



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria
en Ingenierías y Tecnologías
Avanzadas

MarTEOS
“Temporizador”

BARRERA ANGELES DIEGO IVÁN

Sistemas Operativos en Tiempo Real

3MV11



Profesor:
Lamberto Maza Casas

Ciudad de México, México

Diciembre 2019

Desarrollo

El propósito de esta práctica es hacer un temporizador en lenguaje C que corra sobre el sistema operativo en tiempo real MarTEOS. Para ello se utilizó la librería time.h.

En rasgos generales lo que pide el programa es la cantidad de horas, minutos y segundos. Posteriormente convierte los datos a segundos para una facilidad de cálculo. Obtiene el tiempo actual por medio de las funciones de la librería time.h para después sumarle a ese tiempo, los segundos a transcurrir. Una vez que el tiempo pasado sea igual al tiempo actual se imprime en pantalla el mensaje que los segundos han transcurrido.

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

int main(){
    int hr,mn,sc,conv;
    struct timespec acTime;
    struct timespec psTime;
    printf(" TEMPORIZADOR. Este programa manda una alerta en pantalla cuando haya transcurrido el tiempo que el usuario haya especificado.\n");
    printf("\nEscriba las horas: ");
    scanf("%d",&hr);
    printf("\nEscriba los minutos: ");
    scanf("%d",&mn);
    printf("\nEscriba los segundos: ");
    scanf("%d",&sc);

    //Algoritmo
    conv=(hr*3600)+(60*mn)+sc;
    printf("\ntiempo a transcurrir en segundos: %d",conv);
    clock_gettime(CLOCK_REALTIME,&acTime);
    psTime.tv_sc=acTime.tv_sc+conv;
    psTime.tv_nsec=0;
    while(acTime.tv_sc<psTime.tv_sc)
    {
        clock_gettime(CLOCK_REALTIME,&acTime);
    }
    printf("\n\nTIEMPO AGOTADO");
    return 0;
}
```

Figura 1. Programa en C.

Para correr sobre hardware primero tenemos que compilarlos ejecutando el comando *makefile alarma_tarea.exe*, o colocando el nombre del programa con la terminación .exe.

```
diego@DiegoBA:~/myapps/marte_2.0_22Feb2017/examples/time_measurement$ make alarma_tarea.exe
make -C ../../misc time_measurement_posix.o
make[1]: Entering directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'
make[1]: 'time_measurement_posix.o' is up to date.
make[1]: Leaving directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'
make -C ../../misc time_measurement_hwttime.o
make[1]: Entering directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'
make[1]: 'time_measurement_hwttime.o' is up to date.
make[1]: Leaving directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'
make -C ../../misc logger.o
make[1]: Entering directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'
make[1]: 'logger.o' is up to date.
make[1]: Leaving directory '/home/diego/myapps/marte_2.0_22Feb2017/misc'

>> Compiling alarma_tarea.exe: Use of uninitialized value in concatenation (.) or string at /home/di
ego/myapps/marte_2.0_22Feb2017/examples/time_measurement/../../utils/globals.pl line 16.
[OK]
```

Figura 2. Compilación.

Como en la práctica de Xv6 copiamos el archivo .exe en la carpeta MAKEFILE_GRUB_Legacy_Kernel_elf sustituyendo el kernel.elf, ejecutamos make y nos generará el archivo os.iso que quemaremos con el software LiLi.