



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS AVANZADAS



SISTEMAS OPERATIVOS EN TIEMPO REAL

Instalación del Sistema Operativo en Tiempo Real Marte OS

PROFESOR: MAZA CASAS LAMBERTO

GRUPO: 3MV11

ALUMNA:

RAMIREZ PERALTA LILLYAN YAZMIN

Pasos previos a la instalación de Marte OS

Antes de comenzar la instalación del sistema operativo Marte OS es necesario tener un subsistema o distribución de Linux para poder trabajar. En este caso, se muestra la instalación del sistema operativo utilizando Debian. Si se cuenta con Windows, es necesario realizar los pasos que se muestran a continuación para instalar Debian.

1. Correr PowerShell como administrador y escribir lo siguiente:
`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`
2. El programa tardará unos segundos realizando varias acciones y luego te pedirá que reinicies el equipo.
3. Luego de reiniciar, ya es posible instalar Debian, el cual se puede obtener de la liga:
<https://www.microsoft.com/es-mx/p/debian/9msvkqc78pk6?rtc=1&activetab=pivot:overviewtab>
4. Después de descargar Debian es necesario seguir los pasos que indica el instalador.

Una vez que se ha instalado y abierto Debian, es necesario ahora instalar los paquetes *vim*, *qemu*, *git* y *make*, ya que serán necesarios en este procedimiento. Esto se hace escribiendo el siguiente comando en Debian, e ingresando la contraseña del usuario, para cada uno de los paquetes.

```
sudo apt-get install vim
```

Se necesitan además dos archivos, los cuales se pueden descargar de los siguientes enlaces:

- <http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b>
- https://marte.unican.es/marte/marte_2.0_22Feb2017_src.tar.gz

En el primer enlace se encuentra GNAT, que es un compilador de lenguaje Ada, basado en la infraestructura de GCC. Mientras que en el segundo enlace se encuentra el sistema operativo MareteOS.

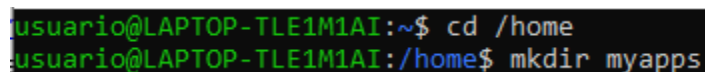
Instalación del sistema operativo MarteOS

El primer paso para instalar el sistema operativo es crear una carpeta con el nombre *myapps* dentro de *home*, en Debian. Para ello, entramos a *home* escribiendo el siguiente comando:

```
cd /home
```

Y para crear la carpeta, se escribe:

```
mkdir myapps
```



```
usuario@LAPTOP-TLE1M1AT:~$ cd /home
usuario@LAPTOP-TLE1M1AT:/home$ mkdir myapps
```

Figura 1. Crear carpeta myapps en home.

Después, será necesario copiar los archivos descargados de GNAT y Marte a la carpeta de *myapps*, esto se hace entrando al directorio donde se guardaron estos archivos. Por ejemplo:

```
usuario@LAPTOP-TLE1M1AI:/home$ pushd /mnt/c/Users/lilly/OneDrive/Escritorio/  
/mnt/c/Users/lilly/OneDrive/Escritorio /home /home/myapps  
usuario@LAPTOP-TLE1M1AI:/mnt/c/Users/lilly/OneDrive/Escritorio$
```

Figura 2. Entrar a un directorio.

Al estar dentro del directorio, se deberán copiar los archivos a la carpeta de *myapps* utilizando los siguientes comandos:

```
cp -v gnat-gpl-2016-x86_64-linux-bin.tar.gz /home/myapps/  
cp -v marte_2.0_22Feb2017_src.tar.gz /home/myapps/
```

Después de hacer esto, regresaremos a home escribiendo *pushd*. Después entraremos a la carpeta *myapps*, de la siguiente manera.

```
usuario@LAPTOP-TLE1M1AI:/mnt/c/Users/lilly/OneDrive/Escritorio$ pushd  
/home /mnt/c/Users/lilly/OneDrive/Escritorio /home/myapps  
usuario@LAPTOP-TLE1M1AI:/home$ cd myapps  
usuario@LAPTOP-TLE1M1AI:/home/myapps$
```

Figura 3. Acceder a la carpeta *myapps*.

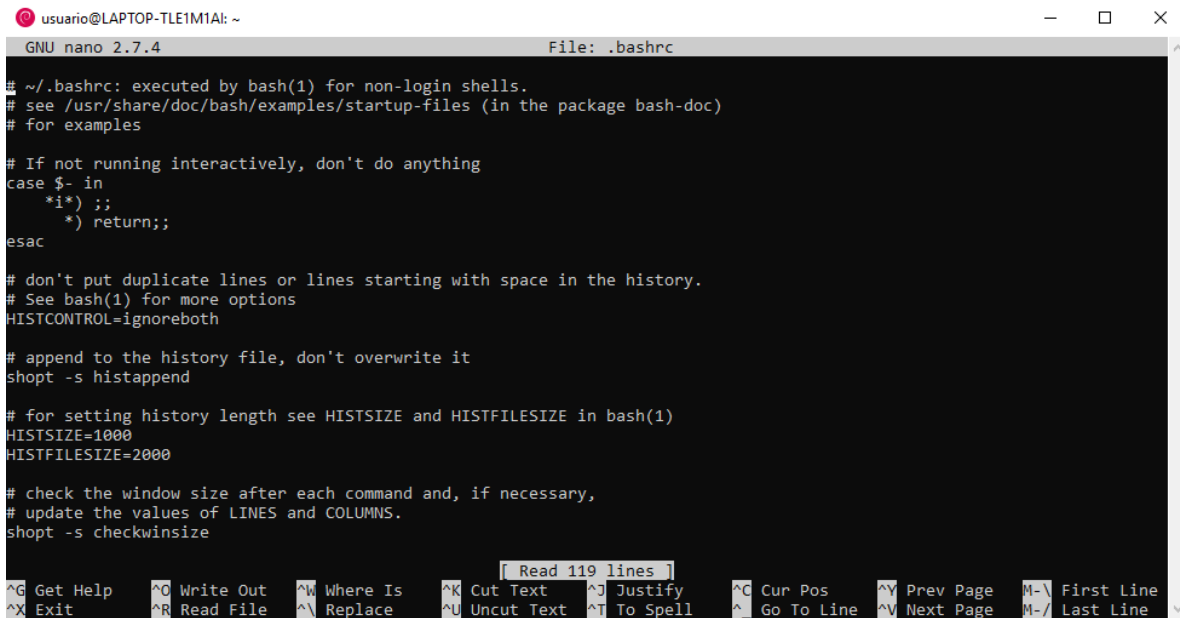
Lo que se tiene que hacer ahora, es descomprimir los dos archivos. Para el primero de ellos, escribiremos el comando *tar xvf gnat-gpl-2016-x86_64-linux-bin.tar.gz*. Esperaremos a que se termine el proceso y luego continuaremos con el segundo archivo, escribiendo el comando *tar xvf marte_2.0_22Feb2017_src.tar.gz*.

Una vez que se han descomprimido los archivos, es necesario modificar el archivo *bashrc* para poder instalar el sistema operativo. Para esto, debemos ir a la raíz del sistema, escribiendo el comando *cd*, ya que nos encontramos ahí, ejecutaremos la instrucción *nano .bashrc*.

```
usuario@LAPTOP-TLE1M1AI:/home/myapps$ cd  
usuario@LAPTOP-TLE1M1AI:~$ nano .bashrc
```

Figura 4. Instrucción *nano .bashrc*.

Al presionar la tecla Enter, se ejecutará la instrucción y entraremos al siguiente archivo.



```
GNU nano 2.7.4 File: .bashrc

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

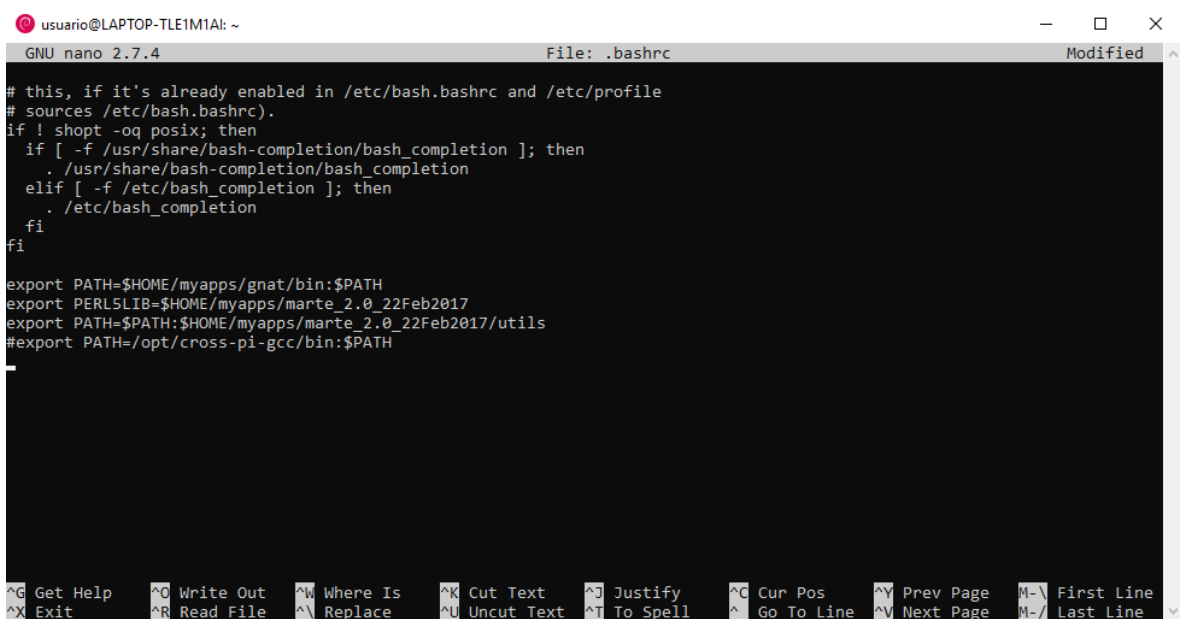
# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

[ Read 119 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    ^Y Prev Page  M- First Line
^X Exit      ^R Read File  ^_ Replace    ^U Uncut Text ^T To Spell   ^G Go To Line  ^V Next Page  M- Last Line
```

Figura 5. Archivo bashrc.

Al final del archivo, se deberá agregar lo siguiente:

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



```
GNU nano 2.7.4 File: .bashrc Modified

# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

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

Figura 6. Agregar cambios al archivo bashrc.

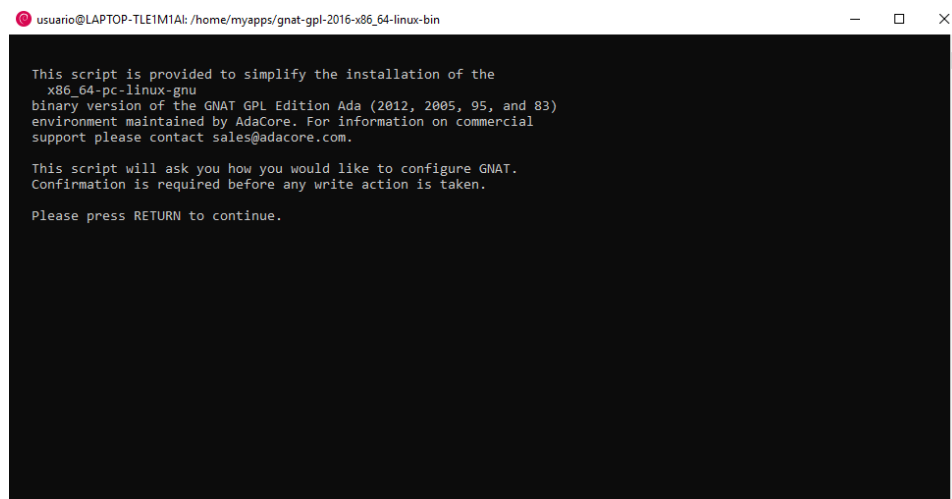
Luego de realizar estos cambios, guardaremos el archivo presionando CRTL+O, y posteriormente saldremos con CTRL+X, tal como se indica en el archivo.

Una vez que hemos salido del archivo *bashrc*, entraremos a la carpeta del compilador GNAT, escribiendo la instrucción `cd gnat-gpl-2016-x86_64-linux-bin/` y posteriormente, escribiremos el comando `./doinstall`.

```
usuario@LAPTOP-TLE1M1AI:/home/myapps$ cd gnat-gpl-2016-x86_64-linux-bin
usuario@LAPTOP-TLE1M1AI:/home/myapps/gnat-gpl-2016-x86_64-linux-bin$ ./doinstall
```

Figura 7. Instrucción `./doinstall`.

Al hacer esto, la ventana de Debian se verá de la siguiente manera, y presionaremos la tecla Enter para continuar.

A screenshot of a terminal window titled 'usuario@LAPTOP-TLE1M1AI: /home/myapps/gnat-gpl-2016-x86_64-linux-bin'. The terminal displays the following text:

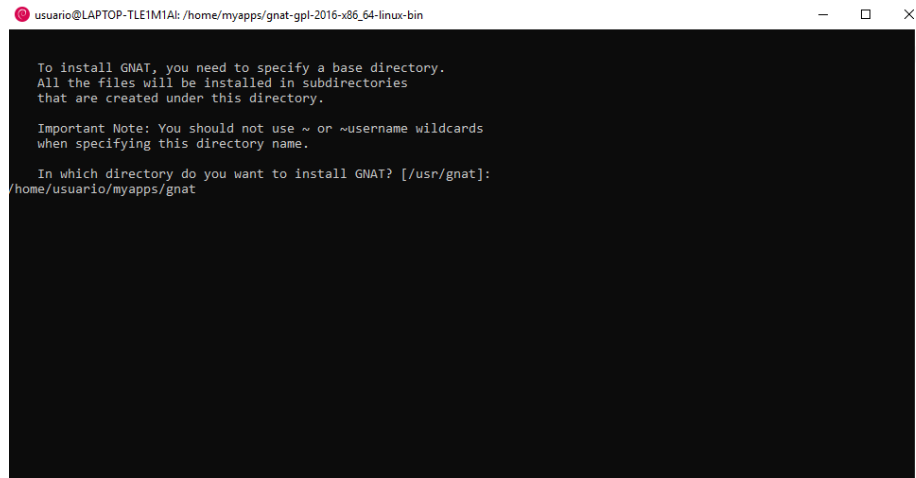
```
This script is provided to simplify the installation of the
x86_64-pc-linux-gnu
binary version of the GNAT GPL Edition Ada (2012, 2005, 95, and 83)
environment maintained by AdaCore. For information on commercial
support please contact sales@adacore.com.

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.
```

Figura 8. Instalación de GNAT.

Después, el programa nos preguntará donde queremos llevar a cabo la instalación de GNAT, entonces escribiremos la ubicación como `/home/NombreDelUsuario/myapps/gnat`, como se muestra a continuación:



```
usuario@LAPTOP-TLE1M1A: /home/myapps/gnat-gpl-2016-x86_64-linux-bin

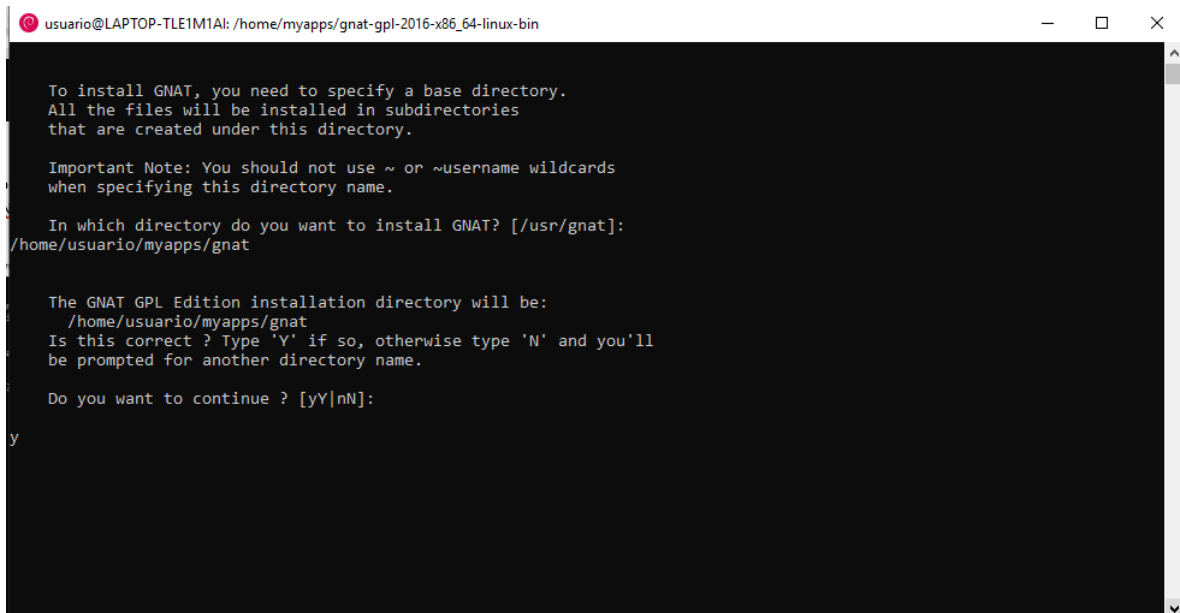
To install GNAT, you need to specify a base directory.
All the files will be installed in subdirectories
that are created under this directory.

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
```

Figura 9. Ingresar nombre de la carpeta de instalación de GNAT

Volveremos a presionar la tecla Enter luego de escribir la ubicación, y el programa nos pedirá confirmar el directorio de instalación, si es correcto escribiremos “y” o “Y” y seguido de esto presionaremos Enter.



```
usuario@LAPTOP-TLE1M1A: /home/myapps/gnat-gpl-2016-x86_64-linux-bin

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.

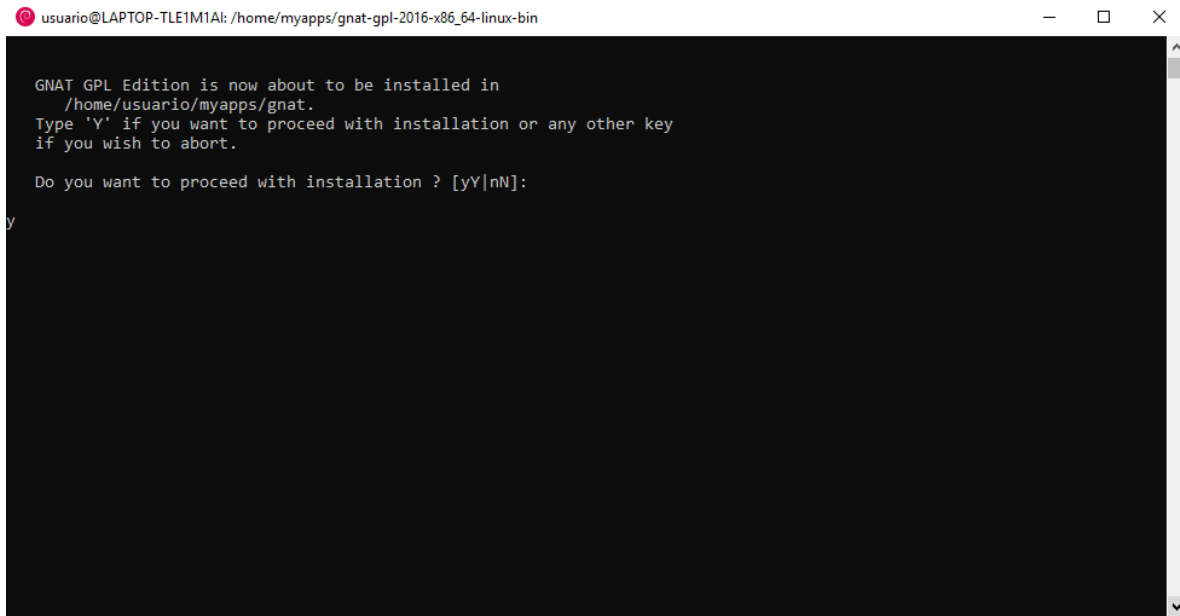
In which directory do you want to install GNAT? [/usr/gnat]:
/home/usuario/myapps/gnat

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]:
y
```

Figura 10. Confirmar carpeta de instalación de GNAT.

El programa nos solicitará nuevamente confirmar para proceder con la instalación, y haremos lo mismo que en el paso anterior.

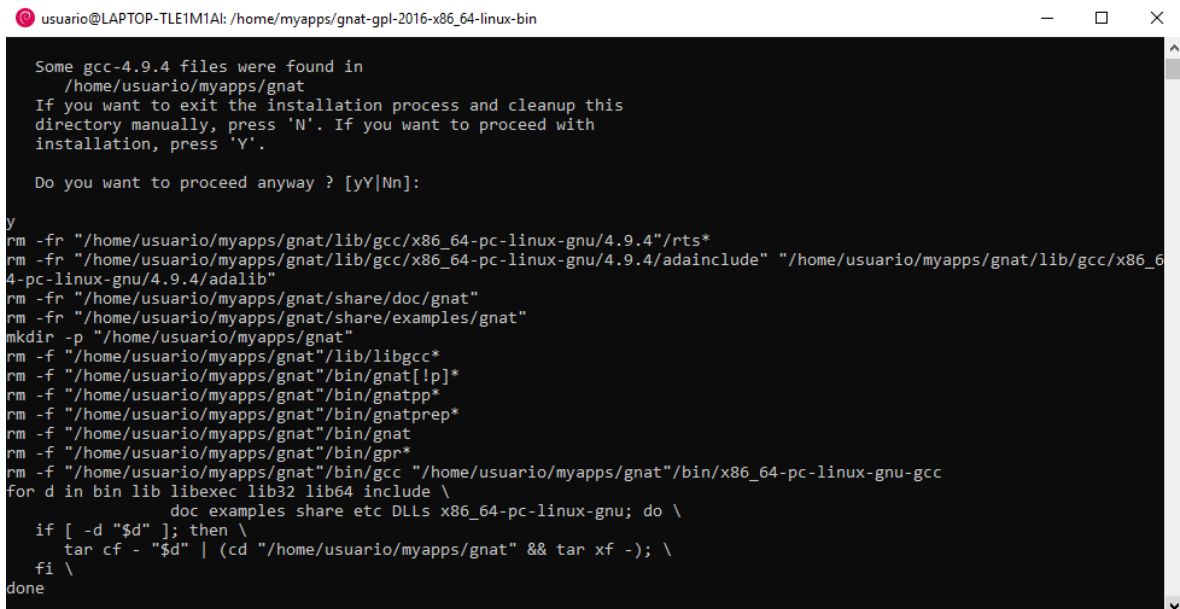
A terminal window titled 'usuario@LAPTOP-TLE1M1A1: /home/myapps/gnat-gpl-2016-x86_64-linux-bin'. The text inside the terminal reads: 'GNAT GPL Edition is now about to be installed in /home/usuario/myapps/gnat. Type 'Y' if you want to proceed with installation or any other key if you wish to abort. Do you want to proceed with installation ? [yY|nN]:'. A lowercase 'y' is visible on the left side of the terminal, indicating the user's input.

```
usuario@LAPTOP-TLE1M1A1: /home/myapps/gnat-gpl-2016-x86_64-linux-bin
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]:
y
```

Figura 11. Confirmar instalación de GNAT.

Después, visualizaremos lo siguiente, lo cual significa que el proceso de instalación ha comenzado.

A terminal window titled 'usuario@LAPTOP-TLE1M1A1: /home/myapps/gnat-gpl-2016-x86_64-linux-bin'. The text inside the terminal reads: 'Some gcc-4.9.4 files were found in /home/usuario/myapps/gnat. If you want to exit the installation process and cleanup this directory manually, press 'N'. If you want to proceed with installation, press 'Y'. Do you want to proceed anyway ? [yY|Nn]:'. A lowercase 'y' is visible on the left side of the terminal. Below the prompt, a series of shell commands are executed to remove old files and create the installation directory structure.

```
usuario@LAPTOP-TLE1M1A1: /home/myapps/gnat-gpl-2016-x86_64-linux-bin
Some gcc-4.9.4 files were found in
/home/usuario/myapps/gnat
If you want to exit the installation process and cleanup this
directory manually, press 'N'. If you want to proceed with
installation, press 'Y'.

Do you want to proceed anyway ? [yY|Nn]:
y
rm -fr "/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4"/rts*
rm -fr "/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4/adainclude" "/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4/adalib"
rm -fr "/home/usuario/myapps/gnat/share/doc/gnat"
rm -fr "/home/usuario/myapps/gnat/share/examples/gnat"
mkdir -p "/home/usuario/myapps/gnat"
rm -f "/home/usuario/myapps/gnat/lib/libgcc*"
rm -f "/home/usuario/myapps/gnat/bin/gnat[!p]*"
rm -f "/home/usuario/myapps/gnat/bin/gnatpp*"
rm -f "/home/usuario/myapps/gnat/bin/gnatprep*"
rm -f "/home/usuario/myapps/gnat/bin/gnat"
rm -f "/home/usuario/myapps/gnat/bin/gpr*"
rm -f "/home/usuario/myapps/gnat/bin/gcc" "/home/usuario/myapps/gnat/bin/x86_64-pc-linux-gnu-gcc"
for d in bin lib libexec lib32 lib64 include \
    doc examples share etc DLLs x86_64-pc-linux-gnu; do \
    if [ -d "$d" ]; then \
        tar cf - "$d" | (cd "/home/usuario/myapps/gnat" && tar xf -); \
    fi \
done
```

Figura 12. Inicio del proceso de instalación de GNAT.

Esperaremos a que termine el proceso de instalación, esto puede demorar algunos minutos, y sabremos que ha terminado con éxito cuando la ventana se vea de la siguiente manera.

```
usuario@LAPTOP-TLE1M1AI: /home/myapps/gnat-gpl-2016-x86_64-linux-bin

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@LAPTOP-TLE1M1AI:/home/myapps/gnat-gpl-2016-x86_64-linux-bin$
```

Figura 13. Fin del proceso de instalación de GNAT.

Después de hacer lo anterior, regresaremos a la carpeta *myapps* con la instrucción “*cd ..*” y entraremos a la carpeta de marte, ejecutando el comando *cd marte_2.0_22Feb2017/*, como se muestra a continuación.

```
usuario@LAPTOP-TLE1M1AI:/home/myapps/gnat-gpl-2016-x86_64-linux-bin$ cd ..
usuario@LAPTOP-TLE1M1AI:/home/myapps$ cd marte_2.0_22Feb2017
usuario@LAPTOP-TLE1M1AI:/home/myapps/marte_2.0_22Feb2017$
```

Figura 14. Entrar a la carpeta de MarteOs.

Ya estando dentro de la carpeta de marte, se ejecutará el comando *./minstall*, con lo cual se visualizará lo siguiente.

```
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017

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@LAPTOP-TLE1M1AI:/home/myapps/gnat-gpl-2016-x86_64-linux-bin$ cd ..
usuario@LAPTOP-TLE1M1AI:/home/myapps$ cd marte_2.0_22Feb2017
usuario@LAPTOP-TLE1M1AI:/home/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 $GNAT_LIBS_PATH("rpi") 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://marte.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...
```

Figura 15. Instrucción *./minstall*.

Presionaremos la tecla Enter para continuar con la instalación, y al finalizar se mostrará lo siguiente.

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

You may want to add "/home/bnm930519/nyapps/marte_2.0_22Feb2017/utls"
to your $PATH environment variable to have direct access to MaRTE
tools (ngnatmake, mgcc, mkmmarte, mkrtsmarteuc, msetcurrentarch, etc.)

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

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

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

$ msetcurrentarch linux_lib && mkrtsmarteuc && mkmmarte

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

Figura 16. Fin del proceso de la instrucción `./minstall`.

El siguiente paso es acceder a la carpeta `utls`, que se encuentra dentro de `marte_2.0_22Feb2017`, como se hizo anteriormente, escribiendo `cd utls/`. Posteriormente, tendremos que definir la arquitectura sobre la cual trabajará el sistema operativo, entonces escribiremos el comando `msetcurrentarch`, con lo cual aparecerán en pantalla las diferentes arquitecturas que podemos utilizar.

```
usuario@LAPTOP-TLE1M1AI:/home/myapps/marte_2.0_22Feb2017$ cd utls/
usuario@LAPTOP-TLE1M1AI:/home/myapps/marte_2.0_22Feb2017/utls$ msetcurrentarch
Use of uninitialized value $GNAT_LIBS_PATH{"rpi"} in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utls/globals.pl line 16.
Current architecture:x86 (Current processor:i386)

Available architectures status:
x86:      RTS (gnat_rts/rts-marteuc_x86): Compiled
Lib MaRTE (objs/x86_obj):      Compiled
linux:    RTS (gnat_rts/rts-marteuc_linux): NOT Compiled
Lib MaRTE (objs/linux_obj):    NOT Compiled
linux_lib: RTS (gnat_rts/rts-marteuc_linux_lib): NOT Compiled
Lib MaRTE (objs/linux_lib_obj): NOT Compiled
rpi:      NOT available
usuario@LAPTOP-TLE1M1AI:/home/myapps/marte_2.0_22Feb2017/utls$
```

Figura 17. Arquitecturas disponibles para Marte.

En este caso se elige la arquitectura x86, ingresando la instrucción `msetcurrentarch x86 i386`. Con ello, se visualizará lo siguiente.

```
rte-kernel-devices_table.ads /home/bnm930519/nyapps/marte_2.0_22Feb2017/kernel/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/na
rte-kernel-file_system.ads /home/bnm930519/nyapps/marte_2.0_22Feb2017/kernel/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/co
nsole_management.adb /home/bnm930519/nyapps/marte_2.0_22Feb2017/misc/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/na
rte-configuration_parameters.ads /home/bnm930519/nyapps/marte_2.0_22Feb2017/ker
nel/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/na
rte-direct_io.ads /home/bnm930519/nyapps/marte_2.0_22Feb2017/kernel/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/ll
bm.a /home/bnm930519/nyapps/marte_2.0_22Feb2017/objs/x86_obj/
ln -s -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/arch/arch_dependent_files/co
nsole_management.ads /home/bnm930519/nyapps/marte_2.0_22Feb2017/misc/
rm -f /home/bnm930519/nyapps/marte_2.0_22Feb2017/lib && ln -s /home/bnm930519/ny
apps/marte_2.0_22Feb2017/objs/x86_obj/ /home/bnm930519/nyapps/marte_2.0_22Feb201
7/lib
cd /home/bnm930519/nyapps/marte_2.0_22Feb2017/gnat_rts/ && rm -f rts && ln -s rt
s-marteuc_x86 rts
msetcurrentarch:x86 set as the default architecture...OK

GNAT/MaRTE RTS for this architecture is not compiled yet.
To compile it execute command: 'mkrtsmarteuc && mkmmarte'
```

Figura 18. Proceso de instalación de Marte luego de elegir la arquitectura.

Como se puede ver, se nos indica que compilemos utilizando los comandos *mkrtsmarteuc* y *mkmarte*. Al ejecutar la primera instrucción, se observará lo siguiente.

```

b.o s-taprop.o s-tarest.o s-tasdeb.o s-tasinf.o s-tasini.o s-taskin.o s-taspr.o
s-tasque.o s-tasres.o s-tasren.o s-tassta.o s-tasuti.o s-tasde.o s-tadeca.o s-
tadert.o s-tataat.o s-tpinop.o s-tpoben.o s-tpobop.o s-tposen.o s-tratas.o threa
d.o s-linux.o a-exetin.o a-exitli.o s-hansup.o a-etgrbu.o a-disedf.o a-diroro.o
ar: creating libgnarl.a
rm *.o
chmod 0444 *.all *.a
cd /home/bnn930519/nyapps/marte_2.0_22Feb2017/gnat_rts/rts-marteuc_x86/adalib/ &
& ln -f -s ../../lib/libmarte.a libmarte.a
cd /home/bnn930519/nyapps/marte_2.0_22Feb2017/gnat_rts/rts-marteuc_x86/adalib/ &
& ln -f -s libmarte.a libc.a
cd /home/bnn930519/nyapps/marte_2.0_22Feb2017/gnat_rts/rts-marteuc_x86/adalib/ &
& ln -f -s ../../lib/multiboot.o multiboot.o
cd /home/bnn930519/nyapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 && rm adainclud
e adalib
cd /home/bnn930519/nyapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 && ln -s rts-na
tive/adainclude adainclude
cd /home/bnn930519/nyapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 && ln -s rts-na
tive/adalib adalib
mkrtsmarteuc: rts-marteuc_x86 done :)

Lib MARTE for this architecture is not compiled yet
Run 'mkmarte'

```

Figura 19. Proceso de la instrucción *mkrtsmarteuc*.

Finalmente, ejecutaremos la segunda instrucción, *mkmarte*, tal como se nos indica, y al finalizar el proceso, veremos lo siguiente.

```

inline void setstate (iostate v) { m_State |= v; }
^
utils.h: In member function 'ustl::ios_base::iostate ustl::ios_base::exceptions(u
stl::ios_base::iostate)':
utils.h:96:80: warning: conversion to 'uint16_t [aka short unsigned int]' from 'u
stl::ios_base::iostate [aka unsigned int]' may alter its value [-Wconversion]
    inline iostate exceptions (iostate v) { return (m_Exceptions = v); }
^
Linking libustl.a ...
ar: creating libustl.a
make[1]: se sale del directorio '/home/bnn930519/nyapps/marte_2.0_22Feb2017/lang
_support/ustl-src'
make -C ustl-src/ install
make[1]: se entra en el directorio '/home/bnn930519/nyapps/marte_2.0_22Feb2017/l
ang_support/ustl-src'
Installing headers to ../../x86_arch/include/ustl ...
Installing libustl.a to ../../lib ...
make[1]: se sale del directorio '/home/bnn930519/nyapps/marte_2.0_22Feb2017/lang
_support/ustl-src'
C++ language support library DONE
mkmarte: work done :-))

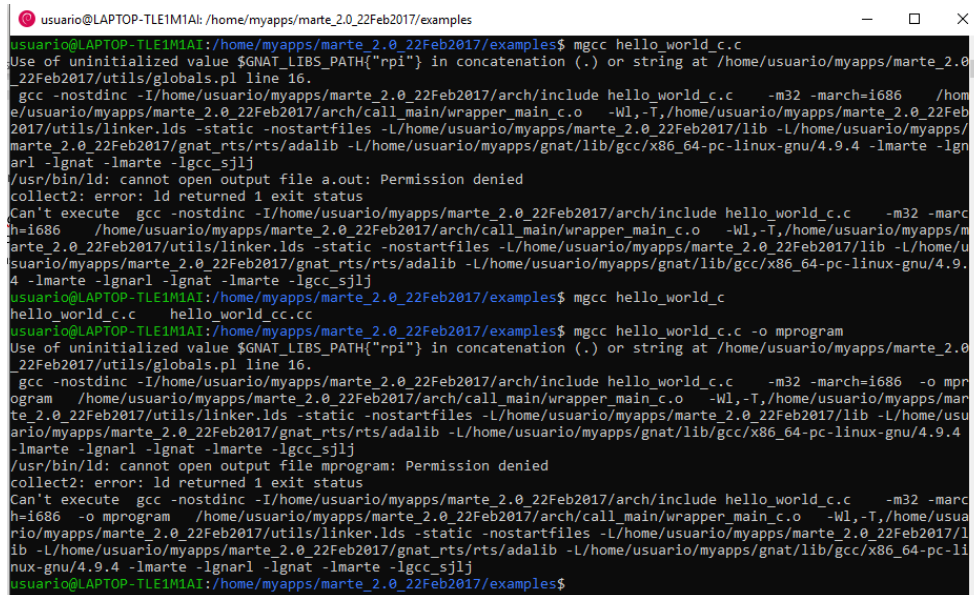
```

Figura 20. Proceso de la instrucción *mkmarte*.

Con esto hemos terminado la instalación del sistema operativo Marte OS, y para comprobar que el proceso se ha realizado correctamente, entraremos a la carpeta *examples* de marte, saliendo primero de *utils*, con el comando *cd ..*, y posteriormente, entrando a la carpeta *examples*: *cd examples/*.

Ya estando dentro de la carpeta, ejecutaremos el comando *ls*, con lo que se visualizarán los nombres de varios archivos en C, C++ y Ada. En este caso, se comprobará el funcionamiento del sistema operativo utilizando el programa *hello_world.c.c*, para ello, se escribirá la instrucción *mgcc hello_world.c.c*, para compilar el archivo. Después, se ejecutará la

instrucción `mgcc hello_world.c.c -o mprogram`, para prear un archivo asociado a la compilación de `hello_world.c.c`, de nombre `mprogram`.



```
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017/examples$ mgcc hello_world.c.c
Use of uninitialized value $GNAT_LIBS_PATH["rpi"] in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utills/globals.pl line 16.
gcc -nostdinc -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/include hello_world.c.c -m32 -march=i686 -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main.c.o -Wl,-T,/home/usuario/myapps/marte_2.0_22Feb2017/utills/linker.lds -static -nostartfiles -L/home/usuario/myapps/marte_2.0_22Feb2017/lib -L/home/usuario/myapps/marte_2.0_22Feb2017/gnat_rts/rtts/adalib -L/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnat -lgcc_sjlj
/usr/bin/ld: cannot open output file a.out: Permission denied
collect2: error: ld returned 1 exit status
Can't execute gcc -nostdinc -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/include hello_world.c.c -m32 -march=i686 -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main.c.o -Wl,-T,/home/usuario/myapps/marte_2.0_22Feb2017/utills/linker.lds -static -nostartfiles -L/home/usuario/myapps/marte_2.0_22Feb2017/lib -L/home/usuario/myapps/marte_2.0_22Feb2017/gnat_rts/rtts/adalib -L/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnat -lgcc_sjlj
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017/examples$ mgcc hello_world.c.c
hello_world.c.c hello_world.cc.cc
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017/examples$ mgcc hello_world.c.c -o mprogram
Use of uninitialized value $GNAT_LIBS_PATH["rpi"] in concatenation (.) or string at /home/usuario/myapps/marte_2.0_22Feb2017/utills/globals.pl line 16.
gcc -nostdinc -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/include hello_world.c.c -m32 -march=i686 -o mprogram -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main.c.o -Wl,-T,/home/usuario/myapps/marte_2.0_22Feb2017/utills/linker.lds -static -nostartfiles -L/home/usuario/myapps/marte_2.0_22Feb2017/lib -L/home/usuario/myapps/marte_2.0_22Feb2017/gnat_rts/rtts/adalib -L/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnat -lgcc_sjlj
/usr/bin/ld: cannot open output file mprogram: Permission denied
collect2: error: ld returned 1 exit status
Can't execute gcc -nostdinc -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/include hello_world.c.c -m32 -march=i686 -o mprogram -I/home/usuario/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main.c.o -Wl,-T,/home/usuario/myapps/marte_2.0_22Feb2017/utills/linker.lds -static -nostartfiles -L/home/usuario/myapps/marte_2.0_22Feb2017/lib -L/home/usuario/myapps/marte_2.0_22Feb2017/gnat_rts/rtts/adalib -L/home/usuario/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnat -lgcc_sjlj
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017/examples$
```


Figura 21. Compilación de `hello_world.c.c`

Además, es necesario crear un archivo ejecutable asociado al archivo C, utilizando la instrucción `make hello_world_c.exe`. Para verificar que éste se ha creado, utilizaremos el comando `ls`.

Para correr el archivo ejecutable sobre qemu, es necesario tener instalado el programa Xming, el cual se podrá descargar del siguiente enlace:

<https://sourceforge.net/projects/xming/>

Luego de realizar los pasos que indica el instalador, ejecutaremos el programa, y posteriormente escribiremos la instrucción `qemu-system-i386 -kernel hello_world_c.exe`, con lo cual se visualizará el archivo ejecutable corriendo sobre el sistema operativo (en qemu).



```
-- 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.

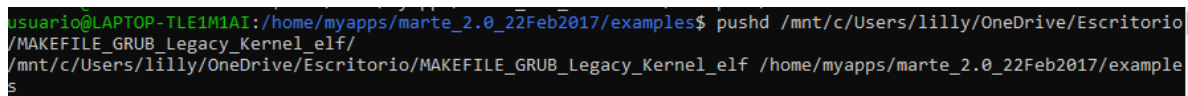
_exit(0) called; rebooting...
Press a key to reboot_
```

Figura 22. Archivo ejecutable corriendo sobre el sistema operativo, en qemu.

El siguiente paso es descargar o clonar la carpeta *MAKEFILE_Grub_Legacy_Kernel* del siguiente repositorio:

https://github.com/sotrteacher/sotr_201808_201812/tree/master/PRACTICA_1_Instalacion_de_un_SOTR/MAKEFILE_GRUB_Legacy_Kernel_elf

Una vez hecho esto, mover la carpeta al escritorio u otro lugar donde se quiera conservar y revisar la ubicación exacta en el explorador de archivos. Copiar el archivo *kernelmemfs* a la carpeta de *MAKEFILE_Grub_Legacy_Kernel*, utilizando la ubicación de esta, como se muestra a continuación.



```
usuario@LAPTOP-TLE1M1AI: /home/myapps/marte_2.0_22Feb2017/examples$ pushd /mnt/c/Users/lilly/OneDrive/Esitorio/MAKEFILE_GRUB_Legacy_Kernel_elf/
/mnt/c/Users/lilly/OneDrive/Esitorio/MAKEFILE_GRUB_Legacy_Kernel_elf /home/myapps/marte_2.0_22Feb2017/example$
```

Figura 23. Acceder al directorio *MAKEFILE_Grub_Legacy_Kernel*.

Volvemos a nuestra ubicación anterior con el comando *pushd*. La siguiente instrucción que debemos ingresar a la terminal es *cp -v hello_world_c.exe `dirs -1 +1`*, con esto se copia el archivo *hello_world_c.exe* a la carpeta de *MAKEFILE_Grub_Legacy_Kernel*. Regresamos nuevamente al directorio anterior utilizando *pushd*. Ya estando ahí, revisamos que se encuentre el archivo ejecutable, esto se hace con la instrucción *ls -lh*.

```

usuario@LAPTOP-TLE1M1AI:/home/myapps/marte_2.0_22Feb2017/examples$ pushd
/mnt/c/Users/lilly/OneDrive/Escritorio/MAKEFILE_GRUB_Legacy_Kernel_elf /home/myapps/marte_2.0_22Feb2017/example
s
usuario@LAPTOP-TLE1M1AI:/mnt/c/Users/lilly/OneDrive/Escritorio/MAKEFILE_GRUB_Legacy_Kernel_elf$ ls -lh
total 4.6M
-rwxrwxrwx 1 usuario usuario 905K Sep 24 15:11 hello_world_c.exe
drwxrwxrwx 1 usuario usuario 4.0K Sep 4 15:29 .
-rwxrwxrwx 1 usuario usuario 673K Dec 2 01:19 kernel.elf
-rwxrwxrwx 1 usuario usuario 673K Dec 2 01:15 kernelmemfs
-rwxrwxrwx 1 usuario usuario 1015K Aug 30 20:12 leds_parallel_port
-rwxrwxrwx 1 usuario usuario 669 Aug 30 20:12 Makefile
-rwxrwxrwx 1 usuario usuario 1.4M Sep 24 15:14 os.iso
usuario@LAPTOP-TLE1M1AI:/mnt/c/Users/lilly/OneDrive/Escritorio/MAKEFILE_GRUB_Legacy_Kernel_elf$

```

Figura 24. Visualizar archivos en el directorio MAKEFILE.

Para sobre escribir el archivo *kernel.elf*, utilizamos la instrucción *cp -vi hello_world_c.exe kernel.elf*. Posteriormente, escribimos *make*, para compilar. Por último, con la instrucción *qemu-system-i386 -cdrom os.iso -m 512* podemos correr Marte OS sobre qemu, para ello se debe ejecutar primero el programa Xming, como se hizo en pasos anteriores. Primero se visualizará lo siguiente:

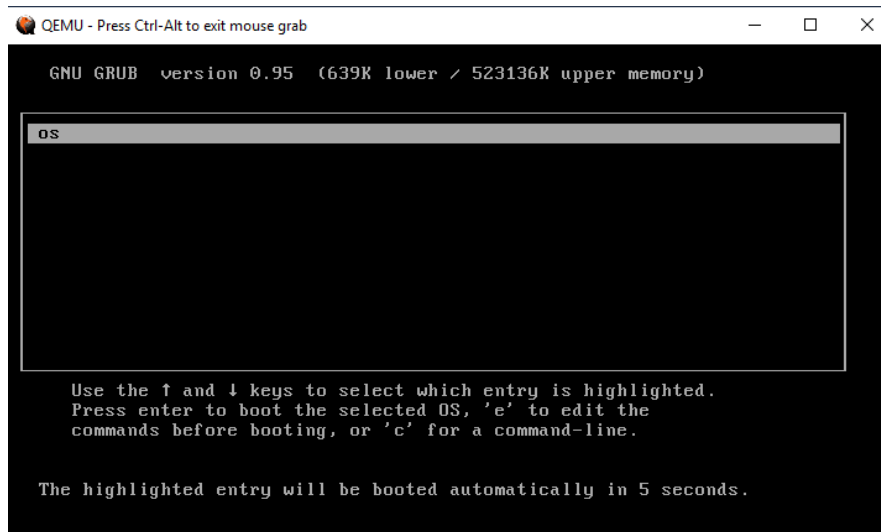


Figura 25. Inicio de la ejecución del sistema operativo en qemu.

Luego de presionar Enter o esperar el comienzo automático del sistema operativo, veremos lo siguiente:

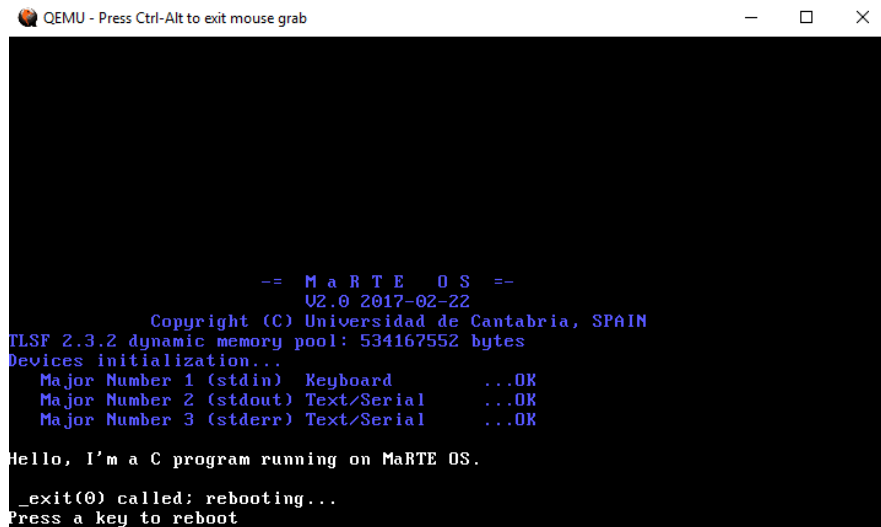


Figura 26. Marte OS con el archivo ejecutable corriendo sobre qemu.

Finalmente, se utiliza el software *Linux Live USB Creator* para crear una *liveUSB* con el sistema operativo xv6. Este software se puede descargar del siguiente enlace:

<https://www.linuxliveusb.com/en/download>

Para grabar el archivo *iso* en la USB, se elige la unidad como indica el software, y posteriormente, se elige la fuente, es decir, el archivo *iso* de la carpeta de *MAKEFILE_Grub_Legacy*. Seguir con los otros pasos que indica el software para crear la *liveUSB*.



Figura 27. Elegir unidad y fuente en Linux Live USB Creator. // Figura 28. Pasos 3 a 5 para crear la liveUSB en el software Linux Live USB Creator.

Luego de realizar todo lo anterior, se puede utilizar este sistema operativo insertando la USB en una computadora, previamente a encenderla. Después, presionar el botón de encendido y presionar F11 o F12 hasta que inicie la ejecución de Marte OS.