Imagen que contiene Icono

Descripción generada automáticamenteDiagrama, Forma

Descripción generada automáticamente

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Redes de Computadoras

**“Analizador de tramas”**

**Versión 1.- LLC**

Alumno:

Hernández Rodríguez Armando Giovanni

Profesora:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

Grupo: 2CM15

Entrega: 9 noviembre 2021

**Introducción al protocolo LLC**

**Definición**

El control del enlace lógico (Logic Link Control) es el protocolo de enlace de las redes de área local, descrito en el estándar IEEE 802.2. LLC está basado en HDLC con formato de trama similar, además pueden estar orientados a conexión o sin conexión.

**Cabecera**

La cabecera LLC incluye dos campos de dirección de 8 bit, llamados Service Access Points o SAPs. El primer campo de 8 bits es el SAP destino y el siguiente el SAP origen. Luego se tiene un campo de control que dependiendo del tipo de trama puede ser de 8 o 16 bits. Y al último se tiene el campo de información.

**Diagrama

Descripción generada automáticamente**

**Estructura de los campos de control de T-I, T-S y T-U**

Imagen que contiene Tabla

Descripción generada automáticamenteAl igual que en HDLC, LLC tiene tramas de información, supervisión y no numeradas.

Tramas I: Se utilizan para transmisión de datos y reconocimiento.

Tramas S: Se utilizan para el control de flujo y de control de errores

Tramas U: Se utilizan para el establecimiento, mantenimiento y terminación de conexión

**Bit P/F**

Cada trama tiene un bit P/F (sondeo / bit final). El bit de sondeo es parte de una trama de comando, mientras que el bit final solo ocurre en las tramas de respuesta. En general, los bits P/F tienen el estado 0. Solamente tiene significado cuando está activo.

**Campo SAPo**

El LSB del SAPo permite saber si se trata de un comando o respuesta. Si C/R es 0 se trata de un comando, por otro lado si C/R es 1 entonces es una respuesta.

**Explicación breve de los procesos de enmascaramiento**

**¿Como obtener el valor de los bits S?**

Para obtener el valor de los bits S en la trama de supervisión, se utilizó un corrimiento a la derecha y un AND. Para ello, primero se ubicó la posición del arreglo unsigned char de la trama en donde se encontraban los bits S, en este caso que corresponde al campo de control (CC) de la cabecera LLC

**t[16] t[17]**

Tabla

Descripción generada automáticamente

Luego a se hizo el corrimiento de dos posiciones a la derecha y se aplicó la operación AND con el número 3 para apagar los demás bits (para que solo quede 000000SS). En la siguiente representación de los datos en bits se puede observar mejor:



Finalmente se toma el valor de los bits S para obtener el comando correspondiente del arreglo

**¿Como obtener el valor de los bits M?**

Ahora bien, para obtener el valor de los bits M en la trama no numerada, se utilizaron corrimientos a la derecha, operaciones AND y OR. Como en la trama anterior, primero se ubicó la posición del arreglo de la trama , la cual es la misma

**t[16]**

**Diagrama

Descripción generada automáticamente con confianza media**

Luego a se hizo el corrimiento de dos posiciones a la derecha y se aplicó la operación AND con el número 3 para apagar los demás bits (para que solo quede 000000MM). Este resultado se le concatenó (o se hizo un OR) con el corrimiento de tres posiciones a la derecha de AND con el número 28 (para que quede 000MMM00, con el OR se obtendrá 000MMMMM). En la siguiente representación de los datos en bits se puede observar mejor:



Finalmente se toma el valor de los bits M para obtener el comando o respuesta correspondiente de los arreglos o según sea el caso.

**Memoria del Programa**

**Texto

Descripción generada automáticamente**

**Captura de pantalla de la salida del programa (imprime tu nombre completo al inicio de la ejecución)**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Conclusiones**

Anteriormente no había trabajado a nivel de bits, sin embargo después de practicar con las operaciones bitwise (XNOR, AND, OR, complemento, corrimientos) he comprobado que se pueden elaborar algoritmos que consideren el gasto de memoria. Como programadores, en ocasiones olvidamos la memoria que podría llegar a ocupar nuestro programa, y sólo nos centramos en que funcione aunque se ocupen muchas variables y arreglos de forma indistinta, sin embargo con el desarrollo de esta práctica me quedó claro la importancia que tiene el análisis antes de realizar cualquier programa, pensar en las variables que se ocuparan y asignar el tipo de dato adecuado.

Pienso que los operadores binarios ofrecen muchas ventajas en el programa pues permiten hacer operaciones que si bien podrían hacerse con multiplicaciones, divisiones, sumas o restas, al realizar operaciones bit a bit se consume mucho menos memoria, además son increíblemente simples y generalmente más rápidas.

Finalmente la práctica me resultó interesante, me pareció retadora ya que me propuse usar las variables que menos se pudieran, el claro reflejo de esto es la memoria del programa en donde solo se ocuparon dos variables. Gracias a la explicación y análisis sobre cómo debía funcionar el programa visto en clase, pude comprender e implementar lo que se requería.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268 | #include<stdio.h>  **Código Fuente**  unsigned char i = 0x00;  void analizaLLC(unsigned char T[]){  char SS[][5] = {"RR", "RNR", "REJ", "SREJ"};  char UC[][6] = {"UI", "SIM", "-", "SARM", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "-", "SABME", "SNRM", "-", "-", "RSET", "-", "-", "-", "XID", "-", "-", "-", "SNRME"}; *// comandos - p*  char UR[][6] = {"UI", "RIM", "-", "DM", "-", "-", "-", "-", "RD", "-", "-", "-", "UA", "-", "-", "-", "-", "FRMR", "-", "-", "-", "-", "-", "XID"}; *// respuestas - f*  printf("\n\n..:::Cabecera LLC:::.."); *//En LLC solo hay SABME T-U 1byte , T-S y T-I 2bytes*  switch(T[16] & 3){ *//0000 0011 -> 3 xxxx xxxx & 0000 0011 -> {0, 1, 2, 3}={00-T-I, 01-T-S, 10 -T-I, 11-TU}*  case 0:  case 2: *//T-I*  printf("\nT-I, N(s)=%d, N(r)=%d",T[16]>>1, T[17]>>1);  if(T[17]&1){  if(T[15]&1){ printf(", -f\n"); }*//LSB SAPo*  else{ printf(", -p\n"); }  }  break;  case 1: *//T-S*  printf("\nT-S: %s, N(r)=%d", SS[(T[16]>>2) & 3], T[17]>>1);  *//printf("\nSS: %s", SS[(T[16]>>2) & 3]); // 0xxxxss & 0000 0011 = 0000 00ss*  if(T[17]&1){ *// p/f encendido?*  if(T[15]&1){ printf(", -f\n"); } *//LSB SAPo*  else{ printf(", -p\n"); }  }  break;  case 3: *//T-U MMMx MM11 T[16]*  *//printf("\nT-U");*  *//((T[16] >> 2) & 3) | ((T[16] >> 3) & 28)*  *//((T[16] >> 2) & 3) | ((T[16] >> 3) << 2) {0, 1, 2,...,31}*  *//printf("\nMMMMM: %s", ((T[16] >> 2) & 3) | ((T[16] >> 3) & 28));*  if(T[16]&16){*// p-f = 1?*  if(T[15]&1){ printf("\nT-U: %s -f\n", UR[((T[16] >> 2) & 3) | ((T[16] >> 3) & 28)] ); }*//LSB SAPo*  else{ printf("\nT-U: %s -p\n", UC[((T[16] >> 2) & 3) | ((T[16] >> 3) & 28)] ); }  }  break;  }  }  void analizaTrama(unsigned char t[]){  printf("\n-------------------------------------------------------------------------------------");  printf("\n\n\tTrama: %d\n", i+1);  printf("\n..:::Cabecera Ethernet:::..\n");  printf("\nMAC Destino: %.2x:%.2x:%.2x:%.2x:%.2x:%.2x", t[0], t[1], t[2], t[3], t[4], t[5]);  printf("\nMAC Origen: %.2x:%.2x:%.2x:%.2x:%.2x:%.2x", t[6], t[7], t[8], t[9], t[10], t[11]);    unsigned short int tot = (t[12]<<8) | t[13]; *// 2bytes*  if(tot<1500){  printf("\nTama%co de cabecera LLC: %d bytes\n", 164, tot);  analizaLLC(t);  }  else{  if(tot == 2048){  printf("\nTipo IP\n");  }  else if(tot == 2054){  printf("\nTipo ARP\n");  }  else{  printf("\nTipo: %.2x.%.2x\n", t[12], t[13]);  }  }  }  int main(){  printf("\n\t<<<Escuela Superior de C%cmputo>>>\nElaborado por: Hern%cndez Rodr%cguez Armando Giovanni\n", 162, 160, 161);    unsigned char t[][256]=  { *// 16 columnas x fila*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,  0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x90,0x6d}, *//trama1*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,  0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x54,0x90,0x6d}, *//trama2*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf0,  0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0xa3,0x90,0x6d}, *//trama3*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xf2,0x90,0x6d}, *//trama4*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,  0x00,0x01,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x41,0x91,0x6d}, *//trama5*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x12,0xf0,0xf0,  0x00,0x03,0x0e,0x00,0xff,0xef,0x17,0x81,0xbc,0x05,0x23,0x00,0x7f,0x00,0x23,0x7f,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x90,0x91,0x6d}, *//trama6*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xdf,0x91,0x6d}, *//trama7*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,  0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0xac,0x92,0x6d}, *//trama8*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0xac,0xf0,0xf0,  0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23,  0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,  0x00,0x77,0x00,0x02,0x50,0x43,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x20,0x50,  0x52,0x4f,0x47,0x52,0x41,0x4d,0x20,0x31,0x2e,0x30,0x00,0x02,0x4d,0x49,0x43,0x52,  0x4f,0x53,0x4f,0x46,0x54,0x20,0x4e,0x45,0x54,0x57,0x4f,0x52,0x4b,0x53,0x20,0x33,  0x2e,0x30,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x4d,0x31,0x2e,0x32,0x58,0x30,0x30,  0x32,0x00,0x02,0x44,0x4f,0x53,0x20,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x32,0x2e,0x31,  0x00,0x02,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x66,0x6f,0x72,0x20,0x57,0x6f,  0x72,0x6b,0x67,0x72,0x6f,0x75,0x70,0x73,0x20,0x33,0x2e,0x31,0x61,0x00,0x02,0x4e,  0x54,0x20,0x4c,0x4d,0x20,0x30,0x2e,0x31,0x32,0x00,0x00,0xfb,0x92,0x6d,0x86,0xdf}, *//trama9*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7b,0x93,0x6d}, *//trama10*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x5f,0xf0,0xf0,  0x02,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,  0xff,0x53,0x4d,0x42,0x72,0x00,0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,  0x11,0x05,0x00,0x02,0x02,0x00,0x01,0x00,0x68,0x0b,0x00,0x00,0x00,0x00,0x01,0x00,  0x7f,0x07,0x00,0x80,0x03,0x02,0x00,0x00,0x00,0xe5,0xfe,0x29,0x25,0x7c,0xc2,0x01,  0x2c,0x01,0x08,0x08,0x00,0x7f,0x07,0x00,0x80,0x32,0x3e,0xb9,0x3d,0x00,0xca,0x93}, *//trama11*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,  0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x94,0x6d}, *//trama12*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x91,0xf0,0xf0,  0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,  0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x82,0x09,  0x0d,0x75,0x00,0x5d,0x00,0x68,0x0b,0x02,0x00,0x00,0x00,0x7f,0x07,0x00,0x80,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x45,  0x53,0x43,0x4f,0x4d,0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2e,0x30,  0x00,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x34,0x2e,0x30,0x00,0x04,0xff,0x00,  0x00,0x00,0x02,0x00,0x02,0x00,0x17,0x00,0x20,0x00,0x5c,0x5c,0x50,0x52,0x4f,0x47,  0x59,0x44,0x45,0x53,0x41,0x5c,0x49,0x50,0x43,0x24,0x00,0x49,0x50,0x43,0x00,0x00}, *//trama13*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x32,0x95,0x6d}, *//trama14*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x46,0xf0,0xf0,  0x04,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,  0xff,0x53,0x4d,0x42,0x73,0x00,0x00,0x00,0x00,0x90,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x82,0x09,  0x03,0x75,0x00,0x29,0x00,0x00,0x00,0x00,0x00,0x02,0xff,0x00,0x00,0x00,0x04,0x00,  0x49,0x50,0x43,0x00,0x00,0x81,0x95,0x6d,0x86,0xcb,0x94,0x6d,0x86,0x0d,0x09,0x0e}, *//trama15*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,  0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x96,0x6d}, *//trama16*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0,  0x06,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,  0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x82,0x0a,  0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x13,0x00,  0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x00,0x5c,  0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x57,0x72,  0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x01,0x00,  0x00,0x10,0xff,0xff,0xff,0xff,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0x6f,0x96,0x6d}, *//trama17*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xbe,0x96,0x6d}, *//trama18*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,  0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x5d,0x97,0x6d}, *//trama19*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0,  0x08,0x08,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23,  0xff,0x53,0x4d,0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x02,0x0b,  0x0e,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x88,0x13,0x00,  0x00,0x00,0x00,0x20,0x00,0x4c,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x00,0x5c,  0x50,0x49,0x50,0x45,0x5c,0x4c,0x41,0x4e,0x4d,0x41,0x4e,0x00,0x68,0x00,0x57,0x72,  0x4c,0x65,0x68,0x44,0x7a,0x00,0x42,0x31,0x36,0x42,0x42,0x44,0x7a,0x00,0x01,0x00,  0x00,0x10,0x00,0x00,0x00,0x80,0x45,0x53,0x43,0x4f,0x4d,0x00,0x00,0xac,0x97,0x6d}, *//trama20*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfb,0x97,0x6d}, *//trama21*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,  0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x4a,0x98,0x6d}, *//trama22*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,  0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x99,0x98,0x6d}, *//trama23*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x45,0x99,0x6d}, *//trama24*  {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xac,0x44,0x4d,0x02,0x00,0x8b,0xf0,0xf0,  0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x42,0x34,0x20,  0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x1b,0x49,0x42,0x4d,  0x53,0x45,0x52,0x56,0x45,0x52,0x20,0x20,0x20,0x20,0x20,0x20,0x00,0xff,0x53,0x4d,  0x42,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x11,0x00,0x00,  0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xe8,0x03,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x06,0x00,0x56,0x00,0x03,0x00,0x01,0x00,0x01,0x00,0x02,0x00,  0x17,0x00,0x5c,0x4d,0x41,0x49,0x4c,0x53,0x4c,0x4f,0x54,0x5c,0x42,0x52,0x4f,0x57,  0x53,0x45,0x00,0x09,0x04,0x33,0x17,0x00,0x00,0x00,0x9b,0x99,0x6d,0x86,0x99,0x98}, *//trama25*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0,  0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23,  0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x01,0x50,  0x00,0x00,0x00,0x45,0xf1,0x99,0x6d,0x86,0x45,0x99,0x6d,0x86,0x1f,0x09,0x52,0x5b}, *//trama26*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x35,0xf0,0xf0,  0x0a,0x0e,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x23,0x7f,  0xff,0x53,0x4d,0x42,0x71,0x00,0x00,0x00,0x00,0x80,0x01,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xc0,0x00,0x00,0x00,0x00,0x01,0x50,  0x00,0x00,0x00,0x00,0x40,0x9a,0x6d,0x86,0x9b,0x99,0x6d,0x86,0x20,0x09,0x75,0x5b}, *//trama27*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,  0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x8f,0x9a,0x6d}, *//trama28*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xde,0x9a,0x6d}, *//trama29*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,  0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0x23,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x2d,0x9b,0x6d}, *//trama30*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,  0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x9b,0x6d}, *//trama31*  {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,  0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xcb,0x9b,0x6d}, *//trama32*  {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,  0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x77,0x9c,0x6d}, *//trama33*    {0xff,0xff,0xff,0xff,0xff,0xff,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x06,0x00,0x04,  0x08,0x00,0x06,0x04,0x00,0x01,0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,  0x00,0x00,0x00,0x00,0x00,0x00,0x94,0xcc,0x39,0xfe }, *///TRAMA a*  {0x00,0x23,0x8b,0x46,0xe9,0xad,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x08,0x06,0x00,0x01, *//TRAMA b*  0x08,0x00,0x06,0x04,0x00,0x02,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x94,0xcc,0x39,0xfe,  0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,0x00,0x00,0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 },    {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x00,0x46,0x00, *// TRAMA c*  0x80,0x42,0x04,0x55,0x34,0x11,0x80,0x11,0x6b,0xf0,0x94,0xcc,0x39,0xcb,0x94,0xcc,  0x67,0x02,0xaa,0xbb,0xcc,0xdd,0x04,0x0c,0x00,0x35,0x00,0x2e,0x85,0x7c,0xe2,0x1a,  0x01,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x77,0x77,0x77,0x03,0x69,  0x73,0x63,0x05,0x65,0x73,0x63,0x6f,0x6d,0x03,0x69,0x70,0x6e,0x02,0x6d,0x78,0x00,  0x00,0x1c,0x00,0x01}  };    for(i=0; i<36; i++){  analizaTrama(t[i]);  }  return 0;  } |