

Práctica 01

Tópicos selectos de Sistemas Digitales

Aldo Alexandro Vargas Meza
18/09/2017



El objetivo de la práctica es desarrollar una interfaz que reciba de un usuario 5 mensajes, los cuales tienen una serie de valores aceptables y unos cuantos valores considerados como validos. Por lo tanto, el desarrollo del programa se divide en etapas.

- Recepción de mensaje
- Validación por mensaje
- Validación de salida
- Impresión del resultado

El mensaje en el programa es una estructura, la cual guarda variables para las señales y también para el numero de mensaje que será configurado.

```
typedef struct mensaje
{
    int num;
    int s1;
    int s2;
} Mensaje;
```

De esta manera, dentro de un ciclo podemos iterar el mensaje que se está llenando, para que el usuario ingrese los valores de los 5 mensajes.

La recepción de mensajes se hace mediante una función que toma los datos del usuario por medio del teclado, después de cada ingreso de ambas señales del mensaje, se lleva a cabo una validación particular, para los rangos validos del mensaje.

```
switch(m[i].num)
{
case 1:
    if(!(m[i].s1 == 1 || m[i].s1 == 0))
    {
        while(!(m[i].s1 == 1 || m[i].s1 == 0))
        {
            printf("Error de Ingreso\n");
            printf("Ingresa la señal 1.\n");
            scanf("%d",&m[i].s1);
        }
        printf("OK");
    }
    else if(!(m[i].s2 >= 0 || m[i].s2 <= 7))
    {
        while(!(m[i].s2 >= 0 || m[i].s2 <= 7))
        {
            printf("Error de Ingreso\n");
            printf("Ingresa la señal 2.\n");
            scanf("%d",&m[i].s2);
        }
        printf("OK");
    }
}
```

Cuando hay un error de ingreso, el programa itera las veces que sea necesaria la instrucción de escribir un valor válido para la señal.

Una vez obtenidos los 5 mensajes válidos, podemos calcular si la salida del sistema será TRUE o FALSE, dependiendo de los valores ya obtenidos.

```
if(m[0].s1 == 1 && m[0].s2 == 0)
{
    if(m[1].s1 == 1 && (m[0].s2>=0 && m[0].s2<=2) )
    {
        if(m[2].s1 == 1 && m[2].s2 == 0)
        {
            if(m[3].s1 == 1 && (m[3].s2>=0 && m[3].s2<=4) )
            {
                if(m[4].s1 == 1 && (m[4].s2>=15 && m[4].s2<=255) )
                {
                    printMsn();
                    printf("\nResultado = TRUE\n");
                    return 0;
                }
            }
        }
    }
}

printMsn();
printf("\nResultado = FALSE\n");
return 0;
```

De esta forma calculamos lógicamente cada variable de cada mensaje.

De manera altera, también hay una función programada para imprimir los valores de cada mensaje en pantalla. Obteniendo como resultado este registro en consola.

```
Mensaje numero [1]
Signal 01: [1]
Signal 02: [1]
-----
Mensaje numero [2]
Signal 01: [1]
Signal 02: [2]
-----
Mensaje numero [3]
Signal 01: [0]
Signal 02: [1]
-----
Mensaje numero [4]
Signal 01: [1]
Signal 02: [2]
-----
Mensaje numero [5]
Signal 01: [0]
Signal 02: [1]
-----
Presione una tecla para continuar . . .

Resultado = FALSE

Process returned 0 (0x0)   execution time : 55.402 s
Press any key to continue.
```

Anexo

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

```
#include <stdlib.h>
```

```
typedef struct mensaje
```

```
{
```

```
    int num;
```

```
    int s1;
```

```
    int s2;
```

```
} Mensaje;
```

```
void printMsn();
```

```
Mensaje m[5];
```

```
int main()
```

```
{
```

```
    printf("Recepcion de mensajes.\n");
```

```
    for(int i = 0; i<5; i++)
```

```
    {
```

```
        m[i].num = i+1;
```

```
        printf("Mensaje numero [%d]\n", m[i].num);
```

```
printf("Ingresa la señal 1.\n");
```

```
scanf("%d",&m[i].s1);
```

```
printf("Ingresa la señal 2.\n");
```

```
scanf("%d",&m[i].s2);
```

```
switch(m[i].num)
```

```
{
```

```
case 1:
```

```
    if(!(m[i].s1 == 1 || m[i].s1 == 0))
```

```
    {
```

```
        while(!(m[i].s1 == 1 || m[i].s1 == 0))
```

```
        {
```

```
            printf("Error de Ingreso\n");
```

```
            printf("Ingresa la señal 1.\n");
```

```
            scanf("%d",&m[i].s1);
```

```
        }
```

```
        printf("OK");
```

```
    }
```

```
    else if(!(m[i].s2 >= 0 || m[i].s2 <= 7))
```

```
    {
```

```
        while(!(m[i].s2 >= 0 || m[i].s2 <= 7))
```

```

        {
            printf("Error de Ingreso\n");
            printf("Ingresa la señal 2.\n");
            scanf("%d",&m[i].s2);
        }

        printf("OK");

    }

    system("pause");
    system("cls");

    break;

case 2:
    if(!(m[i].s1 == 1 || m[i].s1 == 0))
    {
        while(!(m[i].s1 == 1 || m[i].s1 == 0))
        {
            printf("Error de Ingreso\n");
            printf("Ingresa la señal 1.\n");
            scanf("%d",&m[i].s1);
        }
        printf("OK");
    }

```

```
}  
else if(!(m[i].s2 >= 0 || m[i].s2 <= 4))  
{  
    while(!(m[i].s2 >= 0 || m[i].s2 <= 4))  
    {  
        printf("Error de Ingreso\n");  
        printf("Ingresa la señal 2.\n");  
        scanf("%d",&m[i].s2);  
    }  
    printf("OK");  
}
```

```
system("pause");
```

```
system("cls");
```

```
break;
```

```
case 3:
```

```
if(!(m[i].s1 == 1 || m[i].s1 == 0))  
{  
    while(!(m[i].s1 == 1 || m[i].s1 == 0))  
    {  
        printf("Error de Ingreso\n");  
        printf("Ingresa la señal 1.\n");  
        scanf("%d",&m[i].s1);
```



```

    }

    printf("OK");

}

else if(!(m[i].s2 >= 0 || m[i].s2 <= 7))
{
    while(!(m[i].s2 >= 0 || m[i].s2 <= 7))
    {
        printf("Error de Ingreso\n");
        printf("Ingresa la señal 2.\n");
        scanf("%d",&m[i].s2);
    }
    printf("OK");
}

```

```

system("pause");

```

```

system("cls");

```

```

break;

```

```

case 4:

```

```

    if(!(m[i].s1 == 1 || m[i].s1 == 0))
    {
        while(!(m[i].s1 == 1 || m[i].s1 == 0))
        {

```

```
        printf("Error de Ingreso\n");
        printf("Ingresa la señal 1.\n");
        scanf("%d",&m[i].s1);
    }
    printf("OK");

}

else if(!(m[i].s2 >= 0 || m[i].s2 <= 65535))
{
    while(!(m[i].s2 >= 0 || m[i].s2 <= 65535))
    {
        printf("Error de Ingreso\n");
        printf("Ingresa la señal 2.\n");
        scanf("%d",&m[i].s2);
    }
    printf("OK");
}
```

```
system("pause");
```

```
system("cls");
```

```
break;
```

```
case 5:
```

```
    if(!(m[i].s1 == 1 || m[i].s1 == 0))
```

```

{
    while(!(m[i].s1 == 1 || m[i].s1 == 0))
    {
        printf("Error de Ingreso\n");
        printf("Ingresa la señal 1.\n");
        scanf("%d",&m[i].s1);
    }
    printf("OK");

}
else if(!(m[i].s2 >= 0 || m[i].s2 <= 4095))
{
    while(!(m[i].s2 >= 0 || m[i].s2 <= 4095))
    {
        printf("Error de Ingreso\n");
        printf("Ingresa la señal 2.\n");
        scanf("%d",&m[i].s2);
    }
    printf("OK");
}

system("pause");
system("cls");

break;

```

```
    }  
}
```

```
if(m[0].s1 == 1 && m[0].s2 == 0)  
{  
    if(m[1].s1 == 1 && (m[0].s2>=0 && m[0].s2<=2) )  
    {  
        if(m[2].s1 == 1 && m[2].s2 == 0)  
        {  
            if(m[3].s1 == 1 && (m[3].s2>=0 && m[3].s2<=4) )  
            {  
                if(m[4].s1 == 1 && (m[4].s2>=15 &&  
m[4].s2<=255) )  
                {  
                    printMsn();  
                    printf("\nResultado = TRUE\n");  
                    return 0;  
                }  
            }  
        }  
    }  
}
```

```
    printMsn();  
    printf("\nResultado = FALSE\n");  
    return 0;  
}  
  
void printMsn()  
{  
    for(int i = 0; i<5; i++)  
    {  
        printf("Mensaje numero [%d]\n", m[i].num);  
        printf("Signal 01: [%d]\n", m[i].s1);  
        printf("Signal 02: [%d]\n", m[i].s2);  
        printf("-----\n");  
    }  
    system("pause");  
}
```