



Práctica 2.-Control de error

Objetivo: El alumno(s) implementará en lenguaje C una función que reciba una trama (arreglo de caracteres sin signo) y que imprima el checksum correspondiente.

Utilizado para detectar errores en la transmisión de datos a nivel de software.
Usado por los protocolos de TCP/IP.

Vamos a trabajar con la estructura de la cabecera IP; por ahora solo basta recordar en que posición del arreglo se encuentra el Checksum. Si consideramos unicamente el paquete IP, entonces el checksum se encuentra en T[10] y T[11]



Prueba 1. En esta trama los valores del Checksum estan en 0000 ya que apenas se va a calcular (ALICIA)

```
Unsigned char T[]={0x45, 0x00, 0x01, 0xe2, 0xd7, 0xdb, 0x40, 0x00, 0x80,  
0x06, 0x00, 0x00, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd}
```

Prueba 2: Para esta trama ya se tiene el valor del checksum (es la que recibió BETITO) verificar si el checksum es correcto y en caso de no serlo suponer que la trama es correcta y mostrar el Checksum que “debía” haber sido.

45 00 01 9c d7 de 40 00 80 06 88 9d c0 a8 01 43 94 cc 3a dd

Instrucciones.

- Abrir un block de notas y escribir el código correspondiente en C.
- Las tramas se inicializarán en hexadecimal en un arreglo de caracteres sin signo
- Compilar en consola haciendo uso de gcc (en caso de no tenerlo, instalarlo).
- Una vez terminado deberán probar con los dos ejercicios vistos en clase e incluir aquí las capturas de pantalla

Se entrega este documento, incluir:

Todo tu Código,

```
#include <stdio.h>  
  
//Calulo de checksum  
unsigned short calcular_checksum(unsigned char *trama) {
```



Dra. Nidia A. Cortez Duarte



Práctica 2.-Control de error

```
unsigned int suma = 0; // Variable para almacenar la suma de los valores de los bytes de la trama
unsigned short checksum; // Variable para almacenar el checksum final

// Recorre la trama de 20 bytes en bloques de 2 bytes
for(int i = 0; i < 20; i += 2) {
    suma += (trama[i] << 8) | trama[i+1]; // Suma los valores de los dos bytes actuales (con desplazamiento)
}

// Si la suma es mayor a 16 bits, se ajusta para que quepa en 16 bits
while(suma >> 16) {
    suma = (suma & 0xFFFF) + (suma >> 16);
}

// Se calcula el complemento a uno de la suma
checksum = (unsigned short) ~suma;

return checksum;
}

int main() {
    // Prueba 1
    unsigned char T1[] = {0x45, 0x00, 0x01, 0xe2, 0xd7, 0xdb, 0x40, 0x00, 0x80, 0x06, 0x00, 0x00, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};
    unsigned short checksum_calculado = calcular_checksum(T1);

    printf("Prueba 1\n");
    printf("Trama: ");
    for(int i = 0; i < 20; i++) {
        printf("%02x ", T1[i]); // Imprime cada byte de la trama en hexadecimal
    }
    printf("\nChecksum: %04x\n\n", checksum_calculado); // Imprime el checksum calculado en hexadecimal

    // Prueba 2
    unsigned char T2[] = {0x45, 0x00, 0x01, 0x9c, 0xd7, 0xde, 0x40, 0x00, 0x80, 0x06, 0x88, 0x9d, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};
    checksum_calculado = calcular_checksum(T2);
    unsigned short checksum_trama = (T2[10] << 8) | T2[11]; // Combina los bytes 10 y 11 de la trama para obtener el checksum recibido
```




Práctica 2.-Control de error

Ejemplo de mapa de memoria

Unsigned short int CHECKSUM	1	0	0	0	1	1	1	1
	1	0	1	0	0	1	0	1
...	1	0	0	0	1	0	0	0
	1	0	0	1	1	1	0	1
	1	1	1	1	1	0	0	0
	1	0	1	1	0	1	0	0

Finalmente incluir tus **Conclusiones**

¿Habías escuchado sobre funciones de comprobación de trama?

¿Qué opinas sobre la integridad de la información que se transmite en las redes?

Luis Enrique Barros Martinez

No, como tal sobre la trama, en general había interactuado con funciones para comprobar la integridad de la información. Es importante considerar siempre la integridad de los datos transmitidos pues a nivel físico, no sabemos en qué pueda afectar cada bit a la información u que tan grande sea el impacto

Alfredo Bautista Ríos

Hasta antes de la clase teórica y el laboratorio no tenía conocimiento de esta, comprendo lo importante que puede ser comprobar trama para detección de errores, analizar la integridad de los datos y ver que la transmisión este siendo eficiente.

Al trabajar con información en redes es importante que Betito reciba exactamente lo que Alicia le mandó, si la transmisión no es íntegra y ese es un problema constante, dudo que sea una buena comunicación.

Es por eso que es gracias a el checksum que podemos garantizar que lo transmitido sea completo o íntegro, pues puede detectar fallos en la transmisión .

Criterio	Valor	Tu evaluación
El programa se escribe en un block de notas y se compila con gcc	1	1
Se incluye todo el código	1	1
El mapa de memoria refleja todas las variables utilizadas en el programa y han sido seleccionadas de forma consciente.	1	1
Las imágenes de la ejecución son claras	1	1
Se incluyen las conclusiones solicitadas	1	1
TOTAL	5	5

