Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра теоретической и прикладной информатики

Лабораторная работа №4  
“Анализ структуры кадра/фрейма технологии Ethernet”

по дисциплине “Операционные системы и компьютерные сети”

Факультет: ПМИ

Группа: ПМИ-12

Бригада: 5

Студент(-ы): Курдюков И.Н.

Омельницкая Е.И.

Преподаватель: Кобылянский В. Г.

Филиппова Е.В.

Новосибирск

2023

**1) Цель работы**

Спроектировать и реализовать программу, выполняющую анализ структуры кадра/фрейма технологии Ethernet.

**2) Ход работы**

1. Разработать и отладить программу, выполняющую анализ потока кадров. Потоки кадров представлены в виде файлов двоичного формата, место нахождения которых уточняется у преподавателя. В кадрах отсутствует преамбула и контрольная сумма, для исходящего кадра длина может быть меньше минимальной. Каждая бригада выполняет обработку одного файла с именем ethersXX.bin, где ХХ – номер бригады.

При выполнении работы в дистанционном режиме в обязательном порядке выполнить анализ файлов ethers06.bin и ethers07.bin.

**Текст программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <WinSock.h>

#include <locale.h>

#pragma comment(lib, "Ws2\_32.lib")

void MAC\_print(FILE\* out, char\* MAC)

{

for (int i = 0; i < 5; i++)

fprintf(out, "%02X:", (unsigned char)MAC[i]);

fprintf(out, "%02X\n", (unsigned char)MAC[5]);

}

void IP\_print(FILE\* out, char\* IP)

{

for (int i = 0; i < 3; i++)

fprintf(out, "%d.", (unsigned char)IP[i]);

fprintf(out, "%d\n", (unsigned char)IP[3]);

}

int main()

{

setlocale(LC\_ALL, "russian");

FILE\* input = NULL;

char file\_name[20]; //переменная для имени файла

int size = 0; //размер кадра(для каждого кадра)

//переменные для итогового резульатата обработки

int count\_frame = 0; //переменная для подсчёта количества кадров

int DIX\_fr = 0; //число кадров типа DIX

int RAW\_fr = 0; //число кадров типа RAW

int SNAP\_fr = 0; //число кадров типа SNAP

int LLC\_fr = 0; //число кадров типа LLC

char\* DATA;

printf("Введите имя файла формата ethersXX.bin, где XX - номер от 01 до 12: ");

scanf("%s", file\_name);

input = fopen(file\_name, "rb");

if (input == NULL)

{

printf("Ошибка открытия файла. Проверьте правильность ввода имени файла.\n");

system("pause");

return 0;

}

fseek(input, 0, SEEK\_END);

size = ftell(input);

fseek(input, 0, SEEK\_SET);

DATA = (char\*)malloc(size); //выделяем память для массива размером size(в байтах), malloc возвращает указатель на выделенную область памяти и приводит его к типу char\*

fread(DATA, 1, size, input); //считывает size элементов данных, каждый из которых занимает 1 байт, из файла и сохраняет их в массиве DATA.

FILE\* res = fopen("output.txt", "w");

fprintf(res, "Размер файла: %d байтов\n\n", size);

char\* p = DATA;

while (p < DATA + size)

{

count\_frame++;

fprintf(res, "Номер кадра: %d\n", count\_frame);

fprintf(res, "MAC-адрес получателя: ");

MAC\_print(res, p);

fprintf(res, "MAC-адрес отправителя: ");

MAC\_print(res, p + 6);

unsigned short LT = ntohs(\*(unsigned short\*)(p + 12)); //осуществляет перевод целого короткого числа из сетевого порядка байт в порядок байт, принятый на компьютере

if (LT == 0x0800)

{

fprintf(res, "Тип кадра: DIX\n");

fprintf(res, "IP-источник: ");

IP\_print(res, p + 26);

fprintf(res, "IP-адрес: ");

IP\_print(res, p + 30);

LT = ntohs(\*(unsigned short\*)(p + 16)) + 14;

fprintf(res, "Размер кадра: %d\n\n", LT);

DIX\_fr++;

p += LT;

}

else

{

if (LT == 0x0806)

{

fprintf(res, "Тип кадра: DIX\n");

p += 28 + 14; // длина стандартного ARP-пакета является фиксированной и равна 28 байтам, 14 байтов занимают параметры кадра

DIX\_fr++;

fprintf(res, "Размер кадра: %d\n\n", 28 + 14);

}

else {

if (LT > 0x05DC)

{

fprintf(res, "Тип кадра: Ethernet DIX (Ethernet II)\n");

DIX\_fr++;

}

else

{

unsigned short F = ntohs(\*(unsigned short\*)(p + 14));

if (F == 0xFFFF)

{

fprintf(res, "Тип кадра: Raw 802.3 (Novell 802.3)\n");

RAW\_fr++;

}

else if (F == 0xAAAA)

{

fprintf(res, "Тип кадра: Ethernet SNAP\n");

SNAP\_fr++;

}

else

{

fprintf(res, "Тип кадра: 802.3/LLC (Novell 802.2)\n");

LLC\_fr++;

}

}

p += LT + 14;

fprintf(res, "Размер кадра: %d\n\n", LT + 14);

}

}

}

fprintf(res, "Общее число кадров: %d\n", count\_frame);

fprintf(res, "DIX: %d\n", DIX\_fr);

fprintf(res, "RAW: %d\n", RAW\_fr);

fprintf(res, "SNAP: %d\n", SNAP\_fr);

fprintf(res, "LLC: %d\n", LLC\_fr);

printf("Программа успешно выполнена. Результат работы программы сохранён в файле \"output.txt\".\n");

fclose(input);

fclose(res);

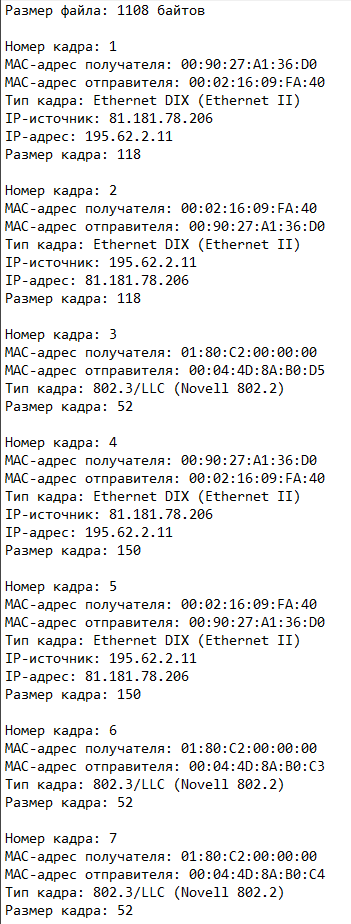
system("pause");

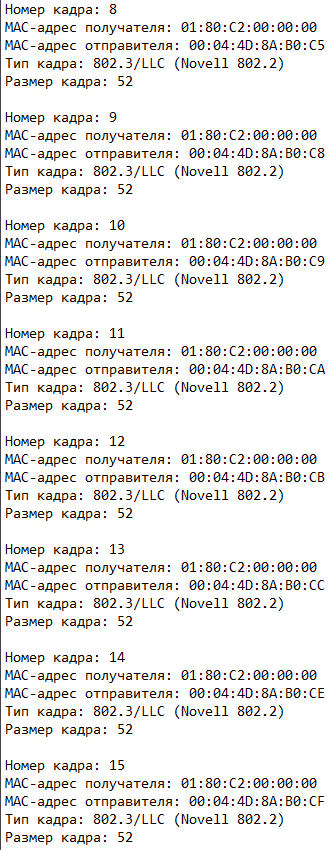
return 0;

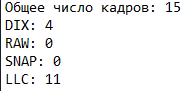
}

**Результат работы программы:**

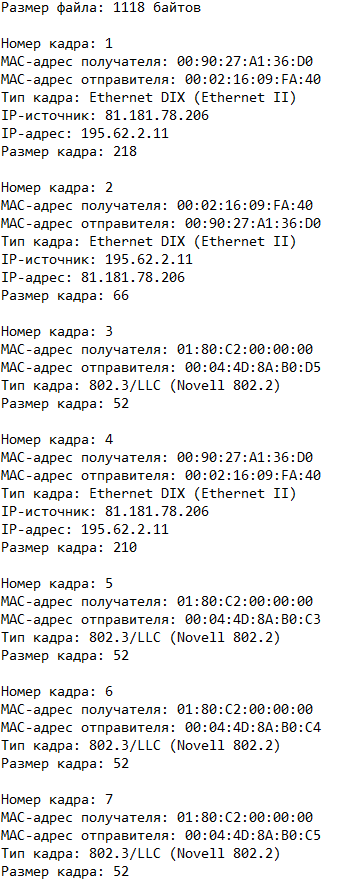
**1. Файл “ethers05.bin”**

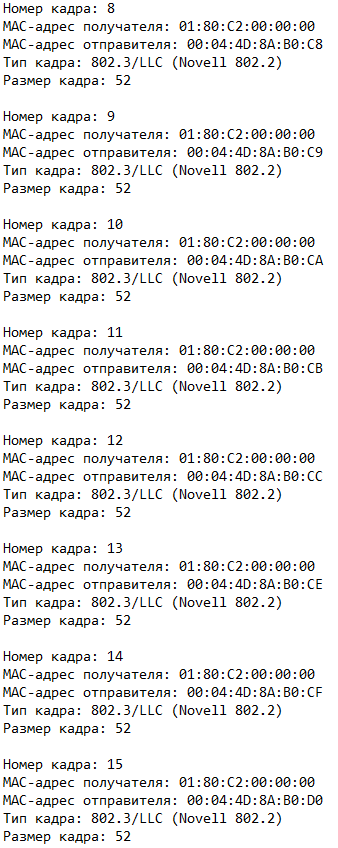


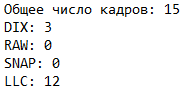




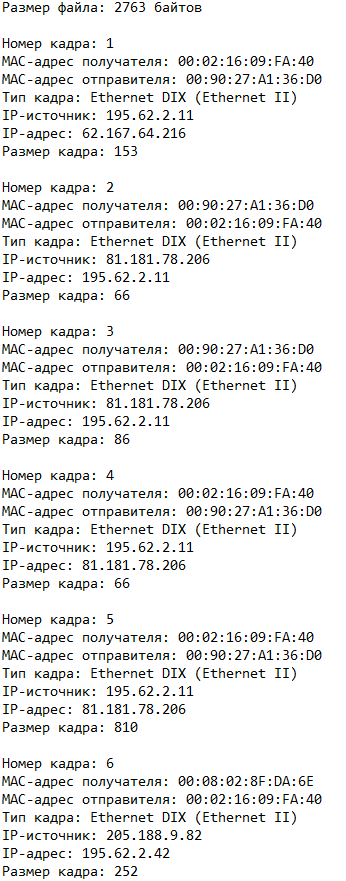
**2. Файл “ethers06.bin”**

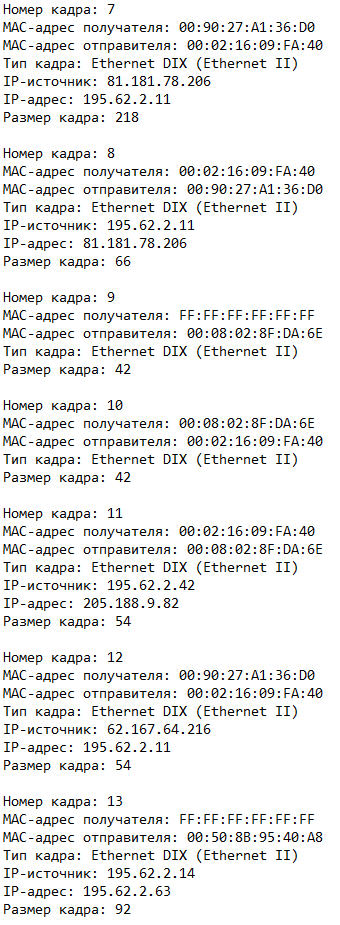
****

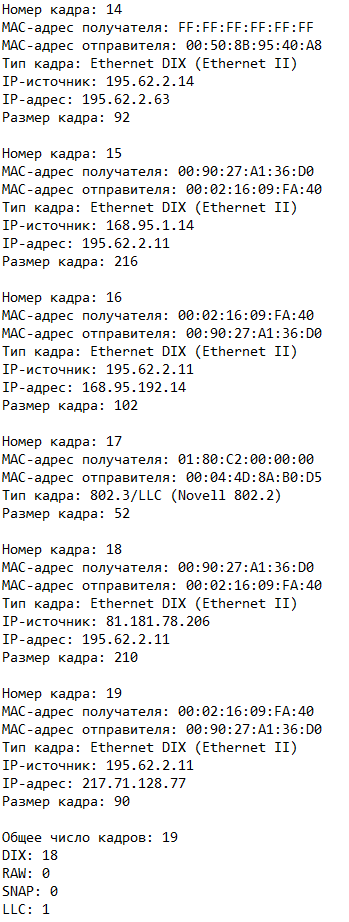
****

****

**3. Файл “ethers07.bin”**

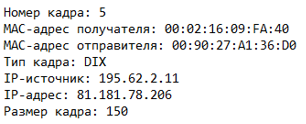




****

2. Выполнить полный анализ кадра с номером ХХ, где ХХ номер Вашей бригады. Анализ можно провести с помощью любого 16-ричного редактора, например, свободно распространяемого редактора HxD.

**Анализ 5 кадра файла “ethers05.bin”:**

****

**Исходный вид кадра:**

00 02 16 09 FA 40 00 90 27 A1 36 D0 08 00 45 00 00 88 31 73 40 00 40 06 A3 30 C3 3E 02 0B 51 B5 4E CE 00 16 81 A0 E8 1C 95 F5 D2 45 26 5D 80 18 BE D4 C8 B1 00 00 01 01 08 0A 35 22 B4 F6 08 95 F1 5A C6 30 28 54 52 E8 9F AB 83 4F 35 06 A5 2A FD 5D 16 87 6B 62 A2 35 F8 2A EF E6 EF 36 69 C4 73 3A 9A A2 5B 08 59 39 3A 8F E4 31 2B 79 8B 6C 56 E9 6E 37 94 45 16 8A 14 4A 0D BD 71 C3 D0 EC 11 39 E0 27 42 9A 62 C2 4F CB 57 2F 87 31 C0 D5 2F 53 E5 4B 04 AA

Размер кадра: 150 байт

МАС-адрес получателя (6 байт): 00 02 16 09 FA 40 = 00:02:16:09:FA:40

МАС-адрес отправителя (6 байт): 00 90 27 A1 36 D0 = 00:90:27:A1:36:D0

Тип протокола/ длина кадра (2 байта): 08 00 = протокол IP Internet

Версия (4 бита): 4 = IPv4

Длина заголовка (4 бита): 5 = 5 слов (1 слово = 4 байта) = 5\*4 = 20 байт

Тип службы (1 байт): 00

Длина дейтаграммы (2 байта): 00 88 = 136 байт (150-6-6-2 = 136).

Идентификатор (2 байта): 31 73

Флаги [3 бита] и смещение фрагмента [13 бит] (2 байта) = 40 00

Время жизни (1 байта): 40 = 64 узлов может пройти дейтаграмма

Протокол верхнего уровня (1 байта): 06 (определяет протокол транспортного уровня)

Контрольная сумма заголовка (2 байта): A3 30

IP-Адрес отправителя (4 байта): C3 3E 02 0B = 195.62.2.11

IP-Адрес получателя (4 байта): 51 B5 4E CE = 81.181.78.206

Данные (116 байт):

TCP-порт отправителя (2 байта): 00 16 = 22

TCP-порт получателя (2 байта): 81 A0= 33184

Код позиции в сообщении (4 байта): E8 1C 95 F5 (порядковый номер первого октета в поле данных пользователя)

Номер октета, который должен прийти следующим (4 байта) : D2 45 26 5D

Hlen (4 бита): 8 = 8\*4 = 32 байта (длина заголовка сегмента)

Резерв (6 бит) и флаги (6 бит): 0 18

Размер окна (2 байта): BE D4 = 48852 число байтов, которые готов принять получатель

Контрольная сумма (2 байта): C8 B1

Указатель важной информации (2 байта): 00 00