НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

ИНСТИТУТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра Управления и информатики

**Лабораторная работа № 4**

**Программирование низкоуровневых задач**

Студент группы А-02-13,

Пак Ольга

**Преподаватель:**

Козлюк Д.

Москва, 2015г.

**Общее задание**

1. Подготовить инструменты для исследований и отладки.

Написать функции для печати отдельных байт и блока данных data размером size байт в шестнадцатеричном и в двоичном представлении.

**void** print\_in\_hex(uint8\_t byte);

**void** print\_in\_hex(**const void**\* data, size\_t size);

**void** print\_in\_binary(uint8\_t byte);

**void** print\_in\_binary(**const void**\* data, size\_t size);

#include <conio.h>

#include <iostream>

#include <vector>

#include <bitset>

using std::cout;

using std::endl;

using std::cin;

using namespace std;

void print\_nibble(uint8\_t nibble)

{

char c;

if (0 <= nibble && nibble <= 9)

{

c = '0' + nibble;

cout << c;

}

else if (nibble <= 15)

{

c = 'A' + (nibble - 10);

cout << c;

}

else cout<< "error";

}

void print\_in\_hex(uint8\_t byte)

{

print\_nibble(byte >> 4);

print\_nibble(byte & 0x0F);

cout << ' ';

}

void print\_in\_hex(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%16==0)

cout<<"\n";

print\_in\_hex(((uint8\_t\*) data)[i]);

}

}

void print\_bit(uint8\_t bit)

{

char c;

if (bit == 0)

{

c = '0';

cout << c;

}

else if (bit == 1)

{

c = '1';

cout << c;

}

else cout<< "error";

}

void print\_in\_binary(uint8\_t byte)

{

for (int i=7; i>=0; i--)

print\_bit((byte>> i)&0x01);

cout << ' ';

}

void print\_in\_binary(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%4==0)

cout<<"\n";

print\_in\_binary(((uint8\_t\*) data)[i]);

}

}

int main()

{

int number;

while (cout<<"Enter number :",cin>>number)

{

cout<<"Number in hexadecimal system : ";

print\_in\_hex(& number, sizeof(number));

cout<<"\n";

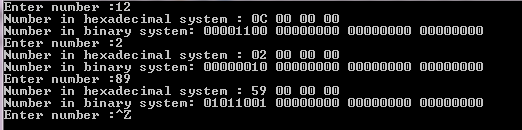
cout<<"Number in binary system: ";

print\_in\_binary(& number, sizeof(number));

cout<<"\n";

}

}



2.Написать программу-калькулятор для побитовых операций.

Пользователь вводит первый операнд, затем оператор (&, | или ^), затем второй операнд. Программа выполняет указанное действие над операндами, и печатает расчет в шестнадцатеричном и двоичном виде. Операнды — двухбайтовые беззнаковые целые числа (uint16\_t).

#include <conio.h>

#include <iostream>

#include <vector>

#include <bitset>

using std::cout;

using std::endl;

using std::cin;

using namespace std;

void print\_nibble(uint8\_t nibble)

{

char c;

if (0 <= nibble && nibble <= 9)

{

c = '0' + nibble;

cout << c;

}

else if (nibble <= 15)

{

c = 'A' + (nibble - 10);

cout << c;

}

else cout<< "error";

}

void print\_in\_hex(uint8\_t byte)

{

print\_nibble(byte >> 4);

print\_nibble(byte & 0x0F);

cout << ' ';

}

void print\_in\_hex(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%16==0)

cout<<"\n";

print\_in\_hex(((uint8\_t\*) data)[i]);

}

}

void print\_bit(uint8\_t bit)

{

char c;

if (bit == 0)

{

c = '0';

cout << c;

}

else if (bit == 1)

{

c = '1';

cout << c;

}

else cout<< "error";

}

void print\_in\_binary(uint8\_t byte)

{

for (int i=7; i>=0; i--)

print\_bit((byte>> i)&0x01);

cout << ' ';

}

void print\_in\_binary(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%4==0)

cout<<"\n";

print\_in\_binary(((uint8\_t\*) data)[i]);

}

}

int main()

{

uint16\_t x, y, z;

char operat;

int a,b;

while (cout<<"Enter x ,operator (&,|,^) and y :\n",cin>>x>>operat>>y)

{

switch(operat)

{

case '&':

z = x & y;

break;

case '|':

z = x | y;

break;

case '^':

z = x ^ y;

break;

}

print\_in\_hex(& x,sizeof(x));

cout<<operat;

print\_in\_hex(& y, sizeof(y));

cout<<"= ";

print\_in\_hex(& z, sizeof(z));

cout << "\n";

print\_in\_binary(& x, sizeof(x));

cout<<operat<<"\n";

print\_in\_binary(& y, sizeof(y));

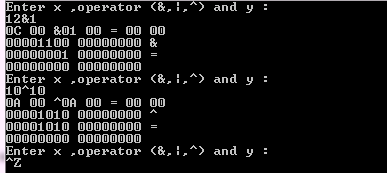
cout<<"=\n";

print\_in\_binary(& z, sizeof(z));

cout << "\n";

}

}



3. Изучить представление и размещение данных в памяти.

3.1. Определить структуру Student, описывающую студента атрибутами:

1) имя длиной не более 16 *печатных (printable)* символов;

2) год поступления;

3) средний балл;

4) пол, представленный одним битом (0 — женский, 1 — мужской);

5) количество пройденных курсов;

6) указатель на структуру Student, описывающую старосту группы (для старосты — нулевой указатель).

*Указание.* Поле размером в несколько бит (не больше, чем бит в определенным целочисленном типе) можно объявить так:

*целочисленный-тип имя-поля* : *число-бит*;

3.2. Объявить и заполнить массив из трех структур Student, описывающий двух студентов одной группы и их старосту.

3.3. Напечатать, занести в отчет и письменно пояснить:

1) адрес и размер массива;

2) адреса и размеры всех элементов массива;

3) для всех полей, кроме пола1, одного из элементов массива (не старосты): адрес, смещение от начала структуры, размер, шестнадцатеричное и двоичное представление;

#include <conio.h>

#include <iostream>

#include <vector>

#include <bitset>

#include <cstddef>

using std::cout;

using std::endl;

using std::cin;

using namespace std;

void print\_nibble(uint8\_t nibble)

{

char c;

if (0 <= nibble && nibble <= 9)

{

c = '0' + nibble;

cout << c;

}

else if (nibble <= 15)

{

c = 'A' + (nibble - 10);

cout << c;

}

else cout<< "error";

}

void print\_in\_hex(uint8\_t byte)

{

print\_nibble(byte >> 4);

print\_nibble(byte & 0x0F);

cout << ' ';

}

void print\_in\_hex(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%16==0)

cout<<"\n";

print\_in\_hex(((uint8\_t\*) data)[i]);

}

cout<<"\n";

}

void print\_bit(uint8\_t bit)

{

char c;

if (bit == 0)

{

c = '0';

cout << c;

}

else if (bit == 1)

{

c = '1';

cout << c;

}

else cout<< "error";

}

void print\_in\_binary(uint8\_t byte)

{

for (int i=7;i>=0;i--)

print\_bit((byte>> i)&0x01);

cout << ' ';

}

void print\_in\_binary(const void\* data, size\_t size)

{

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

{

if (i!=0 && i%4==0)

cout<<"\n";

print\_in\_binary(((uint8\_t\*) data)[i]);

}

cout<<"\n";

}

struct Student

{

char name[16+1];

int year;

double meanb;

uint8\_t sex : 1;

int curs;

Student\* starosta;

};

#define PRINT\_INFO(name) \

cout << "Adres of field " <<" name : " << & students[1].name << "\n";\

cout << "offset : " << offsetof(Student, name) << "\n";\

cout << "size : " << sizeof(students[1].name) << "\n";\

cout << "in the hexadecimal system\n";\

print\_in\_hex(& students[1].name, sizeof(students[1].name));\

cout << "in the binary system\n";\

print\_in\_binary(& students[1].name,sizeof(students[1].name));\

cout<<"\n";

#define PRINT\_ADDR(name) \

cout << #name ": addr. "; \

print\_in\_hex(offsetof(Student, name)); \

cout << "- "; \

print\_in\_hex(offsetof(Student, name) + sizeof(students[i].name) - 1); \

cout << "\n"

int main()

{

Student students[3];

students[0] = Student{"Anton ",2013,5.0,1,3,&students[0]};

students[1] = Student{"Andrey ",2013,4.5,1,3,&students[0]};

students[2] = Student{"Vika ",2013,3.0,0,3,0};

cout << "Array adress ; " << students << "\n";

cout<< "\n";

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

{

cout << "Element's adress: " << & students[i] << "\n";

cout << "Element's size: " << sizeof(students[i]) << "\n";

cout<< "\n";

}

PRINT\_INFO(name);

PRINT\_INFO(year);

PRINT\_INFO(meanb);

PRINT\_INFO(curs);

PRINT\_INFO(starosta);

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

cout << "student[" << i << "]:\n";

cout << "\n";

cout << "x0 x1 x2 x3|x4 x5 x6 x7|x8 x9 xA xB|xC xD xE xF\n";

print\_in\_hex(& students[i], sizeof(students[i]));

cout << "\n";

PRINT\_ADDR(name);

PRINT\_ADDR(year);

PRINT\_ADDR(meanb);

PRINT\_ADDR(curs);

PRINT\_ADDR(starosta);

cout << "\n";

}

}



4) все элементы массива в шестнадцатеричном виде с указанием соответствия блоков байт полям структур.

*Указание.* Смещение поля field структуры *типа* type от начала любого её экземпляра можно определить макросом offsetof(type, field).

4. Написать программу для обработки текстового файла, представляя текст только строками C, размещаемыми в динамической памяти.

4.1. Запросить у пользователя имя файла, сохранив его в массиве символов, размещенном на стеке (не в динамической памяти).

4.2. Проверить, используя функции стандартной библиотеки C++ для работы со строками C, что введенное имя файла корректно:

1) не содержит запрещенных символов: \*, ", <, >, ? или |;

2) если содержит двоеточие, то только вторым символом, которому предшествует буква, и за которым следует обратная косая черта (\).

3) расширение \*.txt (в любом регистре).

*Указание.* Задачи решаются стандартными функциями isalpha(), strchr(), strrchr(), strncmp().

4.3. Загрузить содержимое текстового файла в память целиком:

1) использовать ifstream или fopen() для доступа к файлу;

2) использовать методы seekg() и tellg() либо функции fseek() и ftell() для определения размера файла, переместившись в его конец и получив текущее положение в файле;

3) выделить в динамической памяти массив достаточного размера;

4) загрузить всё содержимое файла в выделенную область памяти методом read() или функцией fread().

4.4. Запросить у пользователя строку, поместив её в массив на стеке.

4.5. Найти введенную строку в загруженном тексте, не используя функций стандартной библиотеки C++ для работы со строками C.

4.6. Разбить текст файла на предложения, поместив их в динамический массив строк типа **char**\*\* (указатель на массив указателей).

4.7. Упорядочить предложения по возрастанию их длины любым способом.

4.8. Записать все предложения в новый текстовый файл.

4.9. Освободить все выделенные в процессе решения блоки памяти.

#include <iostream>

#include "string.h"

#include "stdio.h"

using namespace std;

// функция проверяет корректно ли имя пункт 4.2

bool CorrectName(char\* name)

{

//проверка на некорректные символы (кроме :)

if((strchr(name,'"')!=NULL) or (strchr(name,'\*')!=NULL) or (strchr(name,'<')!=NULL) or (strchr(name,'>')!=NULL) or (strchr(name,'?')!=NULL) or (strchr(name,'|')!=NULL) )

{

cout<<"Uncorrect Name\n";

return false;

}

// проверка на двоеточие,не 2 символом

if(strchr(name,':')!=NULL and strchr(name,':')!=&name[1])

{

cout<<"Uncorrect Name\n";

return false;

}

//проверка на двоеточие,стоящее 2 символом

//условия:перед двоеточием символ,после '\'

if(strchr(name,':')==&name[1])

{

if(isalpha(name[0]==0) or name[2]!='\\' )

{

cout<<"Uncorrect Name\n";

return false;

}

}

int leng=strlen(name);

// проверка на тип файла .txt

if((name[leng-1]!='T' or name[leng-1]!='t') and (name[leng-2]!='x' or name[leng-2]!='X')

and (name[leng-3]!='T' or name[leng-3]!='t') and (name[leng-4]!='.' ))

{

cout<<"Uncorrect Name\n";

return false;

}

return true;

}

int main()

{

char FileName[30];

cout<<"Enter FileName\n";

cin>>FileName;

//п 4.2

while(CorrectName(FileName))

{

// открытие файла

FILE\* f=fopen(FileName,"rb");

if (f!=NULL)

{

//определить размер файла

long size;

fseek(f, 0, SEEK\_END);

size = ftell(f);

fseek(f, 0, SEEK\_SET);

char\* Mass;

Mass = new char [size];

// выгрузка в массив содержимое файла

fread(Mass,1,size,f);

char str[30];

cout<<"enter string\n";

cin>>str;

bool fl=false;

//узнать есть ли строка в файле

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

{

if(Mass[i]==str[0] and size-i>=strlen(str))

{

fl=true;

for(int j=1; j<strlen(str); j++)

{

if(Mass[i+j]!=str[j])

{

fl=false;

break;

}

}

}

if(fl==true)

break;

}

if(fl)

cout<<"string="<<str<<" contained in the file\n";

else

{

cout<<"string="<<str<<" is not contained in the file\n";

}

//пункт 4.6

char\*\* StrMass;// массив строчек(предложений)

int n=0;//количество предложний

// узнаем количество предложений в тексте

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

{

if(Mass[i]=='.' or Mass[i]=='?' or Mass[i]=='!' )

n=n+1;

}

// задаем указатель на количество предложений

StrMass=new char\*[n];

int\* len;

len=new int[n];

n=0;// количество предложений

int k=0;//вспомогательные счетчики

int h=0;//вспомогательные счетчики для разбиения текста в массив по предложениям

// заполняем массив

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

{

if(Mass[i]=='.' or Mass[i]=='?' or Mass[i]=='!' )

{

StrMass[n]=new char[i-k];

len[n]=i-k;

//i-k количество символов в текущем предложении

for(int j=k; j<i+1; j++)

{

StrMass[n][h]=Mass[j];

++h;

}

k=i+1;

h=0;

n=n+1;

}

}

// вывод на экран массива

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

{

cout<<"\n";

for(int j=0; j<=len[i]; j++)

cout<<StrMass[i][j];

}

cout<<"\n---- sorted array---------";

//сортировка массива по длинне,по возрастанию

char\* temp;

int temp\_;

for (int i =1; i<n; ++i)

{

for (int j =n-1; j>=i; --j)

{

if (len[j-1]>len[j])

{

temp\_=len[j-1];

len[j-1]=len[j];

len[j]=temp\_;

temp = StrMass[j-1];

StrMass[j-1] = StrMass[j];

StrMass[j] = temp;

}

}

}

//вывод в консоль

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

{

cout<<"\n";

for(int j=0; j<=len[i]; j++)

cout<<StrMass[i][j];

}

// вывод в файл

FILE\* write\_=fopen("Result.txt","w");

//

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

{

for(int j=0; j<=len[i]; j++)

fwrite(&StrMass[i][j],sizeof(char), 1,write\_);

}

delete[] StrMass;

delete[] Mass;

delete write\_;

delete f;

break;

}

else

{

cout<<"File not open";

break;

}

}

return 0;

}

