



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS



Sistemas Operativos en Tiempo Real

**Guía para instalación de MarteOs y compilación de
programa de alarma.**

Profesor: Maza Casas Lamberto

Alumno: Flores Gutierrez Juan Antonio

Grupo: 3MV11

1- Instalación de MarteOs

MarteOS es un sistema operativo de tiempo real para aplicaciones embebidas que sigue a la mínima en tiempo real POSIX. El entorno de desarrollo se basa en la GNU compiladores GNAT, GCC y GCJ.

Para lograr este punto se seguirán los pasos descritos en una guía de instalación de MarteOs y el compilador GNAT, la cual fue proporcionada por el profesor.

- Como primer paso es necesario descargar los siguientes archivos “gnat-gpl-2016-x86_64-linux-bin.tar.gz” y “marte_2.0_22Feb2017_src.tar.gz”.

El primer archivo podemos encontrarlo en el siguiente enlace:

<http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b>

Mientras que el segundo archivo podemos encontrarlo en el siguiente URL:

https://marte.unican.es/marte/marte_2.0_22Feb2017_src.tar.gz

- Como siguiente paso tenemos que crear la carpeta myapps dentro de “Carpeta Personal”, una vez creada se tienen que guardar en ella los archivos “gnat-gpl-2016-x86_64-linux-bin.tar.gz” y “marte_2.0_22Feb2017_src.tar.gz”. lo cual se realiza con los siguientes comandos sobre la terminal de Debian.

```
cp -rv /mnt/Users/Juan/Desktop/gnat-gpl-2016-x86_64-linux-bin.tar.gz
```

```
cp -rv /mnt/Users/Juan/Desktop/marte_2.0_22Feb2017_src.tar.gz
```

después de esto es necesario crear una carpeta con el nombre gnat dentro de la carpeta myapps, la cual se creara con el comando “*mkdir gnat*”

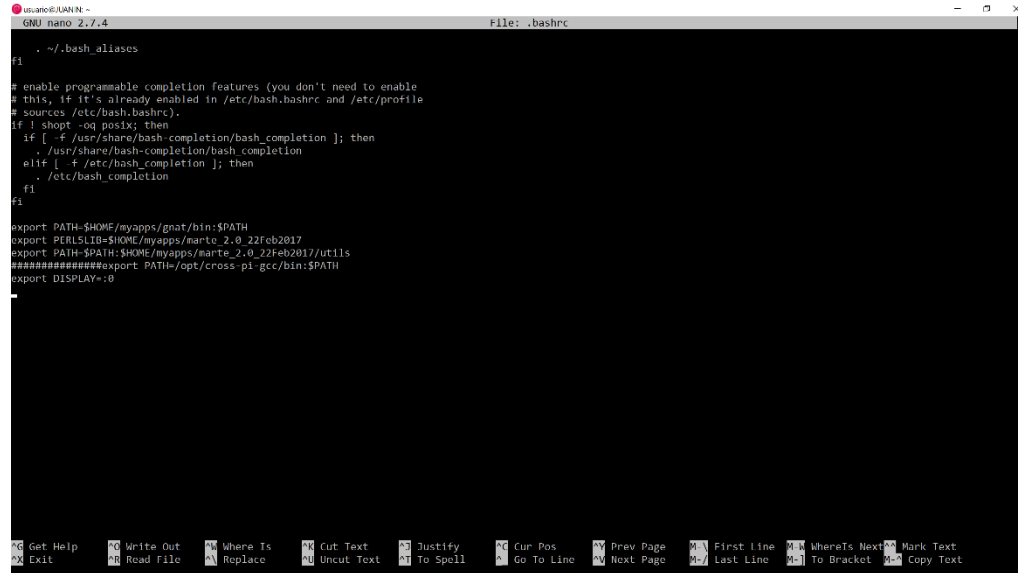
- El siguiente paso es descomprimir los archivos que contienen el compilador GNAT así como el que contiene MarteOs.

Dentro de la carpeta myapps ejecutamos la siguiente instrucción “*tar xvf gnat-gpl-2016-x86_64-linux-bin.tar.gz*”, una vez terminado el proceso, seguimos con la descompresión del archivo que contiene marte y eso lo haremos con la siguiente instrucción “*tar xvf marte_2.0_22Feb2017_src.tar.gz*”.

- Una vez descomprimidos los archivos es necesario modificar el archivo “*bashrc*”, para esto salimos a la raíz del sistema con el comando “*cd*” y después de ello ejecutamos el comando “*nano .bashrc*”

La modificación que se tiene que realizar es agregar las siguientes líneas al final del código del archivo.

```
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
export DISPLAY=:0
```



```
GNU nano 2.7.4 File: .bashrc
~/.bash_aliases

# enable programmable completion features (you don't need to enable
# 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 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
export DISPLAY=:0
```

Para guardar los cambios se teclea el comando (CTRL+O) para guardar cambios y después salir del archivo con (CTRL+X).

- Después dentro de la carpeta myapps entraremos a la carpeta contenedora del compilador GNAT, esto lo hacemos con la instrucción “*cd gnat-gpl-2016-x86_64-linux-bin/*”, ya en esa carpeta ejecutamos la siguiente instrucción “*./doinstall*” con esto se abre una ventana donde se describe la versión del compilador a instalar, para seguir con la instalación se presiona “ENTER”.

Una vez presionado “ENTER” se pregunta la ruta donde se quiere instalar el compilador para ello escribimos la ruta de la carpeta gnat creada anteriormente

- Lo siguiente es volver nuevamente a la carpeta myapps y acceder a la carpeta de marte, esto con la instrucción “*cd marte_2.0_22Feb2017/*”.

Una vez en la carpeta de marte ejecutaremos la instrucción “*./minstall*”

- Una vez realizada la instalación hay que acceder a la carpeta “utils”, la cual se encuentra en la misma carpeta “marte_2.0_22Feb2017” lo que lograremos con instrucción “*cd utils/*”

Para esta instalación, elegimos la arquitectura x86, por lo que ingresaremos la siguiente instrucción en la terminal “*msetcurrentarch x86 i386*”

- Como siguiente paso es necesario ejecutar las instrucciones “*mkrtsmarteuc*” y posteriormente “*mkmarte*” en ese orden, para evitar la visualización de un error.
- En este punto se ha terminado con la instalación de MarteOs, para probar que realmente funcione requeriremos de la aplicación QEMU, si no está instalada, lo siguiente es ejecutar la instrucción “*cd*” para salir a la carpeta raíz y ejecutar la instrucción “*sudo apt-get install qemu*”.

2- Compilación del programa de alarma.

Para el desarrollo del programa alarma se utilizó un código en lenguaje C++, el cual se generó en la carpeta “/myapps/marte_2.0_22Feb2017/examples” con la instrucción “*vim alarma.c*” una vez ahí se desarrolló el siguiente código.

```
#include<stdio.h>
#include<time.h>

int main()
{
    time_t now,s;
    struct tm* current_time;
    int hora,min,seg,hora_actual;
    int ho, minute, sec, t_sec, t_total, dif,r;

    s=time(NULL);
    time(&now);
    current_time=local(&s);

    printf("hora actual: %02d:%02d:%02d\n", current_time->tm_hour,
    current_time->tm_min, current_time->tm_sec);

    printf("introduzca hora de alarma. ");
    printf("hora: ");
    scanf("%02d",&hora);
    printf("minutos: ");
    scanf("%02d",&min);
    printf("segundos: ");
    scanf("%02d",&seg);

    ho=current_time->tm_hour;
    minute=current_time->tm_min;
    sec=current_time->tm_sec;

    if((hora<ho)||((hora==ho&min<minute)||((hora==ho&&min==minute&&seg<sec))
    {
        printf("la hora de alarma ya paso\n");
        r=1;
    }
}
```

```

while(dif>=0)
{
    s=time(NULL);
    time(&sec);
    current_time=localtime(&s);
    ho=current_time->tm_hour;
    minute=current_time->tm_min;
    sec=current_time->tm_sec;

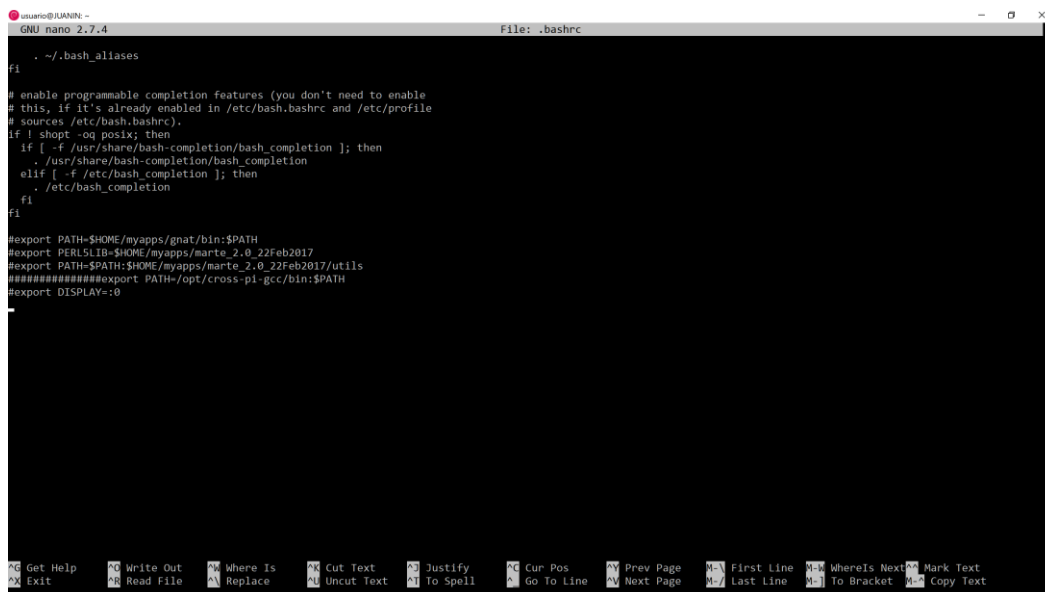
    hora_actual=(ho*3600)+(minute*60)+sec;
    dif=t_sec-hora_actual;
}

if(dif<=0)
{
    printf("ALARMA! ALARMA! ALARMA!\n");
}
else if(r==1)
{
    printf("vualva a correr el programa");
}

return 0;
}

```

Para realizar las pruebas de compilación se realizó una modificación a las líneas de código agregadas anteriormente para la instalación de marte, estas líneas se encuentran en el archivo “bashrc”. Lo único necesario fue comentar todas ellas con el símbolo #.



```

GNU nano 2.7.4 File: .bashrc
# ~/.bash_aliases

# enable programmable completion features (you don't need to enable
# 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

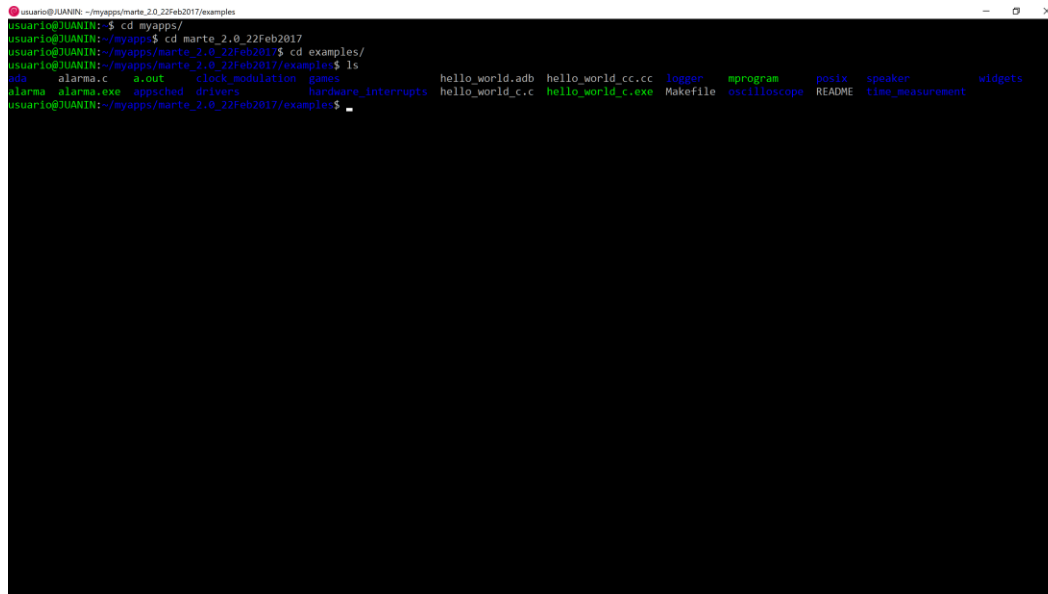
#export PATH=$HOME/myapps/gnat/bin:$PATH
#export PERLSIB=$HOME/myapps/marte_2.0_22Feb2017
#export PATH=$PATH:$HOME/myapps/marte_2.0_22Feb2017/utils
#####export PATH=/opt/cross-pi-gcc/bin:$PATH
#export DISPLAY=:0

```

Una vez verificado que la compilación del archivo alarma.c se modifica nuevamente el archivo “bashrc” y se deja como en principio se menciona en la instalación de MarteOs

Después de ello se ejecuta el comando “*mgcc alarma.c*” esto dentro de la carpeta “/examples” en la carpeta “*martes_2.0_22Feb2017*”.

Lo siguiente es ejecutar la siguiente instrucción “*mgcc alarma.c -o mprogram*” con lo que crearemos un archivo de nombre “mprogram” asociado a la compilación de nuestro archivo “*alarma.c*”.



```
usuario@JUANIN: ~/myapps/marte_2.0_22Feb2017/examples
usuario@JUANIN:~/myapps$ cd marte_2.0_22Feb2017
usuario@JUANIN:~/myapps/marte_2.0_22Feb2017$ cd examples/
usuario@JUANIN:~/myapps/marte_2.0_22Feb2017/examples$ ls
ada      alarma.c  a.out      clock_modulation  games      hello_world.adb  hello_world.cc.cc  logger  mprogram  posix  speaker  widgets
alarma   alarma.exe  appsched  drivers           hardware_interrupts  hello_world.c.c  hello_world_c.exe  Makefile  oscilloscope  README  time_measurement
```

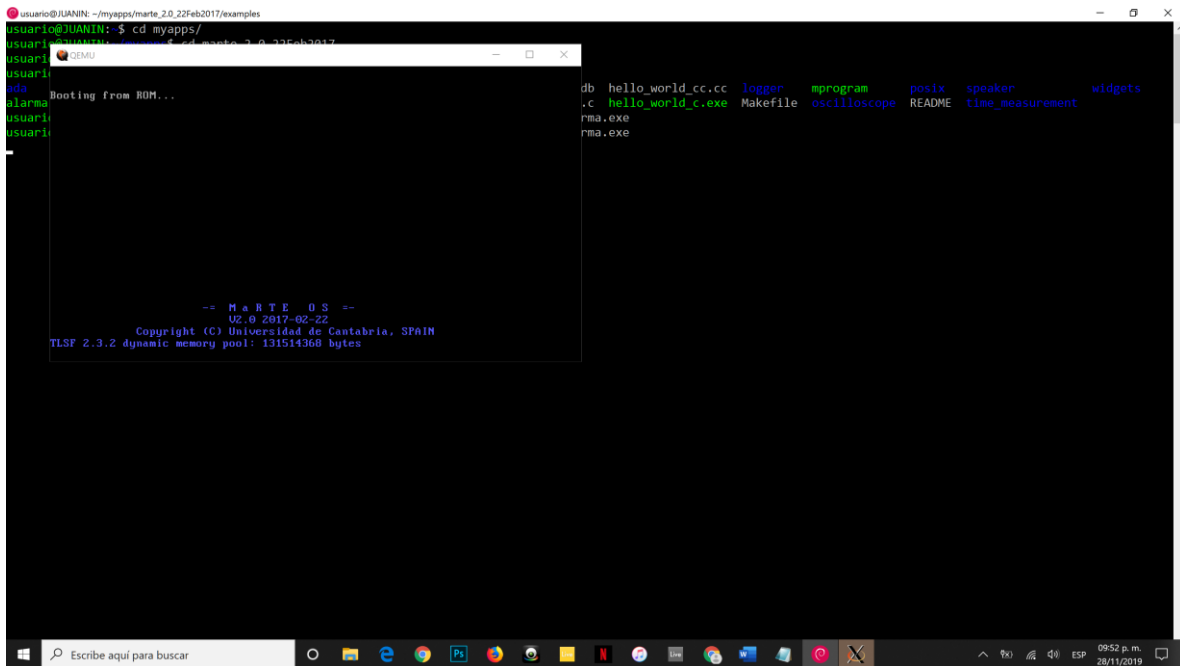
Archivos generados en la carpeta examples/.

Después de ello es necesario crear un archivo ejecutable y esto realiza con el comando “*make alarma.exe*”

Para realizar el siguiente paso y la comprobación del sistema y el programa sin problemas, es necesario descargar el software Xming para visualizar la ventana QEMU. Para descargarlo podemos seguir el siguiente enlace.

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

después es necesario ejecutar la siguiente instrucción “*qemu-system-i386 -kernel alarma.exe*”, esta instrucción nos ayudara a visualizar en una venta de qemu, el archivo ejecutable corriendo sobre MarteOs.



Visualización de MarteOS en la ventana QEMU

Para poder hacer portable el programa es necesario generar un archivo .iso para ello tenemos que descargar la siguiente carpeta

URL:

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

Una vez descargada es necesario mover nuestro archivo "alarma.exe" a la dirección que se especifica en el enlace

movido el archivo a la carpeta ejecutamos el comando "*make clean*" y posteriormente "*cp -v alarma kernel.elf*", esto hace que el archivo kernel del archivo iso sea el archivo Alarma que se compilo. Por último tecleamos el comando "*make*" el cual genera el archivo "OS.iso".

como ultimo paso se utilizara el archivo iso para crear una live usb y poder correr la aplicación como un Sistema operativo en tiempo real.

Para hacer una live usb, se utilizo el software LinuxLive USB Creator.

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

Una vez descargado y en ejecución solo es necesario seleccionar un dispositivo USB, seleccionar el archivo ISO y dar click en el icono de rayo para que comience la instalación. Al finalizar este proceso ya se tiene la memoria USB lista para usarla en cualquier PC.



Para correr la aplicación de MarteOs con el programa alarma solo es necesario utilizar la USB en cualquier PC y la interfaz se vería como en la siguiente imagen.

