|  |  |  |
| --- | --- | --- |
| **ipn** | **INSTITUTO POLITÉCNICO NACIONAL**  **ESCUELA SUPERIOR DE CÓMPUTO** |  |

**Redes de Computadoras**

**“Analizador de tramas”**

**Versión 1.- LLC**

**Por:**

**García Laureano Omar Alejandro**

Profesor:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

Abril 2020

Introducción al protocolo LLC

El Control de Enlace Lógico (Logical Link Control) o LLC es la más alta de las dos subcapas de enlace de datos definidas por el IEEE y la responsable del control de enlace lógico. La subcapa LLC maneja el control de errores, control del flujo, entramado, control de diálogo y del direccionamiento de la MAC.

802.2 define una cabecera especial que incluye una cabecera SNAP (subnetwork access protocol). Algunos protocolos, particularmente los diseñados para OSI networking stack, operan directamente sobre 802.2 LLC, que provee los servicios datagrama y orientado a conexión. Esta cabecera 802.2 está actualmente empotrada en paquetes 802.3 (Ethernet II frames, aka. DIX frames).

La cabecera LLC incluye dos campos de dirección adicionales de 8 bit, llamados service access points or SAPs en terminología OSI; cuando la fuente y el destino SAP son puestos al valor 0xAA, el servicio SNAP es requerido. La cabecera SNAP permite usar valores EtherType con todos los protocolos IEEE 802, así como usar protocolos de espacio de ID privados. En IEEE 802.3x-1997, el estándar IEEE Ethernet fue modificado explícitamente para permitir el uso del campo de 16-bit después de la dirección MAC para utilizarlo como un campo de longitud o de tipo.

Estructura

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Delimitador | Dirección | Control | Información | FCS | Delimitador |
| 8 bits | 8 bits extensible | 8 o 16 bits | variable | 16 o 32 | 8 |

I: Información

1 2 3 4 5 6 7 8

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | N(S) | P/F | N(R) |

S: Supervisión

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | S | P/F | N(R) |

U: No numerada

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | M | P/F | M |

N(S) = Número de secuencia enviado

N(R) = Número de secuencia recibido

S = Bits de función supervisora

M = Bits de función no numerada

P/F = Bit de sondeo/fin

Información

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | N(S) | P/F | N(R) |

Supervisión

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | S | 0 | 0 | 0 | 0 | P/F | N(R) |

Para determinar el tipo o tamaño de la trama nos basamos en 3 valores, primero tenemos el valor de los bytes 12 y 13 de la trama, que si recorremos el byte 12 2 espacios a la izquierda y le sumamos el 13 nos dará un número, con este número comparamos con los valores 1500, 2048 y 2054, si el número es menor a 1500, solo se está dando el tamaño en bytes, si el numero es igual a 2048 nos dice que es IP y si el numero es igual a 2054 nos dice que es ARP.

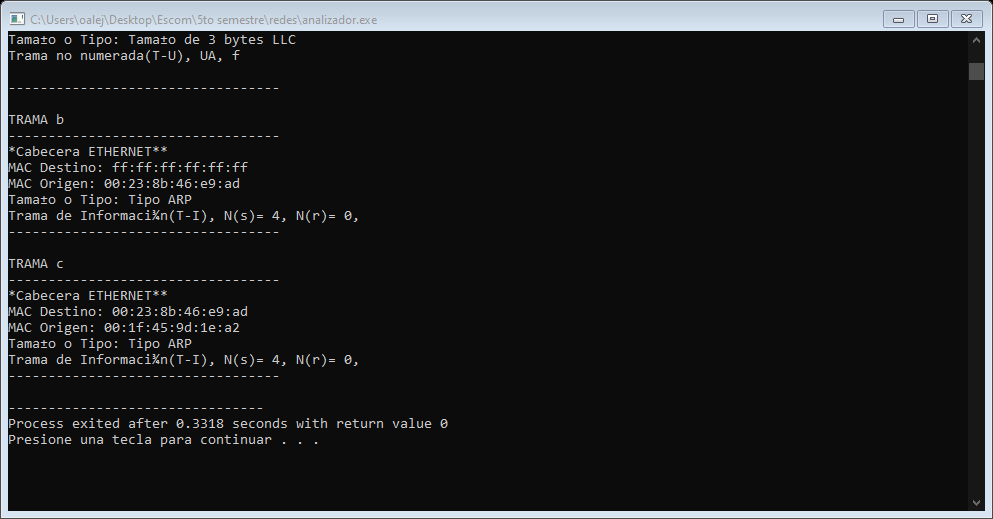
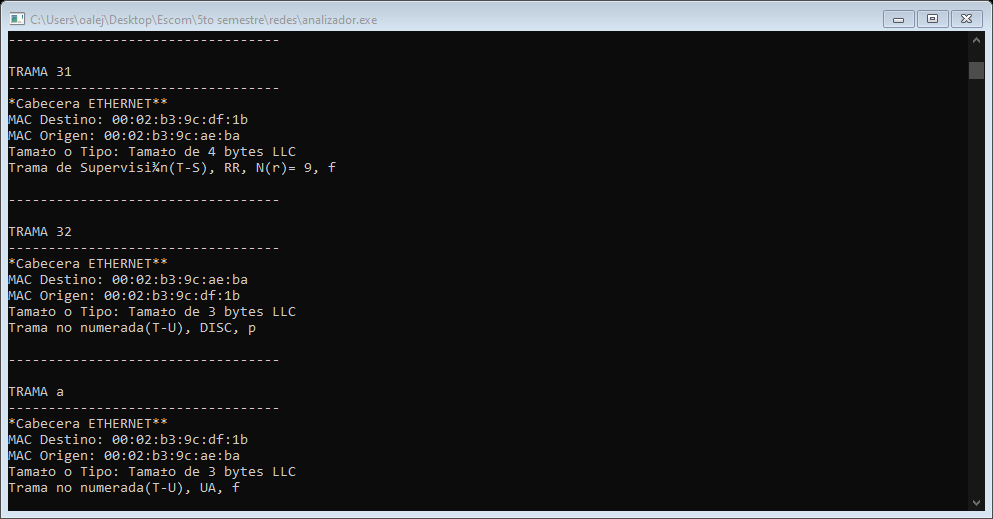
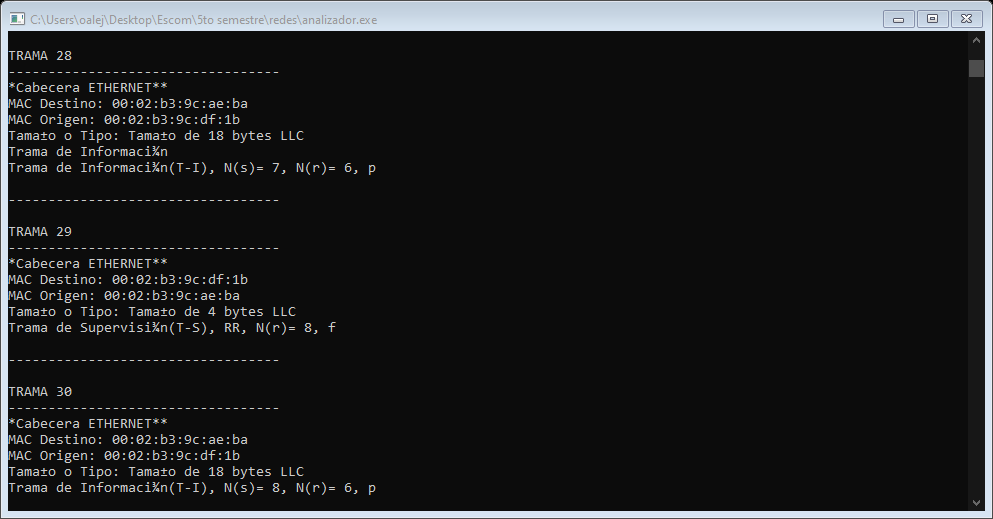
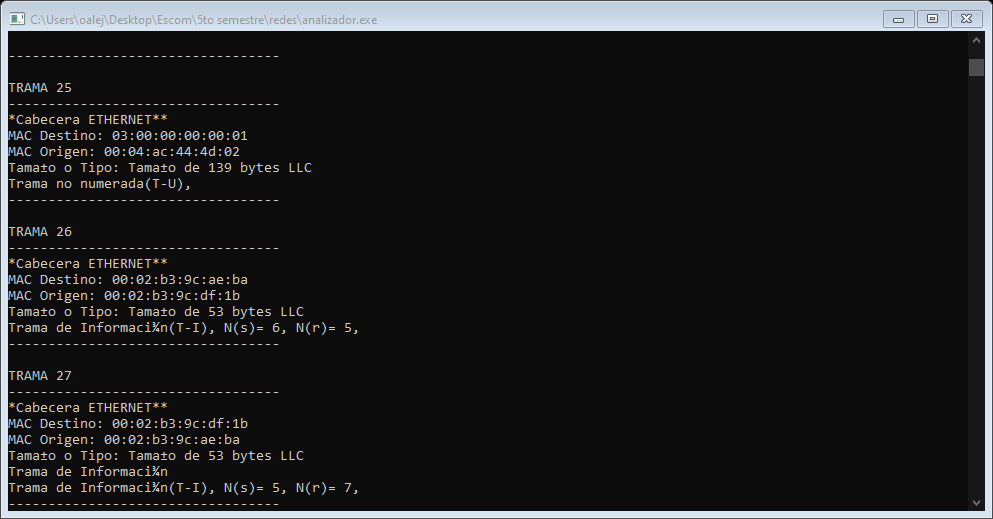
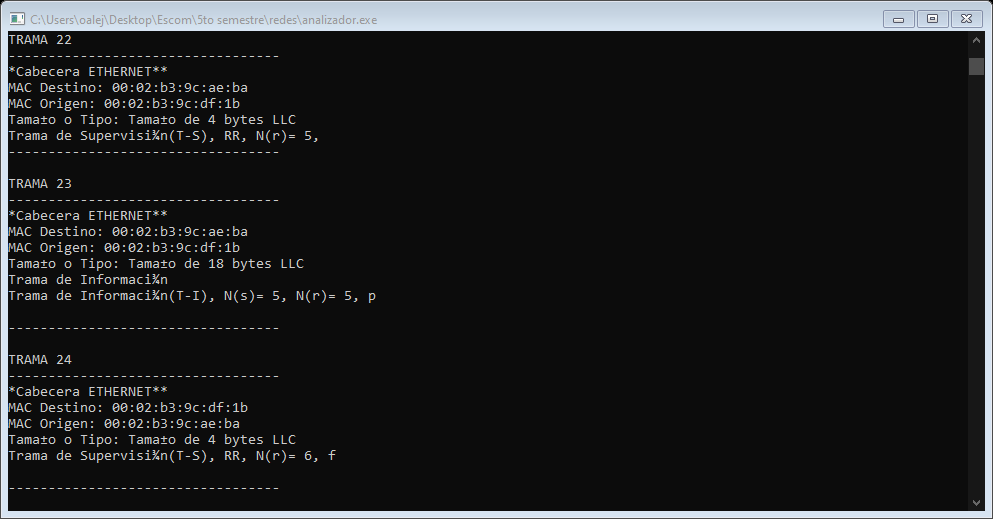
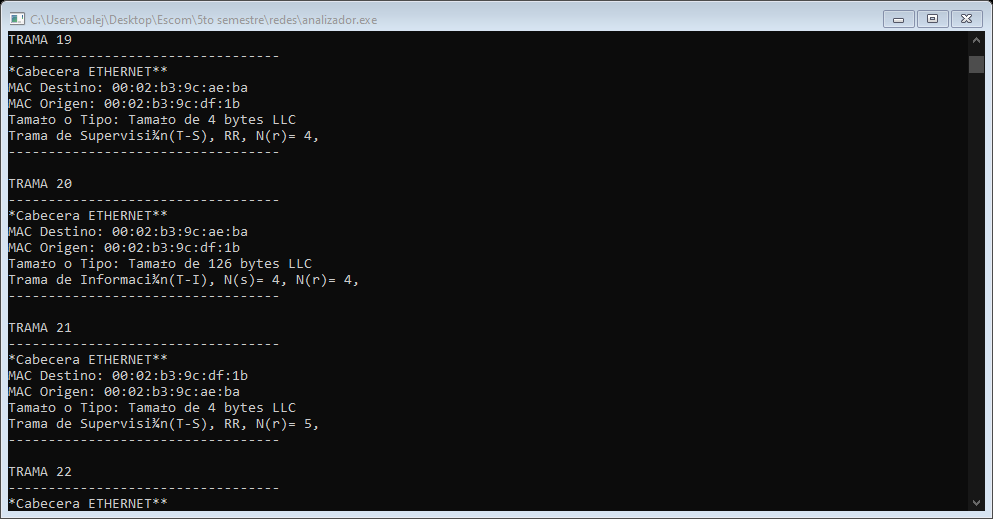
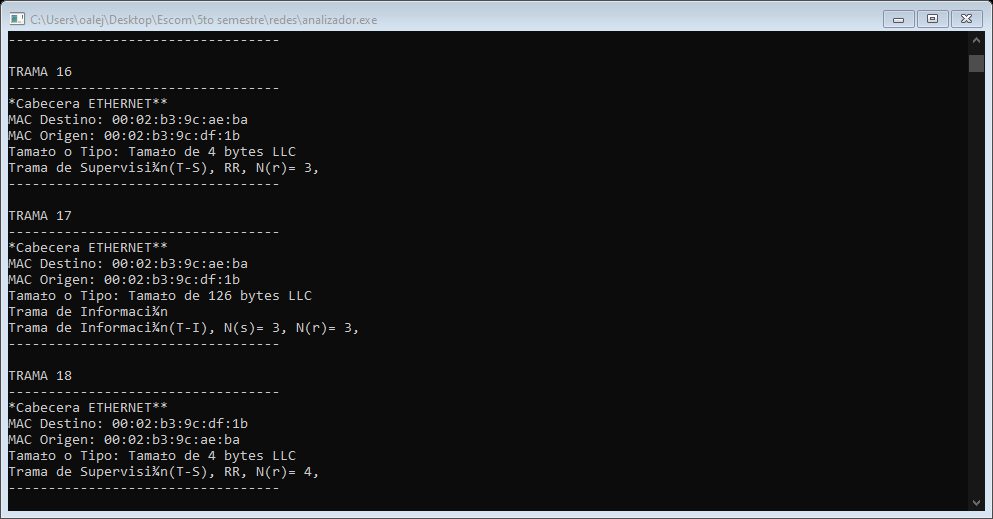
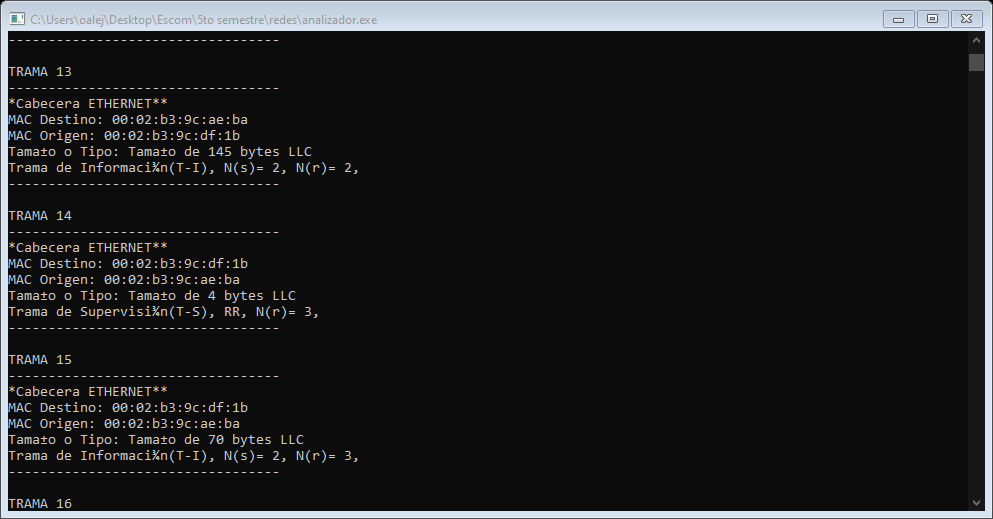
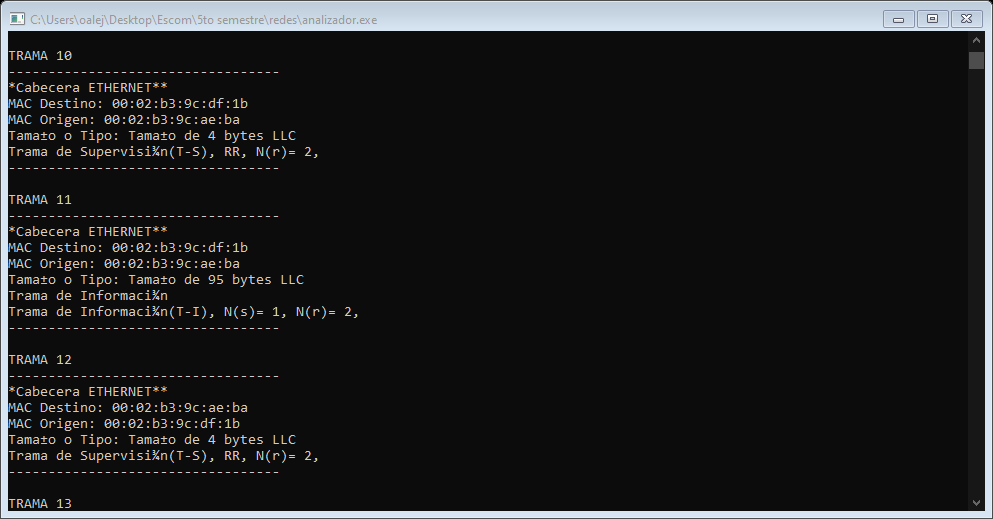
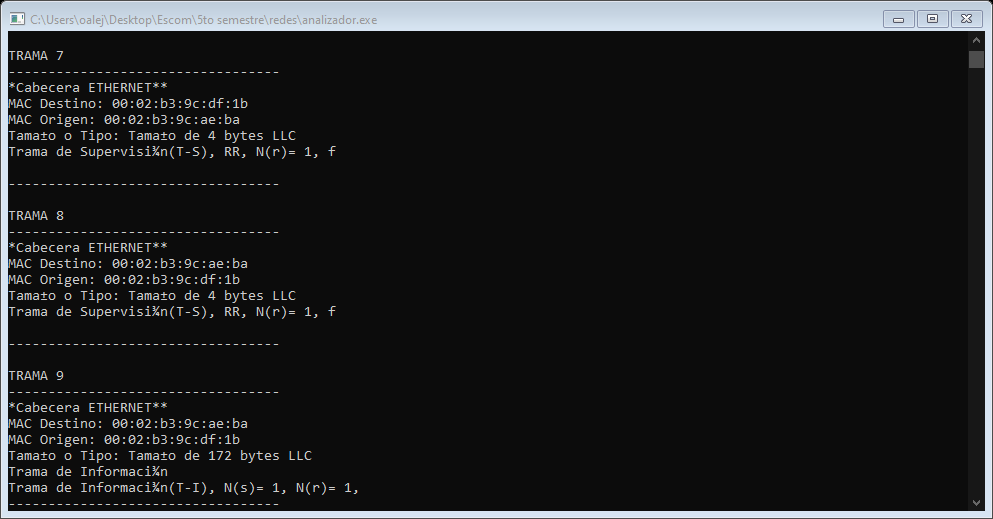
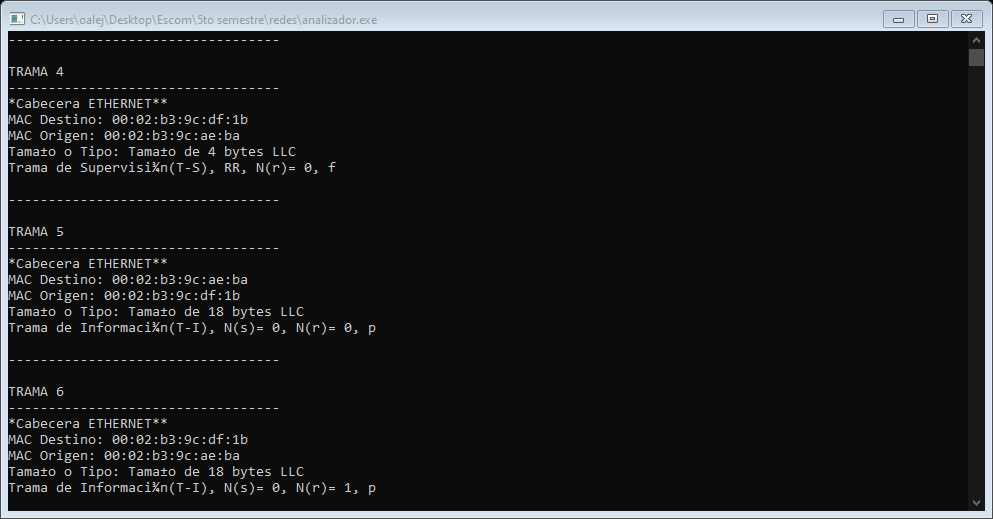
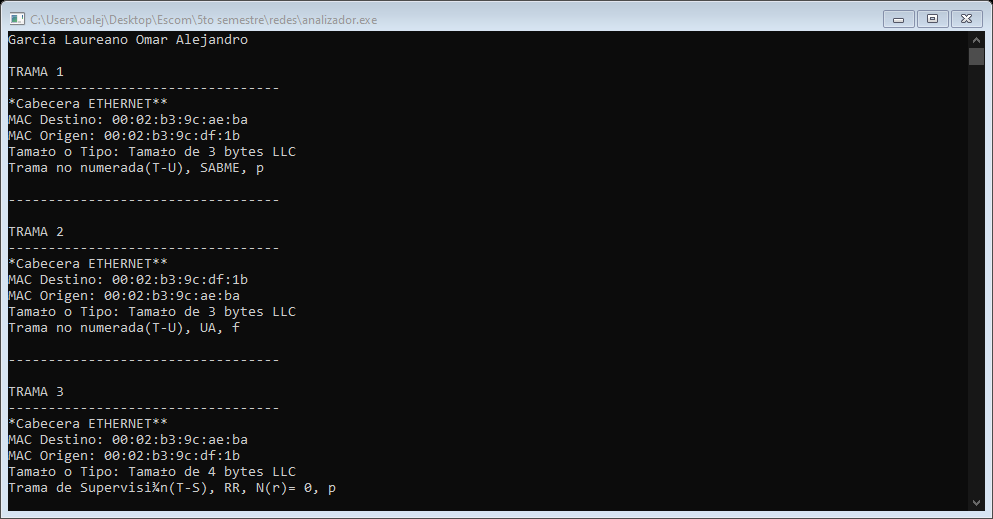
Ahora para determinar si tenemos una trama de información supervisión o no numerada tenemos que hacer una mascara con el byte 16 y el 3 en binario que es 11, esto nos dará 00, 01, 10 o 11 como resultado, donde con la estructura antes podemos saber que 00 y 10 es una trama de información (T-I), 01 es una trama de supervisión (T-S) y 11 corresponde a una trama no numerada (T-U).

Para obtener el valor de los bits S y M que corresponden a trama de Supervisión y No numerada nos tenemos que fijar en diferentes bytes.

Para el primer caso tenemos que fijarnos en el byte 16 de la trama recorrido 2 bits a la derecha y hacer una mascara con 3 (11), lo que nos dará de nuevo 00, 01, 10 o 1 que corresponden a RR, RNR, REJ y SREJ.

En el segundo caso hacemos una mascara con el byte 16 y 16 (10000) pero en este caso, el valor que nos arrojará no es suficiente ya que cambia el significado dependiendo de si es orden o respuesta, para saber esto solo necesitamos comparar el byte 15 con 1 para saber si el bit p/f es 0 o es 1 y en caso de ser 1 vamos ahora a comparar el byte 16 recorrido 2 bits a la derecha y 3 (11) o el byte 16 recorrido 3 bits a la derecha y 28 (11100) y dependiendo del resultado y si es de orden o de respuesta nos dará la respuesta correspondiente a nuestros arreglos (s[ ], UR[ ] y UC[ ]).

Salida del programa:



Código:

1. #include<stdio.h>
3. **void** analizador (unsigned **char** trama[]){
4. **short** i;
5. unsigned **char** s[][5]={"RR", "RNR", "REJ", "SREJ"};
6. unsigned **char** UC[][6]={"UI","SIM","","SARM","UP","","","SABM","DISC","","","SARME","","","","SABME",
7. "SNRM","","","RSET","","","","XID","","","","SNRME","","","","SNRM"};
8. unsigned **char** UR[][5]={"UI","RIM","","DM","","","","","RD","","","","UA","TEST","","",""
9. "","FRMR","","","","","","XID","","","","","TEST","","",""};
10. printf("\*Cabecera ETHERNET\*\*\nMAC Destino: ");
11. **for**(i=0;i<6;i++){
12. **if**(i!=5){
13. printf("%02x:",trama[i]);
14. }**else**{
15. printf("%02x\n",trama[i]);
16. }
17. }
18. printf("MAC Origen: ");
19. **for**(i=6;i<12;i++){
20. **if**(i!=11){
21. printf("%02x:",trama[i]);
22. }**else**{
23. printf("%02x\n",trama[i]);
24. }
25. }
26. printf("Tamaño o Tipo: ");
27. **if**(((trama[12]<<8) + trama[13])<1500)
28. {
29. printf("Tamaño de %d bytes LLC\n", ((trama[12]<<8) + trama[13]));
30. }
31. **else** **if**(((trama[12]<<8) + trama[13])==2048)
32. {
33. printf("Tipo IP\n");
34. }
35. **else** **if**(((trama[12]<<8) + trama[13])==2054)
36. {
37. printf("Tipo ARP\n");
38. }
39. **else**
40. {
41. printf("Otro\n");
42. }
44. **switch**(trama[16]&3){
46. **case** 0: printf("Trama de Información(T-I), N(s)= %d, N(r)= %d, ", (trama[16]>>1), (trama[17]>>1));
47. **if**(trama[17]&1){
48. **if**(trama[15]&1){
49. printf("f\n");
50. }
51. **else**{
52. printf("p\n");
53. }
54. }
55. **break**;
57. **case** 1: printf("Trama de Supervisión(T-S), %s, N(r)= %d, ", s[(trama[16]>>2)&3], trama[17]>>1);
58. **if**(trama[17]&1){
59. **if**(trama[15]&1){
60. printf("f\n");
61. }
62. **else**{
63. printf("p\n");
64. }
65. }
66. **break**;
68. **case** 2: printf("Trama de Información\n");
69. printf("Trama de Información(T-I), N(s)= %d, N(r)= %d, ", (trama[16]>>1), (trama[17]>>1));
70. **if**(trama[17]&1){
71. **if**(trama[15]&1){
72. printf("f\n");
73. }
74. **else**{
75. printf("p\n");
76. }
77. }
78. **break**;
80. **case** 3: printf("Trama no numerada(T-U), ");
81. **if**(trama[16]&16){
82. **if**(trama[15]&1)
83. printf("%s, ", UR[((trama[16]>>2)&3)|((trama[16]>>3)&28)]);
84. **else**
85. printf("%s, ", UC[((trama[16]>>2)&3)|((trama[16]>>3)&28)]);
86. }
87. **if**((trama[16]>>4)&1){
88. **if**(trama[15]&1){
89. printf("f\n");
90. }
91. **else**{
92. printf("p\n");
93. }
94. }
96. **break**;
98. **default**: printf("ERROR");
100. **break**;
102. }

105. }
107. **void** main(){
108. unsigned **char** trama[][200]=
109. {
110. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
111. 0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
112. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
113. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x90,0x6d}, //trama1
114. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,
115. 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
116. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
117. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x54,0x90,0x6d}, //trama2
118. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf0,
119. 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
120. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
121. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0xa3,0x90,0x6d}, //trama3
122. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
123. 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
124. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
125. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xf2,0x90,0x6d}, //trama4
126. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
127. 0x00,0x01,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23,
128. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
129. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0x91,0x6d}, //trama5
130. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x12,0xf0,0xf0,
131. 0x00,0x03,0x0e,0x00,0xff,0xef,0x17,0x81,0xbc,0x05,0x23,0x00,0x7f,0x00,0x23,0x7f,
132. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
133. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x90,0x91,0x6d}, //trama6
134. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
135. 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
136. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
137. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xdf,0x91,0x6d}, //trama7
138. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,
139. 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
140. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
141. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0xac,0x92,0x6d}, //trama8
142. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0xac,0xf0,0xf0,
143. 0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23,
144. 0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
145. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,
146. 0x00,0x77,0x00,0x02,0x50,0x43,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x20,0x50,
147. 0x52,0x4f,0x47,0x52,0x41,0x4d,0x20,0x31,0x2e,0x30,0x00,0x02,0x4d,0x49,0x43,0x52,
148. 0x4f,0x53,0x4f,0x46,0x54,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x53,0x20,0x33,
149. 0x2e,0x30,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x4d,0x31,0x2e,0x32,0x58,0x30,0x30,
150. 0x32,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x32,0x2e,0x31,
151. 0x00,0x02,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x66,0x6f,0x72,0x20,0x57,0x6f,
152. 0x72,0x6b,0x67,0x72,0x6f,0x75,0x70,0x73,0x20,0x33,0x2e,0x31,0x61,0x00,0x02,0x4e,
153. 0x54,0x20,0x4c,0x4d,0x20,0x30,0x2e,0x31,0x32,0x00,0x00,0xfb,0x92,0x6d,0x86,0xdf}, //trama9
154. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
155. 0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
156. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
157. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7b,0x93,0x6d}, //trama10
158. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x5f,0xf0,0xf0,
159. 0x02,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,
160. 0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,
161. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,
162. 0x11,0x05,0x00,0x02,0x02,0x00,0x01,0x00,0x68,0x0b,0x00,0x00,0x00,0x00,0x01,0x00,
163. 0x7f,0x07,0x00,0x80,0x03,0x02,0x00,0x00,0x00,0xe5,0xfe,0x29,0x25,0x7c,0xc2,0x01,
164. 0x2c,0x01,0x08,0x08,0x00,0x7f,0x07,0x00,0x80,0x32,0x3e,0xb9,0x3d,0x00,0xca,0x93}, //trama11
165. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,
166. 0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
167. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
168. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x94,0x6d}, //trama12
169. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x91,0xf0,0xf0,
170. 0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,
171. 0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x00,0x00,
172. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,
173. 0x0d,0x75,0x00,0x5d,0x00,0x68,0x0b,0x02,0x00,0x00,0x00,0x7f,0x07,0x00,0x80,0x00,
174. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x45,
175. 0x53,0x43,0x4f,0x4d,0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2e,0x30,
176. 0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2e,0x30,0x00,0x04,0xff,0x00,
177. 0x00,0x00,0x02,0x00,0x02,0x00,0x17,0x00,0x20,0x00,0x5c,0x5c,0x50,0x52,0x4f,0x47,
178. 0x59,0x44,0x45,0x53,0x41,0x5c,0x49,0x50,0x43,0x24,0x00,0x49,0x50,0x43,0x00,0x00}, //trama13
179. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
180. 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
181. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
182. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x32,0x95,0x6d}, //trama14
183. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x46,0xf0,0xf0,
184. 0x04,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,
185. 0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x90,0x00,0x00,0x00,0x00,0x00,0x00,
186. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x82,0x09,
187. 0x03,0x75,0x00,0x29,0x00,0x00,0x00,0x00,0x00,0x02,0xff,0x00,0x00,0x00,0x04,0x00,
188. 0x49,0x50,0x43,0x00,0x00,0x81,0x95,0x6d,0x86,0xcb,0x94,0x6d,0x86,0x0d,0x09,0x0e}, //trama15
189. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,
190. 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
191. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
192. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x96,0x6d}, //trama16
193. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0,
194. 0x06,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,
195. 0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
196. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x82,0x0a,
197. 0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x13,0x00,
198. 0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x00,0x5c,
199. 0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x57,0x72,
200. 0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x01,0x00,
201. 0x00,0x10,0xff,0xff,0xff,0xff,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0x6f,0x96,0x6d}, //trama17
202. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
203. 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
204. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
205. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xbe,0x96,0x6d}, //trama18
206. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,
207. 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
208. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
209. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x5d,0x97,0x6d}, //trama19
210. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0,
211. 0x08,0x08,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,
212. 0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
213. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x02,0x0b,
214. 0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x13,0x00,
215. 0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x00,0x5c,
216. 0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x57,0x72,
217. 0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x01,0x00,
218. 0x00,0x10,0x00,0x00,0x00,0x80,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0xac,0x97,0x6d}, //trama20
219. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
220. 0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
221. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
222. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfb,0x97,0x6d}, //trama21
223. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,
224. 0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
225. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
226. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x4a,0x98,0x6d}, //trama22
227. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
228. 0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23,
229. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
230. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x99,0x98,0x6d}, //trama23
231. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
232. 0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
233. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
234. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x45,0x99,0x6d}, //trama24
235. {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xac,0x44,0x4d,0x02,0x00,0x8b,0xf0,0xf0,
236. 0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x42,0x34,0x20,
237. 0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x1b,0x49,0x42,0x4d,
238. 0x53,0x45,0x52,0x56,0x45,0x52,0x20,0x20,0x20,0x20,0x20,0x20,0x00,0xff,0x53,0x4d,
239. 0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
240. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x11,0x00,0x00,
241. 0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xe8,0x03,0x00,0x00,0x00,0x00,
242. 0x00,0x00,0x00,0x00,0x06,0x00,0x56,0x00,0x03,0x00,0x01,0x00,0x01,0x00,0x02,0x00,
243. 0x17,0x00,0x5c,0x4d,0x41,0x49,0x4c,0x53,0x4c,0x4f,0x54,0x5c,0x42,0x52,0x4f,0x57,
244. 0x53,0x45,0x00,0x09,0x04,0x33,0x17,0x00,0x00,0x00,0x9b,0x99,0x6d,0x86,0x99,0x98}, //trama25
245. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0,
246. 0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23,
247. 0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,
248. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x01,0x50,
249. 0x00,0x00,0x00,0x45,0xf1,0x99,0x6d,0x86,0x45,0x99,0x6d,0x86,0x1f,0x09,0x52,0x5b}, //trama26
250. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x35,0xf0,0xf0,
251. 0x0a,0x0e,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,
252. 0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x80,0x01,0x00,0x00,0x00,0x00,0x00,
253. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x01,0x50,
254. 0x00,0x00,0x00,0x00,0x40,0x9a,0x6d,0x86,0x9b,0x99,0x6d,0x86,0x20,0x09,0x75,0x5b}, //trama27
255. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
256. 0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23,
257. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
258. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x8f,0x9a,0x6d}, //trama28
259. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
260. 0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
261. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
262. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xde,0x9a,0x6d}, //trama29
263. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
264. 0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0x23,
265. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
266. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x9b,0x6d}, //trama30
267. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
268. 0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
269. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
270. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x9b,0x6d}, //trama31
271. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
272. 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
273. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
274. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xcb,0x9b,0x6d}, //trama32
275. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,
276. 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
277. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
278. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x77,0x9c,0x6d},
280. {0xff,0xff,0xff,0xff,0xff,0xff,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x06,0x00,0x04,
281. 0x08,0x00,0x06,0x04,0x00,0x01,0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,
282. 0x00,0x00,0x00,0x00,0x00,0x00,0x94,0xcc,0x39,0xfe },                          /\*Trama a \*/
284. {0x00,0x23,0x8b,0x46,0xe9,0xad,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x08,0x06,0x00,0x01,  /\*TRAMA b \*/
285. 0x08,0x00,0x06,0x04,0x00,0x02,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x94,0xcc,0x39,0xfe,
286. 0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,0x00,0x00,0x00,0x00,0x00,0x00,
287. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },
289. {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x00,0x46,0x00, /\* TRAMA c \*/
290. 0x80,0x42,0x04,0x55,0x34,0x11,0x80,0x11,0x6b,0xf0,0x94,0xcc,0x39,0xcb,0x94,0xcc,
291. 0x67,0x02,0xaa,0xbb,0xcc,0xdd,0x04,0x0c,0x00,0x35,0x00,0x2e,0x85,0x7c,0xe2,0x1a,
292. 0x01,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x77,0x77,0x77,0x03,0x69,
293. 0x73,0x63,0x05,0x65,0x73,0x63,0x6f,0x6d,0x03,0x69,0x70,0x6e,0x02,0x6d,0x78,0x00,
294. 0x00,0x1c,0x00,0x01}
295. };
297. **short** i;
298. printf("\nGarcia Laureano Omar Alejandro\n");
299. **for**(i=0;i<32;i++){
300. printf("\nTRAMA %d \n", i+1);
301. printf("----------------------------------\n");
302. analizador(trama[i]);
303. printf("\n----------------------------------\n");
304. }
305. printf("\nTRAMA a \n");
306. printf("----------------------------------\n");
307. analizador(trama[32]);
308. printf("\n----------------------------------\n");
309. printf("\nTRAMA b \n");
310. printf("----------------------------------\n");
311. analizador(trama[33]);
312. printf("\n----------------------------------\n");
313. printf("\nTRAMA c \n");
314. printf("----------------------------------\n");
315. analizador(trama[34]);
316. printf("\n----------------------------------\n");
317. }