

PRACTICA 1 Instalación de un Sistema Operativo de Tiempo Real

1 Instalar Debian GNU-Linux de 64 bits en una máquina virtual de Virtualbox, o bien, descargar e instalar gratuitamente la aplicación Debian de la tienda Microsoft Store.

Ajuste del ambiente de compilación Linux:

Instalar los paquetes build-essential, libc6-dev-i386, ejecutando el comando:

```
$ sudo apt-get install build-essential
```

2 Instalación del compilador nativo AdaCore GNAT

MaRTE OS está escrito en Ada, entonces aun si usted no va a programar aplicaciones Ada, usted necesitará el compilador GNAT para compilar el kernel MaRTE. Instalarlo es muy fácil, y se puede hacer en su propio directorio de usuario sin afectar al resto del sistema. Cada versión de MaRTE OS tiene soporte para un conjunto de versiones de compiladores Adacore. Asegúrese de usar el compilador correcto, en caso contrario, el instalador de MaRTE OS se quejará.

Instalar el compilador GNAT de Adacore es muy simple. Primero deberá descargar el archivo de 356 MB:

gnat-gpl-2016-x86_64-linux-bin.tar.gz

A continuación, se plantean dos opciones para descargar este archivo.

OPCION 1 para descargar el archivo gnat-gpl-2016-x86_64-linux-bin.tar.gz

Descargue los archivos archivo01, archivo02, . . . , archivoN

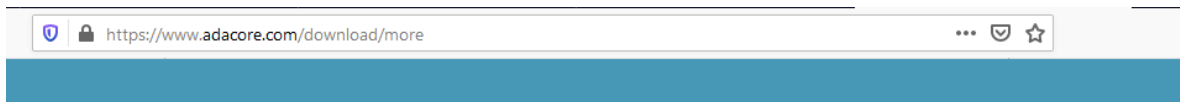
NOTA: La OPCION 1 aún está en construcción.

OPCION 2 para descargar el archivo gnat-gpl-2016-x86_64-linux-bin.tar.gz

Instalación del compilador GNAT de Adacore

Valla al sitio web de Adacore

<https://www.adacore.com/download/more>, seleccione x86 GNU Linux (64 bits) y 2016:



Select your platform

x86 GNU Linux (64 bits) 2016

x86 GNU Linux (64 bits)

GNAT GPL Ada

README.txt

SHA-1: 18da74a0ac9df2753bfd9b3d2dcad6e3379825c2

3.3 KiB

May 16 2016

gnat-gpl-2016-x86_64-linux-bin.tar.gz

SHA-1: 845147a8c6ef6af29a68144d6b3d228fd226268e

355.5 MiB

May 16 2016

[Sources](#)

Una vez descargado el compilador, usted debe descomprimirlo y ejecutar un script simple.

Para la versión GNAT-GPL-2016 ejecute:

```
$ cd $HOME/myapps
```

```
$ tar -zxvf gnat-gpl-2016-x86_64-linux-bin.tar.gz
```

```
$ cd gnat-gpl-2016-x86_64-linux-bin
```

```
$ ./doinstall
```

Elija un directorio de instalación no estándar y especifique dónde quiere tener el compilador (i.e: \$HOME/myapps)

Recuerde poner el directorio donde quedan localizados los binarios de GNAT al frente de su variable PATH:

```
$ export PATH=$HOME/myapps/gnat/bin:$PATH
```

Para hacer esto permanente usted puede agregar el comando previo a su script de inicio de Shell (i.e: \$HOME/.bashrc o \$HOME/.bash_profile).

3 Instalación de MaRTE OS

Descargue el código fuente de MaRTE OS

marTE_2.0_22Feb2017_src.tar.gz

Puede descargarlo de la página web

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

(se incluye imagen de la página de MaRTE OS)



Extraer y ejecutar mininstall

MaRTE OS se instala fácilmente. Para instalarlo, usted debe descomprimirlo y correr un script. Recuerde que el compilador GNAT debe estar al principio de su variable PATH (para comprobarlo puede ejecutar `echo $PATH`). Ejecute los siguientes comandos

```
$ cd $HOME
```

```
$ tar -zxvf marte-version
```

```
$ cd marte_2.0_22Feb2017
```

```
$ ./mininstall
```

mininstall tratará de encontrar su path de instalación de GNAT y hacer unos cuantos enlaces simbólicos.

Compilar las bibliotecas de MaRTE

Después de que mininstall ha finalizado su trabajo proporciona la lista de arquitecturas disponibles (de acuerdo al GNAT/GCC instalado en su sistema). Debe colocar el directorio `utils/` en su variable PATH para tener acceso directo a las herramientas de MaRTE:

```
$ export PATH=$PATH:$HOME/marte_2.0_22Feb2017/utils
```

Para hacer este cambio permanente usted puede agregar el comando previo al script de inicialización de su Shell (i.e: `$HOME/.bashrc` o `$HOME/.bash_profile`).

El siguiente paso es compilar el kernel de MaRTE y bibliotecas. Para hacerlo, establezca la arquitectura usando el script `msetcurrentarch`:

```
$ msetcurrentarch x86 i386
```

(- i386: Intel x86 processor. PIT is used as timer and clock.)

Construir GNAT RTS modificado para MaRTE:

```
$ mkrtsmarteuc
```

Construir la biblioteca `libmarte.a`:

```
$ mkmarte
```

Una vez hecho lo anterior, en este punto, debe existir un ambiente MaRTE OS funcional en su sistema operativo. Para revisar que funciona bien usted puede compilar un ejemplo del directorio example/ de las fuentes de MaRTE.

```
$ cd $HOME/marte_2.0_22Feb2017/example
```

```
$ make hello_world_c.c (para una aplicación de C).
```

Con esto deberá construirse el archivo hello_world_c.exe

También puede usar el comando

```
Mgcc hello_world_c.c -o mprogram
```

Ejecutando la aplicación

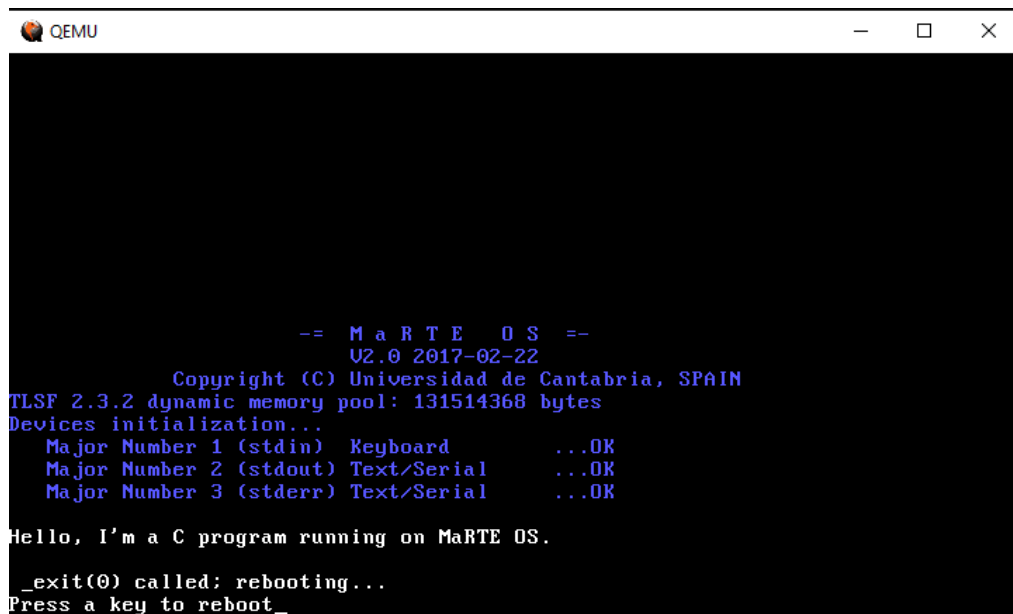
Para ejecutar la aplicación usando qemu puede usar:

```
$ qemu-system-i386 -kernel hello_world_c.exe
```

O también

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

La siguiente figura muestra un ejemplo de ejecución del programa hola mundo sobre MaRTE OS en qemu:



```
-- 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

Hello, I'm a C program running on MaRTE OS.

_exit(0) called; rebooting...
Press a key to reboot_
```