabra error, la función que requerimos para ser comprobado este programa se esta realizando ya que esta generando un gasto en la memoria RAM del qemu.

```
Can't execute gcc -nostdinc -I/home/antonio/marte/arch/include reloj.cc -m32 -march=i686 /home/antonio/marte/arch/call_main/wrapper_main_c.o -WL,-T,/ho

-- M a R T E   O S  --
U2.0 2017-02-22
Copyright (C) Universidad de Cantabria, SPAIN
TLFS 2.3.2 dynamic memory pool: 131514368 bytes
Devices initialization...
Major Number 1 (stdin) Keyboard ...OK
Major Number 2 (stdout) Text/Serial ...OK
Major Number 3 (stderr) Text/Serial ...OK
0:0:0
ERROR:0s42m754u583n:0s48m791u383n ERROR:0s99m807u908n:0s54m373u934n ERROR:0s104
m503u755n:0s60m967u228n ERROR:0s111m500u173n:0s66m568u218n ERROR:0s116m682u953n:
0s79m287u149n ERROR:0s129m825u961n:0s85m542u691n ERROR:0s135m714u417n:0s92m630u4
62n ERROR:0s143m243u863n:0s99m208u670n ERROR:0s149m387u101n:0s106m422u155n ERROR
:0s156m534u376n:0s111m941u11n ERROR:0s162m494u908n:0s121m125u695n ERROR:0s171m28
8u202n:0s125m981u618n ERROR:0s176m572u392n:0s135m412u703n ERROR:0s185m544u200n:0
s141m14u530n ERROR:0s191m619u551n:0s148m144u205n
```



```
Referencias Correspondencia Revisar Vista Ayuda Buscar
Ubuntu 16.04 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
MU
antonio@antonio-VirtualBox: ~/marite/examples
Can't execute gcc -nostdinc -I/home/antonio/marite/arch/include reloj.cc -m32
-march=i686 /home/antonio/marite/arch/call_main/wrapper_main_c.o -WL,-T,/ho
QEMU
83m700u726n ERROR:0s534m374u470n:0s492m73u296n ERROR:0s542m219u41n:0s497m168u914
n ERROR:0s547m675u40n:0s503m861u104n ERROR:0s553m977u515n:0s509m346u436n ERROR:0
s559m467u876n:0s515m848u378n ERROR:0s566m354u504n:0s523m436u491n ERROR:0s573m558
u769n:0s528m100u490n ERROR:0s578m640u140n:0s536m677u556n ERROR:0s586m799u834n:0s
541m192u374n ERROR:0s591m698u500n:0s549m859u116n ERROR:0s599m988u937n:0s554m319u
458n ERROR:0s604m825u584n:0s561m729u895n ERROR:0s611m871u449n:0s567m412u180n ERR
OR:0s617m558u763n:0s574m982u693n ERROR:0s625m536u590n:0s583m525u396n ERROR:0s633
m642u646n:0s590m49u967n ERROR:0s640m587u940n:0s607m383u450n ERROR:0s657m545u956n
:0s612m127u68n ERROR:0s662m686u832n:0s619m554u267n ERROR:0s669m673u193n:0s624m44
u781n ERROR:0s674m550u906n:0s633m749u922n ERROR:0s683m860u467n:0s639m417u121n ER
ROR:0s689m542u752n:0s644m961u120n ERROR:0s695m485u684n:0s656m419u557n ERROR:0s70
6m531u778n:0s661m833u651n ERROR:0s712m359u891n:0s671m394u640n ERROR:0s721m509u37
5n:0s677m396u239n ERROR:0s727m950u136n:0s686m391u514n ERROR:0s736m539u773n:0s691
m55u513n ERROR:0s741m587u620n:0s698m789u455n ERROR:0s748m902u514n:0s705m127u130n
ERROR:0s755m655u884n:0s715m665u338n ERROR:0s765m781u749n:0s721m523u623n ERROR:0
s771m662u663n:0s726m741u603n ERROR:0s777m273u710n:0s734m307u88n ERROR:0s784m420u
147n:0s739m851u925n ERROR:0s790m388u222n:0s747m424u114n ERROR:0s797m538u849n:0s7
52m979u9n ERROR:0s803m483u458n:0s759m621u750n ERROR:0s809m746u543n:0s766m86u816n
ERROR:0s816m597u970n:0s772m676u758n ERROR:0s822m807u417n:0s778m268u529n ERROR:0
s828m383u264n:0s784m221u518n ERROR:0s834m796u368n:0s791m340u298n ERROR:0s841m480
u176n:0s795m979u154n ERROR:0s846m534u728n:0s803m82u849n ERROR:0s853m294u803n:0s8
10m87u647n ERROR:0s860m592u97n:0s817m617u94n ERROR:0s867m731u829n:0s822m278u579n
ERROR:0s872m790u571n:0s829m770u311n ERROR:0s879m894u265n:0s834m361u396n ERROR:0
s884m478u646n:0s839m995u910n ERROR:0s890m553u997n:0s847m21u661n ERROR:0s897m143u
101n:0s853m94u498n ERROR:0s903m598u948n:0s860m574u497n
```

Una vez corriendo el programa, empezó la investigación de como es que se puede averiguar si el programa se esta corriendo sin la necesidad de imprimir datos.

Se llego a la conclusión de pedir ayuda para encontrar la paquetería adecuada y para ello lo único que se requería era poner el siguiente comando.

`qemu-system-i386 -help`



```
antonio@antonio-VirtualBox: ~/marite/examples
antonio@antonio-VirtualBox:~/marite/examples$ qemu-system-i386 -help
QEMU emulator version 2.5.0 (Debian 1:2.5+dfsg-5ubuntu10.39), Copyright (c) 2003
-2008 Fabrice Bellard
usage: qemu-system-i386 [options] [disk_image]

'disk_image' is a raw hard disk image for IDE hard disk 0

Standard options:
-h or -help      display this help and exit
-version         display version information and exit
-machine [type=]name[,prop[=value]][,...]
                  selects emulated machine ('-machine help' for list)
                  property accel=accel1[:accel2[:...]] selects accelerator
                  supported accelerators are kvm, xen, tcg (default: tcg)
                  kernel_irqchip=on|off controls accelerated irqchip support
                  vmport=on|off|auto controls emulation of vmport (default: auto)
                  kvm_shadow_mem=size of KVM shadow MMU
                  dump-guest-core=on|off include guest memory in a core dump (defa
ult=on)
                  mem-merge=on|off controls memory merge support (default: on)
                  iommu=on|off controls emulated Intel IOMMU (VT-d) support (defau
lt=off)
                  igd-passthru=on|off controls IGD GFX passthrough support (defaul
t=off)
                  aes-key-wrap=on|off controls support for AES key wrapping (defau
lt=on)
                  dea-key-wrap=on|off controls support for DEA key wrapping (defau
lt=on)
                  suppress-vmdesc=on|off disables self-describing migration (defau
lt=off)
-cpu cpu         select CPU ('-cpu help' for list)
-smp [cpus=]n[,maxcpus=cpus][,cores=cores][,threads=threads][,sockets=sockets]
```

En el cual nos puede mencionar los diferentes comandos que se pueden ir agregando a comando principal al ser llamado el programa a qemu.



```
antonio@antonio-VirtualBox: ~/mar/e/examples
is ms
unit is ms
-m[emory] [size=]megs[,slots=n,maxmem=size]
    configure guest RAM
    size: initial amount of guest memory
    slots: number of hotplug slots (default: none)
    maxmem: maximum amount of guest memory (default: none)
NOTE: Some architectures might enforce a specific granularity
-mem-path FILE provide backing storage for guest RAM
-mem-prealloc preallocate guest memory (use with -mem-path)
-k language use keyboard layout (for example 'fr' for French)
-audio-help print list of audio drivers and their options
-soundhw c1,... enable audio support
                and only specified sound cards (comma separated list)
                use '-soundhw help' to get the list of supported cards
                use '-soundhw all' to enable all of them
-balloon none disable balloon device
-balloon virtio[,addr=str] enable virtio balloon device (default)
-device driver[,prop[=value][,...]] add device (based on driver)
                                prop=value,... sets driver properties
                                use '-device help' to print all possible drivers
                                use '-device driver,help' to print all possible properties
-name string1[,process=string2][,debug-threads=on|off]
    set the name of the guest
    string1 sets the window title and string2 the process name (on Linux)
When debug-threads is enabled, individual threads are given a separate name (on Linux)
NOTE: The thread names are for debugging and not a stable API.
-uuid %08x-%04x-%04x-%04x-%012x
```

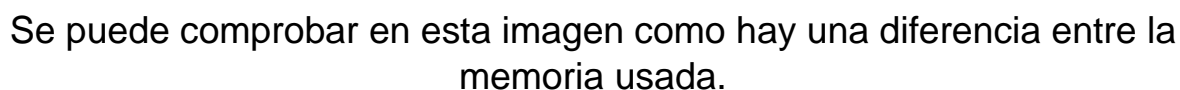
Se logro llegar a estas líneas en la cual mencionan acerca del tamaño del programa que está corriendo en RAM.

Pero al no poderlo compilar de manera correcta ya que me causaban errores. Se logro investigar una forma en la cual se pudiera imprimir el uso de la memoria de Linux.

Para ello se abrieron dos terminales al mismo tiempo. Una en la cual se va a ejecutar el comando con el qemu. Y el otro comando usando el comando free -m.

```
Terminal
< > Carpeta personal Imágenes
Recientes
antonio@antonio-VirtualBox: ~
antonio@antonio-VirtualBox:~$ free -m
              total        used         free       shared  buff/cache   available
Memoria:    1993         617          840           6         536       1197
Swap:        974           0          974
antonio@antonio-VirtualBox:~$
```

Se observa la memoria sin haber corrido el programa.





```
antonio@antonio-VirtualBox: ~
antonio@antonio-VirtualBox:~$ free -m
              total        used        free      shared  bu
Memoria:      1993         617         840          6
Swap:         974           0         974
antonio@antonio-VirtualBox:~$ free -m
              total        used        free      shared  bu
Memoria:      1993         785         670          7
Swap:         974           0         974
antonio@antonio-VirtualBox:~$ free -m
              total        used        free      shared  bu
Memoria:      1993         617         839          6
Swap:         974           0         974
antonio@antonio-VirtualBox:~$
```

Una vez cerrando el qemu se puede comprobar como vuelve a su estado anterior de la memoria.

Con este otro comando también se puede comprobar la diferencia en memoria cuando no se esta corriendo el programa y cuando sí.

cat /proc/meminfo

```
antonio@antonio-VirtualBox: ~
antonio@antonio-VirtualBox:~$ cat /proc/meminfo
MemTotal:      2041112 kB
MemFree:       860004 kB
MemAvailable:  1225924 kB
Buffers:       44900 kB
Cached:       441460 kB
SwapCached:    32 kB
Active:        410316 kB
Inactive:      589952 kB
Active(anon):  42872 kB
Inactive(anon): 477596 kB
Active(file):  367444 kB
Inactive(file): 112356 kB
Unevictable:   48 kB
Mlocked:       48 kB
SwapTotal:     998396 kB
SwapFree:      997616 kB
Dirty:         24 kB
Writeback:     0 kB
AnonPages:     513944 kB
Mapped:        162620 kB
Shmem:         6544 kB
Slab:          62716 kB
SReclaimable:  35348 kB
SUnreclaim:    27368 kB
KernelStack:   6592 kB
PageTables:    28880 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:   2018952 kB
Committed_AS:  3711516 kB
VmallocTotal:  34359738367 kB
```

```
antonio@antonio-VirtualBox: ~
antonio@antonio-VirtualBox:~$ cat /proc/meminfo
MemTotal:      2041112 kB
MemFree:       686332 kB
MemAvailable:  1052344 kB
Buffers:       44908 kB
Cached:       442656 kB
SwapCached:    32 kB
Active:        581004 kB
Inactive:      591108 kB
Active(anon):  213532 kB
Inactive(anon): 478688 kB
Active(file):  367472 kB
Inactive(file): 112420 kB
Unevictable:   48 kB
Mlocked:       48 kB
SwapTotal:     998396 kB
SwapFree:      997616 kB
Dirty:         56 kB
Writeback:     0 kB
AnonPages:     684584 kB
Mapped:        171516 kB
Shmem:         7672 kB
Slab:          62740 kB
SReclaimable:  35348 kB
SUnreclaim:    27392 kB
KernelStack:   6656 kB
PageTables:    29864 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:   2018952 kB
Committed_AS:  4268716 kB
VmallocTotal:  34359738367 kB
```



La imagen de la izquierda es sin el programa corriendo y la de la derecha es corriéndolo.

Este otro comando también se comprobó la diferencia.

`vmstat -s`

```
antonio@antonio-VirtualBox: ~  
antonio@antonio-VirtualBox:~$ vmstat -s  
2041112 K memoria total  
633672 K memoria usada  
410556 K memoria activa  
589900 K memoria inactiva  
858268 K memoria libre  
44964 K memoria de búfer  
504208 K caché de intercambio  
998396 K total de intercambio  
780 K intercambio usado  
997616 K intercambio libre  
192563 tics de CPU de usuario no-«  
217 tics de CPU del usuario «ni  
22987 tics de CPU del sistema  
2070215 tics de CPU de inactividad  
41758 tics de CPU de espera E/S  
0 tics de CPU de IRQ  
691 tics de CPU de softirq  
0 tics de CPU robados  
1113297 páginas en entrada  
5820084 páginas en salida  
16 páginas intercambiadas  
159 páginas cambiadas  
3039629 interrupciones  
5844255 cambios de contexto de CPU  
1560316903 tiempo de arranque  
14651 bifurcaciones  
antonio@antonio-VirtualBox:~$
```

```
antonio@antonio-VirtualBox: ~  
antonio@antonio-VirtualBox:~$ vmstat -s  
2041112 K memoria total  
805096 K memoria usada  
581480 K memoria activa  
591056 K memoria inactiva  
685580 K memoria libre  
44996 K memoria de búfer  
505440 K caché de intercambio  
998396 K total de intercambio  
780 K intercambio usado  
997616 K intercambio libre  
194507 tics de CPU de usuario no-«  
217 tics de CPU del usuario «ni  
23114 tics de CPU del sistema  
2076408 tics de CPU de inactividad  
41764 tics de CPU de espera E/S  
0 tics de CPU de IRQ  
692 tics de CPU de softirq  
0 tics de CPU robados  
1113297 páginas en entrada  
5820292 páginas en salida  
16 páginas intercambiadas  
159 páginas cambiadas  
3066411 interrupciones  
5896644 cambios de contexto de CPU  
1560316903 tiempo de arranque  
14659 bifurcaciones  
antonio@antonio-VirtualBox:~$
```

Y por ultimo se utilizo el comando `top`. El cual se puede demostrar como al utilizar el programa `qemu` que esta corriendo se ve que usa el 98% del CPU.



```
antonio@antonio-VirtualBox: ~  
top - 03:41:06 up 3:19, 1 user, load average: 0.11, 0.55, 0.53  
Tareas: 182 total, 1 ejecutar, 144 hibernar, 2 detener, 0 zombie  
%Cpu(s): 0.3 usuario, 0.2 sist, 0.0 adecuado, 99.5 inact, 0.0 en espera, 0.  
KiB Mem : 2041112 total, 857020 free, 634652 used, 549440 buff/cache  
KiB Swap: 998396 total, 997616 free, 780 used, 1223324 avail Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1064	root	20	0	395124	62668	30672	S	0.3	3.1	2:20.14	Xorg
2100	antonio	20	0	1661132	115524	69364	S	0.3	5.7	3:22.13	compiz
7499	antonio	20	0	678952	42000	28756	S	0.3	2.1	0:34.08	gnome-term+
9748	root	20	0	0	0	0	I	0.3	0.0	0:00.57	kworker/1:2
14672	antonio	20	0	48964	3740	3152	R	0.3	0.2	0:00.10	top
1	root	20	0	119868	5820	3816	S	0.0	0.3	0:02.46	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.27	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:05.78	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.05	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.06	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/1
16	root	20	0	0	0	0	S	0.0	0.0	0:00.58	ksoftirqd/1
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:+
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_+
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper

```
antonio@antonio-VirtualBox: ~  
top - 03:41:27 up 3:19, 1 user, load average: 0.23, 0.55, 0.53  
Tareas: 183 total, 2 ejecutar, 144 hibernar, 2 detener, 0 zombie  
%Cpu(s): 51.3 usuario, 1.8 sist, 0.0 adecuado, 46.8 inact, 0.0 en espera, 0.  
KiB Mem : 2041112 total, 684456 free, 805948 used, 550708 buff/cache  
KiB Swap: 998396 total, 997616 free, 780 used, 1050780 avail Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
14674	antonio	20	0	984808	195056	25484	S	98.0	9.6	0:12.61	qemu-syste+
2100	antonio	20	0	1661132	115524	69364	R	5.7	5.7	3:24.11	compiz
1064	root	20	0	397632	63796	31800	S	3.3	3.1	2:21.11	Xorg
7499	antonio	20	0	678952	42000	28756	S	0.3	2.1	0:34.26	gnome-term+
14672	antonio	20	0	48964	3740	3152	R	0.3	0.2	0:00.15	top
1	root	20	0	119868	5820	3816	S	0.0	0.3	0:02.46	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_+
7	root	20	0	0	0	0	S	0.0	0.0	0:00.27	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:05.81	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.05	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.06	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/1
16	root	20	0	0	0	0	S	0.0	0.0	0:00.58	ksoftirqd/1
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:+
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_+
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper



Conclusiones.

```
antonio@antonio-VirtualBox: ~  
top - 03:41:27 up 3:19, 1 user, load average: 0.23, 0.55, 0.53  
Tareas: 183 total, 2 ejecutar, 144 hibernar, 2 detener, 0 zombie  
%Cpu(s): 51.3 usuario, 1.8 sist, 0.0 adecuado, 46.8 inact, 0.0 en espera, 0.  
KiB Mem : 2041112 total, 684456 free, 805948 used, 550708 buff/cache  
KiB Swap: 998396 total, 997616 free, 780 used. 1050780 avail Mem  


| PID   | USUARIO | PR | NI | VIRT    | RES    | SHR   | S | %CPU | %MEM | HORA+   | ORDEN       |
|-------|---------|----|----|---------|--------|-------|---|------|------|---------|-------------|
| 14674 | antonio | 20 | 0  | 984808  | 195056 | 25484 | S | 98.0 | 9.6  | 0:12.61 | qemu-syste+ |
| 2100  | antonio | 20 | 0  | 1661132 | 115524 | 60364 | R | 5.7  | 5.7  | 3:24.11 | compiz      |


```

Se pudo comprobar el funcionamiento del programa y la validación del uso de la memoria, sin que se requiera la impresión de datos, al correr el programa.



Cibergrafía:

<https://www.adacore.com/download/more>

<https://marte.unican.es/downloads.htm>

https://marte.unican.es/documentation/install/INSTALL_2.0_22Feb2017.txt

<file:///C:/Users/anton/Downloads/Sin%20t%C3%ADtulo.pdf>

https://wiki.osdev.org/GRUB_Legacy#Installing_to_hard_disk

<https://marte.unican.es/documentation/tutorials/booting-marte-howto.pdf>

<https://marte.unican.es/documentation/tutorials/hello-marte-howto.pdf>

<https://es.wikibooks.org/wiki/QEMU/Monitor>

<https://elpuig.xeill.net/Members/vcarceler/articulos/qemu>

<https://dangokyo.me/2018/03/27/qemu-internal-memory-region-address-space-and-qemu-io/>

<https://stackoverflow.com/questions/22300739/wrong-memory-locations-when-debugging-in-qemu-with-gdb>

https://wiki.qemu.org/Documentation/Platforms/PowerPC#Debugging_qemu-system-ppc_with_GDB_on_an_x86_Linux_host

https://qemu.weilnetz.de/doc/qemu-doc.html#pcsys_005fmonitor

<https://www.binarytides.com/linux-command-check-memory-usage/>