Este archivo es una guía en la cual se explicará el proceso para generar un archivo iso de un programa que realiza la acción de una Alarma mediante el uso del sistema operativo en tiempo real MarteOS y el compilador Gnat, este archivo iso se cargará en una memoria USB con el uso del programa Linux Live USB para posteriormente correr el programa como un sistema operativo en una computadora.

El primer paso es tener instalado una plataforma de Linux, en este caso Debian y se debe contar con los siguientes archivos:

* gnat-gpl-2016-x86\_64-linux-bin.tar.gz

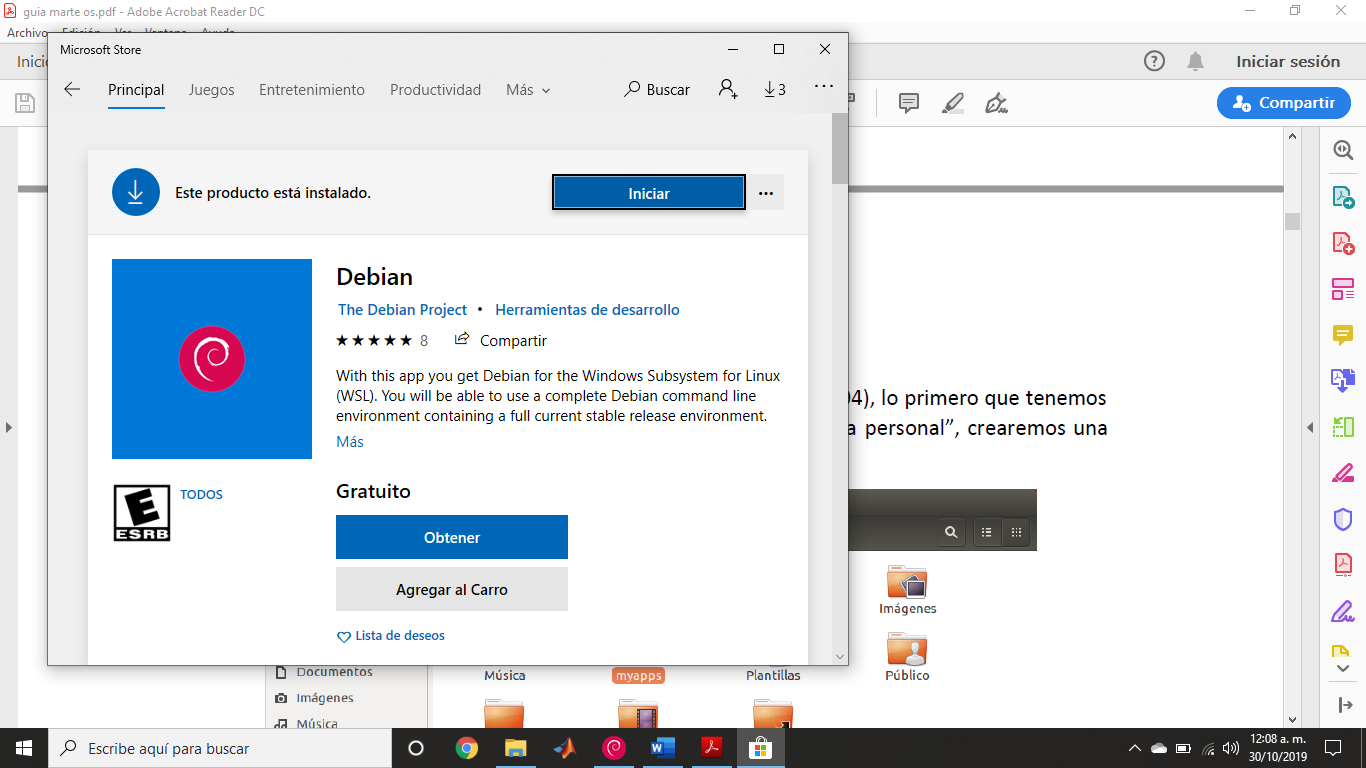
que se puede descargar del siguiente enlace http://mirrors.cdn.adacore.com/art/5739cefdc7a447658e0b016b

* marte\_2.0\_22Feb2017\_src.tar.gz

que se puede encontrar en <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.

Para obtener Debian es necesario considerar la versión debido a problemas de compatibilidad con el compilador gnat, se puede descargar directamente de la tienda de Windows



Una vez descargados los archivos gnat-gpl-2016-x86\_64-linux-bin.tar.gz y marte\_2.0\_22Feb2017\_src.tar.gz los debemos guardar dentro de una carpeta en nuestro ordenador, dentro de la misma creamos una carpeta llamada gnat. Posteriormente procedemos a descomprimir los archivos mediante los siguientes comandos:

*tar xvf gnat-gpl-2016-x86\_64-linux-bin.tar.gz para descomprimir el compilador*

*tar xvf marte\_2.0\_22Feb2017\_src.tar.gz para descomprimir el Sistema operativo en tiempo real*

Antes de comenzar debemos mover la carpeta mediante el comando

Cp -rv /mnt/c/Users/Vic/Downloads/marte/gnat gnat

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, ejecutamos el comando “*nano .bashrc”*,

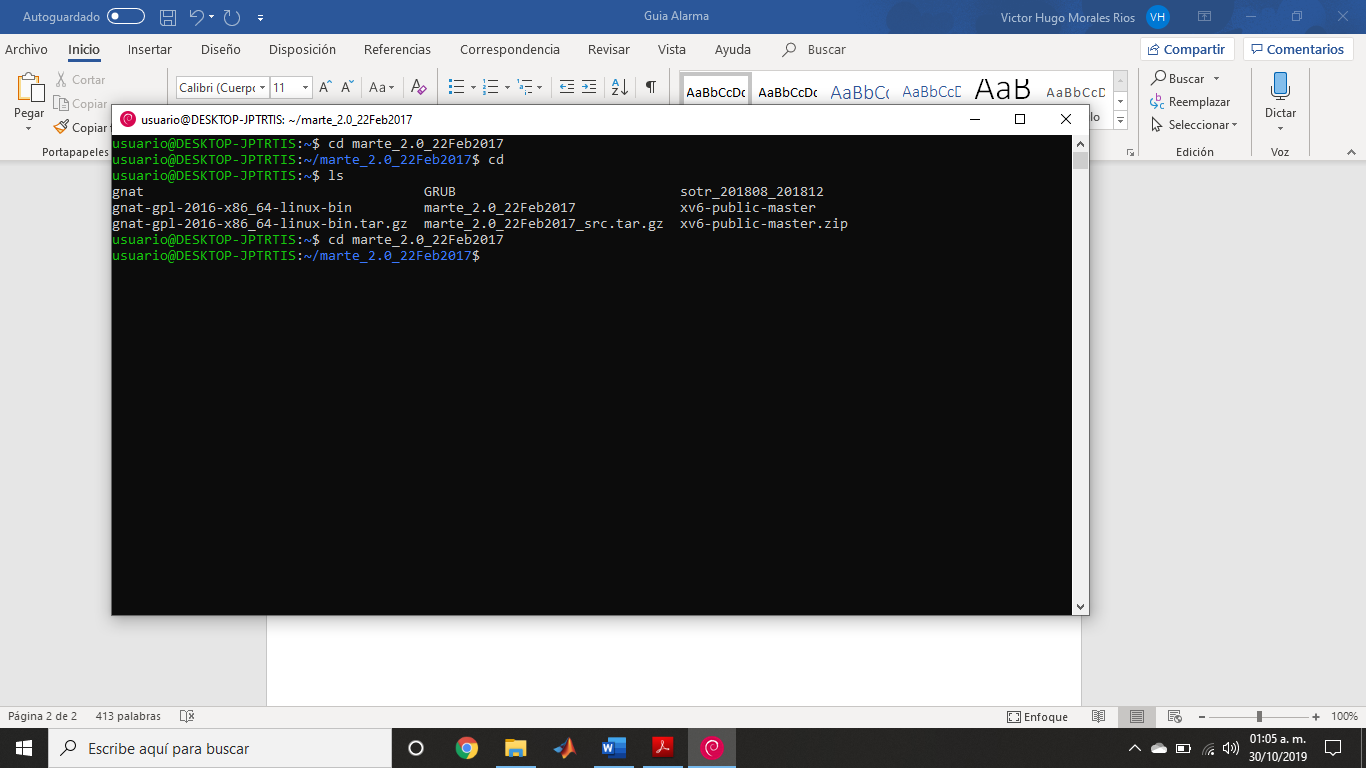
Dentro de este archivo, iremos al final y agregaremos las siguientes líneas:

* • 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

Al terminar la edición del archivo. bashrc debemos guardar los cambios con los comandos control+o y salir con control+x cabe mencionar que cada vez que se realizan cambios en el archivo. bashrc se debe reiniciar Debian para que los cambios se vean reflejados.

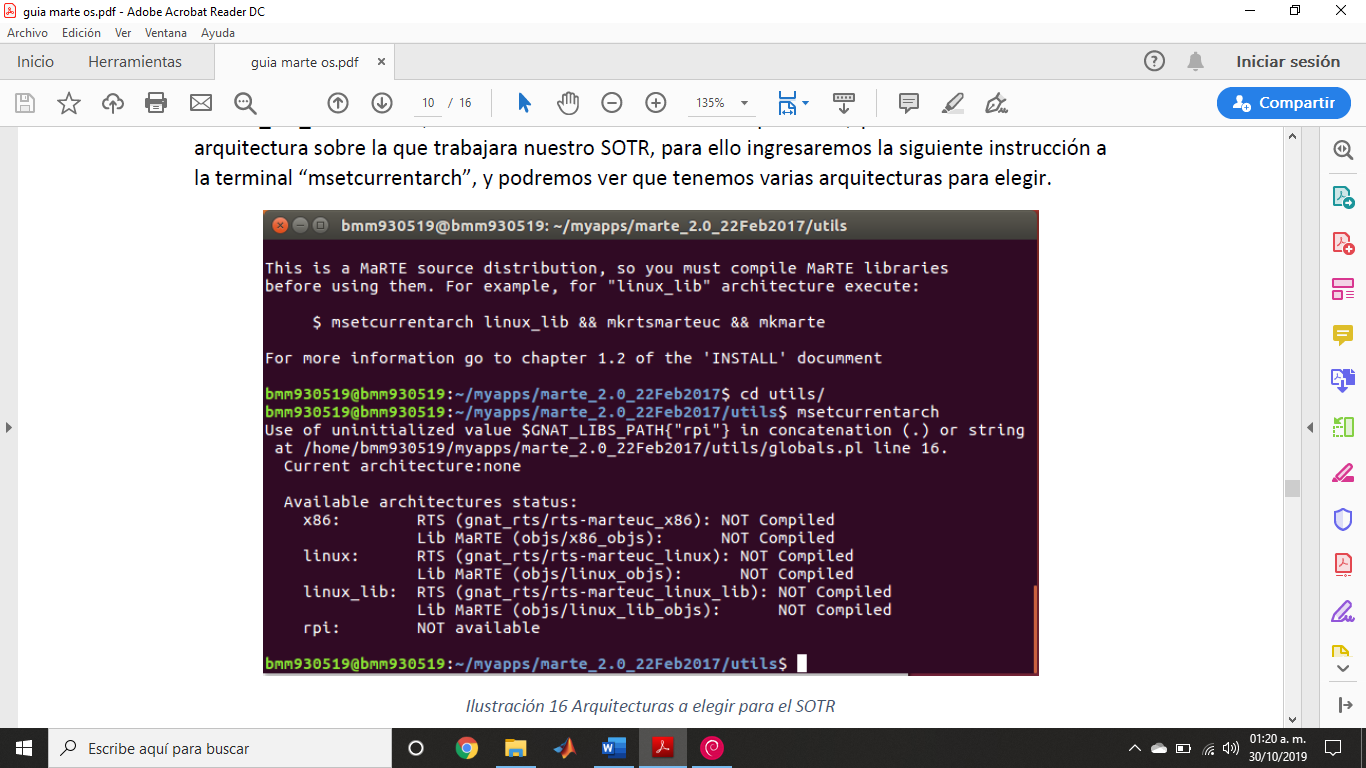
Una vez hecho eso ingresamos al directorio gnat mediante el comando “*cd gnat-gpl-2016-x86\_64-linux-bin/”* una vez situados en esa carpeta ejecutaremos la siguiente instrucción “*./doinstall*” y aceptamos, se nos preguntara en donde queremos instalar el compilador, para ello debemos seleccionar la carpeta gnat que creamos al inicio, para eso debemos ingresar a la carpeta y copiar la ruta de la misma, un ejemplo es /c/Users/Víc/Downloads/marte/gnat/ y damos enter y aceptar hasta que comience la instalación.

Al finalizar se crea el directorio “marte\_2.0\_22Feb2017” y para ingresar a el debemos regresar a home con la instrucción cd+enter y luego ejecutar cd marte\_2.0\_22Feb2017 para ingresar



Una vez ahí procederemos a instalar el sistema operativo en tiempo real, para lo cual ejecutaremos la instrucción “./minstall” y aceptamos hasta que comienza el proceso de instalación.

lo siguientes es acceder a la carpeta “utils”, lo que lograremos con la siguiente instrucción “cd utils/” ya que esta carpeta está dentro de la carpeta “marte\_2.0\_22Feb2017”, una vez estando dentro de la carpeta utils, procederemos a definir la arquitectura sobre la que trabajara nuestro SOTR, para ello ingresaremos la siguiente instrucción a la terminal “msetcurrentarch”, y podremos ver que tenemos varias arquitecturas para elegir.

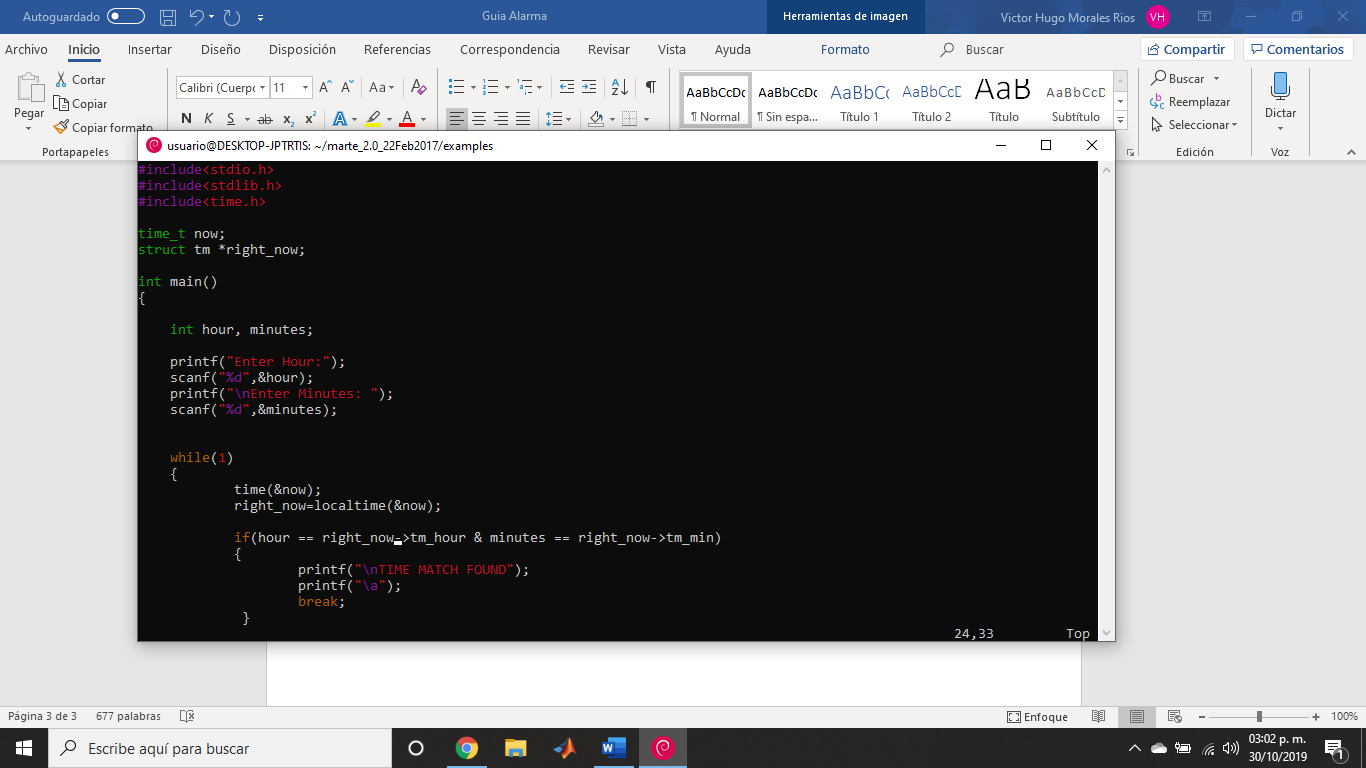


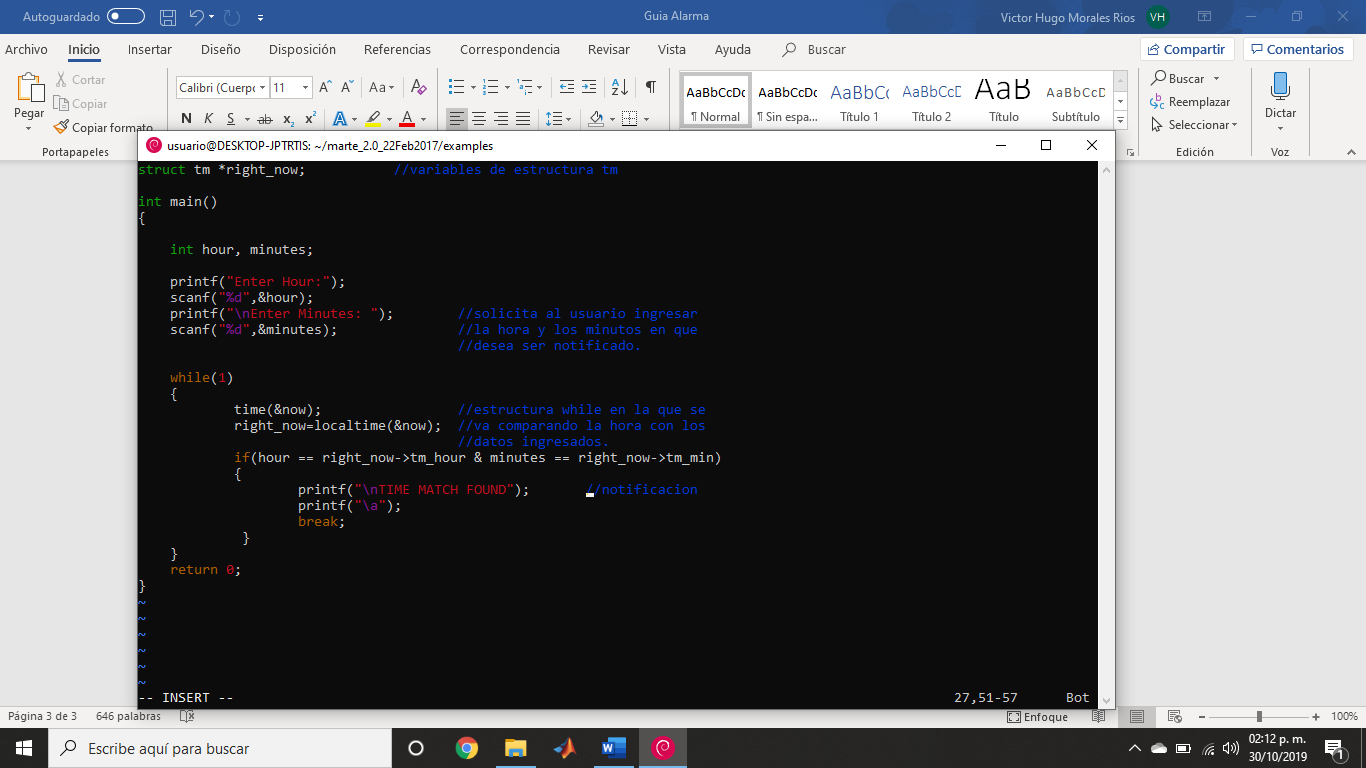
por lo que ingresaremos la siguiente instrucción en la terminal “msetcurrentarch x86 i386” con lo que se ha decidido utilizar la arquitectura x86.

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.

Al ejecutar mkmartre se termina de instalar el Sistema operativo marteOS.

Una vez terminado el proceso de instalación de marteOS procedemos a realizar un código en c que haga la función de una alarma, esto es que toma datos de la hora y solicita al usuario que ingrese la hora y los minutos en que desea se le notifique dicha alarma mediante un mensaje de terminación.

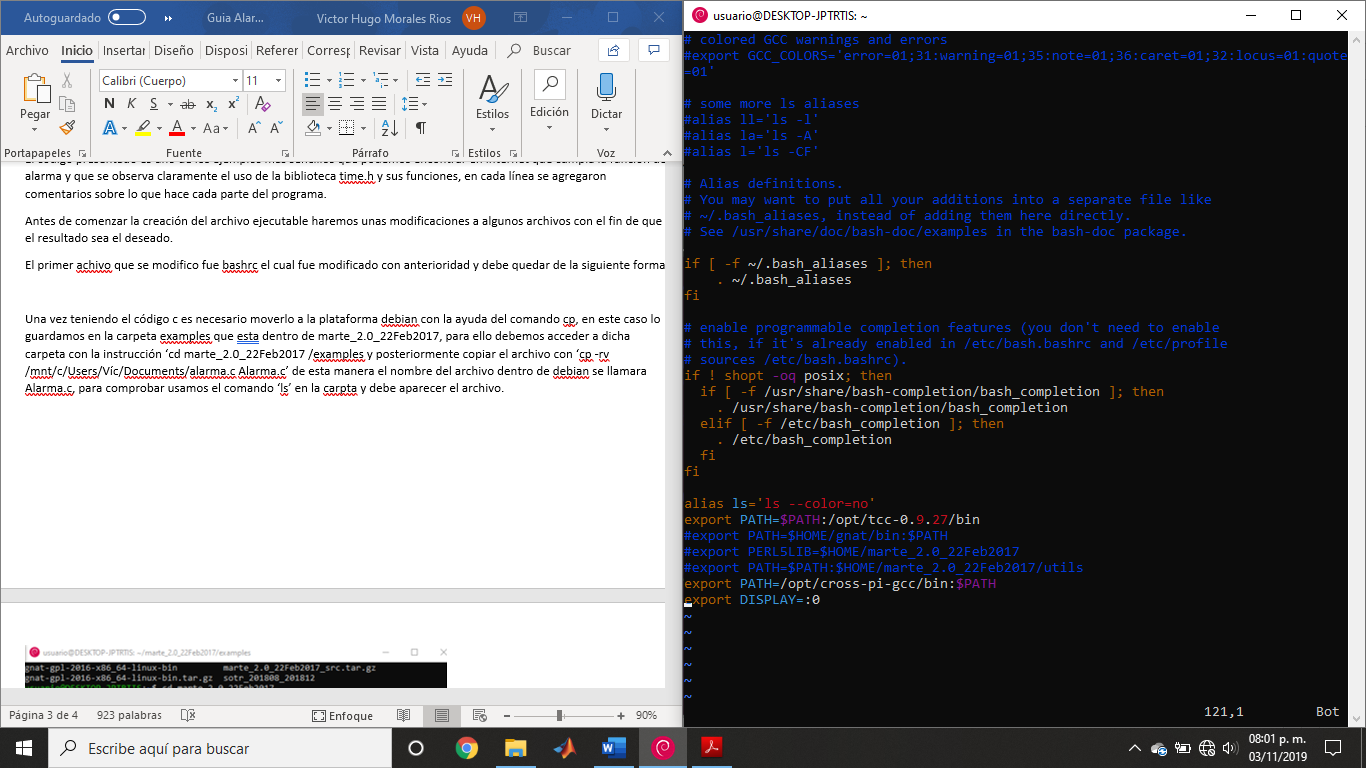
CODIGO EN LENGUAJE C



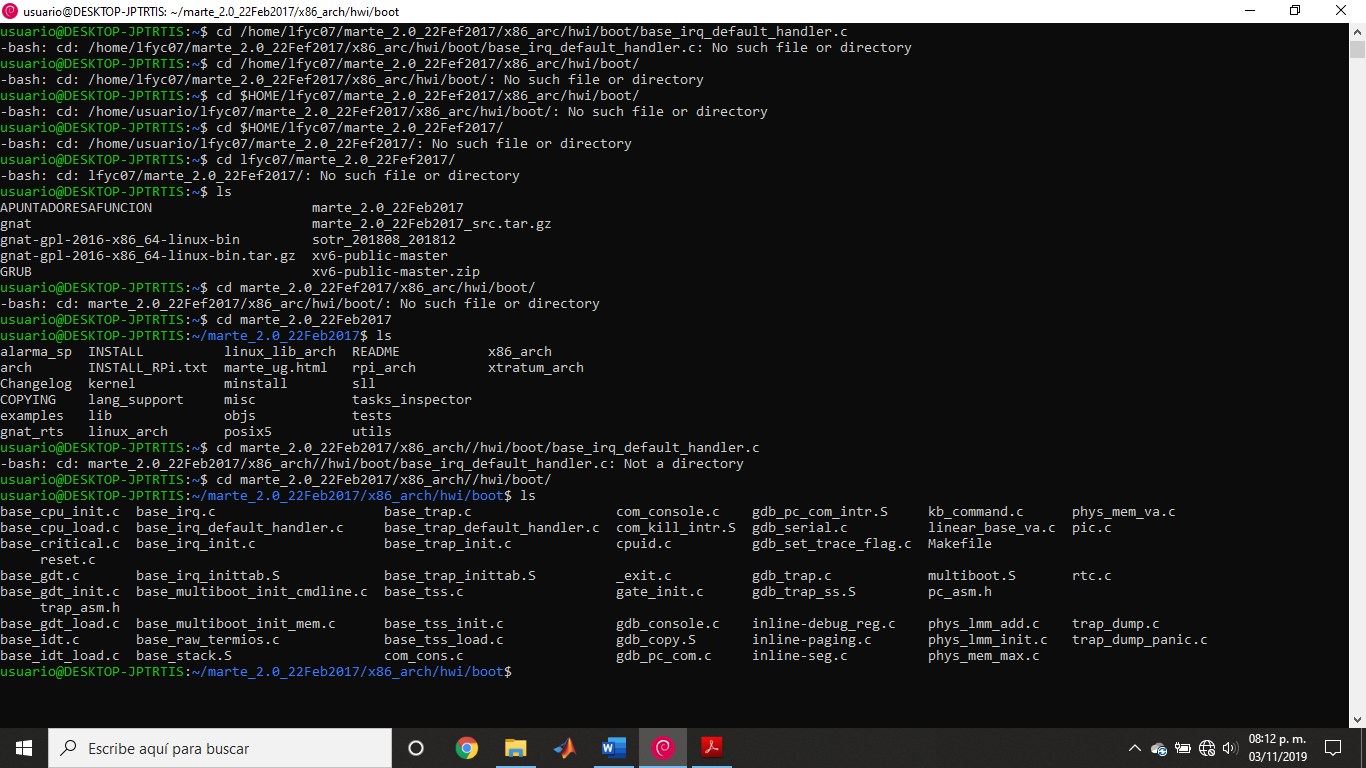
El código presentado es uno de los ejemplos más sencillos que podemos encontrar en internet que cumpla la función de alarma y que se observa claramente el uso de la biblioteca time.h y sus funciones, en cada línea se agregaron comentarios sobre lo que hace cada parte del programa.

Antes de comenzar la creación del archivo ejecutable haremos unas modificaciones a algunos archivos con el fin de que el resultado sea el deseado.

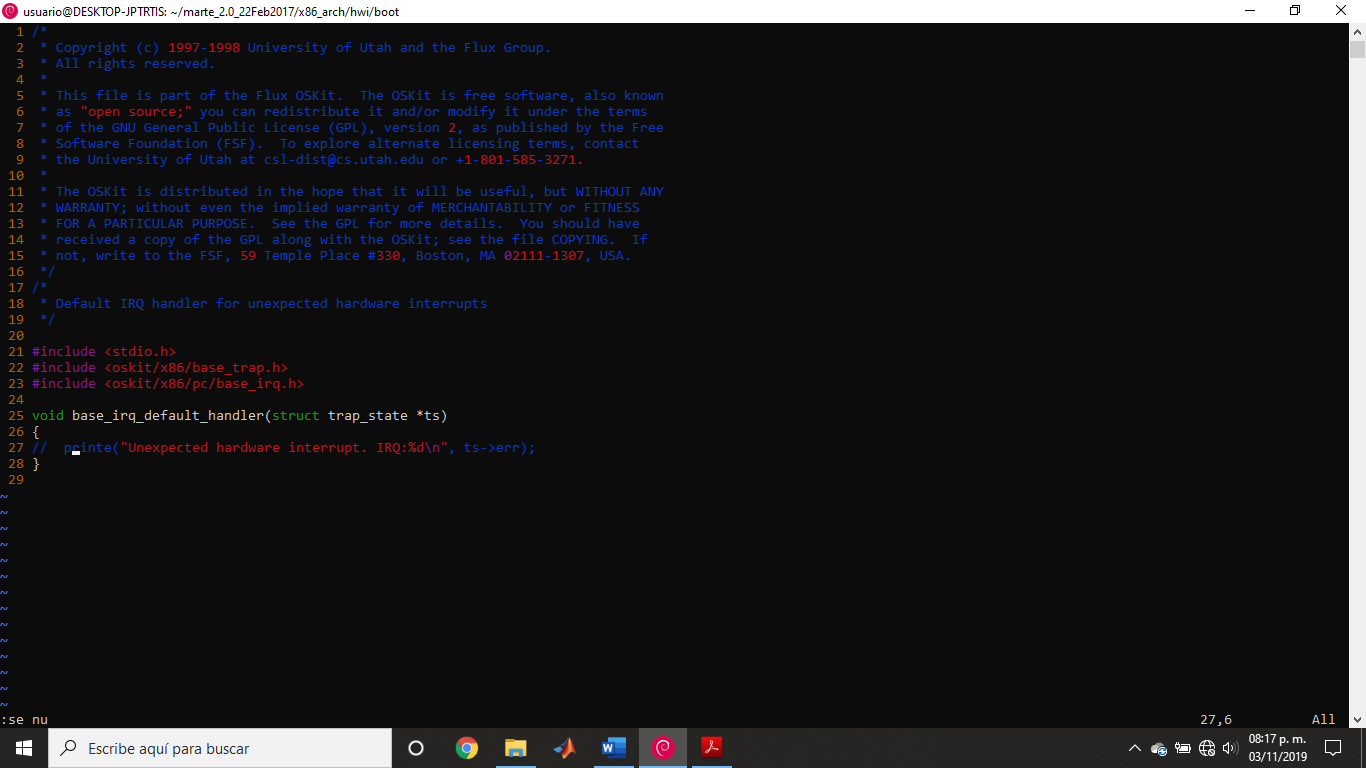
El primer archivo que se modifico fue bashrc el cual fue modificado con anterioridad y debe quedar de la siguiente forma.



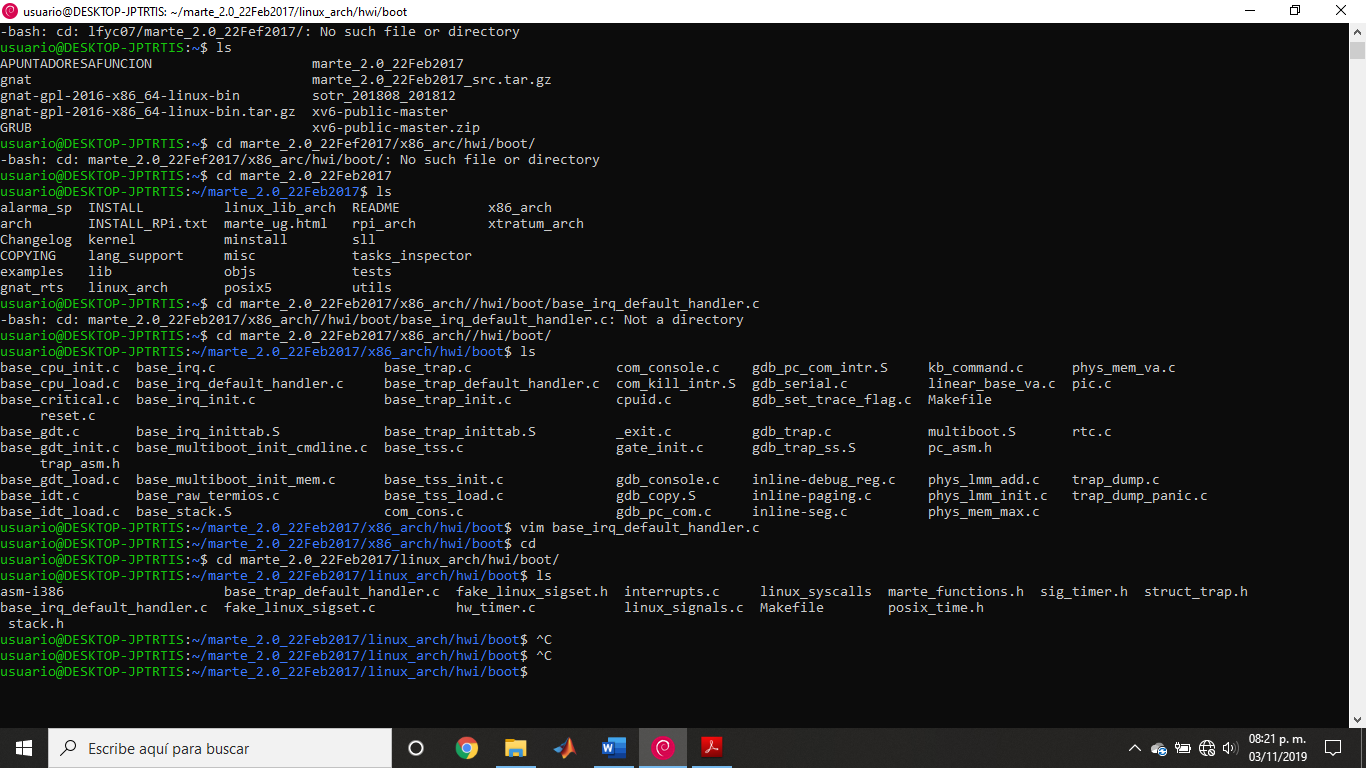
Los siguientes archivos a comentar sirven para evitar que a la hora de ejecutar el archivo no nos envíe un mensaje de error ya que en el tiempo que trascurre para que la alarma notifique el fin de ejecución, para ello debemos acceder a la siguientes rutas “cd marte\_2.0\_22Fef2017/x86\_arc/hwi/boot/” e ingresar al siguiente archivo con la instruccion “vim base\_irq\_default\_handler.c”



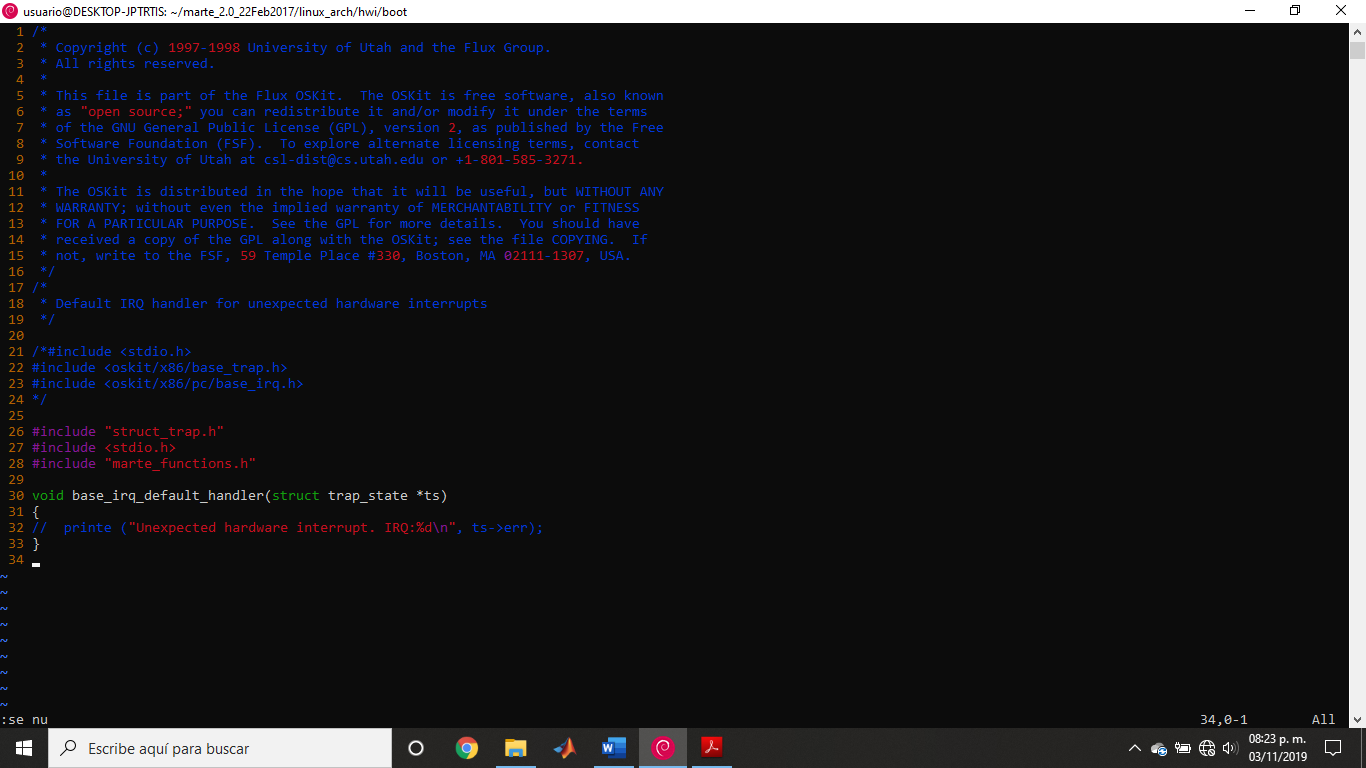
Y comentamos la línea 27



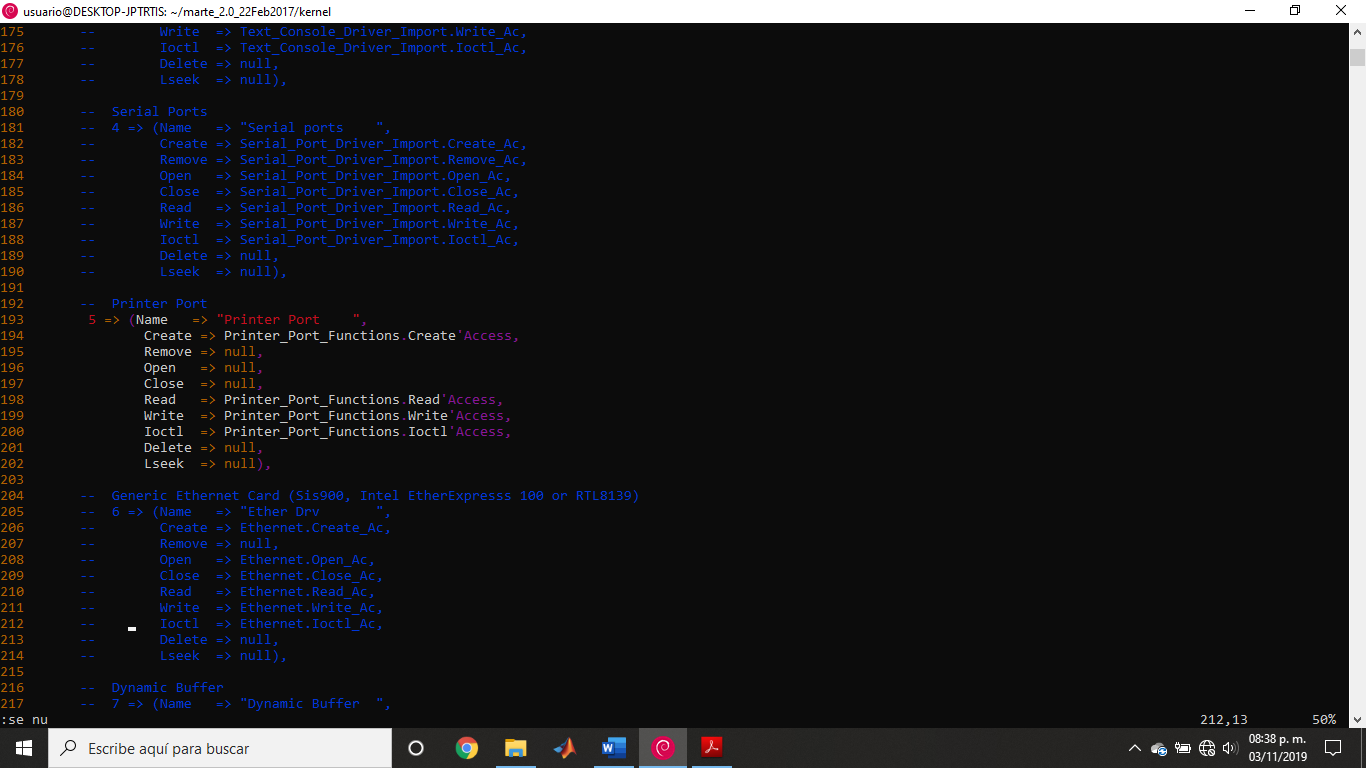
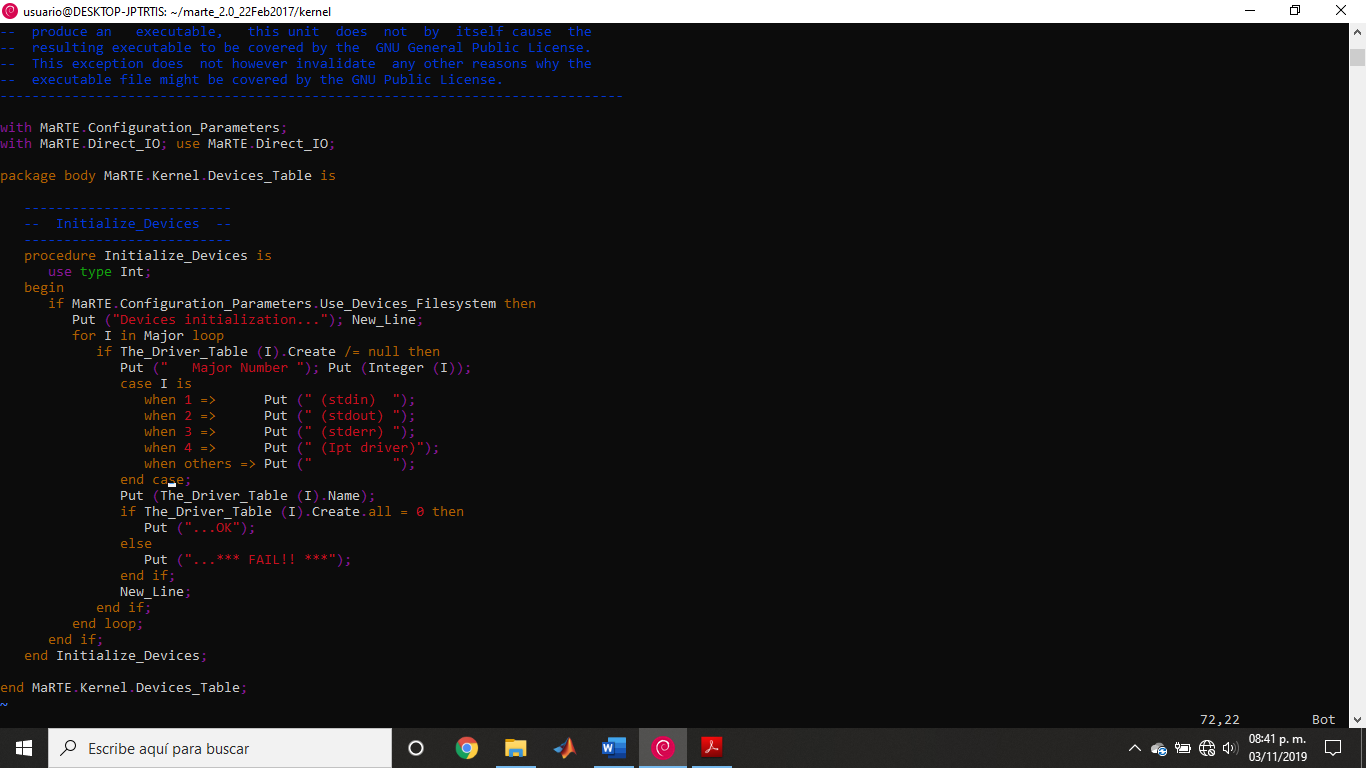
El siguiente es en la ruta “cd marte\_2.0\_22Feb2017/linux\_arch/hwi/boot/” e ingresar al siguiente archivo con la instrucción “vim base\_irq\_default\_handler.c”



Y comentamos la línea 32

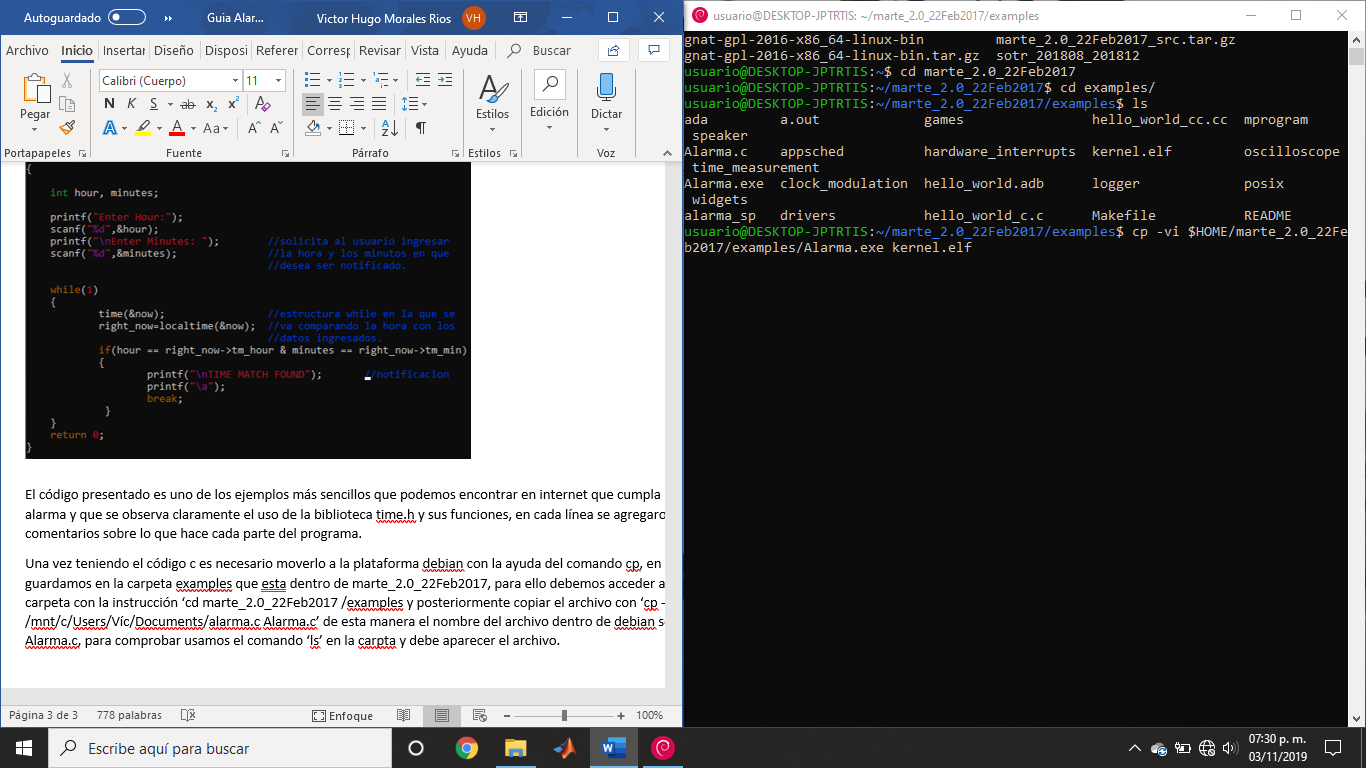


Los últimos archivo a modificar debemos ingresar a la siguiente ruta” cd marte\_2.0\_22Feb2017/kernel/” e ingresar al archivo “vim marte-kernel-devices\_table.ads” y des comentamos todo el bloque del caso 5, al salir ingresamos al archivo “vim marte-kernel-devices\_table.adb” y agregamos el caso “when 4 =>…”

Listo ahora podemos proceder a crear el archivo ejecutable y el archivo iso.

Una vez teniendo el código c es necesario moverlo a la plataforma debian con la ayuda del comando cp, en este caso lo guardamos en la carpeta examples que esta dentro de marte\_2.0\_22Feb2017, para ello debemos acceder a dicha carpeta con la instrucción ‘cd marte\_2.0\_22Feb2017 /examples y posteriormente copiar el archivo con ‘cp -rv /mnt/c/Users/Víc/Documents/alarma.c Alarma.c’ de esta manera el nombre del archivo dentro de debian se llamara Alarma.c, para comprobar usamos el comando ‘ls’ en la carpta y debe aparecer el archivo.



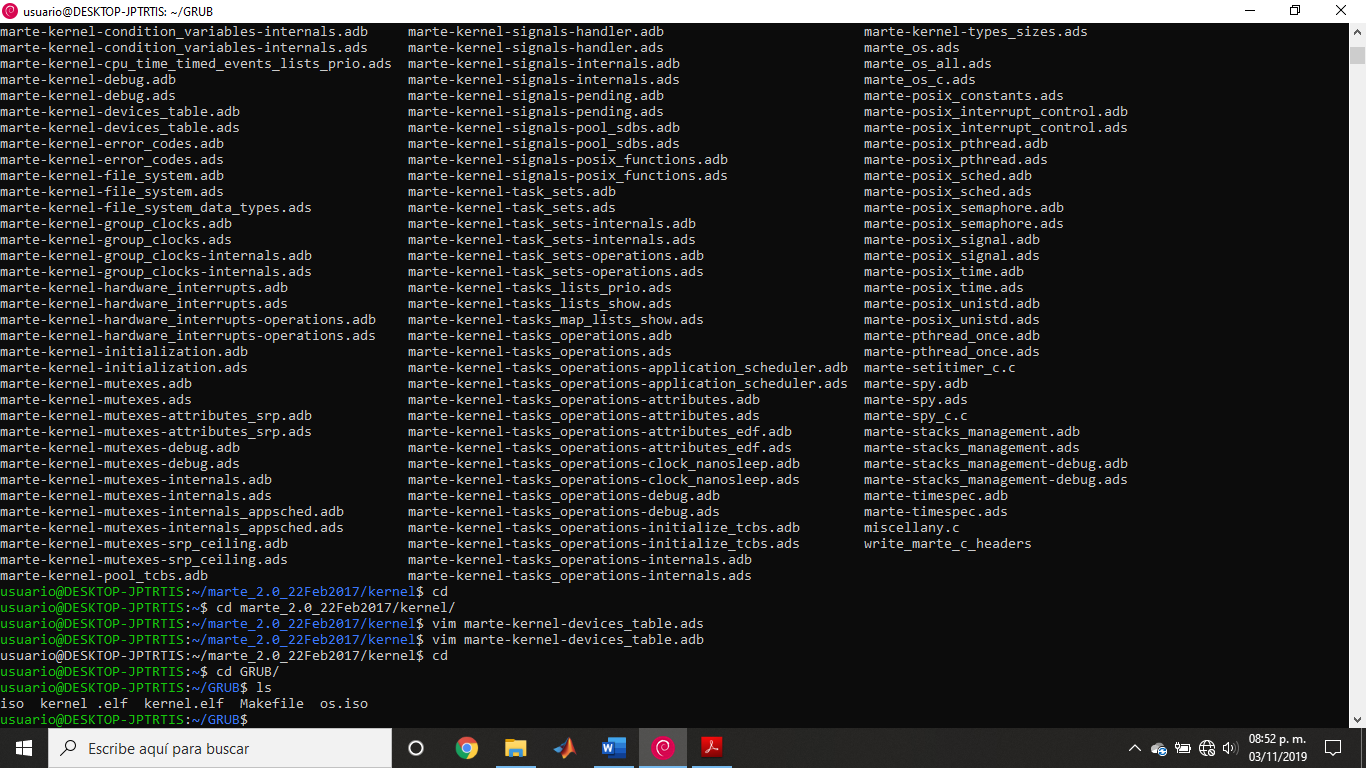
El siguiente paso es compilar el código, utilizamos el comando ‘mgcc Alarma.c’ con esto habremos compilado este archivo y una forma de verificar que la compilación fue exitosa es revisar que se haya creado un archivo de nombre “a.out”, volvemos a ejecutar la instrucción “ls” para verificar que en efecto esto paso.

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”.

Luego de esto crearemos un archivo ejecutable asociado al archivo C, esto lo lograremos con la siguiente instrucción “make Alarma.exe”.

Con eso hemos creado el archivo ejecutable, los ultimos pasos son parqa crear el archivo iso y cargarlo en una memoria.

El primer paso para esto es ingresar a la carpeta GRUB y copiar ahí el archivo Alarma.exe con el nombre kernel.elf para ello ingresamos el comando “cd GRUB” y posteriormente “cp -rv $HOME/marte\_2.0\_22Feb2017/ examples/Alarma.exe kernel.elf” después de ello escribimos “make” para crear el archivo os.iso para comprobarlo utilizamos el comando ls y debe aparecer el archivo kernel.elf y os.iso

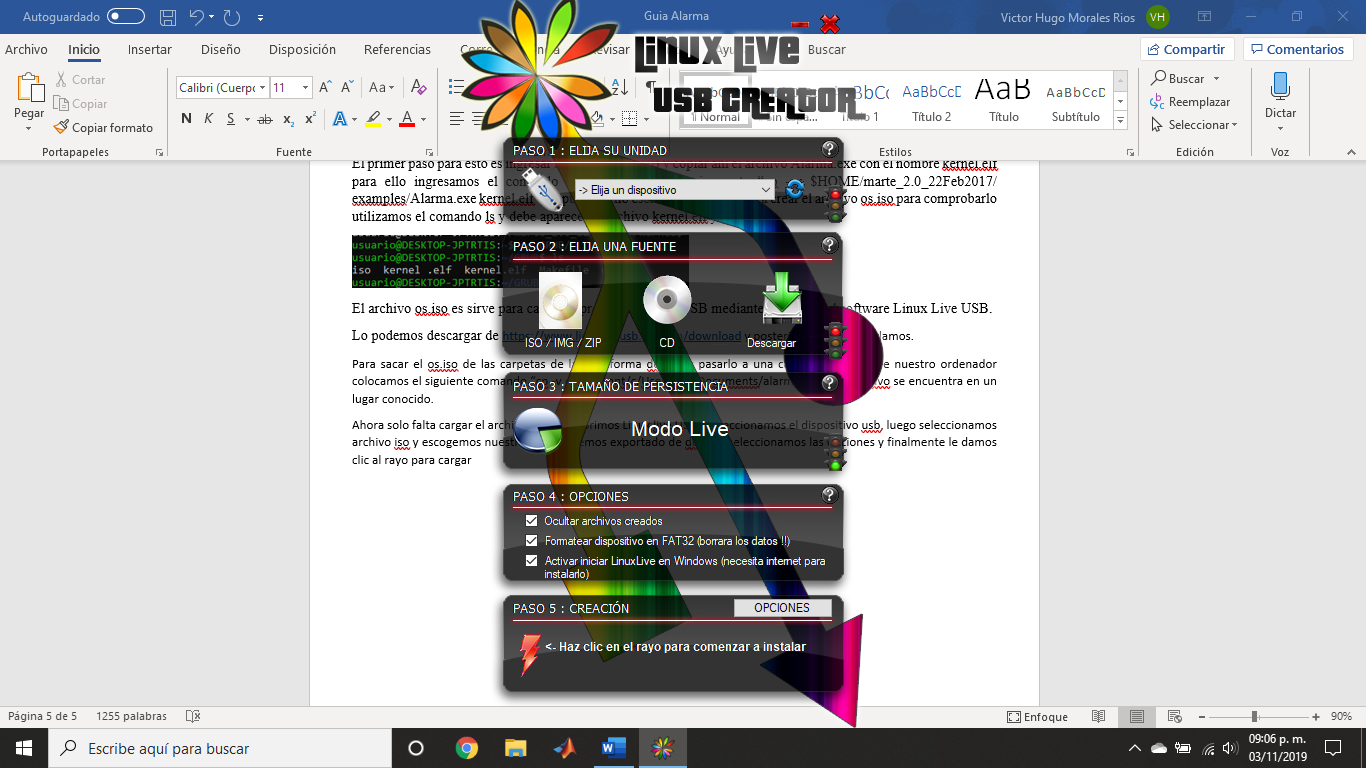


El archivo os.iso es sirve para cargar el programa en una USB mediante la ayuda del software Linux Live USB.

Lo podemos descargar de <https://www.linuxliveusb.com/en/download> y posteriormente lo instalamos.

Para sacar el os.iso de las carpetas de la plataforma debian y pasarlo a una carpeta conocida de nuestro ordenador colocamos el siguiente comando “cp -v os.iso /mnt/c/Users/Víc/Documents/alarma/” y asi el archvo se encuentra en un lugar conocido.

Ahora solo falta cargar el archivo a la usb, abrimos Linux live USB y seleccionamos el dispositivo usb, luego seleccionamos archivo iso y escogemos nuestro iso que hemos exportado de debian, seleccionamos las opciones y finalmente le damos clic al rayo para cargar



Finalmente probamos el programa booteando una computadora colocando la USB y luego iniciando el equipo.