

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

ОТЧЕТ
по лабораторной работе № 11
по дисциплине «Программирование»
Тема: Битовые поля в структурах

Студент гр. 3312

Мохно Даниил.

Преподаватель

Аббас Саддам

Санкт-Петербург

2024

Цель работы.

Целью работы является изучение битовых полей в структурах в языке Си и получение практических навыков в программировании на этом языке.

Задание (вариант 1)

Числовой адрес компьютера в глобальной сети Интернет (ip-адрес) версии 4 состоит из 4-х чисел от 0 до 255, разделенных точками (например, 123.45.67.89). Для записи каждого числа используется 1 байт (октет). Значения битов первого октета определяют т. н. «класс сети».

0xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx — класс А

10xxxxxx xxxxxxxx xxxxxxxx xxxxxxxx — класс В

110xxxxx xxxxxxxx xxxxxxxx xxxxxxxx — класс С

(x — произвольное значение бита — 0 или 1)

Разработать алгоритм и реализовать функции преобразования произвольного адреса IPv4 класса А в адрес класса В и наоборот с использованием битовых полей в структурах и битовых операций.

Адреса вводятся с клавиатуры.

Постановка задачи и описание решения

Создадим объединение ip адреса IPv4, оно будет содержать массив address из четырёх элементов 8-и битного целочисленного беззнакового типа, и структуру, содержащую неименованное поле на 6 бит и целочисленное поле class на 2 бита, которое будет отвечать за последние два бита первого октета.

В главной функции выделяем место под объединение, которое будет хранить ip адрес, вызываем функцию получения ip адреса, которой параметром передаём указатель на объединение, вызываем функцию вывода ip адреса, ей параметром передаём тот же указатель. Затем вызываем функцию смены класса ip адреса, ей так же передаём указатель на объединение, и вызываем функцию вывода ip адреса после чего освобождаем место из-под объединения.

В функции получения ip адреса получаем через форматированный ввод октеты ip адреса, которые сохраняем в ячейках массива нашего объединения.

В функции вывода ip адреса, мы выводим все октеты из массива в объединении, а затем циклом проходимся по массиву вызывая в каждой итерации функцию вывода двоичного представления числа, передавая ей число из массива.

В функции вывода двоичного представления числа мы запускаем цикл на 8 итераций, ставя итератор i равным 7, и пока он больше или равен 0. В цикле выводим число, полученное путём побитового сдвига вправо на i позиций и побитового «и» с единицей. То есть мы сдвинем число до текущего бита, а все остальные биты обнулим. Таким образом получаем текущий бит.

В функции смены класса ip в поле class объединения запишем результат операции исключающего или между текущим значением поля class (последние 2 бита адреса) и 0b10 (битовым представлением 2-и). За тем проверяем является ли второй бит 1-ей путем операции битового «и» между class и 0b10, если да, то записываем полученное в сравнении значение в class.

Описание переменных

Функция – int main():

№	Имя переменной	Тип	Назначение
1	ip	struct IPv4	Указатель на объединение с ip адресом

Функция функция получения ip адреса – void get_ip(IPv4 *ip):

№	Имя переменной	Тип	Назначение
1	ip	struct IPv4	Указатель на объединение с ip адресом

Функция вывода одного байта в битовом формате – void short_to_bin(IPv4 *ip):

№	Имя переменной	Тип	Назначение
1	address	u_int_8t	Однобайтное целое беззнаковое число, содержащее октет ip адреса
2	i	int	Итератор

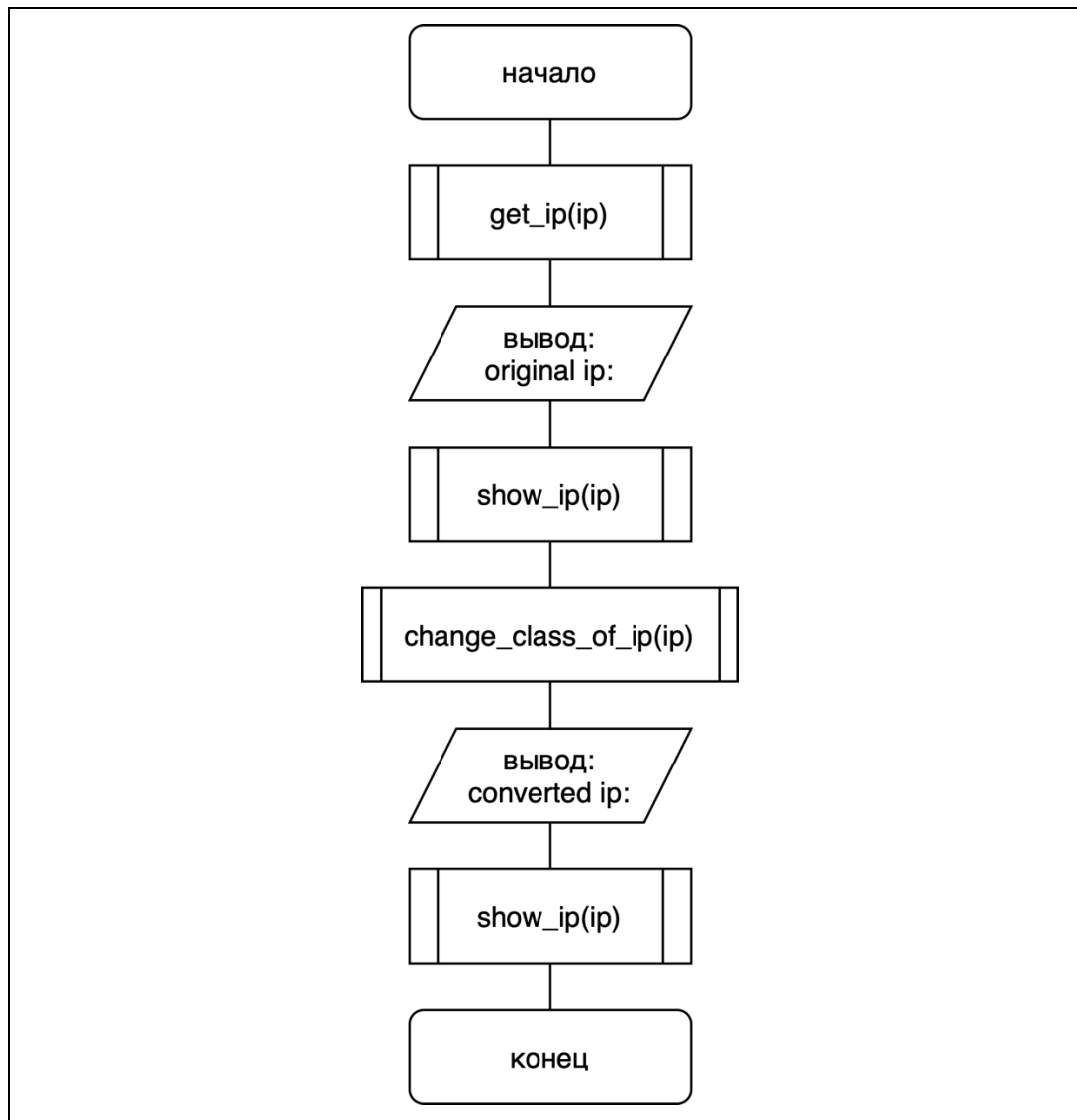
Функция замены двух старших битов первого октета ip адреса – void
change_class_of_ip(IPv4 *ip)

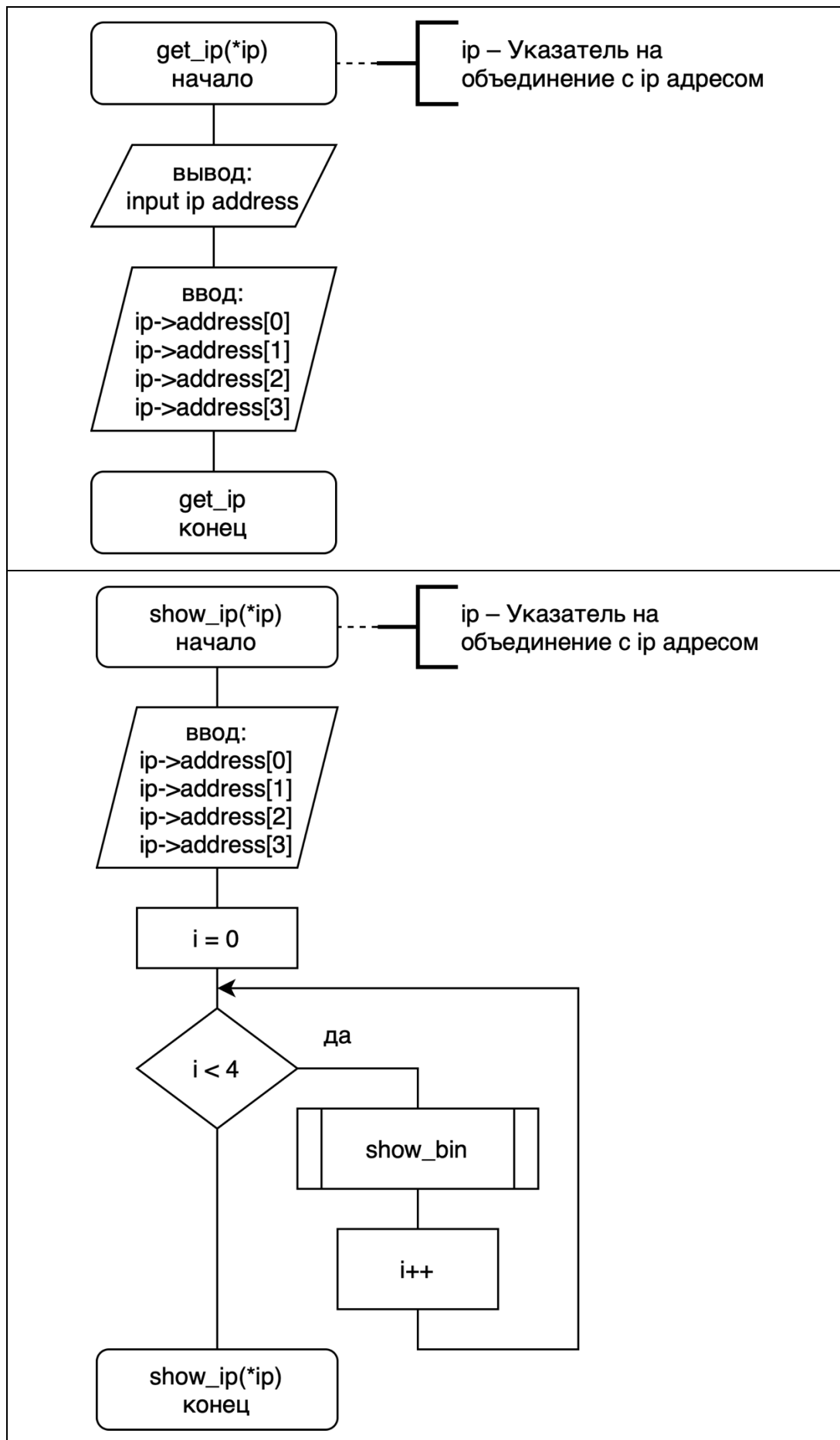
№	Имя переменной	Тип	Назначение
1	ip	struct IPv4	Указатель на объединение с ip адресом

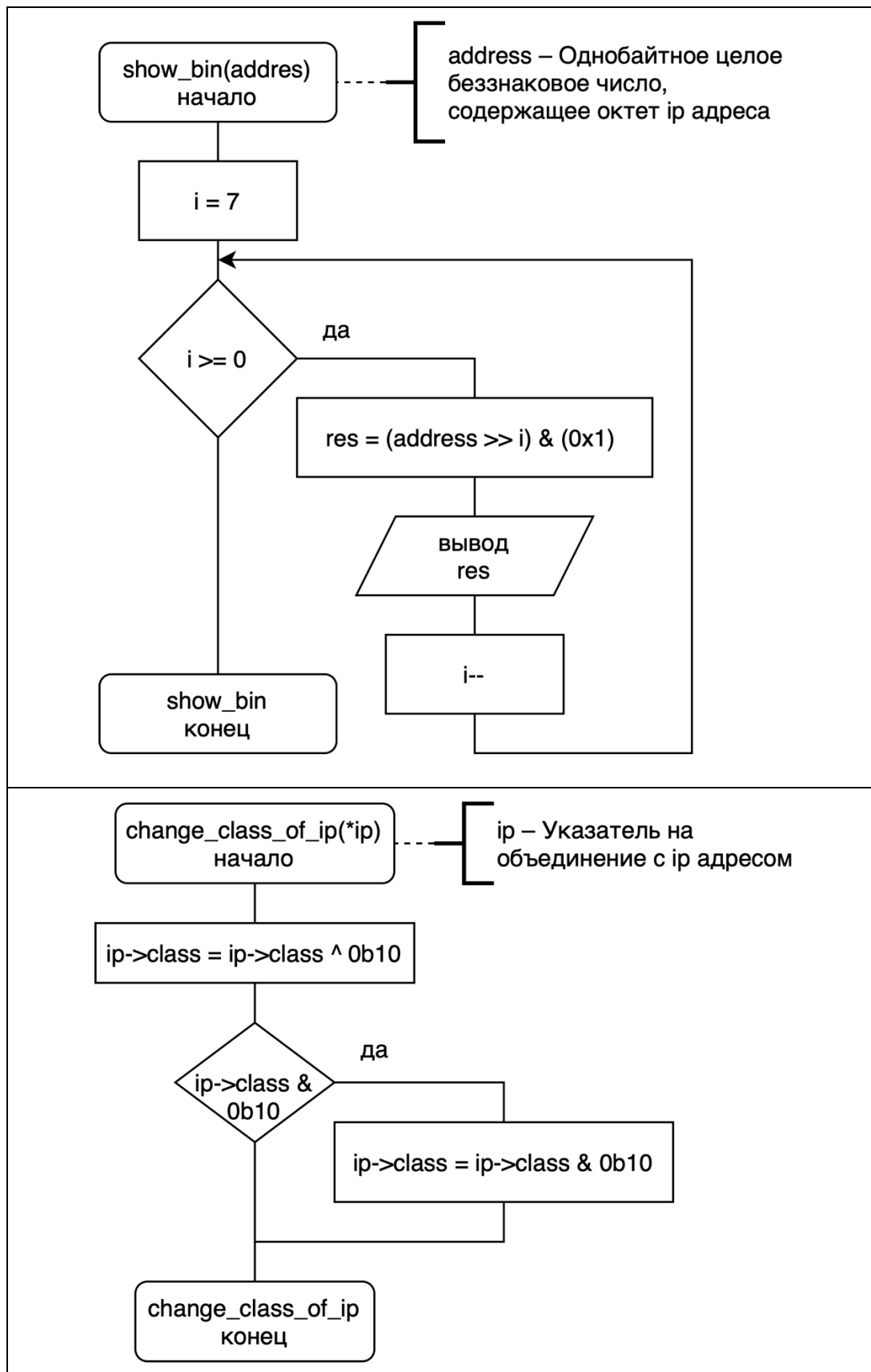
Функция вывода ip адреса и его двоичного битового представления – void
show_ip(IPv4 *ip)

№	Имя переменной	Тип	Назначение
1	ip	struct IPv4	Указатель на объединение с ip адресом
2	i	int	Итератор

Схема алгоритма







Контрольные примеры

№	Входные данные	Выходные данные
1	10.100.100.100	original ip: 10.100.100.100 00001010.01100100.01100100.01100100 converted ip: 138.100.100.100 10001010.01100100.01100100.01100100
2	123.45.67.89	original ip: 123.45.67.89 01111011.00101101.01000011.01011001 converted ip: 187.45.67.89 10111011.00101101.01000011.01011001
3	172.31.255.254	original ip: 172.31.255.254 10101100.00011111.11111111.11111110 converted ip: 44.31.255.254 00101100.00011111.11111111.11111110

Текст программы

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

typedef union IPv4
{
    u_int8_t address[4];
    struct
    {
        int : 6;
        int class: 2;
    };
} IPv4;

void get_ip(IPv4 *ip);
```

```

void show_ip(IPv4 *ip);

void show_bin(u_int8_t address);

void change_class_of_ip(IPv4 *ip);

int main()
{
    IPv4 *ip;

    ip = malloc(sizeof(IPv4));
    get_ip(ip);
    puts("original ip:");
    show_ip(ip);
    change_class_of_ip(ip);
    puts("converted ip:");
    show_ip(ip);
    free(ip);

    return 0;
}

void get_ip(IPv4 *ip)
{
    printf("input ip address\n");
    scanf("%hhu.%hhu.%hhu.%hhu", &(ip->address[0]), &(ip->address[1]), &(ip->address[2]), &(ip->address[3]));
}

void show_ip(IPv4 *ip)
{
    int i;
    printf("%hhu.%hhu.%hhu.%hhu\n", ip->address[0], ip->address[1], ip->address[2], ip->address[3]);
    for (i = 0; i < 4; i++)
    {
        show_bin(ip->address[i]);
        putchar('.');
    }
    putchar(8);
    putchar('\n');
}

void show_bin(u_int8_t address)
{
    int i;
    for (i = 7; i >= 0; i--) printf("%d", (address >> i) & (0x1));
}

void change_class_of_ip(IPv4 *ip)
{
    ip->class = ip->class ^ 0b10;
    if (ip->class & 0b10) ip->class = ip->class & 0b10;
}

```


Примеры выполнения программы

Пример 1:

```
input ip address
10.100.100.100
original ip:
10.100.100.100
00001010.01100100.01100100.01100100
converted ip:
138.100.100.100
10001010.01100100.01100100.01100100
```

Пример 2:

```
input ip address
123.45.67.89
original ip:
123.45.67.89
01111011.00101101.01000011.01011001
converted ip:
187.45.67.89
10111011.00101101.01000011.01011001
```

Пример 3:

```
input ip address
172.31.255.254
original ip:
172.31.255.254
10101100.00011111.11111111.11111110
converted ip:
44.31.255.254
00101100.00011111.11111111.11111110
```

Выводы.

В результате выполнения работы были изучены битовые поля в структурах в языке Си и получены практические навыки в программировании на этом языке.