

**Instituto**

**Politécnico**

**Nacional**

**Escuela Superior de Cómputo**

**Analizador de Tramas**

**Materia:**

Redes de computadoras

**Grupo:**

2CM10

**Grupo:**

Cortez Duarte Nidia Asunción

**Integrantes:**

Castro Cruces Jorge Eduardo

**Fecha:**

Martes, mayo 11, 2020

Código

1. #include<stdio.h>
3. **void** CheckSumIP(unsigned **char** Trama[]){
4. **int** i;
5. unsigned **int** CheckSum=0;
6. **for**(i=14; i<(13+((Trama[14]&15)\*4)); i++){
7. **if**(i==24) i++;
8. **else** CheckSum+=Trama[i++]<<8 | Trama[i];
9. }
10. CheckSum=((CheckSum>>16) + (CheckSum & 0xffff)) ^ 0xffff;
12. **if**(((Trama[24]<<8) | Trama[25]) == (CheckSum)) printf("\nCheckSum: Correct -> 0x%x", CheckSum);
13. **else** printf("\nCheckSum: Incorrect -> 0x%x", CheckSum);
14. }
16. **void** CheckSumTCP(unsigned **char** Trama[]){
17. **int** i;
18. unsigned **int** CheckSum=0;
20. **for**(i=26; i<34; i++)
21. CheckSum+=Trama[i++]<<8 | Trama[i];
22. CheckSum+=Trama[23];
23. CheckSum+=(Trama[((Trama[14]&15)\*4)+26]>>4)\*4;
25. **for**(i=((Trama[14]&15)\*4)+14; i<((((Trama[14]&15)\*4)+14)+(Trama[((Trama[14]&15)\*4)+26]>>4)\*4)-1; i++){ //for(i=Inicio Cabecera TCP; i<Fin Cabecera TCP; i++)
26. **if**(i==((Trama[14]&15)\*4)+30) i++;
27. **else** CheckSum+=Trama[i++]<<8 | Trama[i];
28. }
29. CheckSum=((CheckSum>>16) + (CheckSum & 0xffff)) ^ 0xffff;
31. **if**(((Trama[((Trama[14]&15)\*4)+30]<<8) | Trama[((Trama[14]&15)\*4)+31]) == (CheckSum)) printf("\nCheckSum: Correct -> 0x%x", CheckSum);
32. **else** printf("\nCheckSum: Incorrect -> 0x%x", CheckSum);
33. }
35. **void** LLC(unsigned **char** Trama[]){
36. unsigned **char** S[][5]={"RR","RNR", "REJ", "SREJ"};
37. unsigned **char** UC[][6]={"UI","SIM", "-", "SARM", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "-", "SABME", "SNRM", "-", "-", "RSET", "-", "-", "XID", "-", "-", "-", "SNRME"};
38. unsigned **char** UR[][6]={"UI","RIM", "-", "DM", "-", "-", "-", "SABM", "RD", "-", "-", "-", "UA", "-", "-", "-", "-", "FRMR", "-", "-", "-", "-", "XID", "-", "-", "-", "-"};
40. printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
41. printf("\tLLC\n");
42. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");
44. **switch**(Trama[16] & 3){
45. **case** 0: //T-I
46. printf("\nTrama-Informacion:\n\tN(s): %d\n\tN(r): %d", Trama[16]>>1, Trama[17]>>1);
47. **if**(Trama[17] & 1){
48. **if**(Trama[15] & 1)
49. printf("\nf");
50. **else**
51. printf("\np");
52. }
53. **break**;
54. **case** 1: //T-S
55. printf("\nTrama-Supervision (%s):\n\tN(r): %d", S[(Trama[16]>>2) & 3], Trama[17]>>1);
56. **break**;
57. **case** 2: //T-I
58. printf("\nTrama-Informacion:\n\tN(s): %d\n\tN(r): %d", Trama[16]>>1, Trama[17]>>1);
59. **if**(Trama[17] & 1){
60. **if**(Trama[15] & 1)
61. printf("\nf");
62. **else**
63. printf("\np");
64. }
65. **break**;
66. **case** 3: //T-U
67. printf("\nTrama-No Numerada:");
68. **if**(Trama[16] & 16){ //pf=1?
69. **if**(Trama[15] & 1) //LSB SAPo=1, es respuesta
70. printf("T-U (%s), f", UR[((Trama[16]>>2) & 3) | ((Trama[16]>>3) & 28)]);
71. **else** //es comando
72. printf("T-U (%s), p", UC[((Trama[16]>>2)&3) | ((Trama[16]>>3)&28)]);
73. }
74. **break**;
75. }
76. }
78. **void** ICMP(unsigned **char** Trama[]){
79. **int** i;
80. printf("\n\n\*\*\*\*\*\*\*\*\*\*\n");
81. printf("   ICMP");
82. printf("\n\*\*\*\*\*\*\*\*\*\*");
84. **if**((Trama[((Trama[14]&15)\*4)+14]==8) && (Trama[((Trama[14]&15)\*4)+15]==0)){
85. printf("\nSolicitud ECO");
86. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]);
87. printf("\nIdentificador: %d", Trama[((Trama[14]&15)\*4)+18] + Trama[((Trama[14]&15)\*4)+19]);
88. printf("\nNo. Secuencia: %d", Trama[((Trama[14]&15)\*4)+20] + Trama[((Trama[14]&15)\*4)+21]);
89. printf("\nDatos opcionales: ");
90. **for**(i=((Trama[14]&15)\*4)+22; i<74; i++)
91. printf("0x%x, ", Trama[i]);
92. }
93. **else** **if**((Trama[((Trama[14]&15)\*4)+14]==0) && (Trama[((Trama[14]&15)\*4)+15]==0)){
94. printf("\nRespuesta ECO");
95. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]);
96. printf("\nIdentificador: %d", Trama[((Trama[14]&15)\*4)+18] + Trama[((Trama[14]&15)\*4)+19]);
97. printf("\nNo. Secuencia: %d", Trama[((Trama[14]&15)\*4)+20] + Trama[((Trama[14]&15)\*4)+21]);
98. printf("\nDatos opcionales: ");
99. **for**(i=((Trama[14]&15)\*4)+22; i<74; i++)
100. printf("0x%x, ", Trama[i]);
101. }
102. **else** **if**((Trama[((Trama[14]&15)\*4)+14]==3) && ((Trama[((Trama[14]&15)\*4)+15]>=0) || (Trama[((Trama[14]&15)\*4)+15]<=3))){
103. **if**(Trama[((Trama[14]&15)\*4)+15]==0) printf("\n0 -> Red Inalcanzable");
104. **else** **if**(Trama[((Trama[14]&15)\*4)+15]==0) printf("\n1 -> Host Inalcanzable");
105. **else** **if**(Trama[((Trama[14]&15)\*4)+15]==0) printf("\n2 -> Protocolo Inalcanzable");
106. **else** **if**(Trama[((Trama[14]&15)\*4)+15]==0) printf("\n3 -> Puerto Inalcanzable");
107. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]);
108. }
109. **else** **if**((Trama[((Trama[14]&15)\*4)+14]==3) && ((Trama[((Trama[14]&15)\*4)+15]>=4) || (Trama[((Trama[14]&15)\*4)+15]<=13))){
110. printf("\nDestino inalcanzable");
111. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]);
112. }
113. **else** **if**((Trama[((Trama[14]&15)\*4)+14]==11) && ((Trama[((Trama[14]&15)\*4)+15]==0) || (Trama[((Trama[14]&15)\*4)+15]==1))){
114. printf("\nTiempo extendido");
115. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]);
116. }
117. **else** printf("\nType & Code: Otro");
118. }
120. **void** TCP(unsigned **char** Trama[]){
121. **int** i;
122. printf("\n\n\*\*\*\*\*\*\*\*\*\*\n");
123. printf("   TCP");
124. printf("\n\*\*\*\*\*\*\*\*\*\*");
126. printf("\nSource port: ");
127. **switch**(Trama[((Trama[14]&15)\*4)+14]<<8 | Trama[((Trama[14]&15)\*4)+15]){
128. **case** 7:
129. printf("ECHO");
130. **break**;
131. **case** 19:
132. printf("CHARGEN");
133. **break**;
134. **case** 20:
135. printf("FTP-DATA");
136. **break**;
137. **case** 21:
138. printf("FTP-CONTROL");
139. **break**;
140. **case** 22:
141. printf("SSH");
142. **break**;
143. **case** 23:
144. printf("TELNET");
145. **break**;
146. **case** 25:
147. printf("SMTP");
148. **break**;
149. **case** 53:
150. printf("DOMAIN");
151. **break**;
152. **case** 79:
153. printf("FINGER");
154. **break**;
155. **case** 80:
156. printf("HTTP");
157. **break**;
158. **case** 110:
159. printf("POP3");
160. **break**;
161. **case** 111:
162. printf("SUNRPC");
163. **break**;
164. **case** 119:
165. printf("NNTP");
166. **break**;
167. **case** 139:
168. printf("NETBIOS-SSN");
169. **break**;
170. **case** 143:
171. printf("IMAP");
172. **break**;
173. **case** 179:
174. printf("BGP");
175. **break**;
176. **case** 389:
177. printf("LDAP");
178. **break**;
179. **case** 443:
180. printf("HTTPS (SSL)");
181. **break**;
182. **case** 445:
183. printf("MICROSOFT-DS");
184. **break**;
185. **case** 1080:
186. printf("SOCKS");
187. **break**;
188. **default**:
189. printf("OTHER");
190. **break**;
191. }
193. printf("\nDestination port: ");
194. **switch**(Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]){
195. **case** 7:
196. printf("ECHO");
197. **break**;
198. **case** 19:
199. printf("CHARGEN");
200. **break**;
201. **case** 20:
202. printf("FTP-DATA");
203. **break**;
204. **case** 21:
205. printf("FTP-CONTROL");
206. **break**;
207. **case** 22:
208. printf("SSH");
209. **break**;
210. **case** 23:
211. printf("TELNET");
212. **break**;
213. **case** 25:
214. printf("SMTP");
215. **break**;
216. **case** 53:
217. printf("DOMAIN");
218. **break**;
219. **case** 79:
220. printf("FINGER");
221. **break**;
222. **case** 80:
223. printf("HTTP");
224. **break**;
225. **case** 110:
226. printf("POP3");
227. **break**;
228. **case** 111:
229. printf("SUNRPC");
230. **break**;
231. **case** 119:
232. printf("NNTP");
233. **break**;
234. **case** 139:
235. printf("NETBIOS-SSN");
236. **break**;
237. **case** 143:
238. printf("IMAP");
239. **break**;
240. **case** 179:
241. printf("BGP");
242. **break**;
243. **case** 389:
244. printf("LDAP");
245. **break**;
246. **case** 443:
247. printf("HTTPS (SSL)");
248. **break**;
249. **case** 445:
250. printf("MICROSOFT-DS");
251. **break**;
252. **case** 1080:
253. printf("SOCKS");
254. **break**;
255. **default**:
256. printf("OTHER");
257. **break**;
258. }
260. printf("\nSequence number: ");
261. **for**(i=((Trama[14]&15)\*4)+18; i<((Trama[14]&15)\*4)+22;i++)
262. printf("%d", Trama[i]);
264. printf("\nAcknowledgement number: ");
265. **for**(i=((Trama[14]&15)\*4)+22; i<((Trama[14]&15)\*4)+26;i++)
266. printf("%d", Trama[i]);
268. printf("\nOffset (Header lenght): %d bytes", (Trama[((Trama[14]&15)\*4)+26]>>4)\*4);
270. printf("\nFlags: 0x%x", Trama[((Trama[14]&15)\*4)+27]);
272. printf("\nWindow: 0x%x", Trama[((Trama[14]&15)\*4)+28]<<8 | Trama[((Trama[14]&15)\*4)+29]);
274. CheckSumTCP(Trama);
276. printf("\nUrgent pointer: 0x%x", Trama[((Trama[14]&15)\*4)+32]<<8 | Trama[((Trama[14]&15)\*4)+33]);
277. }
279. **void** UDP(unsigned **char** Trama[]){
280. printf("\n\n\*\*\*\*\*\*\*\*\*\*\n");
281. printf("   UDP");
282. printf("\n\*\*\*\*\*\*\*\*\*\*");
284. printf("\nSource port: ");
285. **switch**(Trama[((Trama[14]&15)\*4)+14]<<8 | Trama[((Trama[14]&15)\*4)+15]){
286. **case** 7:
287. printf("ECHO");
288. **break**;
289. **case** 19:
290. printf("CHARGEN");
291. **break**;
292. **case** 37:
293. printf("TIME");
294. **break**;
295. **case** 53:
296. printf("DOMAIN");
297. **break**;
298. **case** 67:
299. printf("BOOTPS (DHCP)");
300. **break**;
301. **case** 68:
302. printf("BOOTPC (DHCP)");
303. **break**;
304. **case** 69:
305. printf("TFTP");
306. **break**;
307. **case** 137:
308. printf("NETBIOS-NS");
309. **break**;
310. **case** 138:
311. printf("NETBIOS-DGM");
312. **break**;
313. **case** 161:
314. printf("SNMP");
315. **break**;
316. **case** 162:
317. printf("SNMP-TRAP");
318. **break**;
319. **case** 500:
320. printf("ISAKMP");
321. **break**;
322. **case** 514:
323. printf("SYSLOG");
324. **break**;
325. **case** 520:
326. printf("RIP");
327. **break**;
328. **case** 33434:
329. printf("TRACEROUTE");
330. **break**;
331. **default**:
332. printf("OTHER");
333. **break**;
334. }
335. printf("\nDestination port: ");
336. **switch**(Trama[((Trama[14]&15)\*4)+16]<<8 | Trama[((Trama[14]&15)\*4)+17]){
337. **case** 7:
338. printf("ECHO");
339. **break**;
340. **case** 19:
341. printf("CHARGEN");
342. **break**;
343. **case** 37:
344. printf("TIME");
345. **break**;
346. **case** 53:
347. printf("DOMAIN");
348. **break**;
349. **case** 67:
350. printf("BOOTPS (DHCP)");
351. **break**;
352. **case** 68:
353. printf("BOOTPC (DHCP)");
354. **break**;
355. **case** 69:
356. printf("TFTP");
357. **break**;
358. **case** 137:
359. printf("NETBIOS-NS");
360. **break**;
361. **case** 138:
362. printf("NETBIOS-DGM");
363. **break**;
364. **case** 161:
365. printf("SNMP");
366. **break**;
367. **case** 162:
368. printf("SNMP-TRAP");
369. **break**;
370. **case** 500:
371. printf("ISAKMP");
372. **break**;
373. **case** 514:
374. printf("SYSLOG");
375. **break**;
376. **case** 520:
377. printf("RIP");
378. **break**;
379. **case** 33434:
380. printf("TRACEROUTE");
381. **break**;
382. **default**:
383. printf("OTHER");
384. **break**;
385. }
386. printf("\nLength: %d bytes", Trama[((Trama[14]&15)\*4)+18] + Trama[((Trama[14]&15)\*4)+19]);
387. printf("\nChecksum: 0x%x", Trama[((Trama[14]&15)\*4)+20] + Trama[((Trama[14]&15)\*4)+21]);
388. }
390. **void** IP(unsigned **char** Trama[]){
391. **int** i;
392. printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
393. printf("\tIP\n");
394. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");
396. printf("\nVersion: %d", Trama[14]>>4);
398. printf("\nIHL (Internet Header Lenght): %d bytes", (Trama[14]&15)\*4);
400. printf("\nToS (Type of Service): ");
401. **if**(Trama[15]&16) printf("Minimize delay");
402. **else** **if**(Trama[15]&8) printf("Maximize throughout");
403. **else** **if**(Trama[15]&4) printf("Maximize reliability");
404. **else** **if**(Trama[15]&2) printf("Minimize cost");
405. **else** printf("None");
407. printf("\nTotal lenght: %d bytes", Trama[16]<<8 | Trama[17]);
409. printf("\nIndentification: 0x%x", Trama[18]<<8 | Trama[19]);
411. printf("\nFlags: ");
412. **if**(((Trama[20]>>5) & 2) && ((Trama[20]>>5) & 1)) printf("011 -> Don't Fragment / More Fragment");
413. **else** **if**(((Trama[20]>>5) ^ 2) && ((Trama[20]>>5) ^ 1)) printf("000 -> \*\*\*\*\*\*\*\*\*\*\*\*\*\* / \*\*\*\*\*\*\*\*\*\*\*\*\*");
414. **else** **if**(((Trama[20]>>5) ^ 2) && ((Trama[20]>>5) & 1)) printf("001 -> \*\*\*\*\*\*\*\*\*\*\*\*\*\* / More Fragment");
415. **else** **if**(((Trama[20]>>5) & 2) && ((Trama[20]>>5) ^ 1)) printf("010 -> Don't Fragment / \*\*\*\*\*\*\*\*\*\*\*\*\*");
417. printf("\nFragment offset: %d", (Trama[20] & 31)<<8 | Trama[19]);
419. printf("\nTime to live: %d jumps", Trama[22]);
421. printf("\nProtocol: %d -> ", Trama[23]);
422. **switch**(Trama[23]){
423. **case** 1:
424. printf("ICMP");
425. **break**;
426. **case** 2:
427. printf("IGMP");
428. **break**;
429. **case** 6:
430. printf("TCP");
431. **break**;
432. **case** 9:
433. printf("IGRP");
434. **break**;
435. **case** 17:
436. printf("UDP");
437. **break**;
438. **case** 47:
439. printf("GRE");
440. **break**;
441. **case** 50:
442. printf("ESP");
443. **break**;
444. **case** 51:
445. printf("AH");
446. **break**;
447. **case** 57:
448. printf("SKIP");
449. **break**;
450. **case** 88:
451. printf("EIGRP");
452. **break**;
453. **case** 89:
454. printf("OSPF");
455. **break**;
456. **case** 115:
457. printf("L2TP");
458. **break**;
459. **default**:
460. printf("Other");
461. **break**;
462. }
464. CheckSumIP(Trama);
466. printf("\nSource address: ");
467. **for**(i=26; i<30; i++){
468. printf("%d", Trama[i]);
469. **if**(i!=29) printf(".");
470. }
472. printf("\nDestination address: ");
473. **for**(i=30; i<34; i++){
474. printf("%d", Trama[i]);
475. **if**(i!=33) printf(".");
476. }
478. **if**(Trama[23]==1) ICMP(Trama);
479. **else** **if**(Trama[23]==6) TCP(Trama);
480. **else** **if**(Trama[23]==17) UDP(Trama);
481. }
483. **void** ARP(unsigned **char** Trama[]){
484. **int** i;
485. printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
486. printf("\tARP\n");
487. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");
489. **if**((Trama[14]<<8 | Trama[15]) == 1)
490. printf("\nHardware Address Type (Ethernet): %d ", (Trama[14]<<8 | Trama[15]));
491. **else** **if**((Trama[14]<<8 | Trama[15]) == 6)
492. printf("\nHardware Address Type (IEEE 802 LAN): %d ", (Trama[14]<<8 | Trama[15]));
494. **if**((Trama[16]<<8 | Trama[17]) == 2048)
495. printf("\nProtocol Address Type (IPv4): %d", (Trama[16]<<8 | Trama[17]));
497. **if**(Trama[18] == 6)
498. printf("\nHardware Address Lenght (Ethernet/IEEE 802): %d ", Trama[18]);
500. **if**(Trama[19] == 4)
501. printf("\nProtocol Address Lenght (IPv4): %d ", Trama[19]);
503. **if**((Trama[20]<<8 | Trama[21]) == 1)
504. printf("\nOperation (Request): %d ", (Trama[20]<<8 | Trama[21]));
505. **else** **if**((Trama[20]<<8 | Trama[21]) == 2)
506. printf("\nOperation (Reply): %d ", (Trama[20]<<8 | Trama[21]));
507. **else** **if**((Trama[20]<<8 | Trama[21]) == 3)
508. printf("\nOperation (Request Inv): %d ", (Trama[20]<<8 | Trama[21]));
509. **else** **if**((Trama[20]<<8 | Trama[21]) == 4)
510. printf("\nOperation (Reply Inv): %d ", (Trama[20]<<8 | Trama[21]));
512. printf("\nSource Hardware Address: ");
513. **for**(i=22; i<28; i++){
514. printf("%.2x", Trama[i]);
515. **if**(i!=27) printf(":");
516. }
518. printf("\nSource Protocol Address: ");
519. **for**(i=28; i<32; i++){
520. printf("%.2x", Trama[i]);
521. **if**(i!=31) printf(":");
522. }
524. printf("\nTarget Hardware Address: ");
525. **for**(i=32; i<38; i++){
526. printf("%.2x", Trama[i]);
527. **if**(i!=37) printf(":");
528. }
530. printf("\nTarget Protocol Address: ");
531. **for**(i=38; i<42; i++){
532. printf("%.2x", Trama[i]);
533. **if**(i!=41) printf(":");
534. }
535. }
537. **void** Analizador(unsigned **char** Trama[], **int** n){
538. **int** i;
539. //printf("Castro Cruces Jorge Eduardo: %d\n\n", (Trama[((Trama[14]&15)\*4)+26]>>4)\*4);
540. printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");
541. printf("Trama No. %d", n+1);
543. printf("\nMAC destino: ");
544. **for**(i=0; i<6; i++){
545. printf("%.2x", Trama[i]);
546. **if**(i!=5) printf(":");
547. **else** **break**;
548. }
550. printf("\nMAC origen: ");
551. **for**(i=6; i<12; i++){
552. printf("%.2x", Trama[i]);
553. **if**(i!=11) printf(":");
554. **else** **break**;
555. }
557. printf("\nTamanio: %d bytes\n", (Trama[12]<<8 | Trama[13]));
559. **if**((Trama[12]<<8 | Trama[13]) < 1500){
560. LLC(Trama);
561. }
562. **else** **if**((Trama[12]<<8 | Trama[13]) == 2048){
563. IP(Trama);
564. }
565. **else** **if**((Trama[12]<<8 | Trama[13]) == 2054){
566. ARP(Trama);
567. }**else** printf("\nOtro");
568. printf("\n\n\n");
569. }

572. **void** main(){
573. **int** i;
574. unsigned **char** Trama[][84]={
575. {0xff,0xff,0xff,0xff,0xff,0xff,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x06,0x00,0x10,
576. 0x08,0x00,0x06,0x04,0x00,0x04,0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,
577. 0x00,0x00,0x00,0x00,0x00,0x00,0x94,0xcc,0x39,0xfe},                                //t1
578. {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x00,0x46,0x00,
579. 0x80,0x42,0x04,0x55,0x34,0x11,0x80,0x11,0x6b,0xf0,0x94,0xcc,0x39,0xcb,0x94,0xcc,
580. 0x67,0x02,0xaa,0xbb,0xcc,0xdd,0x04,0x0c,0x00,0x35,0x00,0x2e,0x85,0x7c,0xe2,0x1a,
581. 0x01,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x77,0x77,0x77,0x03,0x69,
582. 0x73,0x63,0x05,0x65,0x73,0x63,0x6f,0x6d,0x03,0x69,0x70,0x6e,0x02,0x6d,0x78,0x00,
583. 0x00,0x1c,0x00,0x01},                                                              //t2
584. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
585. 0x09,0x8d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
586. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
587. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7c,0x9b,0x6d},  //t3
588. {0x00,0x23,0x8b,0x46,0xe9,0xad,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x80,0x35,0x00,0x01,
589. 0x08,0x00,0x06,0x04,0x00,0x03,0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x94,0xcc,0x3a,0xe1,
590. 0x00,0x23,0x8b,0x46,0xe9,0xad,0x94,0xcc,0x39,0xcb,0x00,0x00,0x00,0x00,0x00,0x00,
591. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xd8,0xee,0xdf,0xb0},  //t4
592. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
593. 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
594. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
595. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x90,0x6d},  //t5
596. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
597. 0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23,
598. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
599. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x99,0x98,0x6d},  //t6
600. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf1,
601. 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
602. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
603. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x90,0x6d},  //t7
604. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
605. 0x43,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
606. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
607. 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x43,0x05,0x90,0x6d},  //t8
608. {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x00,0x45,0x10,
609. 0x00,0x3c,0x04,0x57,0x00,0x00,0x80,0x01,0x98,0x25,0x94,0xcc,0x39,0xcb,0x94,0xcc,
610. 0x3a,0xe1,0x08,0x00,0x49,0x5c,0x03,0x00,0x01,0x00,0x61,0x62,0x63,0x64,0x65,0x66,
611. 0x67,0x68,0x69,0x6a,0x6b,0x6c,0x6d,0x6e,0x6f,0x70,0x71,0x72,0x73,0x74,0x75,0x76,
612. 0x77,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69},                               // Trama IP-1
613. {0x00,0x1f,0x45,0x9d,0x1e,0xa2,0x00,0x23,0x8b,0x46,0xe9,0xad,0x08,0x00,0x46,0x08,
614. 0x80,0x42,0x04,0x55,0x34,0x11,0x80,0x11,0x6b,0xf0,0x94,0xcc,0x39,0xcb,0x94,0xcc,
615. 0x67,0x02,0xaa,0xbb,0xcc,0xdd,0x04,0x0c,0x00,0x35,0x00,0x2e,0x85,0x7c,0xe2,0x1a,
616. 0x01,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x77,0x77,0x77,0x03,0x69,
617. 0x73,0x63,0x05,0x65,0x73,0x63,0x6f,0x6d,0x03,0x69,0x70,0x6e,0x02,0x6d,0x78,0x00,
618. 0x00,0x1c,0x00,0x01},                                                             // Trama IP-2
619. {0x02,0xFF,0x53,0xC3,0xE9,0xAB,0x00,0xFF,0x66,0x7F,0xD4,0x3C,0x08,0x00,0x45,0x02,
620. 0x00,0x30,0x2C,0x00,0x40,0x00,0x80,0x06,0x4B,0x74,0xC0,0xA8,0x01,0x02,0xC0,0xA8,
621. 0x01,0x01,0x04,0x03,0x00,0x15,0x00,0x3B,0xCF,0x44,0x00,0x00,0x00,0x00,0x70,0x02,
622. 0x20,0x00,0x0C,0x34,0x00,0x00,0x02,0x04,0x05,0xB4,0x01,0x01,0x04,0x02},           // Trama IP-3
623. {0x00,0x14,0xd1,0xc2,0x38,0xbe,0x00,0x18,0xe7,0x33,0x3d,0xc3,0x08,0x00,0x45,0x00,
624. 0x00,0x30,0x94,0x71,0x40,0x00,0x80,0x06,0xf9,0x8c,0xc0,0xa8,0x02,0x3c,0x4a,0x7d,
625. 0x5f,0x68,0x10,0x52,0x00,0x50,0x03,0xc7,0x5a,0xa1,0x00,0x00,0x00,0x00,0x70,0x02,
626. 0x40,0x00,0x67,0x4b,0x00,0x00,0x02,0x04,0x05,0xb4,0x01,0x01,0x04,0x02},            //Protocolo TCP-1
627. {0x00,0x01,0xf4,0x43,0xc9,0x19,0x00,0x18,0xe7,0x33,0x3d,0xc3,0x08,0x00,0x45,0x00,
628. 0x00,0x28,0xf6,0x18,0x40,0x00,0x80,0x06,0x6b,0xa4,0x94,0xcc,0x19,0xf5,0x40,0xe9,
629. 0xa9,0x68,0x08,0x3a,0x00,0x50,0x42,0xfe,0xd8,0x4a,0x6a,0x66,0xac,0xc8,0x50,0x10,
630. 0x42,0x0e,0x00,0x00,0x00,0x00},                                                    //Protocolo TCP-2
631. {0x00,0x01,0xf4,0x43,0xc9,0x19,0x00,0x50,0xba,0xb2,0xf3,0x7b,0x08,0x00,0x45,0x00,
632. 0x00,0x3c,0x09,0x0d,0x00,0x00,0x80,0x01,0x7b,0xfe,0x94,0xcc,0x19,0x1b,0x94,0xcc,
633. 0x73,0x02,0x08,0x00,0x58,0x5a,0x02,0x00,0xf3,0x01,0x61,0x62,0x63,0x64,0x65,0x66,
634. 0x67,0x68,0x69,0x6a,0x6b,0x6c,0x6d,0x6e,0x6f,0x70,0x71,0x72,0x73,0x74,0x75,0x76,
635. 0x77,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69},                                //Trama ICMP solicitud ECO
636. {0x00,0x50,0xba,0xb2,0xf3,0x7b,0x00,0x01,0xf4,0x43,0xc9,0x19,0x08,0x00,0x45,0x00,
637. 0x00,0x3c,0xdf,0x5c,0x00,0x00,0x7c,0x01,0xa9,0xae,0x94,0xcc,0x73,0x02,0x94,0xcc,
638. 0x19,0x1b,0x00,0x00,0x60,0x5a,0x02,0x00,0xf3,0x01,0x61,0x62,0x63,0x64,0x65,0x66,
639. 0x67,0x68,0x69,0x6a,0x6b,0x6c,0x6d,0x6e,0x6f,0x70,0x71,0x72,0x73,0x74,0x75,0x76,
640. 0x77,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69},                                //Trama ICMP respuesta ECO
641. {0x00,0x50,0xba,0xb2,0xf3,0x7b,0x00,0x02,0xd1,0x02,0xd4,0xca,0x08,0x00,0x45,0x00,
642. 0x00,0x38,0x26,0x2d,0x00,0x00,0x40,0x01,0xf8,0x75,0x94,0xcc,0x19,0x6f,0x94,0xcc,
643. 0x19,0x1b,0x03,0x03,0xaa,0x23,0x00,0x00,0x00,0x00,0x45,0x00,0x00,0x3f,0x07,0xdc,
644. 0x00,0x00,0x80,0x11,0xd6,0xaf,0x94,0xcc,0x19,0x1b,0x94,0xcc,0x19,0x6f,0x04,0x0c,
645. 0x00,0x35,0x00,0x2b,0x4e,0x6d},
646. {0x00,0x01,0xf4,0x43,0xc9,0x19,0x00,0x18,0xe7,0x33,0x3d,0xc3,0x08,0x00,0x45,0x00,
647. 0x00,0x28,0xf6,0x18,0x40,0x00,0x80,0x06,0x6b,0xa4,0x94,0xcc,0x19,0xf5,0x40,0xe9,
648. 0xa9,0x68,0x08,0x3a,0x00,0x50,0x42,0xfe,0xd8,0x4a,0x6a,0x66,0xac,0xc8,0x50,0x10,
649. 0x42,0x0e,0x00,0x00,0x00,0x00}};
650. printf("Castro Cruces Jorge Eduardo\n\n");
651. **int** Tam=**sizeof**(Trama)/**sizeof**(Trama[0]);
652. **for**(i=0; i<Tam; i++)
653. Analizador(Trama[i], i);
654. }

Capturas de pantalla





















