

GUIA DE INSTALACION DE MARTEOS

Primeramente para comenzar con la instalación se debe crear la carpeta que almacenará las fuentes.

```
usuario@debian:~$ mkdir myapps
```

Con el comando "ls" observamos la carpeta

```
usuario@debian:~$ ls
Desktop  Downloads  myapps  Public  Videos
Documents  Music      Pictures  Templates
usuario@debian:~$ cd myapps/
```

Posteriormente dentro de esta copiamos los repositorios.

```
usuario@debian:~/myapps$ wget http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b
```

```
usuario@debian:~/myapps$ wget https://mar.te.unican.es/marte/marte_2.0_22Feb2017_src.tar.gz
```

Y creamos una carpeta con el nombre de "gnat"

```
usuario@debian:~/myapps$ mkdir gnat
```

Con el comando "tar xvf" des comprimimos los archivos descargados.

```
5739cefdc7a447658e0b016b  gnat  marte_2.0_22Feb2017_src.tar.gz
usuario@debian:~/myapps$ tar xvf 5739cefdc7a447658e0b016b
```

Se puede visualizar los archivos descomprimidos.

```
5739cefdc7a447658e0b016b  gnat-gpl-2016-x86_64-linux-bin
gnat                      marte_2.0_22Feb2017_src.tar.gz
usuario@debian:~/myapps$ tar xvf marte_2.0_22Feb2017_src.tar.gz
```

Posteriormente escribimos "cd" para ir a la raíz del sistema y escribimos el comando ".bashrc"

```
usuario@debian:~/myapps$ cd
usuario@debian:~$ nano .bashrc
```

Ya dentro del sistema procedemos a escribir las siguientes líneas de código al final de todo el texto.

```
export PATH=$HOME/myapps/gnat/bin:$PATH
export PERL5LIB=$HOME/myapps/marte_2.0_22Feb2017
export PATH=$PATH:$HOME/myapps/marte_2.0_22Feb2017/utils
#export PATH=/opt/cross-pi-gcc/bin:$PATH
```

Continuamos con la instalación regresando a la carpeta “myapps” y entramos a las fuentes de gnat.

```
|usuario@debian:~/myapps$ cd gnat-gpl-2016-x86_64-linux-bin/
```

Dentro escribimos el comando “./doinstall”

```
|usuario@debian:~/myapps/gnat-gpl-2016-x86_64-linux-bin$ ./doinstall
```

Por lo tanto se mostrara la siguiente ventana.

```
This script is provided to simplify the installation of the
x86_64-pc-linux-gnu
binary version of the GNAT GPL Edition Ada (2012, 2005, 95, and 83)
environment maintained by AdaCore. For information on commercial
support please contact sales@adacore.com.
```

```
This script will ask you how you would like to configure GNAT.
Confirmation is required before any write action is taken.
```

```
Please press RETURN to continue.
```

Presionamos enter para continuar. Y muestra el siguiente mensaje.

```
To install GNAT, you need to specify a base directory.
All the files will be installed in subdirectories
that are created under this directory.
```

```
Important Note: You should not use ~ or ~username wildcards
when specifying this directory name.
```

```
In which directory do you want to install GNAT? [/usr/gnat]:
/home/usuario/myapps/gnat
```

```
The GNAT GPL Edition installation directory will be:
/home/usuario/myapps/gnat
Is this correct ? Type 'Y' if so, otherwise type 'N' and you'll
be prompted for another directory name.
```

```
Do you want to continue ? [yY|nN]:
```

Presionamos Y y enter .

```
GNAT GPL Edition is now about to be installed in
/home/usuario/myapps/gnat.
Type 'Y' if you want to proceed with installation or any other key
if you wish to abort.
```

```
Do you want to proceed with installation ? [yY|nN]:
```

Con esto gnat estará instalado y mostrara el siguiente mensaje.

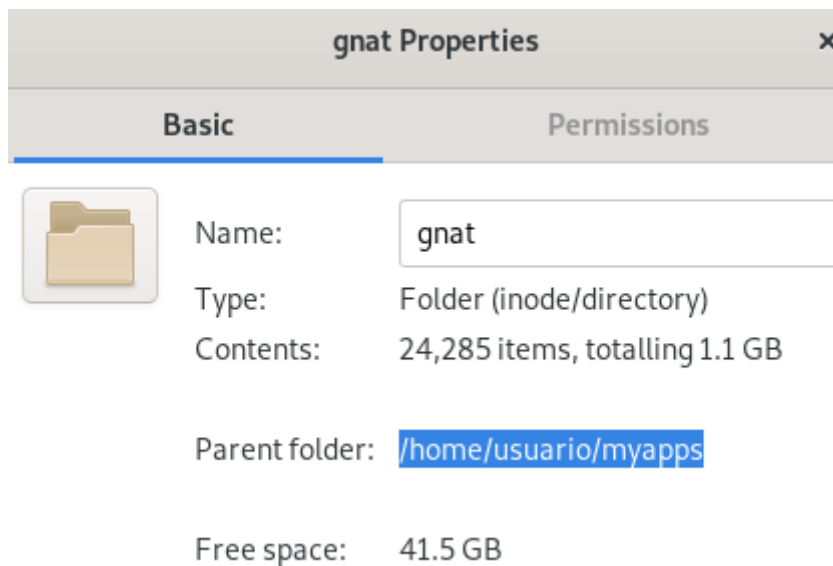
```
GNAT GPL is now installed. To launch it, you must put
/home/usuario/myapps/gnat/bin
in front of your PATH environment variable. The following
commands enable you to do this:
  PATH="/home/usuario/myapps/gnat/bin:$PATH"; export PATH  (Bourne shell)
  setenv PATH "/home/usuario/myapps/gnat/bin:$PATH"        (C shell)
Thank you for installing GNAT GPL Edition!
```

```
usuario@debian:~/myapps/gnat-gpl-2016-x86_64-linux-bin$
```

Posteriormente procederemos a instalar marte, regresamos a “myapps” y entramos en las fuentes de marte

```
usuario@debian:~/myapps/gnat-gpl-2016-x86_64-linux-bin$ cd ..
usuario@debian:~/myapps$ ls
5739cefdc7a447658e0b016b      marte_2.0_22Feb2017
gnat                          marte_2.0_22Feb2017_src.tar.gz
gnat-gpl-2016-x86_64-linux-bin
usuario@debian:~/myapps$ cd marte_2.0_22Feb2017/
```

En una ventana aparte buscamos la carpeta de gnat y observamos la dirección de esta en el sistema.



Ya con esto escribimos el comando “./minstall” y nos mostrara el siguiente mensaje, en el cual presionamos enter para continuar.

```

usuario@debian:~/myapps/marte_2.0_22Feb2017$ ./minstall
main::check_if_gnat_is_present() called too early to check prototype at ./minstall line 30.
Use of uninitialized value in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utils/templates/globals.pl line 16.

Welcome to MaRTE OS installation process
=====
http://martel.unican.es/
Universidad de Cantabria, SPAIN

To perform successfully this installation compilers:
- GNAT-GPL-2016-x86_64 (for x86, linux and linux_lib architectures)
- GNAT-GPL-2016-arm-elf (for rpi architecture)
should already be installed in your system, and GNAT 'bin/' directory
should be in front of your $PATH environment variable.

You also need to have write permission on the GNAT compiler directory.

Press any key to start the installation...

```

Se instalaran los paquetes y mostrará el siguiente mensaje.

```

--== :-) MaRTE OS installation script finished :-) ==--

You may want to add "/home/usuario/myapps/marte_2.0_22Feb2017/utils"
to your $PATH environment variable to have direct access to MaRTE
tools (mgnatmake, mgcc, mkmarte, mkrtsmarteuc, msetcurrentarch, etc.)

In this installation, MaRTE OS can generate applications for the
following architectures:

- linux: Linux operating system
- x86: x86 bare machine
- linux_lib: Linux operating system (using Linux file system)

```

This is a MaRTE source distribution, so you must compile MaRTE libraries before using them. For example, for "linux" architecture execute:

```
$ msetcurrentarch linux && mkrtsmarteuc && mkmarte
```

For more information go to chapter 1.2 of the 'INSTALL' document

```
usuario@debian:~/myapps/marte_2.0_22Feb2017$
```

Posteriormente entramos en "utils"

```
usuario@debian:~/myapps/marte_2.0_22Feb2017$ cd utils/
```

E escribimos el siguiente comando “msetcurrentarch” y posteriormente “msetcurrentarch x86 i386”.

```
usuario@debian:~/myapps/marte_2.0_22Feb2017/utils$ msetcurrentarch
Use of uninitialized value in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utils/globals.pl line 16.
Current architecture:none

Available architectures status:
x86:      RTS (gnat_rts/rts-marteuc_x86): NOT Compiled
          Lib MaRTE (objs/x86_objs):      NOT Compiled
linux:    RTS (gnat_rts/rts-marteuc_linux): NOT Compiled
          Lib MaRTE (objs/linux_objs):     NOT Compiled
linux_lib: RTS (gnat_rts/rts-marteuc_linux_lib): NOT Compiled
          Lib MaRTE (objs/linux_lib_objs): NOT Compiled
rpi:      NOT available

usuario@debian:~/myapps/marte_2.0_22Feb2017/utils$ msetcurrentarch x86 i386
```

Mostrando el siguiente mensaje, en este se debe el comando “mkrtsmarteuc”

```
msetcurrentarch:x86 set as the default architecture...OK

GNAT/MaRTE RTS for this architecture is not compiled yet.
To compile it execute command: 'mkrtsmarteuc && mkmarte'
usuario@debian:~/myapps/marte_2.0_22Feb2017/utils$
```

Y posteriormente “mkmarte”

```
cd /home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4
ve/adalib      adalib
mkrtsmarteuc: rts-marteuc_x86 done :)

Lib MaRTE for this architecture is not compiled yet
Run 'mkmarte'
usuario@debian:~/myapps/marte_2.0_22Feb2017/utils$
```

Al finalizar se mostrara el mensaje de que fue instalado exitosamente.

```
Linking libustl.a ...
ar: creating libustl.a
make[1]: Leaving directory '/home/usuario/myapps/marte_2.0_22Feb2017/lang_support/ustl-src'
make -C ustl-src/ install
make[1]: Entering directory '/home/usuario/myapps/marte_2.0_22Feb2017/lang_support/ustl-src'
Installing headers to ../../x86_arch/include/ustl ...
Installing libustl.a to ../../lib ...
make[1]: Leaving directory '/home/usuario/myapps/marte_2.0_22Feb2017/lang_support/ustl-src'
C++ language support library DONE

mkmarte: work done :-)
usuario@debian:~/myapps/marte_2.0_22Feb2017/utils$
```

Posteriormente procedemos a ingresar a la carpeta de ejemplos.

```
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ ls
ada                hardware_interrupts  Makefile            time_measurement
appsched           hello_world.adb      oscilloscope        widgets
clock_modulation   hello_world.c.c      posix
drivers            hello_world_cc.cc    README
games              logger               speaker
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$
```

Y compilamos el archivo “hello_world_c.c” de la siguiente forma, con lo cual se creara un archivo .out.

```
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ mgcc hello_world_c.c
Use of uninitialized value in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utils/globals.pl line 16.
gcc -nostdinc -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/include hello_world_c.c -m32 -march=i686 /home/usuario/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main.c.o -WL,-T,/home/usuario/myapps/marte_2.0_22Feb2017/utils/linker.lds -static -nostartfiles -L/home/usuario/myapps/marte_2.0_22Feb2017/lib -L/home/usuario/myapps/marte_2.0_22Feb2017/gnat_rts/rts/adalib -L/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnarl -lgnat -lmarte -lgcc_sjlj
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ ls
ada                games                logger               speaker
a.out              hardware_interrupts  Makefile            time_measurement
appsched           hello_world.adb      oscilloscope        widgets
clock_modulation   hello_world.c.c      posix
drivers            hello_world_cc.cc    README
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$
```

Posteriormente creamos mediante make el archivo ejecutable de este.

```
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ make hello_world_c.exe
>> Compiling hello_world_c.exe: Use of uninitialized value in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utils/globals.pl line 16.
[OK]
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ ls
ada                games                hello_world_c.exe    README
a.out              hardware_interrupts  logger               speaker
appsched           hello_world.adb      Makefile             time_measurement
clock_modulation   hello_world.c.c      oscilloscope         widgets
drivers            hello_world_cc.cc    posix
usuario@debian:~/myapps/marte_2.0_22Feb2017/examples$ qemu-system-i386 -kernel h
sello_world_c.exe
```

Y procedemos a emularlo con qemu usando el comando “qemu-system-i386 -kernel hello_world_c.exe” con lo cual tendremos marteos ejecutándose exitosamente.

```

                -= M a R T E   O S   -=
                U2.0 2017-02-22
                Copyright (C) Universidad de Cantabria, SPAIN
TLSF 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_
```