INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
Nombre: Ramirez Benitez Brayan

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.

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}
```

```
#include<stdio.h>

unsigned short ObtenerChk(unsigned char *, unsigned char);
void VerificarChk(unsigned char *,unsigned char, unsigned short);
void MostrarChk(unsigned short);

int main(){
        unsigned short Checksum = 0;

        unsigned char T[20] = {0x45, 0x00, 0x01, 0xe2, 0xd7, 0xdb, 0x40, 0x00, 0x80,
        0x06, 0x00, 0x00, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};

        Checksum = ObtenerChk(T, 20);
        MostrarChk(Checksum);
        //VerificarChk(T1, 20, Checksum);

        return 0;
}

unsigned short ObtenerChk(unsigned char *T, unsigned char tam){
        unsigned char i;
        unsigned int chk = 0x00;

        for(i = 0; i<tam ;i+=2){
                if(!(i == 10))
                        chk += (T[i] << 8) + T[i + 1];
        }

        return (~((chk&65535) + (chk >> 16)))&65535;
}

void VerificarChk(unsigned char *T, unsigned char tam, unsigned short Checksum){

        if(Checksum == (T[10] << 8) + T[11])
                printf("El Checksum 0x%.4x es correcto");
        else{
                printf("El Checksum 0x%.4x no es correcto\n", (T[10] << 8) + T[11]);
                MostrarChk(Checksum);
        }
}

void MostrarChk(unsigned short Checksum){
        printf("El Checksum es: 0x%.4x\n", Checksum);
}
```

```
C:\Users\braya\Downloads\Redes\Practica 2\Codigo>gcc chksum.c -o chksum

C:\Users\braya\Downloads\Redes\Practica 2\Codigo>chksum
El Checksum es: 0x8fa5
```

M. en C. Nidia A. Cortez Duarte

Práctica 2.-Checksum

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
```

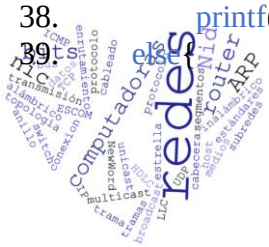
```c
#include<stdio.h>

unsigned short ObtenerChk(unsigned char *, unsigned char);
void VerificarChk(unsigned char *,unsigned char, unsigned short);
void MostrarChk(unsigned short);

int main(){
        unsigned short Checksum = 0;

        unsigned char T[20] = {0x45, 0x00, 01, 0x9c, 0xd7, 0xde, 0x40, 0x00, 0x80,
        0x06, 0x88, 0x9d, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};

        Checksum = ObtenerChk(T, 20);
        VerificarChk(T, 20, Checksum);

        return 0;
}

unsigned short ObtenerChk(unsigned char *T, unsigned char tam){
        unsigned char i;
        unsigned int chk = 0x00;

        for(i = 0; i<tam ;i+=2){
                if(!(i == 10))
                        chk += (T[i] << 8) + T[i + 1];
        }

        return (~((chk&65535) + (chk >> 16)))&65535;
}

void VerificarChk(unsigned char *T, unsigned char tam, unsigned short Checksum){

        if(Checksum == (T[10] << 8) + T[11])
                printf("El Checksum 0x%.4x es correcto");
        else{
                printf("El Checksum 0x%.4x no es correcto\n", (T[10] << 8) + T[11]);
                MostrarChk(Checksum);
        }
}

void MostrarChk(unsigned short Checksum){
        printf("El Checksum es: 0x%.4x\n", Checksum);
}
```

```
C:\Users\braya\Downloads\Redes\Practica 2\Codigo>gcc chksum.c -o chksum

C:\Users\braya\Downloads\Redes\Practica 2\Codigo>chksum
El Checksum 0x889d no es correcto
El Checksum es: 0x8fe8

C:\Users\braya\Downloads\Redes\Practica 2\Codigo>
```

Práctica 2.-Checksum

Código en C

```c
#include<stdio.h>

unsigned short ObtenerChk(unsigned char *, unsigned char);
void VerificarChk(unsigned char *,unsigned char, unsigned short);
void MostrarChk(unsigned short);

int main(){
	unsigned short Checksum = 0;

	unsigned char T[20] = {0x45, 0x00, 0x01, 0xe2, 0xd7, 0xdb, 0x40, 0x00, 0x80,
	0x06, 0x00, 0x00, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};

	unsigned char T1[20] = {0x45, 0x00, 01, 0x9c, 0xd7, 0xde, 0x40, 0x00, 0x80,
	0x06, 0x88, 0x9d, 0xc0, 0xa8, 0x01, 0x43, 0x94, 0xcc, 0x3a, 0xdd};

	Checksum = ObtenerChk(T, 20);
	MostrarChk(Checksum);
	//VerificarChk(T1, 20, Checksum);

	return 0;
}

unsigned short ObtenerChk(unsigned char *T, unsigned char tam){
	unsigned char i;
	unsigned int chk = 0x00;

	for(i = 0; i<tam ;i+=2){
		if(!(i == 10))
			chk += (T[i] << 8) + T[i + 1];
	}

	return (~((chk&65535) + (chk >> 16)))&65535;
}

void VerificarChk(unsigned char *T, unsigned char tam, unsigned short Checksum){

	if(Checksum == (T[10] << 8) + T[11])
		printf("El Checksum 0x%.4x es correcto");
	else{
```



M. en C. Nidia A. Cortez Duarte

Práctica 2.-Checksum

```
40.             printf("El Checksum 0x%.4x no es correcto\n", (T[10] << 8) + T[11]);
41.             MostrarChk(Checksum);
42.         }
43. }
44.
45. void MostrarChk(unsigned short Checksum){
46.         printf("El Checksum es: 0x%.4x\n", Checksum);
47. }
```

Mapa de memoria utilizado en el programa (considerando registros de 8 bits)

Unsigned short Checksum	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
Unsigned int chk	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
Unsigned char i	0	0	0	0	0	0	0	0
Unsigned char tam	0	0	0	1	0	1	0	0

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
TOTAL	**4**	4