



**INSTITUTO POLITÉCNICO NACIONAL**



**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA  
Y TECNOLOGÍAS AVANZADAS**

**UPIITA**

**Reporte: Uso y comprobación de MarteOS**

**SISTEMAS OPERATIVOS EN TIEMPO REAL**

**3MV9**

**REYES GARCÍA MAURICIO ITZAE**

## INTRODUCCIÓN

QEMU es un emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped). QEMU también tiene capacidades de virtualización dentro de un sistema operativo, ya sea GNU/Linux, Windows, o cualquiera de los sistemas operativos admitidos; de hecho es la forma más común de uso. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (x86, x86-64, PowerPC, MIPS, SPARC, etc.). Está licenciado en parte con la LGPL y la GPL de GNU.

El objetivo principal es emular un sistema operativo dentro de otro sin tener que reparticionar el disco duro, empleando para su ubicación cualquier directorio dentro de éste.

El programa no dispone de GUI, pero existe otro programa llamado QEMU manager que puede hacer de interfaz gráfica si se utiliza QEMU desde Windows. También existe una versión para GNU/Linux llamada qemu-launcher. En Mac OS X puede utilizarse el programa Q que dispone de una interfaz gráfica para crear y administrar las máquinas virtuales.

QEMU emula un sistema informático completo, incluyendo procesador y varios periféricos. Este puede ser usado para proveer hosting virtual a varios ordenadores virtuales en un único ordenador. QEMU puede arrancar varios sistemas operativos, incluyendo entre otros Linux, Microsoft Windows, DOS, y BSD. Admite además la emulación de varias plataformas de hardware, incluyendo x86, AMD64, Alpha, Mips, y Sparc.

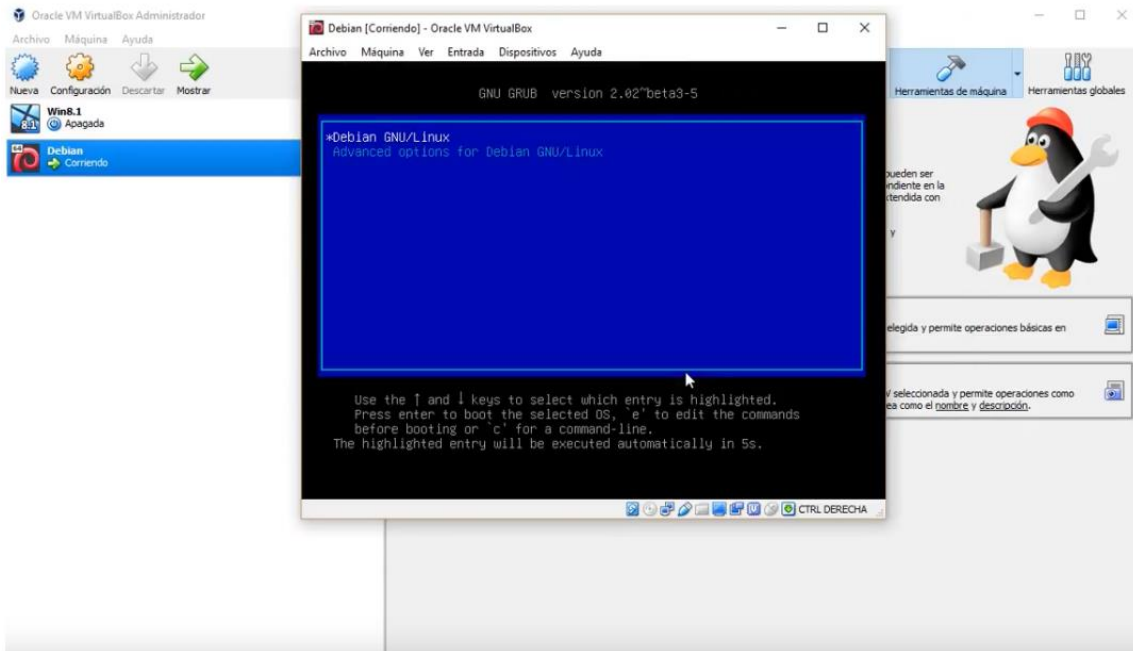
MarteOS es un sistema operativo de tiempo real para aplicaciones embebidas que sigue a la mínima en tiempo real POSIX.13. El proyecto se desarrolla por el Grupo de computación y Tiempo Real en la Universidad de Cantabria, aunque también existen colaboradores en distintos lugares.

El entorno de desarrollo se basa en la GNU compiladores GNAT, GCC y GJC. La mayor parte de su código está escrito en Ada con algunas partes C y ensamblador.

## DESARROLLO

El ejercicio, comienza con la instalación del compilador marte OS y la inicialización de una máquina virtual en Windos. Una vez inicializada, se prepara el archivo instalador de Debian, ofrecido por Linux, descargado de su página oficial, para su corrimiento en una máquina virtual.

Dentro de este, al crear una nueva máquina virtual, se selecciona el tipo de sistema (Debian) y a continuación, se le asigna un nombre para identificar al sistema. Una vez instalado y después de haber creado a los usuarios, se procede a la ejecución del Debian dentro del ambiente del Virtual Box



Teniendo ya instalado Debian, se procede con la instalación de marte OS, creando un acceso de este dentro del ambiente de la máquina virtual.

Posteriormente, teniendo ya instalada MarteOS, se procede con la instalación del QEMU, para la generación de la terminal con la cual se va a estar trabajando.

Una vez teniendo la terminal, se procede a crear una imagen ISO por medio de este QEMU, para que este pueda ser montado y escrito las veces que queramos. Las instrucciones de cómo se realiza este tipo de acciones se obtuvieron de la siguiente fuente:

[https://wiki.archlinux.org/index.php/QEMU\\_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/index.php/QEMU_(Espa%C3%B1ol))

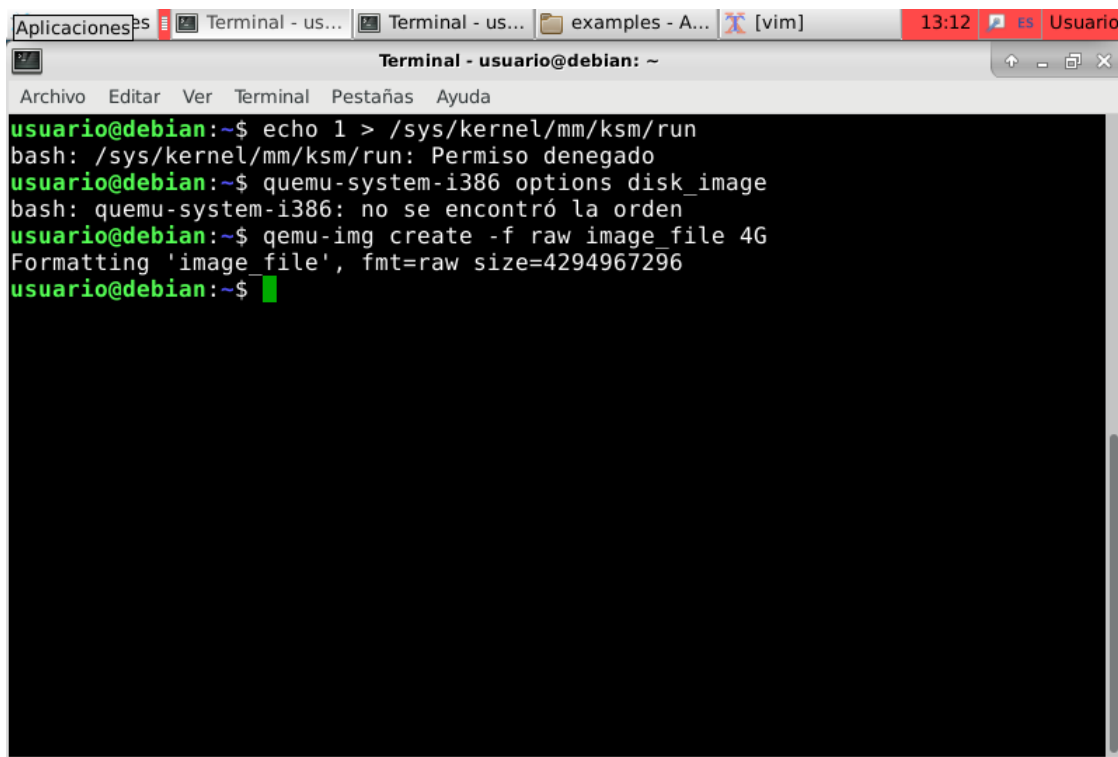
Dichas instrucciones se muestran a continuación, tal y como se encontraron en la página web.

La imagen de disco puede ser en "crudo", de manera que, literalmente, byte por byte es lo mismo que el cliente ve, y siempre utilizará toda la capacidad del disco duro del disco duro invitado en el

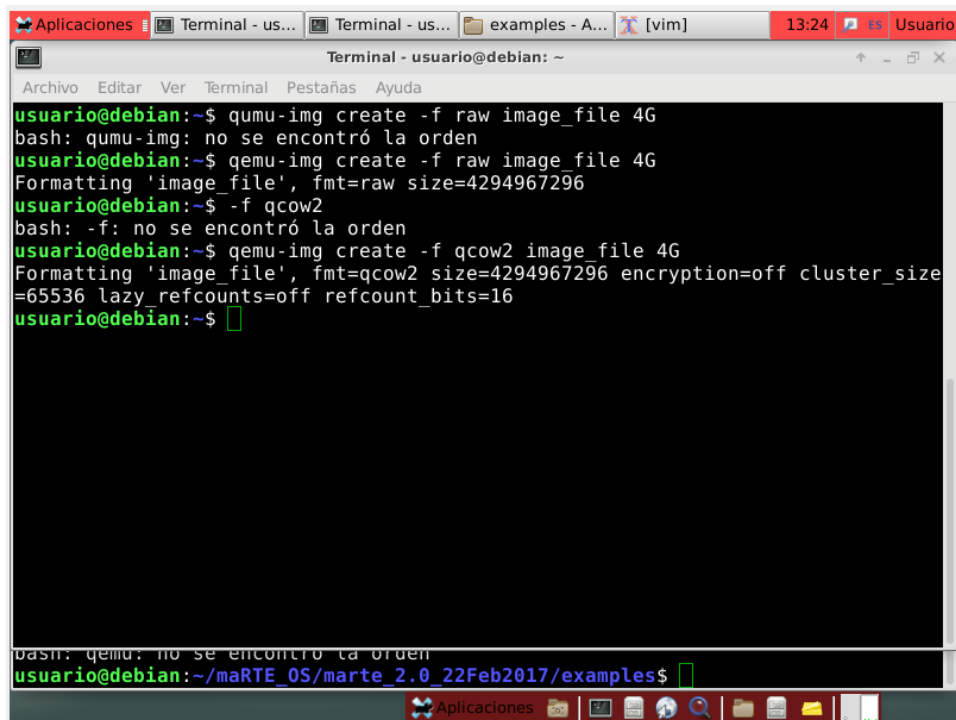
host. Este método proporciona la menor sobrecarga de Entrada / Salida, pero puede desperdiciar una gran cantidad de espacio, ya que el espacio no utilizado por el invitado no se puede utilizar en el host.

Por otra parte, la imagen de disco duro puede estar en un formato tal como el de qcow2 el cuál únicamente asigna espacio a el archivo de la imagen cuando el SO invitado está escribiendo en los sectores del disco virtual. La imagen aparece como el tamaño total del sistema operativo huésped, a pesar que puede tomar hasta una cantidad muy pequeña de espacio en el sistema host. El uso de este formato en lugar del "crudo" probablemente afecte el rendimiento.

QEMU proporciona el `qemu-img` comando para crear imágenes de disco. Por ejemplo, para crear una imagen de 4GB en formato "crudo":

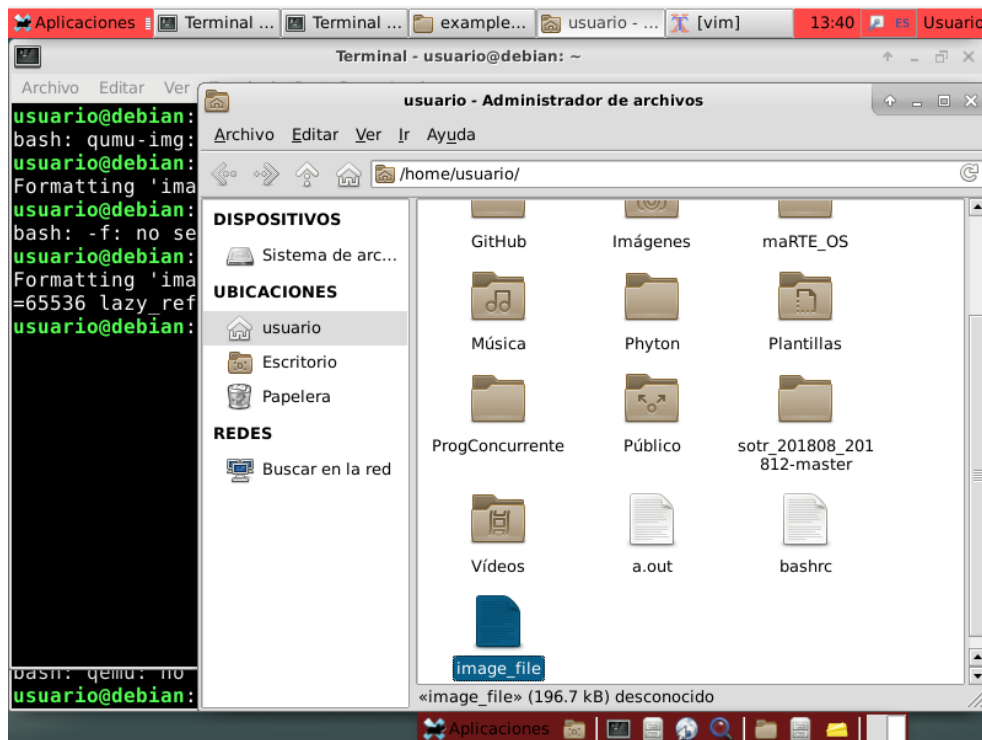
A screenshot of a terminal window titled "Terminal - usuario@debian: ~". The window shows a series of commands and their outputs. The first command is `echo 1 > /sys/kernel/mm/ksm/run`, which results in a "Permiso denegado" (Permission denied) error. The second command is `qemu-system-i386 options disk_image`, which results in a "no se encontró la orden" (command not found) error. The third command is `qemu-img create -f raw image_file 4G`, which successfully creates a 4GB raw image file, displaying the message "Formatting 'image\_file', fmt=raw size=4294967296". The terminal window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The top status bar shows the time as 13:12 and the user as Usuario.

Se puede utilizar `-f qcow2` para crear un disco `qcow2` en su lugar. Y se realiza de la siguiente manera.



```
usuario@debian:~$ qumu-img create -f raw image_file 4G
bash: qumu-img: no se encontró la orden
usuario@debian:~$ qemu-img create -f raw image file 4G
Formatting 'image_file', fmt=raw size=4294967296
usuario@debian:~$ -f qcow2
bash: -f: no se encontró la orden
usuario@debian:~$ qemu-img create -f qcow2 image_file 4G
Formatting 'image_file', fmt=qcow2 size=4294967296 encryption=off cluster_size
=65536 lazy_refcounts=off refcount_bits=16
usuario@debian:~$
```

La forma en la que se comprueba la ejecución de este ejemplo, sin la necesidad de visualizarlo sobre el QEMU, es mediante la exploración en la carpeta de archivos del usuario, aquí se establecieron los archivos creados.



## **CONCLUSIÓN**

La verificación, en este caso, para la creación y corrimiento de un ejemplo, como lo es, la creación de una imagen ISO, montable para cualquier fin y su respectiva creación en la carpeta de archivos de la máquina, en el sistema Operativo. Es una manera de comprobar la compilación y ejecución del QEMU en Marte OS.

Ejemplos como éstos hay muchos, pero la creación inminente de archivos asegura que la correlación entre plataformas están activas, por lo cual es una señal de que los sistemas operativos, también pueden ser monitoreados en tiempo real, así mismo, también se puede verificar car sus rastros de ellos mediante la exploración de archivos o buscando en registros de la memoria RAM.