Reporte de construcción de un programa ejecutable en MarteOS.

Raúl Morales Cendón

1- Instalación de GNAT y MarteOS.

Una vez descargados los archivos y colocados en una carpeta llamada “myapps”, se entra a esa carpeta desde la consola de Debian.

```
raul_3698@LAPTOP-V5I0ODL8:~$ cd /mnt/c/Users/RAUL3/Desktop/myapps/
raul_3698@LAPTOP-V5I0ODL8:/mnt/c/Users/RAUL3/Desktop/myapps$
```

Posteriormente se descomprimen estas carpetas utilizando las instrucciones: ***“tar xvf gnat-gpl-2016-x86_64-linux-bin.tar.gz”*** y ***“tar xvf marte_2.0_22Feb2017_src.tar.gz”.***

Una vez descomprimidas las carpetas, se tiene que editar un archivo para poder instalar el compilador y el sistema, por lo cual se ingresa en la terminal el comando ***cd*** para regresar a la raíz, y desde ahí se introduce la instrucción ***nano. bashrc.***

Al final del archivo se agregan las líneas mostradas en la figura siguiente.

```
export PATH=/mnt/c/Users/RAUL3/Desktop/myapps/gnat/bin:$PATH
export PERL5LIB=/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017
export PATH=$PATH:/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/utils
#export PATH=/opt/cross-pi-gcc/bin:$PATH
```

Se guardan los cambios (CTRL+O) y se sale del archivo (CRTL+X). Para que se guarden estos cambios es importante cerrar la terminal y luego volverla abrir.

Cuando se vuelve a abrir la consola, con el comando ***cd*** se accede a la carpeta ***gnat-gpl-2016-x86_64-linux-bin/***, la cual se encuentra en la carpeta que se creó al inicio llamada ***myapps***.

Una ves en esta carpeta se ejecuta el comando ***./doinstall,*** el cual comienza con la instalación de gnat. Se siguen las instrucciones en pantalla hasta que la instalación haya finalizado.

Después, se accede a la carpeta ***marte_2.0_22Feb2017*** para instalar MarteOS. Se comenzara con la instalación ejecutando la instrucción ***./minstall.***

Una vez finalizada la instalación, se accede a la carpeta ***utils/*** y se ejecuta el comando ***msetcurrentarch x86 i386***.

Posteriormente se ejecutan los comandos ***mkrtsmarteuc*** y ***mkmarte,*** en ese orden, para finalizar la instalación.

2- Construcción de ejecutable.

Para poder ejecutar el archivo desde una BIOS sin recibir un mensaje de error en la pantalla edite un archivo para ocultar este mensaje de interrupción. El archivo es ***"base_irq_default.handler.c"***, el cual se encuentra en la carpeta ***"marte_2.0_22Feb2017/linux_arch/hwi/boot***/". En este archivo se comentó el comando ***printe*** para ocultar el mensaje de error.

```
30  void base_irq_default_handler(struct trap_state *ts)
31  {
32    //printe ("Unexpected hardware interrupt. IRQ:%d\n", ts->err);
33  }
34
```

Para guardar este cambio, se ejecuta nuevamente el comando ***mkmarte***.

Posteriormente se capturo el programa de alarma. Crear el archivo se introduce en la consola la instrucción ***vim tarea_alarma.c.***

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

int main(){
        int x;
        float tiempo;
        bool y = true;
        time_t t0;
        time_t t1;

        printf("Tarea 2: Alarma \n");

        printf("Introduzca la duracion de la alarma en minutos:");
        scanf("%i",&x);

        printf("Recibira una mensaje en %i minutos\n",x);
        t0 = time(NULL);

        while(y){

                t1 =time(NULL);
                tiempo=t1-t0;

                if (tiempo >= x*60){
                        y=false;
                        }

        }

        printf("Ha transcurrido %f minutos \n",tiempo/60);
        getchar();
}
```

En el programa se incluyeron las librerías “stdio.h”, “stdbool.h” y “time.h”. En el main, declaré cinco variables.

La variable “x” se utiliza para guardar el dato que el usuario ingresa especificando la duración de la alarma en minutos. Posteriormente imprimo en la consola un mensaje confirmando la duración de la alarma.

Después un ciclo while inicia y no termina hasta que se la condición de tiempo se cumple. La función time(NULL) devuelve el numero de segundos desde el epoch, se utiliza para contar los segundos desde que se recibe el tiempo deseado por el usuario, hasta que se llega a ese tiempo. Por lo que utiliza el tiempo de inicio, menos otra lectura por tiempo dentro del ciclo while, cuando la resta de estas dos variables de segundos desde el epoch, alcanza el tiempo que indica el usuario, el ciclo while termina y lo anuncia en una consola.

Posteriormente se sale del editor vim con presionando ***ESC*** e ingresando las instrucciones ***:w*** y ***:x***.

En la consola se escribe ***mgcc tarea_alarma.c***, después ***mgcc tarea_alarma.c -o program***

```
raul_3698@LAPTOP-V5I0ODL8:/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/examples$ mgcc tarea_alarma.c
Use of uninitialized value in concatenation (.) or string at /mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/utils
/globals.pl line 16.
 gcc -nostdinc -I/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/arch/include tarea_alarma.c    -m32 -march=i686
  /mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main_c.o   -Wl,-T,/mnt/c/Users/RAUL3/Desk
top/myapps/marte_2.0_22Feb2017/utils/linker.lds -static -nostartfiles -L/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Fe
b2017/lib -L/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/gnat_rts/rts/adalib -L/mnt/c/Users/RAUL3/Desktop/myapp
s/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnarl -lgnat -lmarte -lgcc_sjlj
raul_3698@LAPTOP-V5I0ODL8:/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/examples$ mgcc tarea_alarma.c -o program
Use of uninitialized value in concatenation (.) or string at /mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/utils
/globals.pl line 16.
 gcc -nostdinc -I/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/arch/include tarea_alarma.c    -m32 -march=i686
-o program   /mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/arch/call_main/wrapper_main_c.o   -Wl,-T,/mnt/c/Users
/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/utils/linker.lds -static -nostartfiles -L/mnt/c/Users/RAUL3/Desktop/myapps/mar
te_2.0_22Feb2017/lib -L/mnt/c/Users/RAUL3/Desktop/myapps/marte_2.0_22Feb2017/gnat_rts/rts/adalib -L/mnt/c/Users/RAUL3/De
sktop/myapps/gnat/lib/gcc/x86_64-pc-linux-gnu/4.9.4 -lmarte -lgnarl -lgnat -lmarte -lgcc_sjlj
```

Después se crea el archivo ejecutable con ***make tarea_alarma.exe.***

Con esto se crea el ejecutable y posteriormente creé el archivo iso que se carga a una live usb. Para esto se copia el archivo ***“tarea_alarma.exe”***, en la carpeta ***MAKEFILE_GRUB_Legacy_Kernel_elf***.

Se accede a la carpeta por medio de Debian, y ejecuté el comando ***make***, y en esa carpeta se crea el archivo ***os.iso***, el cual se carga a la memoria USB con el programa “Lili USB Linux”.

Posteriormente el archivo ***“os.iso”*** se ejecuta desde la BIOS de una computadora.