



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



REDES DE COMPUTADORAS

PRÁCTICA 2:
“Checksum”

NOMBRE DEL ESTUDIANTE:
HERNÁNDEZ RODRÍGUEZ ARMANDO GIOVANNI

GRUPO: 2CM15

PROFESORA: CORTEZ DUARTE NIDIA ASUNCIÓN



Práctica 2.-Checksum

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 mostrar el correcto.

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, sugiero utilizar alguna herramienta para dar formato como:

<http://www.planetb.ca/syntax-highlight-word>

Nota: Para darle formato al código se utilizó carbon:

<https://carbon.now.sh/>



M. en C. Nidia A. Cortez Duarte



Práctica 2.-Checksum

Anexo Código Fuente

```
// Elaborado por: Hernández Rodríguez Armando Giovanni
#include<stdio.h>

unsigned short int checksum = 0x0000; // 2 bytes = 16 bits

unsigned short int getChecksum(unsigned char frame[] ){
    unsigned char i = 0x00;
    unsigned char aux = 0x00;
    checksum = 0x0000; // 16 bits = 2 bytes

    // Suma hexadecimal con bloques de 2bytes
    for(i=0; i<20; i++){
        // Suma la concatenacion de 1 byte el byte siguiente de una trama
        checksum += (frame[i]<<8) | frame[++i];
        //Acarreo fuera de los 16 bits
        if(i==17){
            aux = checksum>>12;
        }
    }
    // Suma hexadecimal con el acarreo
    checksum += aux;
    // Complementar a uno el resultado
    checksum = ~(checksum);

    return checksum;
}

void correctChecksum(unsigned char frame[]){
    checksum = 0x0000;
    // Obtener el checksum correspondiente a una trama para ello se asigna un valor de 0 al checksum
    frame[10] = 0x00;
    frame[11] = 0x00;
    checksum = getChecksum(frame);

    /* Si el checksum dado es igual al calculado entonces el checksum es correcto
    sino el checksum es incorrecto y se indica el valor del checksum correcto.
    */
    if((checksum >> 8) == frame[10] & (checksum & 0x00ff) == frame[11]){
        printf("\nChecksum correcto");
    }else{
        printf("\nChecksum incorrecto");
        printf("\nEl checksum correcto es: %.4x", checksum);
    }
}

int main(){

    unsigned char frame1[] = {0x45, 0x00, 0x01, 0xe2, 0xd7, 0xdb, 0x40, 0x00, 0x80, 0x06, 0x00, 0x00,
    0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};
    unsigned char frame2[] = {0x45, 0x00, 0x01, 0x9c, 0xd7, 0xde, 0x40, 0x00, 0x80, 0x06, 0x88, 0x9d,
    0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};
    checksum = 0x0000;

    // Prueba 1
    checksum = getChecksum(frame1);
    printf("\nPrueba1\nChecksum de la trama1: %.4x\n",checksum);

    // Prueba 2
    printf("\nPrueba2");
    correctChecksum(frame2);

    return 0;
}
```

