



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



REDES DE COMPUTADORAS

“REPORTE PRÁCTICA 1-CALCULADORA IP”

Abstract

In this report we will see some of characteristics in order to learn about your IP, in this report there is a program whose purpose is gives you class' classification, mask, type, host range, broadcast and you may repeat the action with other direction.

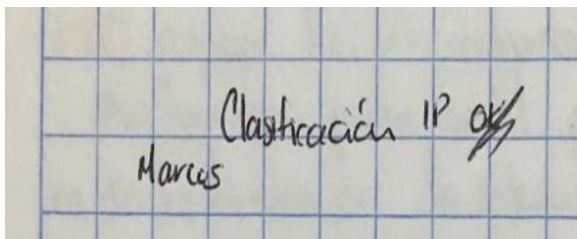
By:

MARCOS OSWALDO VÁZQUEZ MORENO

Professor:

MSc. NIDIA ASUNCIÓN CORTEZ DUARTE

SEPTEMBER 2018



Índice

Contenido

Introducción	1
Marco Teórico	1
Software (librerías, paquetes, herramientas)	2
Procedimiento	2
Resultado	3
Discusión:	4
Conclusiones:.....	4
Referencias	5
Código	6

Introducción

En el siguiente reporte se hablará del protocolo IPv4 que nos permite comunicarnos entre redes y algunas características que lo conforman, de esta manera, daremos paso a la creación de un programa el cual permite que a la inserción de una dirección IP te devuelva como resultado a qué clasificación pertenece, cuál es su máscara de red, cuál es su tipo, cuál es su rango de host y en caso de querer repetir el proceso, hacerlo de esta manera ingresando la misma o distinta dirección IP. [1]

Marco Teórico

Quizás te has preguntado alguna vez lo que significa la "IP" que aparece en algunas palabras como VoIP. IP es el acrónimo de "Internet Protocol" o Protocolo de Internet en castellano. También se le conoce simplemente como protocolo IP pero, ¿qué es eso de protocolo IP? Con este artículo vamos a intentar ayudarte a entender cómo funciona el protocolo IP y por qué cosas como la voz se puede transmitir mediante este protocolo IP.

Un protocolo es un conjunto de normas que rigen el funcionamiento de las cosas en una determinada tecnología, por lo que de esta forma se consigue que exista algún tipo de estandarización. Cuando hablamos de comunicaciones de red, un protocolo es el conjunto de normas que rigen cómo los paquetes de comunicación se transmiten a través de la red. Cuando tienes un protocolo, puedes estar seguro de que todas las máquinas de una red (o del mundo, cuando se trata de Internet), por muy diferentes que sean, hablan el mismo idioma y pueden integrarse en cualquier sistema.

Desarrollado durante la década de 1970, el protocolo IP es el protocolo de red fundamental usado a través de Internet, las redes domésticas y las redes empresariales. El protocolo IP se utiliza a menudo junto con el protocolo de control de transporte (Transport Control Protocol o TCP) y entonces se les llama de manera intercambiable tanto protocolo IP como protocolo TCP/IP. [2]

Dentro de las características de una IP tenemos las diferentes clasificaciones, las cuales se definen por el primer octeto de bits:

- A.- 0.0.0.0 al 127.255.255.255
- B.- 128.0.0.0 al 191.255.255.255
- C.- 192.0.0.0 al 223.255.255.255
- D.- 224.0.0.0 al 239.255.255.255
- E.- 240.0.0.0 al 255.255.255.255

Continuando, se inicializaron 4 arreglos para las máscaras de red en 0. Posteriormente se pide al usuario que ingrese una IP válida, distribuyendo de esta manera la cadena en octetos separados por el punto, a continuación, entra a una validación de la primera posición del arreglo en comparación con 128 para la clasificación A, 64 para la clasificación B, 32 para la clasificación C, 16 para la clasificación D y 8 para la clasificación E.

De esta manera se imprime a que clasificación pertenece, aunque tenemos el caso en el que al entrar a la clasificación A, B y/o C se despliegan los siguientes resultados:

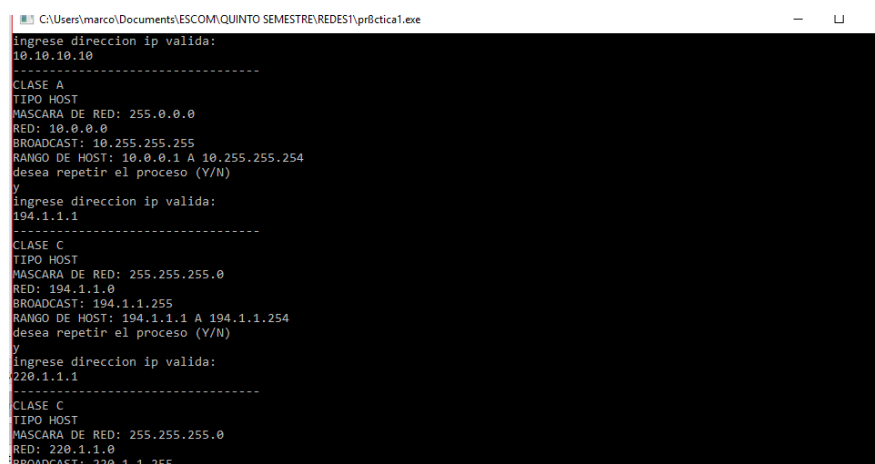
- **Mascara de Red:** Imprimiendo la inicialización correspondiente de la Máscara asignada para cada uno de los casos.
 - Clase A: MR[0]= 255; MR[1]= 0; MR[2]= 0; MR[3]= 0.
 - Clase B: MR[0]= 255; MR[1]= 255; MR[2]= 0; MR[3]= 0;
 - Clase C: MR[0]= 255; MR[1]= 255; MR[2]= 255; MR[3]= 0;
- **Red:** Imprimiendo una compuerta AND entre el arreglo IP[] y el arreglo de máscara MR[], esto mediante el uso del operador bit a bit “&”.
- **Broadcast:** Imprimiendo una compuerta OR entre el arreglo IP[] y un casteo a unsigned char del arreglo MR[], esto mediante el uso bit a bit inclusivo “|”.
- **Rango de host:** Imprimimos una compuerta AND entre el arreglo IP[] y el arreglo de máscara MR[] sin contar la posición 3 de cada arreglo en la cual para el rango inicial se suma uno y en el final se resta uno.

Teniendo por último un ciclo para repetir el proceso de ser deseado por el usuario.

Resultado

A continuación, se muestran algunas imágenes en las que se puede apreciar el correcto funcionamiento de dicho programa.

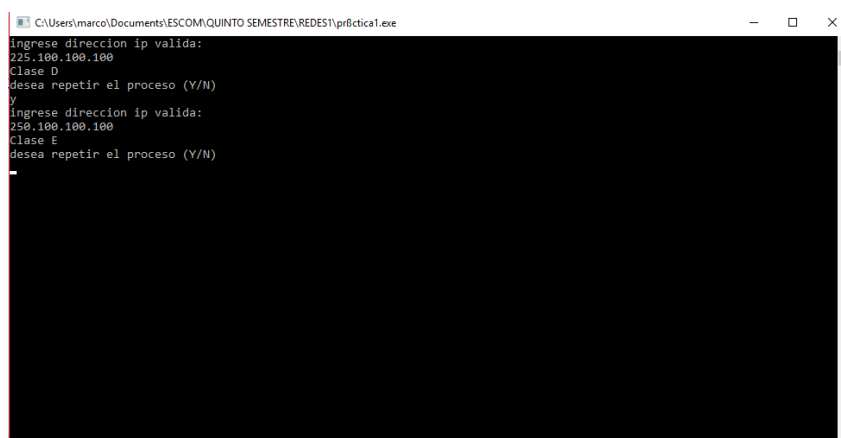
En la imagen 2 se puede apreciar un resultado de la inserción de una dirección IP.



```
C:\Users\marco\Documents\ESCOM\QUINTO SEMESTRE\REDES1\pr8ctica1.exe
Ingrese direccion ip valida:
10.10.10.10
-----
CLASE A
TIPO HOST
MASCARA DE RED: 255.0.0.0
RED: 10.0.0.0
BROADCAST: 10.255.255.255
RANGO DE HOST: 10.0.0.1 A 10.255.255.254
desea repetir el proceso (Y/N)
y
Ingrese direccion ip valida:
194.1.1.1
-----
CLASE C
TIPO HOST
MASCARA DE RED: 255.255.255.0
RED: 194.1.1.0
BROADCAST: 194.1.1.255
RANGO DE HOST: 194.1.1.1 A 194.1.1.254
desea repetir el proceso (Y/N)
y
Ingrese direccion ip valida:
220.1.1.1
-----
CLASE C
TIPO HOST
MASCARA DE RED: 255.255.255.0
RED: 220.1.1.0
BROADCAST: 220.1.1.255
```

Imagen 2. Resultados.

En la imagen 3 se muestra el funcionamiento de las clasificaciones D y E.



```
C:\Users\marco\Documents\ESCOM\QUINTO SEMESTRE\REDES1\prfctica1.exe
Ingrese direccion ip valida:
225.100.100.100
Clase D
desea repetir el proceso (Y/N)
y
Ingrese direccion ip valida:
250.100.100.100
Clase E
desea repetir el proceso (Y/N)
```

Imagen 3. Resultados Clasificación D y E.

Discusión:

Interpretando los resultados del estudio hecho anteriormente me hace recordar que sin duda tenemos un lenguaje de nivel medio alto el cual permite realizar trabajos bit a bit y claro a nivel de bits, es preciso decir que se tiene el resultado con operadores a nivel de bits que quizá utilizabas para otras cosas en el lenguaje.

Por otro lado, tenemos el tema de cómo se distribuida una dirección IP, en 4 octetos de bits y esto se hace para tener de manera más rápida y fácil las direcciones IP de cualquier dispositivo con tarjeta de red. Las direcciones IP están distribuidas por regiones y por formas de acceso en la red, datos móviles y distintos distribuidores de servicios de internet.

Dicho lo anterior, es importante mencionar que la teoría vista en clase se puede llevar a cabo en la lógica de la programación, así como los que inventaron la forma de conectividad en red de esta manera lo lograron conseguir.

Aun cuando no es mi fortaleza la programación en este lenguaje logré llevarlo a cabo, claro que no es el más difícil, pero me gustó mucho como a nivel de bits lo logré hacer y sin ningún problema.

Conclusiones:

En conclusión, puedo decir que aprendí a usar de manera adecuada los operadores a nivel de bits, que ahora sé que al usar algunas técnicas de programación puedo implementar técnicas usadas en las redes de computadora, así como se puede implementar para otras ramas de la computación claro está decirlo, sin embargo, todo esto tiene aplicaciones reales por lo que me permito mencionar el como un router no puede asignar a un dispositivo con tarjeta de red una dirección asignada para red o para broadcast, por lo que tiene un rango dependiendo de la clasificación para la que se ha asignado.

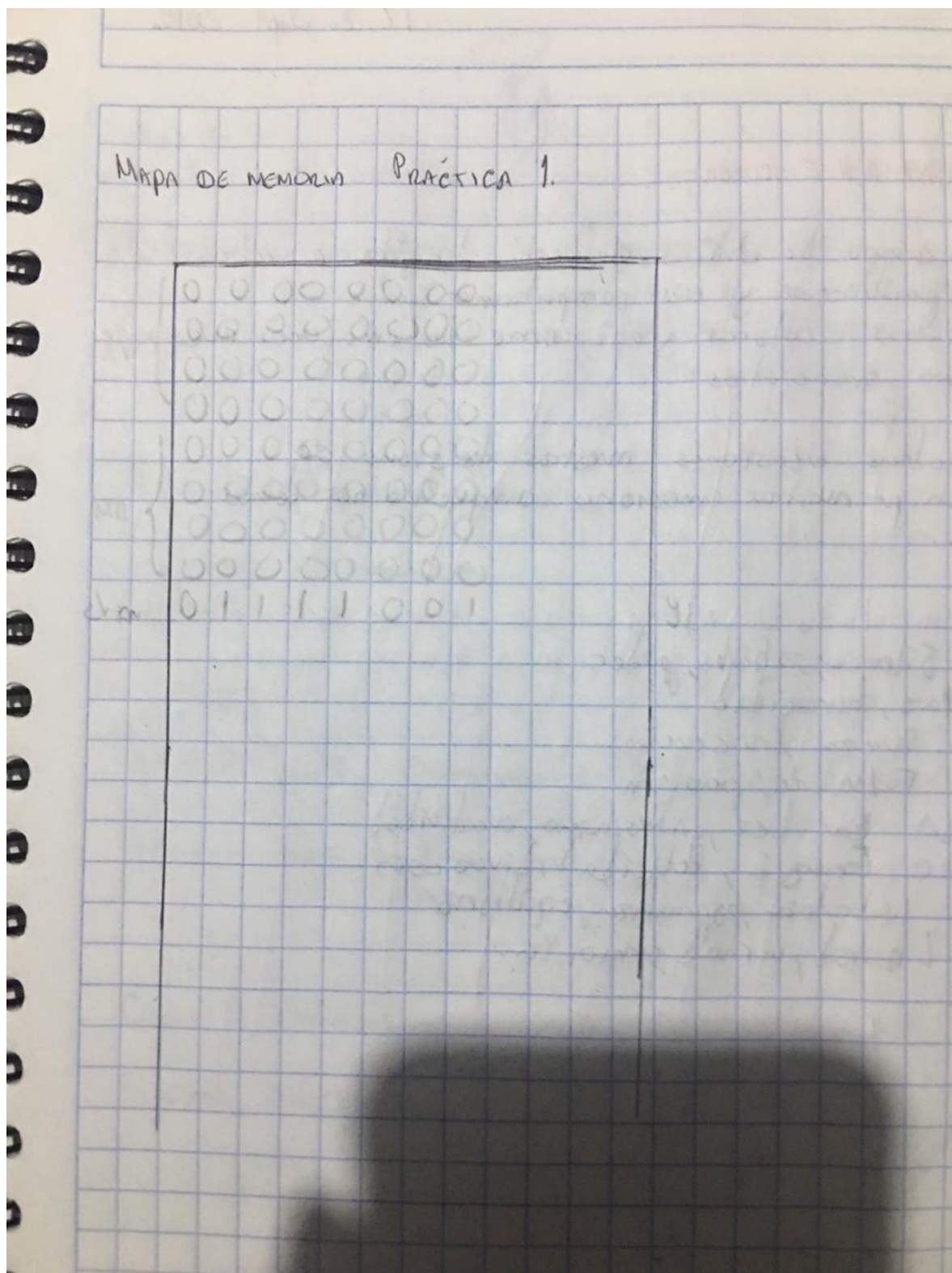
Afortunadamente y gracias a la lógica que manejé no tuve casi ningún error o complicación, solo que al momento de obtener el apartado de broadcast no estaba casteando el tipo de dato que estaba ocupando directo del arreglo MR[] destinado a las máscaras de red.

Por último, pero no lo menos importante, es decir que los resultados de la aplicación fueron los esperados, fueron exitosos, así como el aprendizaje que me deja.

Referencias

- [1] N. Córtez, «Explicación del problema,» CDMX, 2018.
- [2] U. I. d. Valencia, «<https://www.universidadviu.com/funciona-protocolo-ip/>,» [En línea]. Available: <https://www.universidadviu.com/funciona-protocolo-ip/>.
- [3] B. O. Moncada, Programando en C a bajo nivel, Buenos Aires: Facultad de Ingeniería, 2011.
- [4] Google, «Syntax Highlight Code In Word Documents,» Google, 2012. [En línea]. Available: <http://www.planetb.ca/syntax-highlight-word>. [Último acceso: 22 Septiembre 2018].

Mapa de memoria



Código

El código que se muestra en seguida fue el implementado para el programa llamado Calculadora IP. [4]

```
1.  /*
2.  VAZQUEZ MORENO MARCOS OSWALDO
3.  REDES DE COMPUTADORAS
4.  EJERCICIO 1
5.  CLASIFICACION DE CLASES EN UNA DIRECCION IP, MASCARA DE RED, TIPO, RANGO DE HOST,
6.  BROADCAST, REPETIR
7.  */
8.  #include<stdio.h>
9.  #include<stdlib.h>
10.
11. //int validacion(int ip);
12.
13. int main(){
14.
15.
16. //arreglo de cuatro octetos
17. unsigned char IP[4]= {0,0,0,0}, MR[4]= {0,0,0,0};
18. char a= 'y';
19. do{
20.
21.     printf("ingrese direccion ip valida: \n");
22.     scanf("%u.%u.%u.%u",&IP[0], &IP[1], &IP[2], &IP[3]);
23.     fflush(stdin);
24. //validacion para entrar al distinto tipo de casos
25. if(IP[0] & 128) //clase A
26.     {
27.         if(IP[0] & 64) //clase B
28.         {
29.             if(IP[0] & 32) //clase C
30.             {
31.                 if(IP[0] & 16) //clase D
32.                 {
33.                     if(IP[0] & 8) //clase E
34.                     {
35.                         printf("Clase E\n");
36.                     }
37.                     else
38.                     {
39.                         printf("Clase E\n");
40.                     }
41.                 }
42.                 else
43.                 {
44.                     printf("Clase D\n");
45.                 }
46.             }
47.             else
48.             {
49.                 printf("-----\n");
50.                 printf("CLASE C\n");
51.                 MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
52.                 MR[1]= 255;
53.                 MR[2]= 255;
54.                 MR[3]= 0;
```

```

55.         if(IP[3]==255){
56.             printf("TIPO BROADCAST\n");
57.         }
58.         else if(IP[3]==0){
59.             printf("TIPO RED\n");
60.         }
61.         else
62.             printf("TIPO HOST\n");
63.         printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0],MR[1], MR[2], MR[3])
        ;
64.         printf("RED: %u.%u.%u.%u\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2]
        , IP[3]&MR[3]);
65.         printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0],
        IP[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned ch
        ar)~MR[3]); //CASTEO DE ENTERO A UNSIGNED CHAR
66.         printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0]
        , IP[1] & MR[1], IP[2] & MR[2], (IP[3] &MR[3])+1, IP[0] | (unsigned char)~MR[0], I
        P[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned ch
        ar)~MR[3])-1);
67.     }
68. }
69. else
70. {
71.     printf("-----\n");
72.     printf("CLASE B\n");
73.     MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
74.     MR[1]= 255;
75.     MR[2]= 0;
76.     MR[3]= 0;
77.     if(IP[3]==255 && IP[2]==255){
78.         printf("TIPO BROADCAST\n");
79.     }
80.     else if(IP[3]==0 && IP[2]==0){
81.         printf("TIPO RED\n");
82.     }
83.     else
84.         printf("TIPO HOST\n");
85.     printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0],MR[1], MR[2], MR[3]);
86.     printf("RED: %u.%u.%u.%u\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2], IP
        [3]&MR[3]);
87.     printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0], IP[1]
        | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned char)~
        MR[3]);
88.     printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0], IP
        [1] & MR[1], IP[2] & MR[2], (IP[3] &MR[3])+1, IP[0] | (unsigned char)~MR[0], IP[1]
        | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned char)~
        MR[3])-1);
89. }
90. }
91. else
92. {
93.     printf("-----\n");
94.     printf("CLASE A\n");
95.     MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
96.     MR[1]= 0;
97.     MR[2]= 0;
98.     MR[3]= 0;
99.     if(IP[3]==255 && IP[2]==255 && IP[1]==255){
100.         printf("TIPO BROADCAST\n");
101.     }
102.     else if(IP[3]==0 && IP[2]==0 && IP[1]==0){

```

```

103.                printf("TIPO RED\n");
104.            }
105.            else
106.                printf("TIPO HOST\n");
107.
108.                printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0], MR[1], MR[2], MR[3]
109.            );
110.                printf("RED: %u.%u.%u.%u\n", IP[0] & MR[0], IP[1] & MR[1], IP[2] &
111.            MR[2], IP[3] & MR[3]);
112.                printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0],
113.            IP[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned ch
114.            ar)~MR[3], IP[4] | (unsigned char)~MR[4]);
115.                printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0]
116.            , IP[1] & MR[1], IP[2] & MR[2], (IP[3] & MR[3])+1, IP[0] | (unsigned char)~MR[0], I
117.            P[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned ch
118.            ar)~MR[3])-1);
119.            };
120.
121.            printf("desea repetir el proceso (Y/N)\n");
122.            scanf("%c", &a);
123.            fflush(stdin);
124.
125.            }while(a == 'Y' || a == 'y');
126.            return 0;
127.
128.        }

```