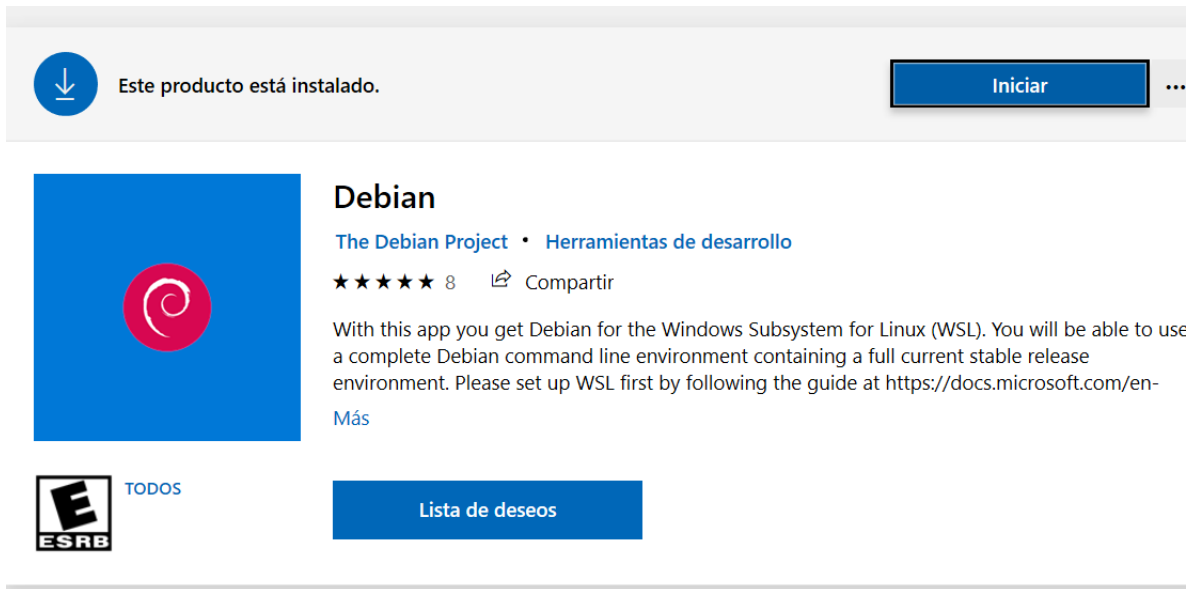


## Creación de una live USB con sistema operativo Marte OS corriendo un programa de alarma desde Debian

Como primer paso debemos descargaremos la distribución Debian desde la Microsoft store



Una vez instalado debían creamos una carpeta llama myapps, e ingresamos a la misma

```
omamoga@DESKTOP-5UM39DF: ~/myapps$
```

A continuación, descargaremos los dos archivos necesarios para la instalación de Marte OS que son

- gnat-gpl-2016-x86\_64-linux-bin.tar.gz
- marte\_2.0\_22Feb2017\_src.tar.gz

El primer archivo podemos encontrarlo en el siguiente enlace:

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

y los descargamos por medio del comando wget

```
omamoga@DESKTOP-5UMB9DF:~$ ls
gnat-gpl-2016-x86_64-linux-bin  alarma1  alarma.c  bashrc.save  myapps  sotr-201608-201617
5739cefdc7a447658e0b016b      a.out   a.out     myapps
omamoga@DESKTOP-5UMB9DF:~$ wget http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b
--2019-11-16 07:25:40-- http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b
Resolving mirrors.cdn.adacore.com (mirrors.cdn.adacore.com)... 205.232.38.142
Connecting to mirrors.cdn.adacore.com (mirrors.cdn.adacore.com)|205.232.38.142|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 372813330 (356M) [application/x-gzip]
Saving to: '5739cefdc7a447658e0b016b'

5739cefdc7a447658e0b016b  0%[
```

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

[https://marte.unican.es/marte/marte\\_2.0\\_22Feb2017\\_src.tar.gz](https://marte.unican.es/marte/marte_2.0_22Feb2017_src.tar.gz)

El primer archivo como es de suponerse es el compilador GNAT, mientras que el segundo es el Sistema operativo en tiempo real.

Entonces nos queda dentro de la carpeta los dos archivos descargados anteriormente

```
omamoga@DESKTOP-5UMB9DF:~/myapps$ ls
5739cefdc7a447658e0b016b  gnat-gpl-2016-x86_64-linux-bin  marte_2.0_22Feb2017_src.tar.gz
gnat-gpl-2016-x86_64-linux-bin  marte_2.0_22Feb2017_src.tar.gz
omamoga@DESKTOP-5UMB9DF:~/myapps$
```

El siguiente paso es descomprimir los archivos que contienen el compilador GNAT así como el SOTR, comenzamos con el compilador GNAT, dentro de la carpeta myapps damos click derecho y abrimos un nuevo terminal, acto seguido ingresamos la siguiente instrucción “`tar xvf 5739cefdc7a447658e0b016b`”, una vez terminado este proceso, procederemos a descomprimir el archivo contenedor del SOTR y eso lo haremos con la siguiente instrucción “`tar xvf marte_2.0_22Feb2017_src.tar.gz`”.

Una vez que hemos descomprimido los archivos contenedores del compilador y el SOTR, lo siguiente es modificar el archivo bashrc para poder instalar el SOTR sin ningún problema, para ello haremos lo siguiente, en la misma terminal, será necesario salir a la raíz del sistema, esto lo lograremos con la instrucción “`cd`”, una vez ahí, ejecutaremos el comando “`nano .bashrc`”, como se muestra a continuación.

```
omamoga@DESKTOP-5UMB9DF:~$ nano .bashrc
```

Una vez en la raíz y abriendo el archivo bashrc agregaremos al final del archivo las siguientes líneas:

```
export PATH=$HOME/myapps/gnat/bin:$PATH
export PERLLIB=$HOME/myapps/marte_2.0_22Feb2017
export PATH=$PATH:$HOME/myapps/marte_2.0_22Feb2017/Utils
export DISPLAY=:0
```

Una vez modificado el archivo salimos del editor y guardamos lo cambio que recién realizamos, para eso presionamos CTRL + O para guardar y después salimos con CTRL + X.

Una vez realizado esto cerramos la terminal actual para que los cambios realizados se vean reflejados, una vez mas entramos a la carpeta de my apps, una vez dentro ingresamos a la carpeta del gnat-gpl-2016-x86\_64-linux-bin

```
omamoga@DESKTOP-5UM39DF:~/myapps$ cd gnat-gpl-2016-x86_64-linux-bin/  
omamoga@DESKTOP-5UM39DF:~/myapps/gnat-gpl-2016-x86_64-linux-bin$
```

Una vez dentro ejecutamos la instrucción `./doinstall` para comenzar la instalación de gnat.

```
omamoga@DESKTOP-5UM39DF:~/myapps/gnat-gpl-2016-x86_64-linux-bin$ ./doinstall
```

Una vez ejecutado el comando anterior se mostrará 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.
```

A continuación, debemos especificar un directorio en donde deseamos instalar GNAT, si desconocemos la dirección de la carpeta donde deseamos instalar podemos suspender un momento la instalación y nos dirigimos al directorio donde deseamos instalar el gnat y tecleamos el comando `"pwd"`

```
omamoga@DESKTOP-5UM39DF:~/myapps/gnat$ pwd  
/home/omamoga/myapps/gnat  
omamoga@DESKTOP-5UM39DF:~/myapps/gnat$
```

Una vez que conocemos el directorio procedemos a continuar con la instalación de GNAT, una vez terminada la instalación procedemos a salir de la carpeta e ingresar a la carpeta de marte.

```
omamoga@DESKTOP-5UM39DF:~/myapps$ cd marte_2.0_22Feb2017  
omamoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017$
```

Ya dentro de la carpeta de marte ejecutamos la instrucción de `"./minstall"`

```
omamoga@DESKTOP-5UM39DF:~/myapps$ cd marte_2.0_22Feb2017  
omamoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017$ ./minstall
```

En cuanto ejecutamos el código anterior se mostrará el siguiente texto

```

omamoga@DESKTOP-5UMB9DF:~/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/omamoga/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...

```

Presionamos la tecla enter para comenzar la instalación de marte OS , una vez terminado el proceso accedemos a la carpeta “utils” y ejecutamos la instrucción “msetcurrentarch” para definir la arquitectura .

```

omamoga@DESKTOP-5UMB9DF:~/myapps/marte_2.0_22Feb2017$ ls
arch      gnat_rts      lang_support  marte_ug.html  posix5      tasks_inspector  xtratum_arch
Changelog INSTALL      lib          minstall       README      tests
COPYING   INSTALL_RPi.txt linux_arch    misc          rpi_arch    utils
examples  kernel       linux_lib_arch  objs         sll         x86_arch

omamoga@DESKTOP-5UMB9DF:~/myapps/marte_2.0_22Feb2017$ cd utils/
omamoga@DESKTOP-5UMB9DF:~/myapps/marte_2.0_22Feb2017/utils$ msetcurrentarch
Use of uninitialized value $GNAT_LIBS_PATH{"rpi"} in concatenation (.) or string at /home/omamoga/myapps/marte_2.0_22Feb2017/Utils/templates/globals.pl line 16.
Current architecture:none

Available architectures status:
x86:      RTS (gnat_rts/rts-marteuc_x86): Compiled
          Lib MaRTE (objs/x86_objs):      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

```

Volvemos a ejecutar el comando pero de la siguiente forma “msetcurrentarch x86 i386 ”

```

omamoga@DESKTOP-5UMB9DF:~/myapps/marte_2.0_22Feb2017/utils$ msetcurrentarch x86 i386
Use of uninitialized value $GNAT_LIBS_PATH{"rpi"} in concatenation (.) or string at /home/omamoga/myapps/marte_2.0_22Feb2017/Utils/templates/globals.pl line 16.

```

Para continuar, seguiremos las indicaciones que vienen hasta abajo, que es ejecutar las instrucciones “mkrtsmarteuc” y posteriormente “mkmarte”, es importante que sea en ese orden, de lo contrario visualizaremos una ventana de error.

Hasta este punto, terminamos con la instalación del SOTR MaRTE OS.

## Desarrollo del código de alarma

Para la lógica del programa lo que hacemos es primero pedirle la hora al sistema y después procedemos a convertir tanto la hora de alarma como la hora actual en segundos y después las restamos mientras que la diferencia sea positiva se seguirá preguntando la hora convirtiéndola a segundos y restándola hasta que la diferencia sea cero o menor, una vez llegado a ese punto se lanza un mensaje de alarma. A continuación de anexa el código del programa.

```
#include <stdio.h>
#include <time.h>
// driver code
int main()
{
    time_t seconds,s, val = 1;
    struct tm* current_time;
    int hora,minutos,segundos,hora_actual;
    int hour,min,sec,t_sec,tiempo_total;
    int p = 0,diferencia;
    // time in seconds
    s = time(NULL);
    time(&seconds);

    // to get current time
    current_time = localtime(&s);

    // print time in minutes,
    // hours and seconds
    printf("La hora actual es : %02d:%02d:%02d\n",
        current_time->tm_hour,
        current_time->tm_min,
        current_time->tm_sec);

    printf ("A que hora desea programar su alarma?\n");
    printf ("introduzca la hora \n");
    scanf ("%02d",&hora);
    printf ("introduzca los minutos\n");
    scanf ("%02d",&minutos);
    printf ("introduzca los segundos\n");
    scanf ("%02d",&segundos);

    hour = current_time->tm_hour;
    min = current_time->tm_min;
    sec = current_time->tm_sec;

    if ((hora < hour)|| (hora == hour && minutos < min )|| (hora == hour && minutos == min && segundos < sec ))
    {
        printf ("la hora ya paso\n");
    }
    else
    {
        t_sec = ((hora)*3600)+((minutos)*60)+(segundos);
    }
}
```

```
//tiempo_total = time(&seconds) + t_sec;

while ( diferencia >= 0 )
{
    s=time(NULL);
    time(&seconds);
    current_time=localtime(&s);
    hour = current_time->tm_hour;
    min = current_time->tm_min;
    sec = current_time->tm_sec;

    hora_actual = (hour*3600)+(min*60)+sec;

    diferencia = t_sec - hora_actual;

    //      printf("%d\n",diferencia);
}

if (diferencia <= 0)
{
    printf("ALARMA!!!!\n");
    printf("ALARMA!!!!\n");
    printf("ALARMA!!!!\n");
}

return 0;
}
```

Ahora una vez que tenemos nuestro código procedemos a compilarlo por medio del comando `gcc` como se muestra en la imagen siguiente.

```
namoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017/examples$ gcc timer.c -o mprogram
```

Ya compilado el código generamos un archivo ejecutable correspondiente a nuestro código.

```
omamoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017/examples$ make timer.exe
```

Antes de proceder a los siguientes pasos necesitamos descargar o clonar del repositorio de la materia de SOTR y por conveniencia la ubicamos en el escritorio de la computadora.

Una vez que tenemos nuestro ejecutable y la carpeta en el escritorio procedemos a copiarlo en la siguiente ruta .

```
cp -v timer.exe /mnt/c/Users/Asus/Desktop/sotr_201808_201812-master/PRACTICA_1_Instalacion_de_un_SOTR/MAKEFILE_GRUB_Legacy_Kernel_elf/
```

Ya copiado el archivo accedemos a esa ruta .

```
cd /mnt/c/Users/Asus/Desktop/sotr_201808_201812-master/PRACTICA_1_Instalacion_de_un_SOTR/MAKEFILE_GRUB_Legacy_Kernel_elf/
```








Ejecutamos la siguiente línea de código

```
omamoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017/examples$ cp -v timer.exe kernel.elf
```

Y finalmente generamos el archivo iso que incluya el código de la alarma con el comando “make”

```
omamoga@DESKTOP-5UM39DF:~/myapps/marte_2.0_22Feb2017/examples$ make
```

Finalmente entramos a la carpeta de SOTR y deberíamos encontrar un archivo ISO listo para montar a una memoria USB

 iso	9/5/2019 12:29 PM	Carpeta de archivos	
 hello_world_c	9/17/2019 3:33 PM	Aplicación	905 KB
 kernel.elf	10/30/2019 1:56 PM	Archivo ELF	906 KB
 kernelmemfs	9/6/2019 3:55 PM	Archivo	673 KB
 Makefile	9/5/2019 12:29 PM	Archivo	1 KB
 os	10/30/2019 1:57 PM	Archivo WinRAR	1,370 KB
 timer	10/30/2019 1:56 PM	Aplicación	906 KB

Se recomienda montar una live usb por medio del programa LiLi USB creator o por medio de Rufus (no probado en clase)

