ESCOM-IPN

Práctica 1
Calculadora de IP

Redes de Computadoras

Laura Andrea Morales López
MSc. NIDIA ASUNCIÓN CORTEZ DUARTE

Marzo 2018

Resumen

In this report we will see some of the caracteristics of the IP and provide a program that will calculate the IP Class, Range of host, Network and Broadcast.

Índice

1. Indroducción

2. Problema

* What question(s) are you trying to answer? * Include any preliminary observations or background information about the subject

3. Hypothesis:

* Write a possible solution for the problem. * Make sure this possible solution is a complete sentence. * Make sure the statement is testable, an if-then statement is recommended to illustrate what criteria will support your hypothesis (and what data would no support the hypothesis).

4. Software

- Librería stdlib.h
- Librería stdio.h
- Librería stdbool.h

5. Procedimiento

- Leemos la ip, buscamos darle formato con el mismo escaner de la entrada, checamos que cada uno de los cuatro valores esten entre 0 y 225.
- Buscamos la manera más efiiente de guardar esta información que en este caso es un entero sin signo, colocamos la información y pasamos a procesarla.
- Checamos la clase, esto se puede hacer con los bits más significativos del cuarteto de bytes. Podemos darnos cuenta de que lo importante es ver que bits entan en 1 y cuales no, asi que podemos checarlos fácilmente. Para la clase A, el primer bit debe de ser 0. Para la clase B el segundo bit debe de ser 0, y así sucesivamente.
- Despues de esto la marcara la podemos obtener por definición de cada clase y con ella realizamos un OR con la IP y la Mascara negada.
- Realizamos el calculo de la Red con un and binario.

- El rango se obtiene simplemente sumando a la Red +1 y restando al Broadcast −1.
- Sabemos que tipo es por una comparación sencilla, comparamos con la red y el broadcast, si ninguno entra entonces es un tipo host.
- Por ultimo creamos una funcion que nos muestre por pantalla la IP con su formato.

6. Resultados

Tenemos los siguientes resultados:

7. Conclusión

Aprender a realizar calculos bit a bit fue bastante interesante, normalmente nos dicen ese tipo de cosas pero al no aplicarlas no vemos el gran potencial que pueden tener.

Por ejemplo usar un unsigned int para almacenar 32 bits de una IP me parece realmente interesante, o usar los corrimientos y los operadores binarios una manera sencilla, elegante y eficiente de realizar este tipo de calculos.

Cuando lo implemente tuve varios errores, algunos más dificiles de corregir que otros, uno que me costo trabajo ver fue la jerarquía de los operadores binarios, me realizaba cosas extrañas.

La función que se me hizo dificil fue mostrarlo, pues lo guarde como un entero completo, si embargo realizar los calculos fue muy sencillo, esta manera me gustó mas que realizar 4 variables para una IP.

Uno de los errores de mi programa es cuando le mandas caracteres, no coloca nada dentro de mi IP entonces se queda con el 0.0.0.0 default.

Una ultima mejora a realizar es la comparación para definir la clase puedo realizarla de mejor marea sin necesidad de realizar un corrimiento y ademas la operación. Veremos esta mejora en la siguiente versión de esta práctica.

8. Código

```
1  /*########################  TRAMA ANALIZER  ########################
2  * author Laura Andrea Morales* version 0.1
3  * team CompilandoConocimiento* date 4/03/2018
4  * compile "gcc -std=c11 IP.c -o IP"* run "./IP "
5  */
6  #include <stdlib.h>
7  #include <stdio.h>
8  #include<stdbool.h>
9  typedef unsigned int uint;
10 typedef unsigned char byte;
11
12      byte ArrayIP[4]={0,0,0,0};
13      byte ArrayMask[4]={0,0,0,0};
14
15
16
17 //*****************************Only for reserch uses*****************************************
18 void binprintf(int IP)                                          //==============PRINT AS BINARY
       ==========
19 {
20      unsigned int aux=1<<((sizeof(int)<<3)-1);                  //Change the pointer
21      while(aux) {                                               //While aux is true
22          printf("%d", (IP&aux ? 1 : 0));                        //Print it
23          aux >>= 1;                                             //Move it baby
24      }
25 }
26      /*//*****************************Only for reserch uses*****************************************
27 //=================HELLO DARKNESS MY OLD FRIEND=============================================
28      //scanf("%hhu.%hhu.%hhu.%hhu", &ArrayIP[0], &ArrayIP[1], &ArrayIP[2], &ArrayIP[3]); //What is
            your IP?
29      printf("IP\n");
30      ToIPPrint(IP);
31      printf("\n");
32 *///*****************************Only for reserch uses*****************************************
33
34  int POW( byte Exp)                  //==========     POWER BASE 2     ==========
35 {
36      int Pow = 1;                                               //Initialize
```

```
37      for(byte i = 0; i < Exp; i++)                                   //For the number of the exponent
38      {
39          Pow *= 2;                                                   //Power
40         // printf("Pow %d\n",Pow );
41          //binprintf(Pow);
42      }
43      return Pow;
44  }
45
46
47  uint CreateIP(byte ArrayIP[4]) {              // ======= CREATE AN IP =====
48      uint IPAdress = 0;                                                  //Start by cleaning it all**********
49      IPAdress |= ArrayIP[0] << (32 - 8);                                 //Let's put a info in 0-7
50      IPAdress |= ArrayIP[1] << (32 - 16);                                //Let's put b info in 8-15
51      IPAdress |= ArrayIP[2] << (32 - 24);                                //Let's put c info in 16-31
52      IPAdress |= ArrayIP[3];                                             //The info in d just fit
53
54      return IPAdress;                                    //Go little butterfly
55  }
56
57  void Scan(){                                                            //Scan my elements
58      bool flag=true;                                                     //We will need this
59      unsigned short int a = 0 ,b=0,c=0,d=0;
60
61      printf("IP Calculator v1.0\n");
62
63      do{
64          printf("Please give me your IP\n\n");
65      scanf("%hu.%hu.%hu.%hu", &a, &b, &c, &d);                               //Scan my adress
66      if(a<=255 && b<=255 && c<=255 && d<=255) {                          //If youre pretty
67          ArrayIP[0]=a;                                                   //Push into my bytes
68          ArrayIP[1]=b;                                                   //Push into my bytes
69          ArrayIP[2]=c;                                                   //Push into my bytes
70          ArrayIP[3]=d;                                                   //Push into my bytes
                                    //Let me out
71      }
72      else printf("Wrong IP, The values are between 0-255 please try again.\n");//Let me know im fool
73      a=0;
74      b=0;
75      c=0;
76      d=0;
77      printf("Please give me your Mask\n");
78      scanf("%hu.%hu.%hu.%hu", &a, &b, &c, &d);                               //Scan my adress
79      //printf("%hu.%hu.%hu.%hu\n",a,b,c,d);
80      if(a<=255 && b<=255 && c<=255 && d<=252) {                          //If youre pretty
81          ArrayMask[0]=a;                                                 //Push into my bytes
82          ArrayMask[1]=b;                                                 //Push into my bytes
83          ArrayMask[2]=c;                                                 //Push into my bytes
84          ArrayMask[3]=d;                                                 //Push into my bytes
85          flag=false;                                                     //Let me out
86      }
87      else printf("Wrong Mask, The values are between 0-255 please try again.\n");//Let me know im
            fool
88
89  }while(flag);
90
91  }
92  void ToIPPrint(uint IP)                                 //==========Let's print it pretty
        ==========
93  {
94      printf("%d.%d.%d.%d\n", ((IP >> 24) & 0xFF), ((IP >> 16) & 0xFF), ((IP >> 8) & 0xFF), (IP & 0xFF
            ));//Print it
95  }
96
97  byte CountBits(uint Address){
98      bool flag= true;
99      uint aux=2147483648;
100     byte Number=0;
101     byte NumberOfOnes=0;
102
103     for(byte j=0;j<32;j++){
104         if(flag){
105                 if (Address&(aux>>j))
106                 {
107                         Number++;
108                     }
109                     else{
110                         flag=false;
111                         NumberOfOnes=Number;
112                     }}
113         else{
114             if (Address&(aux>>j))
115             {
116                 Number++;
117             }
118         }
119
120 }
```

```
121     if (Number==NumberOfOnes)
122     {
123         return Number;
124 
125     }
126     else return 0;
127 }
128 
129 
130 int main(int argc, char **argv)
131 {
132     Scan();                                                     //Scan my IP
133 
134     uint Mask=CreateIP(ArrayMask);                              //Give me my mask
135     while(!CountBits(Mask)){                                    //The mask is OK?
136         printf("Mask wrong: Please try again.\n");              //If not the Scan again till OK
137         Scan();
138         Mask=CreateIP(ArrayMask);
139     }
140 
141 
142     uint IP=CreateIP(ArrayIP);                                  //Give me my IP
143 
144 //=============================If youre a Class A=============================================
145     if(!(IP&2147483648)){
146         printf("\nClass A\n");                                 //Print your class
147 
148         if(CountBits(Mask)<8){
149             printf("Wrong Mask for class A\n");
150 
151         }
152         else{
153 
154             printf("\nBroadcast: \n");                        //Print your boradcast
155             ToIPPrint(IP|~(Mask));
156 
157             printf("Network\n");                             //Print you Network
158             ToIPPrint(IP&Mask);
159 
160             printf("Range:\n");                              //Print your Range
161             ToIPPrint((IP&Mask)+1);
162             printf("-\n");
163             ToIPPrint((IP|~(Mask))-1);
164 
165             //*************************SOME INFO WE NEED TO KNOW*****************************
166             printf("Sub Network Bits (Borrow bits): %hhu\n",(CountBits(Mask)-8) );
167 
168             printf("Network Bits: 8\n");
169 
170 
171             printf("Host Bits: %hhu\n",(32-CountBits(Mask)) );
172 
173 
174             printf("Number of SubNetworks: %d\n",(POW((CountBits(Mask)-8))));
175 
176             printf("Number of Host by SubNetwork :%d\n",(POW((32-CountBits(Mask)))-2) );
177 
178             printf("\n");
179             //*************************TIPE OF IP*****************************************
180             if (IP==(IP&Mask))printf("Its an IP Network\n");
181             else if(IP==(IP|~(Mask))) printf("Its an IP Broadcast\n");
182             else printf("Its an IP Host\n");
183         }
184     }
185     else if(!(IP&1073741824)){
186         printf("\nClass B\n");                                 //Print your class
187         printf("Mask\n");                                    // and your mask
188         ToIPPrint(Mask);
189 
190         if(CountBits(Mask)<16){
191             printf("Wrong Mask for class B\n");
192 
193         }
194         else{
195 
196 
197             printf("\nBroadcast: \n");                        //Print your boradcast
198             ToIPPrint(IP|~(Mask));
199 
200             printf("Network\n");                             //Print you Network
201             ToIPPrint(IP&Mask);
202 
203             printf("Range:\n");                              //Print your Range
204             ToIPPrint((IP&Mask)+1);
205             printf("-\n");
206             ToIPPrint((IP|~(Mask))-1);
207 
208 
```

```
209             printf("Sub Network Bits (Borrow bits): %hhu\n",(CountBits(Mask)-16) );
210
211             printf("Network Bits: 16\n");
212
213
214             printf("Host Bits: %hhu\n",(32-CountBits(Mask)) );
215
216             printf("Number of SubNetworks: %d\n",(POW((CountBits(Mask)-16))));
217
218             printf("Number of Host by SubNetwork :%d\n",(POW((32-CountBits(Mask)))-2));
219
220             printf("\n");
221             if (IP==(IP&Mask))printf("Its an IP Network\n");
222             else if(IP==(IP|~(Mask))) printf("Its an IP Broadcast\n");
223             else printf("Its an IP Host\n");
224         }
225
226     }
227     else if(!(IP&536870912)){
228         printf("\nClass C\n");                                    //Print your class
229         printf("Mask\n");                                        // and your mask
230         ToIPPrint(Mask);
231
232
233
234         if(CountBits(Mask)<24){
235             printf("Wrong Mask for class C\n");
236
237         }
238         else{
239
240
241             printf("\nBroadcast: \n");                                //Print your broadcast
242             ToIPPrint(IP|~(Mask));
243
244             printf("Network\n");                                     //Print you Network
245             ToIPPrint(IP&Mask);
246
247             printf("Range:\n");                                      //Print your Range
248             ToIPPrint((IP&Mask)+1);
249             printf("-\n");
250             ToIPPrint((IP|~(Mask))-1);
251
252
253             printf("Sub Network Bits (Borrow bits): %hhu\n",(CountBits(Mask)-24) );
254
255             printf("Network Bits: 24\n");
256
257
258             printf("Host Bits: %hhu\n",(32-CountBits(Mask)) );
259
260             printf("Number of SubNetworks: %d\n",(POW((CountBits(Mask)-24))));
261
262             printf("Number of Host by SubNetwork :%d\n",(POW((32-CountBits(Mask)))-2) );
263
264
265
266             printf("\n");
267             if (IP==(IP&Mask))printf("Its an IP Network\n");
268             else if(IP==(IP|~(Mask))) printf("Its an IP Broadcast\n");
269             else printf("Its an IP Host\n");
270         }
271     }
272     else if(!(IP&268435456)){                                  //Print your class
273         printf("Class D\n");
274         printf("Multicast\n");                                       //Tell me about it
275         printf("The range its from 224.0.0.0 to 239.255.255.255 \n it is reservated for multicast.\n
                ");
276
277
278     }
279     else{
280         printf("Class E\n");                                         //Print your class
281         printf("Reserch and development\n");                         //Tel me about it
282         printf("This IP class is reserved for experimentation\nonly for R&D or study. The IP
                adress for this class\nits from 240.0.0.0 to 255.255.255.254\n");
283
284
285     }
286
287
288 }
```

9. Mapa de memoria

Referencias

[1] Axel Ernesto Moreno Cervantes *Redes de Computación*. ESCOM, 2018.

[2] Nidia Cortez. *Redes de Computadoras* ESCOM, 2018.