D.L L.O

Sistemas operativos en tiempo real

Reporte alarma

Lino Olvera Diego Leonardo

# Alarma en marte 2

## Carpeta GRUB

Se descarga la carpeta MAKEILE\_GRUB\_Legacy\_Kernel\_elf de la siguiente URL: <https://github.com/sotrteacher/sotr_201808_201812/tree/master/PRACTICA_1_Instalacion_de_un_SOTR>. Una vez realizado esto, se traslada la carpeta a Devian, utilizando el siguiente comando:

Cp –rv /mnt/c/Users/D.L/Downloads/MAKEILE\_GRUB\_Legacy\_Kernel\_elf GRUB

De esta manera, la carpeta se movio a los directorios de Devian con el renombre de GRUB, que es más corto y fácil de usar.

## Instalación de marte

Se deben descargar los archivos:

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

Para esto, se utilizan los siguientes enlaces:

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

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

Una vez realizado esto, se procede a moverlos a los directorios internos de debían. Par ello, se utilizan las instrucciones “cp –rv /mnt/c/Users/D.L/Downloads/ gnat-gpl-2016-x86\_64-linux-bin.tar.gz”, “cp –rv /mnt/c/Users/D.L/Downloads/ marte\_2.0\_22Feb2017\_src.tar.gz”. También se crea una carpeta con el nombre de gnat.

El siguiente paso es descomprimir los archivos que contienen el compilador GNAT así como el SOTR, comenzamos con el compilador GNAT, se ingresa la siguiente instrucción “*tar xvf gnat-gpl-2016-x86\_64-linux-bin.tar.gz”*, 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”*. Como se muestra a continuación.

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.

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

Finalmente, como lo indica el archivo, hay que guardar los cambios efectuados (CTRL+O) para después salir del archivo (CTRL+X).

Es importante que luego de hacer esto, cerremos la terminal en la que nos encontrábamos, esto para que los cambios que acabamos de realizar puedan efectuarse exitosamente.

Se entra a la carpeta de GNAT, esto lo lograremos con la instrucción “cd gnat-gpl-2016-x86\_64-linux-bin/”, una vez situados en esa carpeta ejecutaremos la siguiente instrucción “./doinstall” y se nos presentara una ventana donde se nos describe la versión del compilador que vamos a instalar, para seguir con la instalación solo presionamos “ENTER”.

Posteriormente, se nos preguntara la dirección de la carpeta donde queremos instalar el compilador. En este caso, se trata de la carpeta “gnat”, entonces la cadena que ingresaremos, quedara de la siguiente manera “/home/gnat”, con esto estaremos indicando exactamente donde queremos que se instale el compilador.

Para proseguir solo oprimimos la tecla “ENTER”. Se nos preguntara si la ruta de instalación es la correcta, a lo que solo debemos oprimir “Y” y luego “ENTER”.

Acto seguido se nos preguntara si queremos proceder con la instalación, y haremos lo mismo que anteriormente presionamos la tecla “Y” y seguido de eso la tecla “ENTER”.

Lo siguiente es salir a la carpeta “myapps” y acceder a la carpeta “marte\_2.0\_22Feb2017”, para lograr eso, ejecutaremos la instrucción “cd ..” y luego de ello, ejecutaremos la instrucción “cd marte\_2.0\_22Feb2017/”.

Una vez ahí procederemos a instalar el SOTR, para lo cual ejecutaremos la instrucción “./minstall”, con lo que visualizaremos lo siguiente.

Para continuar presionamos la tecla “ENTER”. Y a diferencia del compilador este proceso es bastante rápido, por lo que pronto visualizaremos lo siguiente, en señal de que el proceso ha terminado exitosamente.

Estamos cerca de terminar la instalación, lo siguientes es acceder a la carpeta “utils”, lo que lograremos con la siguiente instrucción “cd utils/” ya que esta carpeta esta 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.

Entre ellas están:

* • X86
* • Linux
* • Linux\_lib
* • Rpi

Para esta instalación, elegimos la arquitectura x86, por lo que ingresaremos la siguiente instrucción en la terminal “msetcurrentarch x86 i386”, una vez ejecutada, visualizaremos lo siguiente.

Como podemos ver, la arquitectura que elegimos ha sido puesta como default para el funcionamiento de nuestro SOTR.

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.

La primera instrucción tardará un poco en ejecutarse pero es importante no cerrar la ventana de la terminal. Una vez que visualicemos lo siguiente, podemos continuar a ejecutar la instrucción “mkmarte”.

## Ejecución del archivo

Teniendo el archivo Alarma2.c se realiza lo siguiente:

Se deben modificar las fuentes de marte para evitar que aparezca el error “Unexpected hardware interrupt”. Para ello se deben modificar los siguientes archivos:

* marte\_2.0\_22Feb2017/x86\_arch/hwi/boot/base\_irq\_default\_handler.c
* marte\_2.0\_22Feb2017/x86\_arch/Linux\_arch/hwi/boot/base\_irq\_default\_handler.c

Dentro de estos archivos, se debe comentar la línea que se muestra en la siguiente figura:

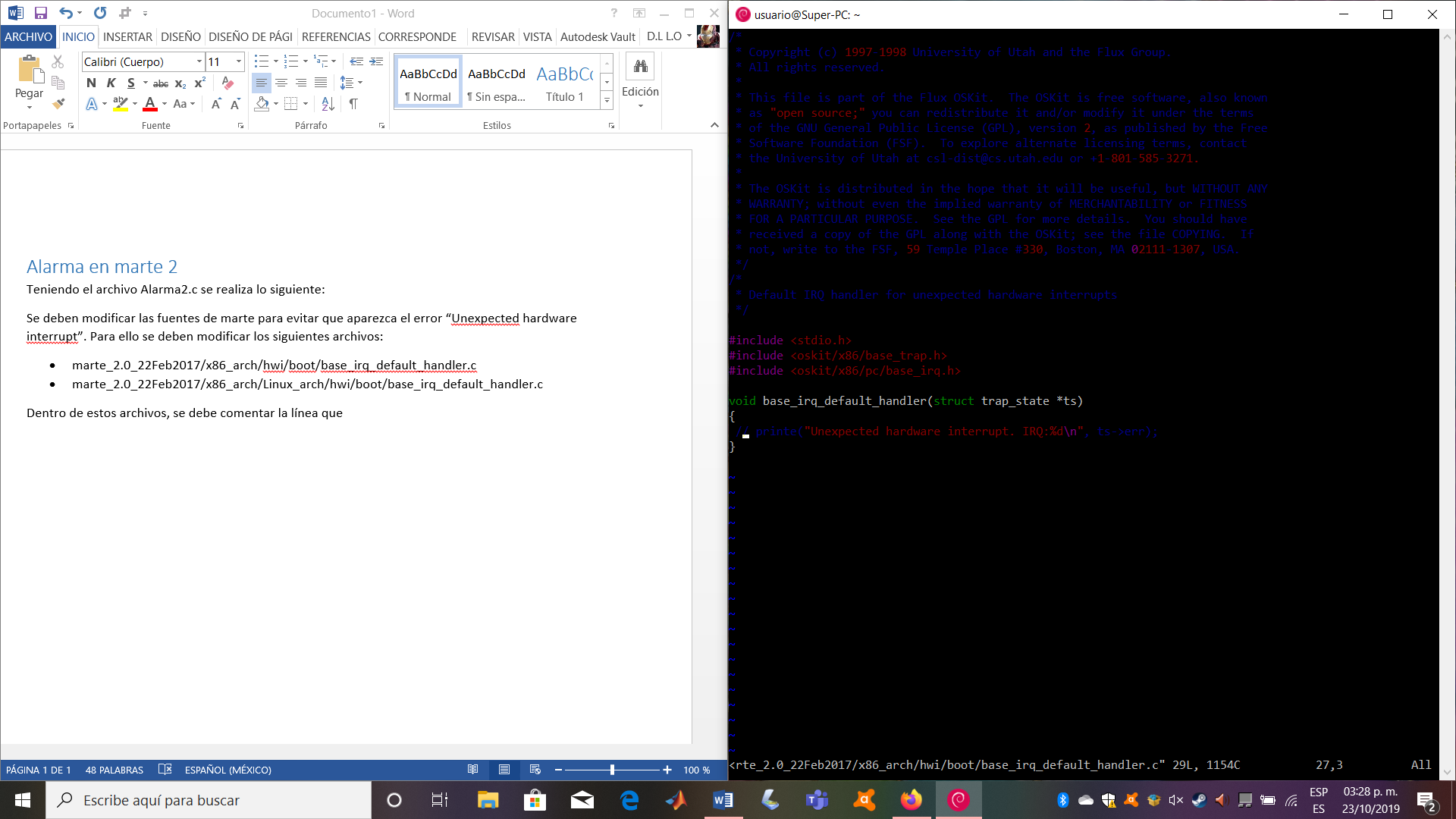


Figura Línea comentada

Esto debido a que si se deja sin realizar, el programa funcionará correctamente pero mostrará constantemente mensajes de error marcando “Unexpected hardware interrup” lo que opacaría el funcionamiento del programa. Una vez hecho esto, nos vamos a la carpeta de examples en marte: cd marte/exampes, donde se encuentra el programa alarma2.c.

También se modifica el los archivos marte-kernel-devices\_table.adb y marte-kernel-devices\_table.ads. Ambos se encuentran en el directorio marte\_2.0\_22Feb2017/kernel. En el archivo .ads se descomentan las siguientes líneas de código:

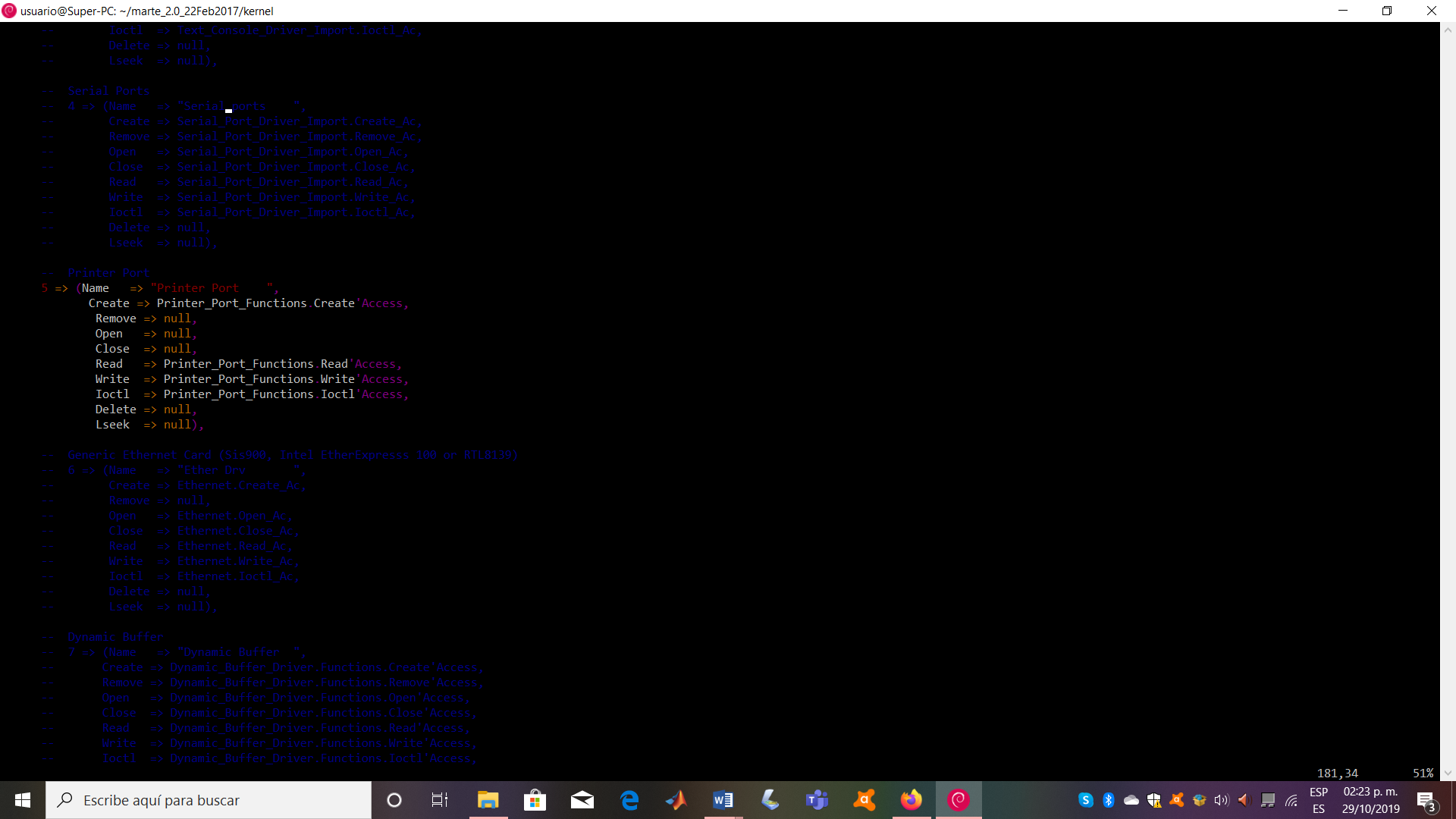


Ilustración Se descomento la lista 5

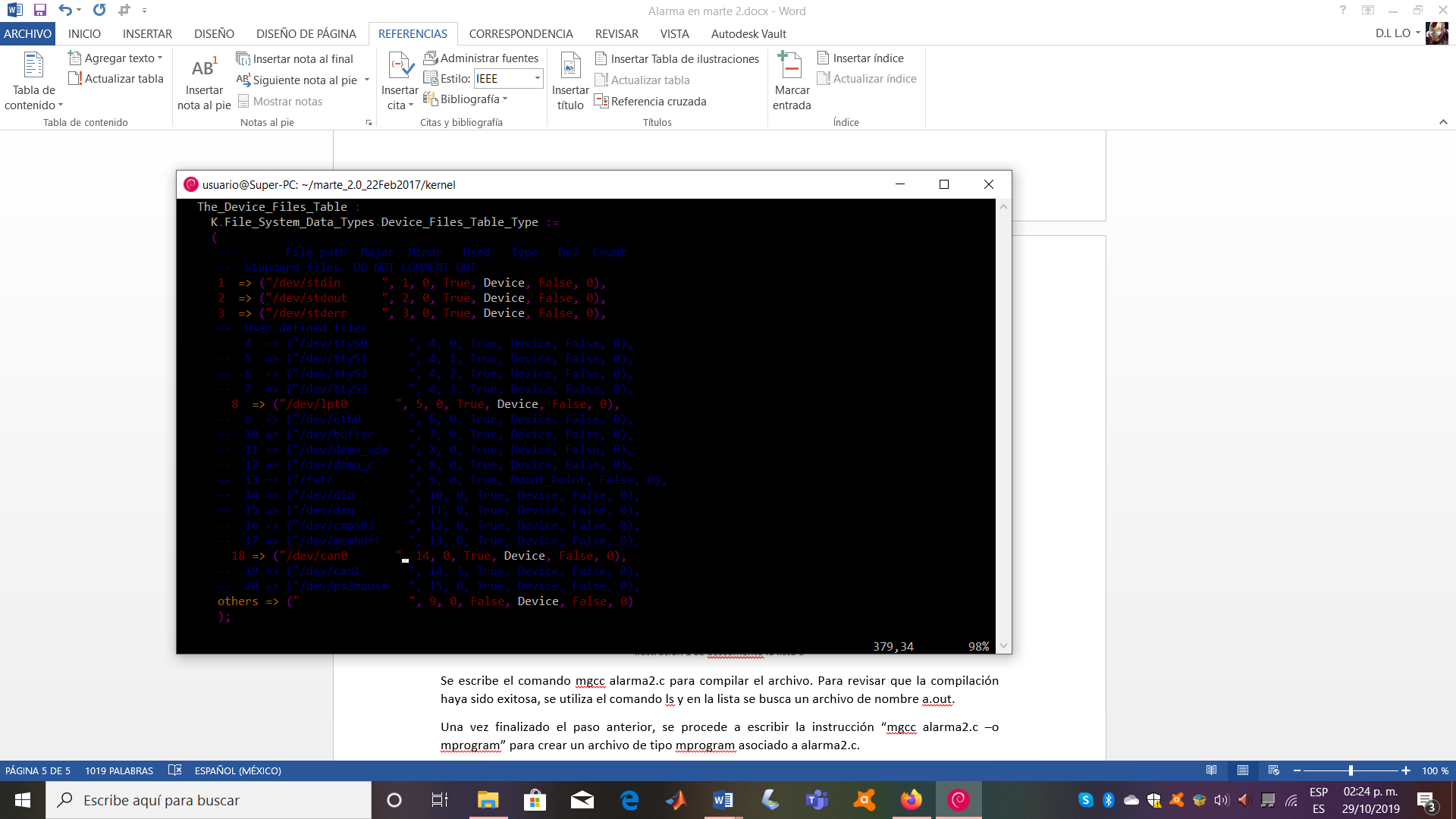


Ilustración Hasta el b fina del archivo se descomentan los puntos 8 y 18

En el archivo .adb se modifica lo siguiente:

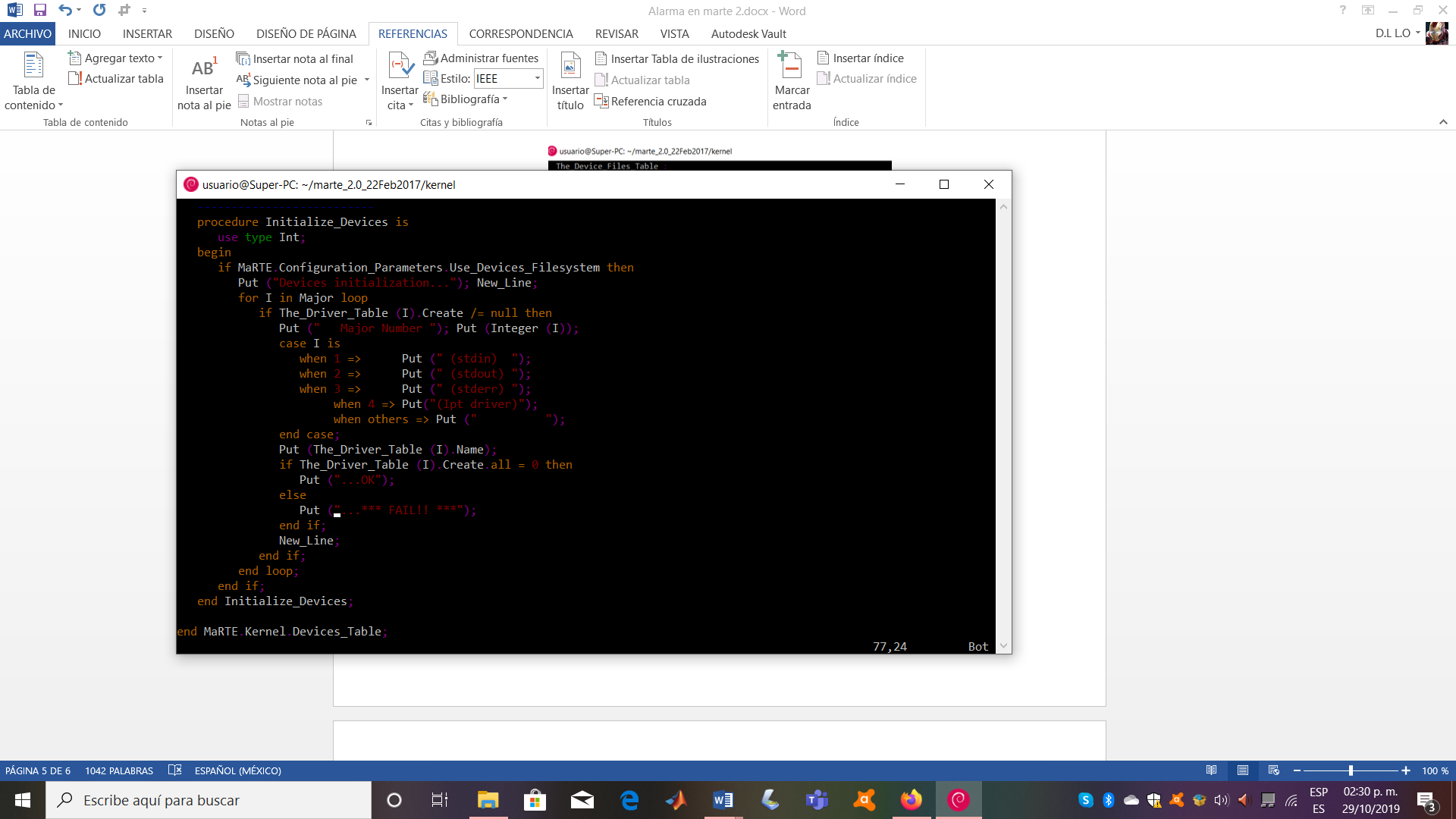


Ilustración Se agregó la opcion when4

Se escribe el comando mgcc alarma2.c para compilar el archivo. Para revisar que la compilación haya sido exitosa, se utiliza el comando ls y en la lista se busca un archivo de nombre a.out.

Una vez finalizado el paso anterior, se procede a escribir la instrucción “mgcc alarma2.c –o mprogram” para crear un archivo de tipo mprogram asociado a alarma2.c.

Ahora, se escribe la instrucción make alarma2.exe para construir un ejecutable. Este se debe mover a la carpeta GRUB para poder crear el os.iso. Para ello, nos colocamos dentro de GRUB y escribimos el siguiente comando: cp –v ../ marte\_2.0\_22Feb2017/examples/alarma2.exe kernel.elf. Esto copiará el archivo alarma2.c a la carpeta GRUB y lo renombrará como kermel.elf.

Una vez realizado esto, se procede a escribir la instrucción make dentro de GRUB, lo que creará el archivo os.iso.