

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

Отчет по лабораторной работе № 13
по дисциплине «Программирование»
Тема: битовые поля в структурах

Студент гр. 9305

_____ Есин А.Ю

Преподаватель

_____ Перязева Ю.В

Санкт-Петербург

2019

Содержание

Цель.....	3
Задание	3
Постановка задачи и описание решения	3
Описание структуры	4
Структура вызова функций	5
Текст программы	9
Пример работы программы	14
Заключение	15

Цель

Получить практические навыки в разработке алгоритмов с использованием битовых полей в структурах на языке Си.

Задание

Разработать алгоритм и реализовать функции преобразования статуса MAC-адреса сетевой карты заданного производителя с источника пакетов для конкретных адресатов на источник широковещательных пакетов и обратно с использованием битовых полей в структурах и битовых операций.

Постановка задачи и описание решения

Дано: MAC-адрес, введенный с клавиатуры.

Требуется получить: MAC-адрес с выбранным пользователем статусом.

Сначала считываем данные с клавиатуры. После того как пользователь выберет, какой статус он хочет установить, программа меняет первый справа бит шестого (справа) октета в соответствии с этим выбором: устанавливает «0», если пользователь в качестве статуса выбрал «источник пакетов для конкретных адресатов» и «1» - если «источник широковещательных пакетов»

Описание структуры

Таблица 1. Описание структуры (typedef struct address {...} mac).

Имя поля	Тип	Назначение
a1	unsigned	Элемент MAC-адреса (первый октет)
a2	unsigned	Элемент MAC-адреса (первый октет)
b1	unsigned	Элемент MAC-адреса (второй октет)
b2	unsigned	Элемент MAC-адреса (второй октет)
c1	unsigned	Элемент MAC-адреса (третий октет)
c2	unsigned	Элемент MAC-адреса (третий октет)
d1	unsigned	Элемент MAC-адреса (четвертый октет)
d2	unsigned	Элемент MAC-адреса (четвертый октет)
e1	unsigned	Элемент MAC-адреса (пятый октет)
e2	unsigned	Элемент MAC-адреса (пятый октет)
f1	unsigned	Элемент MAC-адреса (шестой октет)
f2	unsigned	Элемент MAC-адреса (шестой октет)

Структура вызова функций

1.Main

Описание:

Является точкой входа в программу.

Прототип:

```
Int main();
```

Пример вызова:

```
main();
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Локальная	first	mac*	Экземпляр структуры mac
Локальная	flag	int	Выбор статуса сетевого интерфейса

Возвращаемое значение:

2.setMac

Описание:

Считывание MAC-адреса с клавиатуры и сохранение данных в address.

Прототип:

```
void setMac(mac *address);
```

Пример вызова:

```
setMac(first);
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Формальный аргумент	address	mac*	Указатель на экземпляр структуры типа mac
Локальная	inp1	unsigned short	Первое число октета
Локальная	inp2	unsigned short	Второе число октета

Возвращаемое значение:

3.printMac

Описание:

Вывод MAC-адреса address на экран в шестнадцатеричной и двоичной формах.

Прототип:

```
void printMac(mac *address);
```

Пример вызова:

```
printMac(first);
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Формальный аргумент	address	mac*	Указатель на экземпляр структуры типа mac

Возвращаемое значение:

4.status_to_wide

Описание:

Изменение статуса MAC-адреса на источник широковещательных пакетов.

Прототип:

```
void status_to_wide(mac *address);
```

Пример вызова:

```
status_to_wide (first);
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Формальный аргумент	address	mac*	Указатель на экземпляр структуры типа mac

Возвращаемое значение:

5.status_to_certain

Описание:

Изменение статуса MAC-адреса на источник пакетов для конкретных адресов.

Прототип:

```
void status_to_certain(mac *address);
```

Пример вызова:

```
status_to_certain (first);
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Формальный аргумент	address	mac*	Указатель на экземпляр структуры типа mac

Возвращаемое значение:

6.printBin

Описание:

Вывод на экран числа num в двоичном представлении.

Прототип:

```
void printBin(int num);
```

Пример вызова:

```
printBin(address->a1);
```

Описание переменных:

Вид переменной	Имя поля	Тип	Назначение
Формальный аргумент	num	int	Число из MAC-адреса, которое необходимо вывести

Возвращаемое значение:

Текст программы

1) Файл *main.c*

```
#include <stdio.h>
#include <stdlib.h>
#include "struct.h"

int main() {
    mac *first = NULL;
    int flag;

    puts("Enter the MAC");
    puts("MAC represented as a1a2-b1b2-c1c2-d1d2-e1e2-f1f2");
    first = (mac*)malloc(sizeof(mac));
    setMac(first);
    printf("\n");
    printMac(first);
    puts("\n\n1 - status = broadcast packages source");
    puts("2 - status = specific address packages source");
    scanf("%d",&flag);
    if (flag == 1) {
        status_to_wide(first);
    } if (flag == 2) {
        status_to_certain(first);
    }
    puts("\nByte changed:\n");
    printMac(first);
    puts("\nPress ENTER to exit...");
    getchar();
    free(first);
    first = NULL;
    getchar();
}
```

2) Файл struct.c

```
#include "struct.h"
#include <stdio.h>

void setMac(mac *address)
{
    unsigned short inp1,inp2;

    puts("Enter first octet (a1 a2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->a1 = inp1;
    address->a2 = inp2;
    puts("Enter second octet (b1 b2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->b1 = inp1;
    address->b2 = inp2;
    puts("Enter third octet (c1 c2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->c1 = inp1;
    address->c2 = inp2;
    puts("Enter forth octet (d1 d2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->d1 = inp1;
    address->d2 = inp2;
    puts("Enter fifth octet (e1 e2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->e1 = inp1;
    address->e2 = inp2;
    puts("Enter sixth octet (f1 f2, with the space):");
    scanf("%hx %hx",&inp1,&inp2);
    address->f1 = inp1;
    address->f2 = inp2;
```

```
}
```

```
void printMac(mac *address)
{
    puts("HEX form:");
    printf("%x",address->a1);
    printf("%x-",address->a2);
    printf("%x",address->b1);
    printf("%x-",address->b2);
    printf("%x",address->c1);
    printf("%x-",address->c2);
    printf("%x",address->d1);
    printf("%x-",address->d2);
    printf("%x",address->e1);
    printf("%x-",address->e2);
    printf("%x",address->f1);
    printf("%x",address->f2);

    puts("\n\nbyte form:");
    printBin(address->a1);
    printf(" ");
    printBin(address->a2);
    printf(" ");
    printBin(address->b1);
    printf(" ");
    printBin(address->b2);
    printf(" ");
    printBin(address->c1);
    printf(" ");
    printBin(address->c2);
    printf(" ");
    printBin(address->d1);
    printf(" ");
    printBin(address->d2);
    printf(" ");
    printBin(address->e1);
    printf(" ");
    printBin(address->e2);
    printf(" ");
    printBin(address->f1);
    printf(" ");
    printBin(address->f2);
}
```

```

    printBin(address->e1);
    printf(" ");
    printBin(address->e2);
    printf(" ");
    printBin(address->f1);
    printf(" ");
    printBin(address->f2);
    printf(" ");
}

```

```

void printBin(int num)
{
    int i, v = 128;

    for (i = 1; i <= 8; i++) {
        if (num >= v) {
            printf("%d", 1);
            num -= v;
        }
        else {
            printf("%d", 0);
        }
        v /= 2;
    }
}

```

```

void status_to_certain(mac *address)
{
    address->a2 &= ~(1 << 0);
}

```

```

void status_to_wide(mac *address)
{
    address->a2 |= (1 << 0);
}

```

3) Файл *struct.h*

```
#ifndef LAB13_STRUCT_H
#define LAB13_STRUCT_H

typedef struct address {
    unsigned a1:4;
    unsigned a2:4;
    unsigned b1:4;
    unsigned b2:4;
    unsigned c1:4;
    unsigned c2:4;
    unsigned d1:4;
    unsigned d2:4;
    unsigned e1:4;
    unsigned e2:4;
    unsigned f1:4;
    unsigned f2:4;
} mac;

void setMac(mac *address);
void printMac(mac *address);
void status_to_wide(mac *address);
void status_to_certain(mac *address);
void printBin(int num);

#endif //LAB13_STRUCT_H
```

Пример работы программы

Исходные данные:

a1 = A

a2 = A

b1 = 3

b2 = 8

c1 = C

c2 = 7

d1 = B

d2 = 2

e1 = 3

e2 = 4

f1 = A

f2 = 5

Вывод программы:

```
C:\Users\Artem\CLionProjects\lab13\cmake-build-debug\lab13.exe
Enter second octet (b1 b2, with the space):
B 8
Enter third octet (c1 c2, with the space):
c 7
Enter forth octet (d1 d2, with the space):
b 2
Enter fifth octet (e1 e2, with the space):
B 4
Enter sixth octet (f1 f2, with the space):
a 5

HEX form:
aa-38-c7-b2-34-a5

byte form:
00001010 00001010 00000011 00001000 00001100 00000111 00001011 00000010 00000011 00000100 00001010 00000101

1 - status = broadcast packages source
2 - status = specific address packages source
1

Byte changed:

HEX form:
ab-38-c7-b2-34-a5

byte form:
00001010 00001011 00000011 00001000 00001100 00000111 00001011 00000010 00000011 00000100 00001010 00000101
Press ENTER to exit...
```

Заключение

Выводы:

При выполнении лабораторной работы были получены практические навыки в разработке алгоритмов с использованием битовых полей в структурах посредством написания программы на языке Си.