**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе № 1**

**по дисциплине «Организация ЭВМ и систем»**

**Тема: ИССЛЕДОВАНИЕ ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ РАЗЛИЧНЫХ ФОРМАТОВ ДАННЫХ**

| Студент гр. 3311 |  | Сапронов К. Д. |
| --- | --- | --- |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2024

**Цель работы**

Знакомство с внутренним представлением различных типов данных,

используемых компьютером при их обработке.

**Задание (вариант 18)**

1. Разработать алгоритм ввода с клавиатуры типов данных unsigned int и double и показать на экране их внутреннее представление в двоичной системе счисления.
2. В соответствии с следующим заданием дополнить разработанный ранее алгоритм: Поменять местами значения рядом стоящих бит в парах. Количество пар и номер младшего разряда правой пары задаётся с клавиатуры.

**Описание решения**

1. Для вывода двоичного представления unsigned int цикл проходится по числу в обратном порядке, так как иначе оно будет выведено наоборот. Создается маска, в которой единица сдвигается влево на текущий индекс, после чего маска побитово сравнивается с самим числом, таким образом получая значение текущего бита, которое выводится на экран

2. Для вывода двоичного представления double был использован тип union с полями числа типа double и типа unsigned long long. Цикл же работает аналогично циклу из п.1.

3. Для реализации задания также используется тип union с полями double и unsigned long long. В цикле для каждой пары битов создаются соответствующие маски, с помощью которых получаются значения этих битов. Если они не равны, то эти биты инвертируются с помощью маски.

При инверсии маска сравнивается с двоичных представление числа. Все нулевые биты маски не меняют значения битов (0 и 0 -> 0, 0 и 1 -> 1), а ненулевой бит инвертирует нужный бит числа (1 и 0 -> 1, 1 и 1 -> 0)

**Двоичные представления float и double**

Тип float занимает в памяти компьютера 4 байта памяти

Первый бит - знаковый: 0 - число положительное, 1 - отрицательное..

Следующие 8 бит - порядок числа, остальные 23 — мантисса.

Пример: число 5.75

0 10000001 01110000000000000000000

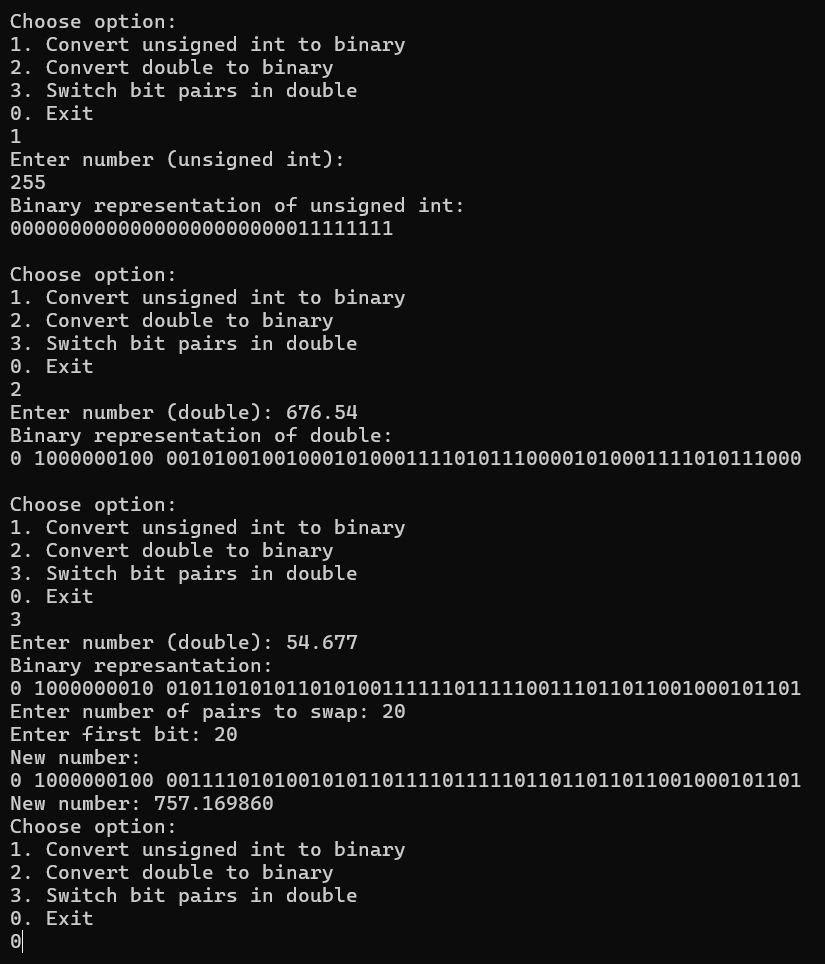
Тип double занимает в памяти компьютера 8 байт памяти

Первый бит также является знаковым, следующие 11 - порядок числа, остальные 52 - мантисса.

Пример: число -5.75

1 10000000001 0111000000000000000000000000000000000000000000000000

**Пример запуска программы**

****

**Текст программы.**

#include <stdio.h>

void printBinaryUnsignedInt(unsigned int num) {

puts("");

for (int i = ((sizeof(unsigned int) \* 8) - 1); i >= 0; i--) {

unsigned int mask = 1 << i;

printf("%c", (num & mask) ? '1' : '0');

}

puts("");

}

void printBinaryDouble(double num) {

puts("");

union {

double d;

unsigned long long i;

} u;

u.d = num;

for (int i = ((sizeof(double) \* 8)-1); i >= 0; i--) {

if (i==62 || i==52){

printf(" ");

}

unsigned long long mask = 1ULL << i;

printf("%c", (u.i & mask) ? '1' : '0');

}

puts("");

}

double swapAdjacentBits(double num, int numPairs, int startBit) {

union {

double d;

unsigned long long i;

} u;

u.d = num;

for (int pair = 0; pair < numPairs; pair++) {

int bit1 = startBit + 2 \* pair;

int bit2 = bit1 + 1;

unsigned long long mask1 = 1ULL << bit1;

unsigned long long mask2 = 1ULL << bit2;

int bit1Value = (u.i & mask1) ? 1 : 0;

int bit2Value = (u.i & mask2) ? 1 : 0;

if (bit1Value != bit2Value) {

u.i ^= mask1;

u.i ^= mask2;

}

}

return u.d;

}

int main() {

int numPairs, startBit, answer;

unsigned int num;

double num2;

do {

puts("\nChoose option:");

puts("1. Convert unsigned int to binary");

puts("2. Convert double to binary");

puts("3. Switch bit pairs in double");

puts("0. Exit");

scanf("%d", &answer);

getchar();

switch (answer) {

case 1:

printf("Enter number (unsigned int):\n");

scanf("%u", &num);

printf("Binary representation of unsigned int:");

printBinaryUnsignedInt(num);

break;

case 2:

printf("Enter number (double): ");

scanf("%lf", &num2);

printf("Binary representation of double:");

printBinaryDouble(num2);

break;

case 3:

printf("Enter number (double): ");

scanf("%lf", &num2);

printf("Binary represantation:");

printBinaryDouble(num2);

printf("Enter number of pairs to swap: ");

scanf("%d", &numPairs);

printf("Enter first bit: ");

scanf("%d", &startBit);

double newNum = swapAdjacentBits(num2, numPairs, startBit);

printf("New number:");

printBinaryDouble(newNum);

printf("New number: %lf", newNum);

break;

default:

break;

}

} while (answer!=0);

return 0;

}