МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Организация ЭВМ и систем»

**Лабораторная работа №2.**

**Определение конфигурации компьютера программными средствами**

Выполнил:

студент группы ИВТАПбд-21

Кондратьев П.С.

Проверила:

Лылова А.В.

Ульяновск, 2018

**Цель работы:** научиться определять конфигурацию и параметры функциональных устройств персонального компьютера.

Определение конфигурации компьютера, является необходимой задачей, при написании программ, ориентированных на максимально возможное использование ресурсов вычислительной системы. Определив, программным способом, наличие и возможности тех или иных устройств, программа может включать тот или иной набор команд или алгоритмы, оптимизированные под конкретную архитектуру.

Для определения конфигурации вычислительной системы существует множество средств, встроенных как в саму операционную систему, так и в низкоуровневые средства (BIOS) [1].

***Целью лабораторной работы*** является получение навыков написания аппаратно-зависимого программного кода. Для выполнения лабораторной работы был взять язык высокого уровня с++, так как в нем присутствует возможность получать системные характеристики из реестара, благодаря подключенной библиотеки Lmcons.h, которая позволяет выбирать нужные параметры системы из реестра.

Вариант Задания:

|  |  |
| --- | --- |
| 5 | Определить модель компьютера (BIOS INT 15h) и версию BIOS. |

Библиотека Lmcons.h, предоставляет нам возможность получить тип и данные для указанного имени значения, связанного с открытым ключом реестра.

Синтакси:

LONG WINAPI RegQueryValueEx(

\_In\_        HKEY    hKey,

\_In\_opt\_    LPCTSTR lpValueName,

\_Reserved\_  LPDWORD lpReserved,

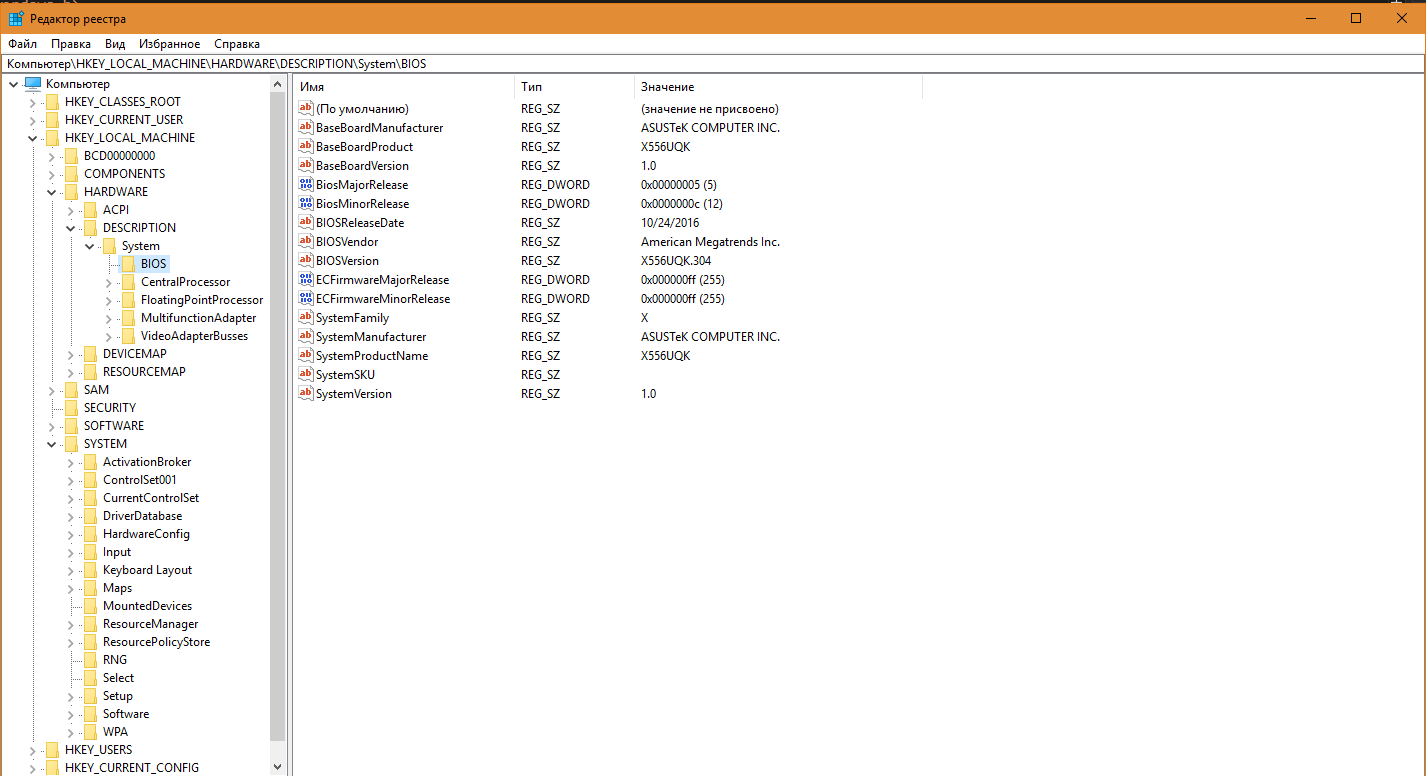
\_Out\_opt\_   LPDWORD lpType,

\_Out\_opt\_   LPBYTE  lpData,

\_Inout\_opt\_ LPDWORD lpcbData

);

Для нахождения BIOSVersion мы указывали его путь в реестре в данном методе этой библиотеки.



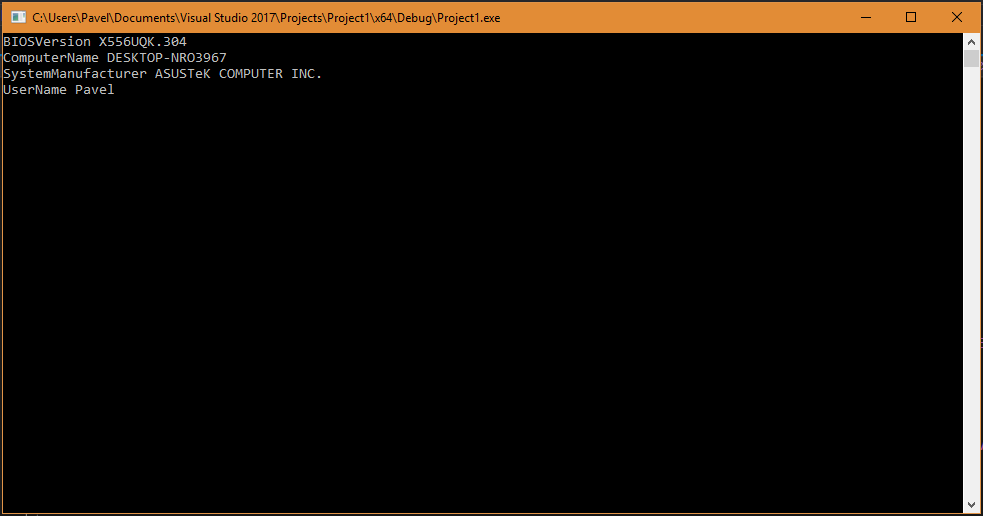
RegOpenKeyEx(HKEY\_LOCAL\_MACHINE, "HARDWARE\\DESCRIPTION\\System\\BIOS", NULL, KEY\_QUERY\_VALUE, &rKey);

RegQueryValueEx(rKey, "BIOSVersion", NULL, NULL, (LPBYTE)&Reget, &RegetPath);

cout << "BIOSVersion " << Reget << "\n";

Проделаем те же самые операции и с ComputerName, SystemManufacturer, UserName, указывая путь к ним и то значение которое нужно.

**Пример работы:**



**Вывод по проделанной работе:**

В ходе выполнения данной лабораторной работы изучили библиотеку Lmcons.h, предоставляет нам возможность получить тип и данные для указанного имени значения, связанного с открытым ключом реестра. Также был рассмотрен GNU Compiler Collection (Набор компиляторов для различных языков программирования), для получения из с++ кода ассемблерный.

**Исходный код:**

#include <windows.h>

#include <iostream>

#include <Lmcons.h>

using namespace std;

int main() {

HKEY rKey, dkey;

TCHAR Reget[256], Reget1[256], Reget2[256];

DWORD RegetPath = sizeof(Reget);

TCHAR name[UNLEN + 1];

DWORD size = UNLEN + 1;

RegOpenKeyEx(HKEY\_LOCAL\_MACHINE, "HARDWARE\\DESCRIPTION\\System\\BIOS", NULL, KEY\_QUERY\_VALUE, &rKey);

RegQueryValueEx(rKey, "BIOSVersion", NULL, NULL, (LPBYTE)&Reget, &RegetPath);

cout << "BIOSVersion " << Reget << "\n";

RegetPath = sizeof(Reget1);

RegOpenKeyEx(HKEY\_LOCAL\_MACHINE, "SYSTEM\\CurrentControlSet\\Control\\ComputerName\\ComputerName", NULL, KEY\_QUERY\_VALUE, &dkey);

RegQueryValueEx(dkey, "ComputerName", NULL, NULL, (LPBYTE)&Reget1, &RegetPath);

cout << "ComputerName " << Reget1 << "\n";

RegetPath = sizeof(Reget2);

RegOpenKeyEx(HKEY\_LOCAL\_MACHINE, "SYSTEM\\CurrentControlSet\\Control\\SystemInformation", NULL, KEY\_QUERY\_VALUE, &dkey);

RegQueryValueEx(dkey, "SystemManufacturer", NULL, NULL, (LPBYTE)&Reget2, &RegetPath);

cout << "SystemManufacturer " << Reget2 << "\n";

GetUserName((TCHAR\*)name, &size);

cout << "UserName " << name << "\n";

getchar();

return 0;

}

Код ассемблера:

.file "Source.cpp".text.section .rdata,"dr"

\_ZStL19piecewise\_construct: space 1

.lcomm \_ZStL8\_\_ioinit,1,1.def \_\_main; .scl 2; .type 32; .endefalign 8

.LC0:

.ascii "HARDWARE\\DESCRIPTION\\System\\BIOS\0"

.LC1:

.ascii "BIOSVersion\0"

.LC2:

.ascii "BIOSVersion \0"

.LC3:

.ascii "\12\0"

.align 8

.LC4:

.ascii "SYSTEM\\CurrentControlSet\\Control\\ComputerName\\ComputerName\0"

.LC5:

.ascii "ComputerName\0"

.LC6:

.ascii "ComputerName \0"

.align 8

.LC7:

.ascii "SYSTEM\\CurrentControlSet\\Control\\SystemInformation\0"

.LC8:

.ascii "SystemManufacturer\0"

.LC9:

.ascii "SystemManufacturer \0"

.LC10:

.ascii "UserName \0"

.text

.globl main

.def main; .scl 2; .type 32; .endef

.seh\_proc main

main:

.LFB5954:

pushq %rbp

.seh\_pushreg %rbp

subq $1120, %rsp

.seh\_stackalloc 1120

leaq 128(%rsp), %rbp

.seh\_setframe %rbp, 128

.seh\_endprologue

call \_\_main

movl $256, 204(%rbp)

movl $257, -68(%rbp)

leaq 984(%rbp), %rax

movq %rax, 32(%rsp)

movl $1, %r9d

movl $0, %r8d

leaq .LC0(%rip), %rdx

movq $-2147483646, %rcx

movq \_\_imp\_RegOpenKeyExA(%rip), %rax

call \*%rax

movq 984(%rbp), %rax

leaq 204(%rbp), %rdx

movq %rdx, 40(%rsp)

leaq 720(%rbp), %rdx

movq %rdx, 32(%rsp)

movl $0, %r9d

movl $0, %r8d

leaq .LC1(%rip), %rdx

movq %rax, %rcx

movq \_\_imp\_RegQueryValueExA(%rip), %rax

call \*%rax

leaq .LC2(%rip), %rdx

movq .refptr.\_ZSt4cout(%rip), %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movq %rax, %rcx

leaq 720(%rbp), %rax

movq %rax, %rdx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

leaq .LC3(%rip), %rdx

movq %rax, %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movl $256, 204(%rbp)

leaq 976(%rbp), %rax

movq %rax, 32(%rsp)

movl $1, %r9d

movl $0, %r8d

leaq .LC4(%rip), %rdx

movq $-2147483646, %rcx

movq \_\_imp\_RegOpenKeyExA(%rip), %rax

call \*%rax

movq 976(%rbp), %rax

leaq 204(%rbp), %rdx

movq %rdx, 40(%rsp)

leaq 464(%rbp), %rdx

movq %rdx, 32(%rsp)

movl $0, %r9d

movl $0, %r8d

leaq .LC5(%rip), %rdx

movq %rax, %rcx

movq \_\_imp\_RegQueryValueExA(%rip), %rax

call \*%rax

leaq .LC6(%rip), %rdx

movq .refptr.\_ZSt4cout(%rip), %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movq %rax, %rcx

leaq 464(%rbp), %rax

movq %rax, %rdx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

leaq .LC3(%rip), %rdx

movq %rax, %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movl $256, 204(%rbp)

leaq 976(%rbp), %rax

movq %rax, 32(%rsp)

movl $1, %r9d

movl $0, %r8d

leaq .LC7(%rip), %rdx

movq $-2147483646, %rcx

movq \_\_imp\_RegOpenKeyExA(%rip), %rax

call \*%rax

movq 976(%rbp), %rax

leaq 204(%rbp), %rdx

movq %rdx, 40(%rsp)

leaq 208(%rbp), %rdx

movq %rdx, 32(%rsp)

movl $0, %r9d

movl $0, %r8d

leaq .LC8(%rip), %rdx

movq %rax, %rcx

movq \_\_imp\_RegQueryValueExA(%rip), %rax

call \*%rax

leaq .LC9(%rip), %rdx

movq .refptr.\_ZSt4cout(%rip), %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movq %rax, %rcx

leaq 208(%rbp), %rax

movq %rax, %rdx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

leaq .LC3(%rip), %rdx

movq %rax, %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

leaq -68(%rbp), %rdx

leaq -64(%rbp), %rax

movq %rax, %rcx

movq \_\_imp\_GetUserNameA(%rip), %rax

call \*%rax

leaq .LC10(%rip), %rdx

movq .refptr.\_ZSt4cout(%rip), %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

movq %rax, %rcx

leaq -64(%rbp), %rax

movq %rax, %rdx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

leaq .LC3(%rip), %rdx

movq %rax, %rcx

call \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

call getchar

movl $0, %eax

addq $1120, %rsp

popq %rbp

ret

.seh\_endproc

.def \_\_tcf\_0; .scl 3; .type 32; .endef

.seh\_proc \_\_tcf\_0

\_\_tcf\_0:

.LFB6439:

pushq %rbp

.seh\_pushreg %rbp

movq %rsp, %rbp

.seh\_setframe %rbp, 0

subq $32, %rsp

.seh\_stackalloc 32

.seh\_endprologue

leaq \_ZStL8\_\_ioinit(%rip), %rcx

call \_ZNSt8ios\_base4InitD1Ev

nop

addq $32, %rsp

popq %rbp

ret

.seh\_endproc

.def \_Z41\_\_static\_initialization\_and\_destruction\_0ii; .scl 3; .type 32; .endef

.seh\_proc \_Z41\_\_static\_initialization\_and\_destruction\_0ii

\_Z41\_\_static\_initialization\_and\_destruction\_0ii:

.LFB6438:

pushq %rbp

.seh\_pushreg %rbp

movq %rsp, %rbp

.seh\_setframe %rbp, 0

subq $32, %rsp

.seh\_stackalloc 32

.seh\_endprologue

movl %ecx, 16(%rbp)

movl %edx, 24(%rbp)

cmpl $1, 16(%rbp)

jne .L6

cmpl $65535, 24(%rbp)

jne .L6

leaq \_ZStL8\_\_ioinit(%rip), %rcx

call \_ZNSt8ios\_base4InitC1Ev

leaq \_\_tcf\_0(%rip), %rcx

call atexit

.L6:

nopaddq $32, %rsp

popq %rbp

ret

.seh\_endproc

.def \_GLOBAL\_\_sub\_I\_main; .scl 3; .type 32; .endef

.seh\_proc \_GLOBAL\_\_sub\_I\_main

\_GLOBAL\_\_sub\_I\_main:

.LFB6440:

pushq %rbp

.seh\_pushreg %rbp

movq %rsp, %rbp

.seh\_setframe %rbp, 0

subq $32, %rsp

.seh\_stackalloc 32

.seh\_endprologue

movl $65535, %edx

movl $1, %ecx

call \_Z41\_\_static\_initialization\_and\_destruction\_0ii

nop

addq $32, %rsp

popq %rbp

ret.seh\_endproc.section .ctors,"w".align 8.quad \_GLOBAL\_\_sub\_I\_main

.ident "GCC: (x86\_64-posix-seh-rev0, Built by MinGW-W64 project) 7.3.0"

.def \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc; .scl 2; .type 32.endef

.def getchar; .scl 2; .type 32; .endef

.def \_ZNSt8ios\_base4InitD1Ev; .scl 2; .type 32; .endef

.def \_ZNSt8ios\_base4InitC1Ev; .scl 2; .type 32; .endef

.def atexit; .scl 2; .type 32; .endef

.section .rdata$.refptr.\_ZSt4cout, "dr"

.globl .refptr.\_ZSt4cout

.linkonce discard

.refptr.\_ZSt4cout:

.quad \_ZSt4cout

# Используемые ресурсы

1. Library Windows: [Электронный ресурс] URL: https://msdn.microsoft.com/enus/library/windows/desktop/ms724911(v=vs.85).aspx (дата обращения: 05.05.2018).
2. Подключенные устройства и их список: [Электронный ресурс] URL: http://www.cyberforum.ru/win-api/thread1970481.html (дата обращения: 05.05.2018).
3. Лабораторная работа Лылова А.В. «Организация ЭВМ и систем»