



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



REDES DE COMPUTADORAS

PRÁCTICA 1:

“Calculadora IP versión 1”

NOMBRE DEL ESTUDIANTE:

HERNÁNDEZ RODRÍGUEZ ARMANDO GIOVANNI

GRUPO: 2CM15

PROFESORA: CORTEZ DUARTE NIDIA ASUNCIÓN



Práctica 1.-Calculadora IP

Objetivo: el alumno programará una calculadora IP versión 1 en C. Haciendo uso de un block de notas y compilando en gcc

Dada una dirección IP (inicializada en hexadecimal)

Output: Clase, Dirección IP de Red y dirección de Broadcast.

Las IP de prueba son las siguientes, favor de llenar la tabla

	Dirección IP	Dir en hexadecimal	Clase	Dir. IP Red	Broadcast
Z	10.250.1.1	0X0A.0XFA.0X01.0X01	A	10.0.0.0	10.255.255.255
Y	150.10.15.0	0X96.0X0A.0X0F.0X00	B	150.10.0.0	150.10.255.255
X	192.14.2.0	0XC0.0X0E.0X02.0X00	C	192.14.2.0	192.14.2.255
W	148.17.9.1	0X94.0X11.0X09.0X01	B	148.17.0.0	148.17.255.255
V	193.0.0.255	0XC1.0X00.0X00.0XFF	C	193.0.0.0	193.0.0.255
U	220.200.23.1	0XDC.0XC8.0X17.0X01	C	220.200.23.0	220.200.23.255
T	177.100.18.4	0XB1.0X64.0X12.0X04	B	177.100.0.0	177.100.255.255
S	95.0.21.0	0X5F.0X00.0X15.0X00	A	95.0.0.0	95.255.255.255
R	111.111.111.111	0X6F.0X6F.0X6F.0X6F	A	111.0.0.0	111.255.255.255
Q	219.26.51.0	0XDB.0X1A.0X33.0X00	C	219.26.51.0	219.26.51.255
P	167.0.0.0	0XA7.0X00.0X00.0X00	B	167.0.0.0	167.0.255.255
O	240.1.1.1	0XF0.0X01.0X01.0X01	E	-	-
N	10.1.1.1	0X0A.0X01.0X01.0X01	A	10.0.0.0	10.255.255.255
M	225.1.1.1	0XE1.0X01.0X01.0X01	D	-	-

Instrucciones.

- Abrir un block de notas y escribir el código correspondiente en C.
- La dirección IP se inicializará en hexadecimal en un arreglo de caracteres sin signo
- La máscara de red se inicializará en decimal 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 las 14 direcciones IP de la tabla. Tomar captura de pantalla de la salida para cada IP y pegarlas al final de este documento.

Se entrega este documento, incluir:

Todo tu Código, sugiero utilizar alguna herramienta para dar formato como:

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

Deberan crear y mostrar **el mapa de memoria** utilizado en su programa (considerar registros de 8 bits)



M. en C. Nidia A. Cortez Duarte



Práctica 1.-Calculadora IP

Conclusiones

Mediante el desarrollo de esta práctica comprendí la importancia del uso de operadores binarios en C, particularmente el operador AND, NOT y OR. Anteriormente no había hecho uso de los operadores binarios en algún programa, por lo que la práctica 0 y la práctica 1 me han servido para entender su funcionamiento. En lo personal considero que algunas veces su implementación es la mejor forma de operar sin tener que escribir mucho código, en específico cuando se está operando bit a bit. De hecho, en procesadores de bajo costo, estas operaciones son mucho más eficiente y rápidas que la división, adición, substracción y multiplicación.

Ahora bien, en cuanto a la implementación tuve problemas con los resultados que se obtenían del operador NOT (~), debido a que aunque la IP y MR (máscara de red) estaban declaradas como arreglos de carácter sin signo tuve que hacer un cast (unsigned char) para forzar a que los resultados con este operador fueran los correctos, sino me devolvía números negativos.

Finalmente comparando el costo computacional sobre el uso de if y else si se usa el operador & y un valor numérico o && y un rango de valores numéricos, se observa que el uso de & es mucho mejor debido a que solo se está evaluando una condición a nivel bit a bit, la cual si es verdadera devuelve un valor entero diferente a cero. En cambio al utilizar && y un rango de valores se establece una comparación entre cada uno de los valores que se tiene en ese rango, ya no se hace la comparación de un valor sino de un conjunto de valores que si es verdadero devuelve un 1. Esto me enseña que al evaluar la complejidad de los algoritmos, reside únicamente en el problema real que debemos solucionar. Por lo que debemos centrarnos primero en las propiedades del “enunciado” y jugar con ellas experimentalmente, para conseguir la mejor estrategia (algoritmo).





Práctica 1.-Calculadora IP

Anexo Código Fuente

```
1  #include<stdio.h>
2
3  int main(){
4      unsigned char MR[4] = {0, 0, 0, 0};
5      unsigned char ip[4] = {0X0A, 0XFA, 0X01, 0X01}; // 10.250.1.1
6      //unsigned char ip[4] = {0X96, 0X0A, 0X0F, 0X00}; // 150.10.15.0
7      //unsigned char ip[4] = {0XC0, 0X0E, 0X02, 0X00}; // 192.14.2.0
8      //unsigned char ip[4] = {0X94, 0X11, 0X09, 0X01}; // 148.17.9.1
9      //unsigned char ip[4] = {0XC1, 0X00, 0X00, 0XFF}; // 193.0.0.255
10     //unsigned char ip[4] = {0XDC, 0XC8, 0X17, 0X01}; // 220.200.23.1
11     //unsigned char ip[4] = {0XB1, 0X64, 0X12, 0X04}; // 177.100.18.4
12     //unsigned char ip[4] = {0X5F, 0X00, 0X15, 0X00}; // 95.0.21.0
13     //unsigned char ip[4] = {0X6F, 0X6F, 0X6F, 0X6F}; // 111.111.111.111
14     //unsigned char ip[4] = {0XDB, 0X1A, 0X33, 0X00}; // 219.26.51.0
15     //unsigned char ip[4] = {0XA7, 0X00, 0X00, 0X00}; // 167.0.0.0
16     //unsigned char ip[4] = {0XF0, 0X01, 0X01, 0X01}; // 240.1.1.1
17     //unsigned char ip[4] = {0X0A, 0X01, 0X01, 0X01}; // 10.1.1.1
18     //unsigned char ip[4] = {0XE1, 0X01, 0X01, 0X01}; // 225.1.1.1
19
20     if(ip[0] & 128){ //Resultado es diferente de cero 1
21         if(ip[0] & 64){ // 11
22             if(ip[0] & 32){ // 111
23                 if(ip[0] & 16){ // 1111
24                     printf("\nClase E");
25                     printf("\nIP:%d.%d.%d.%d\nSin mascara de red", ip[0], ip[1],
26                         ip[2], ip[3]);
27                     return 0;
28                 }else{ // 1110
29                     printf("\nClase D");
30                     printf("\nIP:%d.%d.%d.%d\nSin mascara de red", ip[0], ip[1],
31                         ip[2], ip[3]);
32                     return 0;
33                 }
34             }else{ // 110
35                 MR[0] = 255;
36                 MR[1] = 255;
37                 MR[2] = 255;
38
39                 printf("\nClase C");
40             }
41         }else{ // 10
42             MR[0] = 255;
43             MR[1] = 255;
44             printf("\nClase B");
45         }
46     }
```

M. en C. Nidia A. Cortez Duarte





Práctica 1.-Calculadora IP

```
46     }else{// Resultado es cero 0
47         MR[0] = 255;
48         printf("\nClase A");
49     }
50
51     printf("\nIP:%d.%d.%d.%d", ip[0], ip[1], ip[2], ip[3]);
52     printf("\nIP Broadcast: %d.%d.%d.%d", ip[0]| (unsigned char)~MR[0],
53     ip[1]| (unsigned char) ~MR[1], ip[2]| (unsigned char)~MR[2],
54     ip[3]| (unsigned char)~MR[3]);
55     printf("\nIP Red: %d.%d.%d.%d\n", ip[0] & MR[0], ip[1] & MR[1], ip[2] & MR[2],
56     ip[3] & MR[3]);
57     return 0;
58 }
```

Evaluación

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 y se hace uso de operadores binarios	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 son claras	1	1
Se incluyen las conclusiones solicitadas	1	1
TOTAL	5	5

