



Instituto Politécnico Nacional



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

Sistemas Operativos en tiempo real

Guía de compilación de la aplicación Alarma

Profesor: Maza Casas Lamberto

- Rodriguez Ramirez Marco Antonio

Grupo: 3MV11

15/Noviembre/2019

Guía de compilación de Alarma

Esta guía tiene el propósito de explicar cómo se programó y compiló un SO de una alarma, la cual consiste en dar la hora actual del sistema y se le ingresan valores de horas y minutos, al llegar a la hora indicada se envían mensajes a pantalla indicando que ha llegado a la hora programada.

Se tomó como base la guía proporcionada por el profesor, la cual fue elaborada por los alumnos: Baldemar Martínez Morales y Jonathan Alexis Salas Soto, y tiene como objetivo la instalación de marte OS y de gnat.

URL:

https://github.com/sotrteacher/sotr_201808_201812/blob/master/PRACTICA_1_Instalacion_de_un_SOTR/guia%20marte%20os.pdf

Pre instalación de marte OS:

Para poder realizar la alarma, es importante mencionar que se realizó usando una distribución de LINUX, más específicamente en UBUNTU 16.04, esto debido a que las distribuciones de GNAT que se usaron para compilar tienen un error al compilar los archivos en otras distribuciones de UBUNTU y de DEBIAN.

Instalación de marte OS y gnat:

1. Lo primero que tenemos que hacer es crear una carpeta dentro de “Carpeta personal”, en este caso la llamaremos **carpetaejemplo**.
2. Una vez creada esta carpeta debemos guardar en ella los archivos “gnat-gpl-2016-x86_64-linuxbin.tar.gz” y “marte_2.0_22Feb2017_src.tar.gz”, a su vez, dentro de la carpeta **carpetaejemplo** crearemos una carpeta de nombre **gnat**.
3. El siguiente paso es descomprimir los archivos que contienen el compilador GNAT así como marte OS, comenzamos con el compilador GNAT, dentro de la carpeta que creamos **carpetaejemplo** damos clic derecho y abrimos un nuevo terminal, acto seguido ingresamos la siguiente instrucción *tar xvf gnat-gpl-2016-x86_64-linuxbin.tar.gz*, una vez terminado este proceso, procederemos a descomprimir el archivo contenedor de marte OS y eso lo haremos con la siguiente instrucción *tar xvf marte_2.0_22Feb2017_src.tar.gz*.
4. Una vez que hemos descomprimido los archivos contenedores del compilador y marte OS, lo siguiente es modificar el archivo **bashrc** para poder instalar marte OS 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*.
5. Dentro de este archivo, iremos al final y agregaremos las siguientes líneas:
 - a. *export PATH=\$HOME/carpetaejemplo/gnat/bin:\$PATH*
 - b. *export PERL5LIB=\$HOME/carpetaejemplo/marte_2.0_22Feb2017*
 - c. *export PATH=\$PATH:\$HOME/carpetaejemplo/marte_2.0_22Feb2017/utils*
 - d. *#export PATH=/opt/cross-pi-gcc/bin:\$PATH*

6. 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. Luego en la carpeta *carpetaejemplo* nuevamente abrimos un nuevo terminal. Donde entraremos a la carpeta contenedora del compilador 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*.
7. Posteriormente, se nos preguntara la dirección de la carpeta donde queremos instalar el compilador. Es aquí donde entra en juego la carpeta *gnat* que creamos en un inicio dentro de la carpeta *carpetaejemplo*. Para poder obtener la dirección de esta carpeta sólo es necesario seleccionar la carpeta *gnat* y dar click derecho sobre ella, seleccionar la opción “*propiedades*” y tendremos la vista de una ventana que nos dirá donde se encuentra nuestra carpeta con una etiqueta de nombre “*Lugar*”.
8. Para este caso en particular, la carpeta de interés, tiene lugar en la dirección */home/marco/carpetaejemplo*, para poder ingresar la dirección correctamente, agregaremos el directorio *gnat*, entonces la cadena que ingresaremos, quedara de la siguiente manera */home/bmm930519/myapps/gnat*, con esto estaremos indicando exactamente dónde queremos que se instale el compilador.
9. 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*.
10. El proceso anterior puede demorar varios minutos, por lo que es importante no cerrar la ventana hasta que podamos visualizar la señal de que el proceso ha culminado.
11. Lo siguiente es salir a la carpeta *carpetaejemplo* y acceder a la carpeta *marte_2.0_22Feb2017*. Una vez ahí procederemos a instalar marte OS, para lo cual ejecutaremos la instrucción *./minstall*.
12. Lo siguiente es acceder a la carpeta *utils*, una vez estando dentro de la carpeta, procederemos a definir la arquitectura sobre la que trabajara marte OS, para ello ingresaremos la siguiente instrucción a la terminal *msetcurrentarch*, y podremos ver que tenemos varias arquitecturas para elegir. Entre ellas están:
 - a. *X86*
 - b. *Linux*
 - c. *Linux_lib*
 - d. *Rpi*
13. Para esta instalación, elegimos la arquitectura x86, por lo que ingresaremos la siguiente instrucción en la terminal *msetcurrentarch x86 i386*.
14. 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 termine podemos continuar a ejecutar la instrucción *mkmarTE*.

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

Programación de la aplicación de Alarma:

La aplicación de la alarma se programo en C++, y se compilo usando marte OS, con el comando *mgcc*.

El código de la alarma es el siguiente:

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

int main ()
{
    time_t tiempo;
    struct tm *hora;
    int h=0, m=0, s=0, i, dur=0;
    bool alarm=0, fin=0;
    time(&tiempo);
    hora = localtime(&tiempo);

    for(;;){
        printf("Esta es la hora actual: %s", asctime(hora));
        printf("\n\n A que hora quieres que suene la alarma ?: \n\n
        Hora: ");
        scanf("%d", &h);
        printf("\n Minuto: ");
        scanf("%d", &m);
        printf("\n Duracion de la alarma: ");
        scanf("%d", &dur);

        i=time(NULL);

        while(fin == 0)
        {
            if(time(NULL)!=i && alarm!=1)
            {
                printf("%s", asctime(hora));
                i=time(NULL);
            }

            time(&tiempo);
            hora = localtime( &tiempo);
```

```

        if((*hora).tm_hour==h && (*hora).tm_min==m && (*hora).tm_sec
==s )
        {
            printf("%s \a \n ALARMA !!!", asctime(hora));
            s++;
            alarm=1;
        }
        if(s>=dur){
            alarm=0;
            s=0;
            fin = 1;
        }
    }
}
}

```

Una vez terminado de programarse se procedió a mover el archivo con extensión .cpp se compilo usando el comando `gcc Alarma.cpp -o Alarma`, una vez teniendo el archivo ya compilado, se procede a moverlo a la carpeta proporcionada por el profesor, en donde se tiene un ejemplo de creación de archivos iso.

URL:

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

Una vez movido el archivo a esta carpeta se teclea el siguiente 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, después se teclea el comando `make` el cual genera el archivo OS.iso, este es el iso que se usara para crear una live usb y poder correr la aplicación como un Sistema operativo en tiempo real.

Para hacer una live usb, se usó el software LinuxLive USB Creator, el cual se puede descargar desde su página oficial.

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

Los pasos a seguir son los siguientes:

- Escoger el dispositivo USB
- Seleccionar el archivo ISO
- Dar clic en el icono de rayo para que comience la creación

Finalizado este proceso, se tiene la memoria USB lista para usarla en cualquier PC.

Para ejecutar la aplicación se conecta la USB con la computadora apagada, se enciende la computadora y se presiona F2 para cambiar el menú de arranque del sistema y se selecciona la opción en donde este nuestra USB, después de esto veremos funcionar nuestra Alarma.

Capturas de la aplicación de la alarma funcionando en una computadora

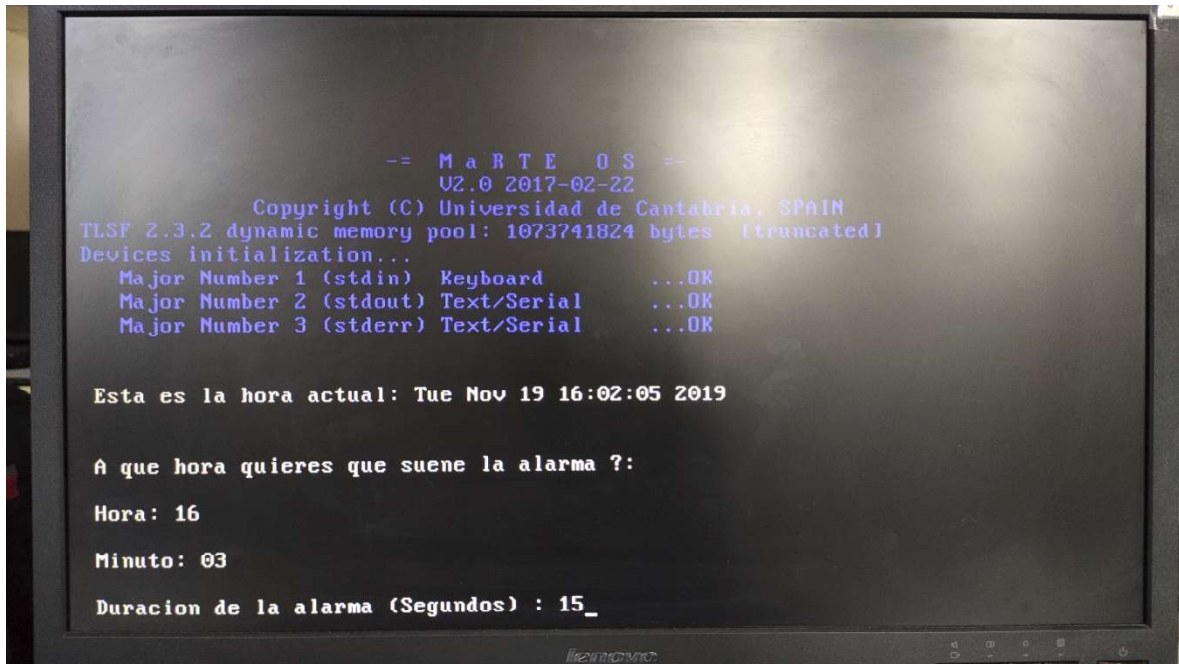


Ilustración 1: Inicio de la aplicación

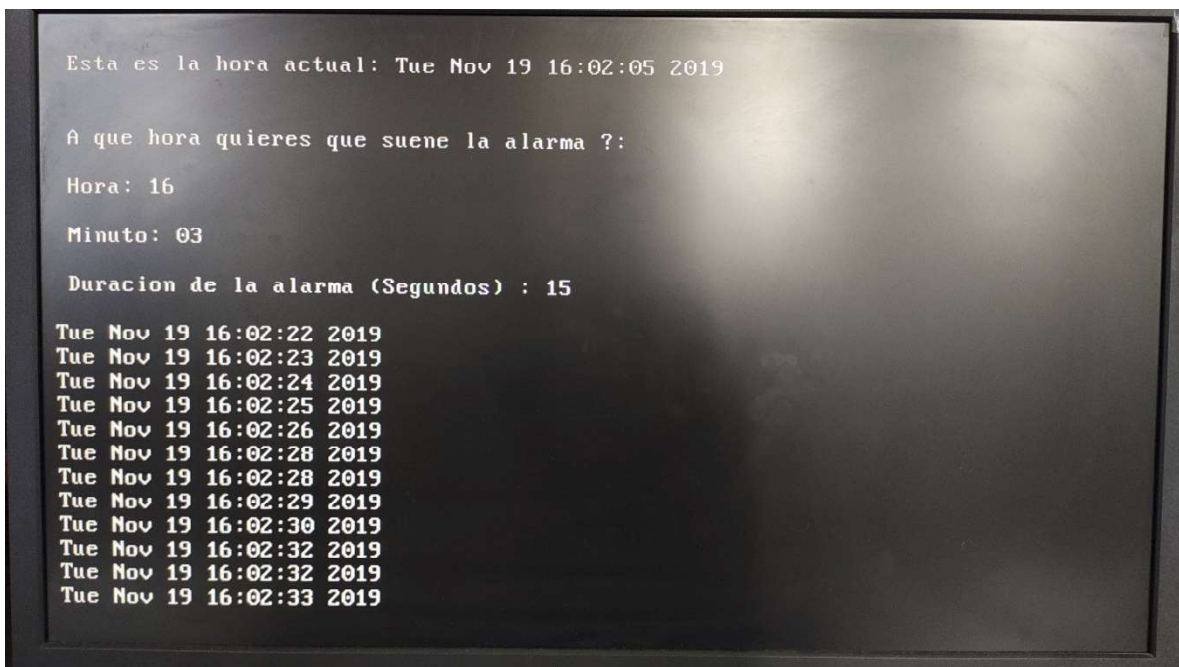


Ilustración 2: Muestreo de la hora del sistema, manda un mensaje a pantalla en cada actualización del tiempo

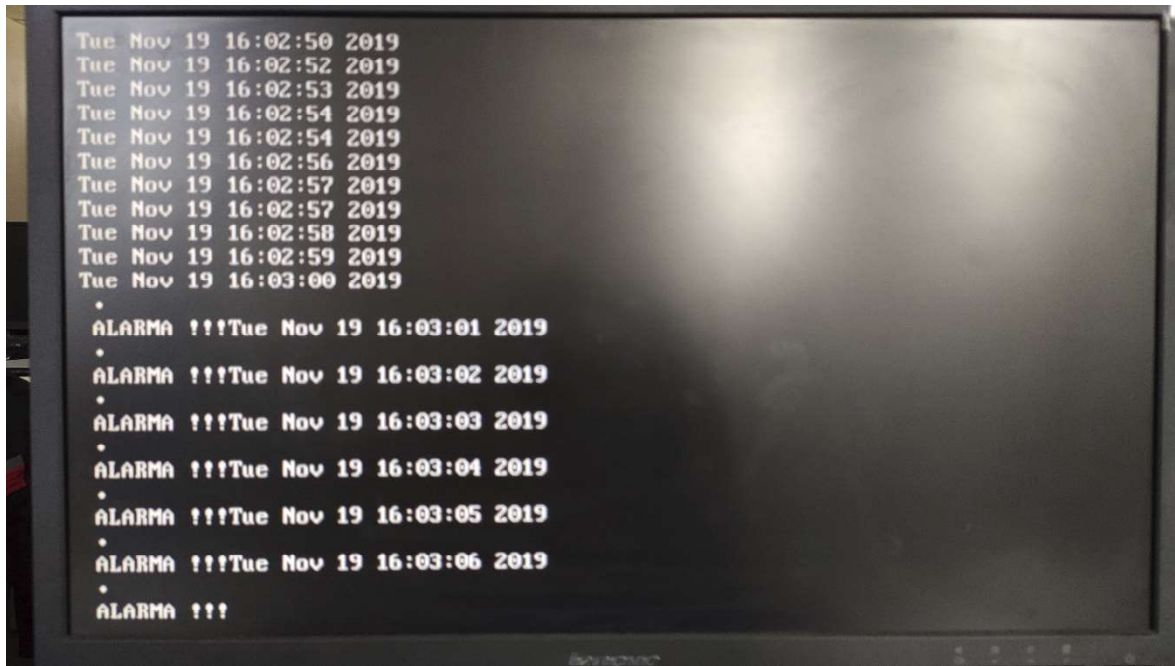


Ilustración 3: Al llegar a la hora ingresada se muestra el mensaje de alarma

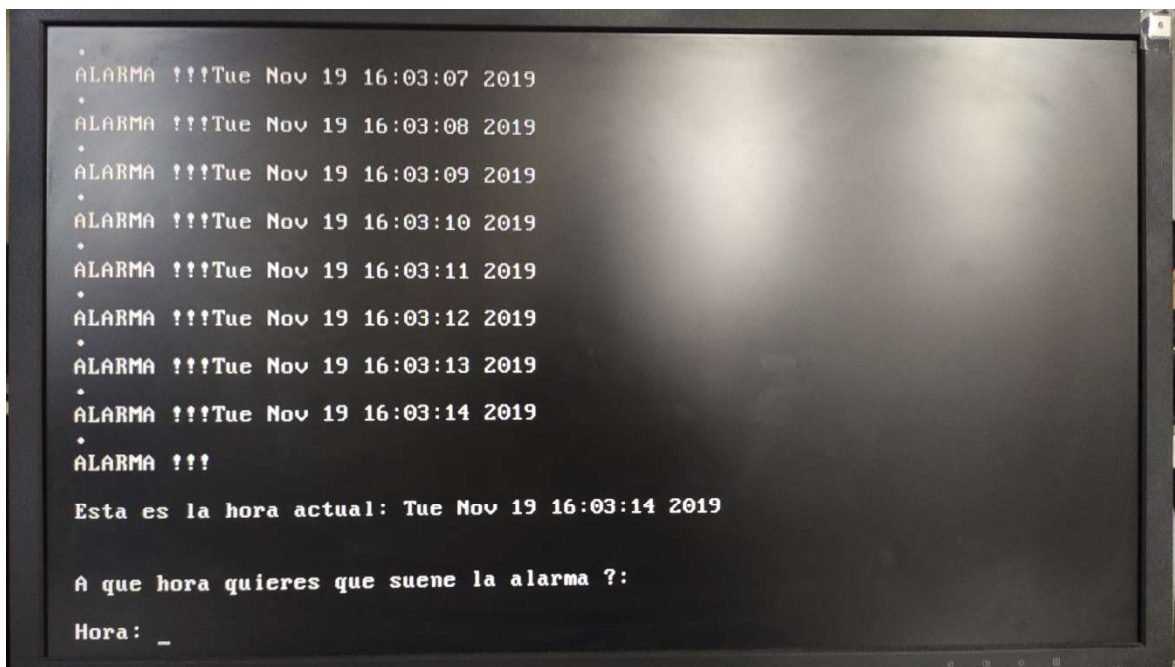


Ilustración 4: Al finalizar el tiempo que se muestra el mensaje de alarma, se vuelve a repetir el ciclo de funcionamiento