

Міністерство освіти і науки, молоді та спорту України

Національний аерокосмічний університет ім. М.Є.
Жуковського

Кафедра 503

Лабораторна робота № 1
з дисципліни

«Системне програмування»

Тема: «Изучение Win32 API для получения информации об
ошибке, системе, для перекодирования текстов из одного
формата в другой»

Виконав: ст. гр. 535а

Целоусов А.А.

Перевірив: асистент каф.

503 Мозговий М.В.

Харків 2020

1. Задание I:

Необходимо написать программу, которая бы генерировала ошибку в ходе выполнения системного вызова и выдавала системное описание данной ошибки. Вторым режимом работы данной программы должен быть вывод информации о состоянии системы. Режим запуска программы определяется ключом, передаваемым в командной строке (-e – печать ошибки, -s – печать информации о системе).

2. Текст программы:

```
#include "pch.h"
#include <iostream>
#include "windows.h"
#include "stdio.h"
#include "string.h"
#include <locale.h>

void PrintError(); void
PrintSystemInfo();

int main( int argc, char* argv[])
{ setlocale(LC_ALL, "Russian");
  printf("***** KEYS *****\n");
  printf("-e - show error code and
  description\n"); printf("-s - show system
  information\n"); if (argc != 2)
  { printf_s("No input! Try again\n");
  }
  else
  { if (strcmp(argv[1], "-e") == 0)
    {
      PrintError();
    }
    else if (strcmp(argv[1], "-s") == 0)
    {
      PrintSystemInfo();
    }
    else
    { printf("No such key. Input the right one and try again!\n");
    }
  }
  system("pause");
  return 0;
}

void PrintError()
{
  MEMORYSTATUS memory_status;
  GlobalMemoryStatus(&memory_status);
  if (LocalAlloc(LPTR, memory_status.dwTotalPhys * (3)) == NULL) {
    LPVOID error_message;
    DWORD error_code = GetLastError();
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
    NULL, error_code,
```

```

        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&error_message, 0, NULL);
    printf("***** ERROR *****\n");
    wprintf(L"Error code: 0x%x\n", error_code); wprintf(L>About
error: %s\n", (char*)error_message); }
}

void PrintSystemInfo()
{
    SYSTEM_INFO system_info;
    GetSystemInfo(&system_info);
    printf("***** SYSTEM INFO
*****\n"); printf("Oem Id: %u\n",
system_info.dwOemId);
    printf("The processor architecture: %u\n", system_info.wProcessorArchitecture);
    printf("Reserverd member for future use: %u\n", system_info.wReserved);
    printf("Page size: %u\n", system_info.dwPageSize);
    printf("MIN application address: %lx\n", system_info.lpMinimumApplicationAddress);
    printf("MAX application address: %lx\n", system_info.lpMaximumApplicationAddress);
    printf("Active processor mask: %u\n", system_info.dwActiveProcessorMask);
    printf("Number of processors: %u\n", system_info.dwNumberOfProcessors);
    printf("Processor type: %u\n", system_info.dwProcessorType);
    printf("Allocation granularity address: %u\n", system_info.dwAllocationGranularity);
}

```

3.Скриншот:

```

C:\Users\Acer>cd /d C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication1\x64\Debug

C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication1\x64\Debug>ConsoleApplication1.exe -e
***** KEYS *****
-e - show error code and description
-s - show system information
***** ERROR *****
Error code: 0x8
About error: Недостаточно ресурсов памяти для обработки этой команды.

Для продолжения нажмите любую клавишу . . .

C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication1\x64\Debug>ConsoleApplication1.exe -s
***** KEYS *****
-e - show error code and description
-s - show system information
***** SYSTEM INFO *****
Oem Id: 9
The processor architecture: 9
Reserverd member for future use: 0
Page size: 4096
MIN application address: 10000
MAX application address: fffeffff
Active processor mask: 15
Number of processors: 4
Processor type: 8664
Allocation granularity address: 65536
Для продолжения нажмите любую клавишу . . .

```

4.Задание II.

Написать программу, позволяющую выполнять перекодировку текста из ASCII в Юникод и обратно. В качестве входных данных выступает файл с текстом. Программа при запуске получает параметр командной строки определяющий исходную кодировку файла (-a – ANSI файл, -u – Юникод файл).

5.Текст программы

```
#include "pch.h"
#include <iostream>
#include "windows.h"
#include "stdio.h"
#include "string.h"
#include <locale.h>

void ANSI_To_Unicode(LPCSTR);
void Unicode_To_ANSI(LPCSTR);
void ShowError();

int main(int argc, char* argv[])
{ setlocale(LC_ALL, "rus");
  printf("Введите ключ необходимой операции:\n");
  printf("\t* ключи исходной кодировки файла:\n\t\t-a - ANSI файл;\n \t\t-u - Unicode файл.\n"); printf("\t* Полный путь к файлу с занесенными данными.\n"); printf("Пример: -x 'C:\\path\\to\\file' \n");

  if (argc != 3)
  { printf("Ошибка! При вводе данных!\n");
    printf("\t* ключи исходной кодировки файла:\n\t\t-a - ANSI файл;\n \t\t-u - Unicode файл.\n"); printf("\t* Полный путь к файлу с занесенными данными.\n"); printf("Пример: -x 'C:\\path\\to\\file' \n");
    return 0;
  }
  else
  { if (strcmp(argv[1], "-a") == 0)
    ANSI_To_Unicode(argv[2]);
    else if (strcmp(argv[1], "-u") == 0)
    Unicode_To_ANSI(argv[2]); else
    printf("Ошибка ввода ключа\n"); } return 0;
}

void ANSI_To_Unicode(LPCSTR fileName)
{
  WCHAR* unicodeBuff;
  DWORD countRdChars, countWrWChars;
  HANDLE handleSourceFile, handleDestinationFile;
  CHAR ansiBuff[1024];
  BOOL resultFile = true;
  CHAR nameDestinationFile[] =
"C:\\Users\\Acer\\source\\repos\\ConsoleApplication2\\ANSI_Text_To_Unicode.txt";
  printf("\nРезультат можно увидеть -> %s\n", nameDestinationFile);
```

```

        handleSourceFile = CreateFile(fileName, GENERIC_READ, FILE_SHARE_READ,
        NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

        if (handleSourceFile == INVALID_HANDLE_VALUE)
        {
            ShowError();
            exit(EXIT_FAILURE); }

        handleDestinationFile = CreateFile(nameDestinationFile, GENERIC_WRITE,
        0, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);

        if (handleDestinationFile == INVALID_HANDLE_VALUE)
        {
            ShowError();
            exit(EXIT_FAILURE); }

        while (((resultFile = ReadFile(handleSourceFile, &ansiBuff,
        sizeof(ansiBuff), &countRdChars, NULL)) != false)
        && (countRdChars != 0))
        { int sizeToAllocate = MultiByteToWideChar(CP_UTF8,
        0, ansiBuff, countRdChars, NULL, 0); unicodeBuff = (WCHAR*)calloc(sizeToAllocate,
        sizeof(WCHAR));

            MultiByteToWideChar(CP_UTF8, 0, ansiBuff,
        countRdChars, unicodeBuff, sizeToAllocate); if (!WriteFile(handleDestinationFile,
        unicodeBuff,
        sizeToAllocate * sizeof(WCHAR), &countWrWChars, NULL)) ||
        (countWrWChars != sizeToAllocate *
        sizeof(WCHAR)))
            {
                ShowError();
                exit(EXIT_FAILURE); }
            free(unicodeBuff);
        }

        if (resultFile == false && countRdChars != 0)
        {
            ShowError();
            exit(EXIT_FAILURE);
        }

        CloseHandle(handleSourceFile);
        CloseHandle(handleDestinationFile);
    }

    void Unicode_To_ANSI(LPCSTR fileName)
    {
        WCHAR* unicodeBuff;
        DWORD countReadBytes, countWriteBytes;
        HANDLE handleSourceFile, handleDestinationFile;
        CHAR rawBytesBuff[1024];
        CHAR* ansiBuff;
        BOOL resultFile = true;
        CHAR nameDestinationFile[] =
        "C:\\Users\\Acer\\source\\repos\\ConsoleApplication2\\Unicode_Text_To_ANSI.txt";
        printf("\nРезультат можно увидеть -> %s\n", nameDestinationFile);
    }

```

```

        handleSourceFile = CreateFile(fileName, GENERIC_READ, FILE_SHARE_READ,
        NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

        if (handleSourceFile == INVALID_HANDLE_VALUE)
        {
            ShowError();
            exit(EXIT_FAILURE); }
        handleDestinationFile = CreateFile(nameDestinationFile, GENERIC_WRITE,
0, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);

        if (handleDestinationFile == INVALID_HANDLE_VALUE)
        {
            ShowError();
            exit(EXIT_FAILURE);
        } if (!ReadFile(handleSourceFile,
&rawBytesBuff, 2, &countReadBytes, NULL))
        {
            ShowError();
            exit(EXIT_FAILURE); }

        while (((resultFile = ReadFile(handleSourceFile, &rawBytesBuff,
sizeof(rawBytesBuff), &countReadBytes, NULL)) != false)
        && (countReadBytes != 0))
        { int unicodeLength = MultiByteToWideChar(CP_UTF8, 0,
rawBytesBuff, countReadBytes, NULL, 0); unicodeBuff = (WCHAR*)calloc(unicodeLength,
sizeof(WCHAR));

            MultiByteToWideChar(CP_UTF8, 0, rawBytesBuff,
countReadBytes, unicodeBuff, unicodeLength); int ascilenght =
            WideCharToMultiByte(CP_ACP, 0,
unicodeBuff, unicodeLength, NULL, 0, NULL, NULL); ansiBuff = (CHAR*)calloc(ascilenght,
sizeof(CHAR)); WideCharToMultiByte(CP_ACP, 0,
unicodeBuff,
unicodeLength, ansiBuff, ascilenght, NULL, NULL); if (!WriteFile(handleDestinationFile,
ansiBuff,
ascilenght, &countWriteBytes, NULL)) || (countWriteBytes != ascilenght))
            {
                ShowError();
                exit(EXIT_FAILURE);
            }
            free(unicodeBuff);
            free(ansiBuff); }

        if (resultFile == false && countReadBytes != 0)
        {
            ShowError();
            exit(EXIT_FAILURE);
        }

        CloseHandle(handleSourceFile);
        CloseHandle(handleDestinationFile);
    }
    void ShowError()
    { setlocale(LC_ALL, "rus");
        LPVOID error_message;
        DWORD error_code = GetLastError();
        FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL, error_code,

```

```

        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&error_message, 0, NULL);
printf("***** ERROR *****\n");
wprintf(L"Error code: 0x%x\n", error_code); wprintf(L>About
error: %s\n", (char*)error_message); }

```

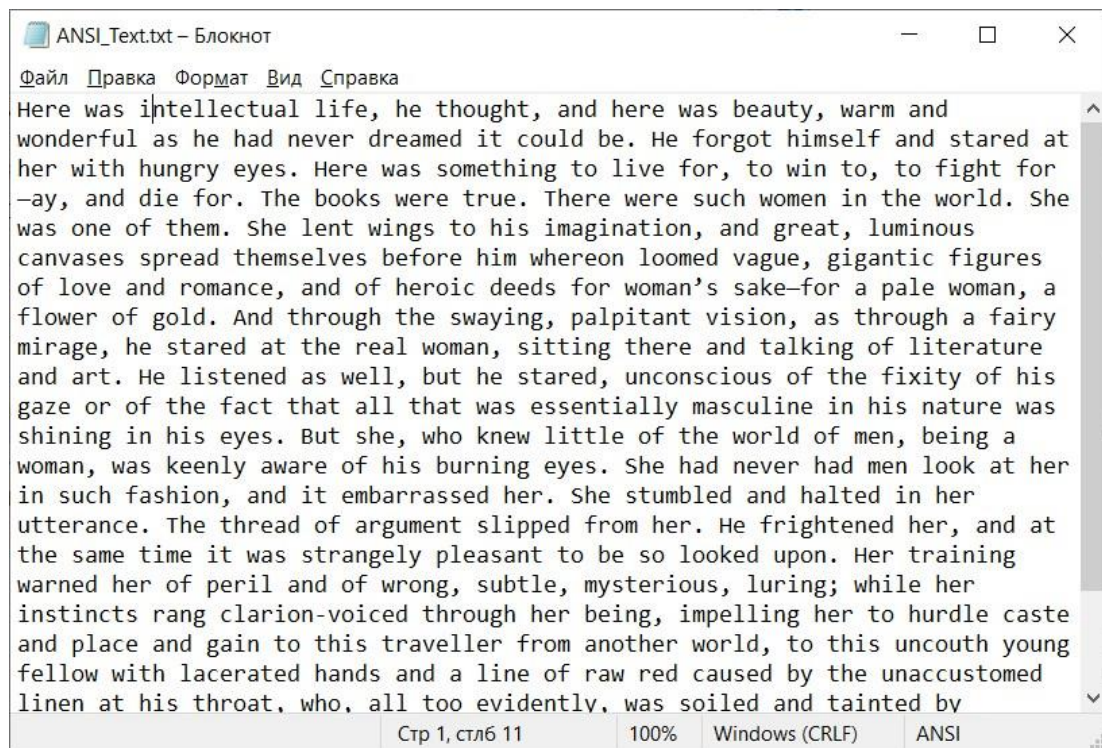
6.Скриншот - ANSI to UNICODE

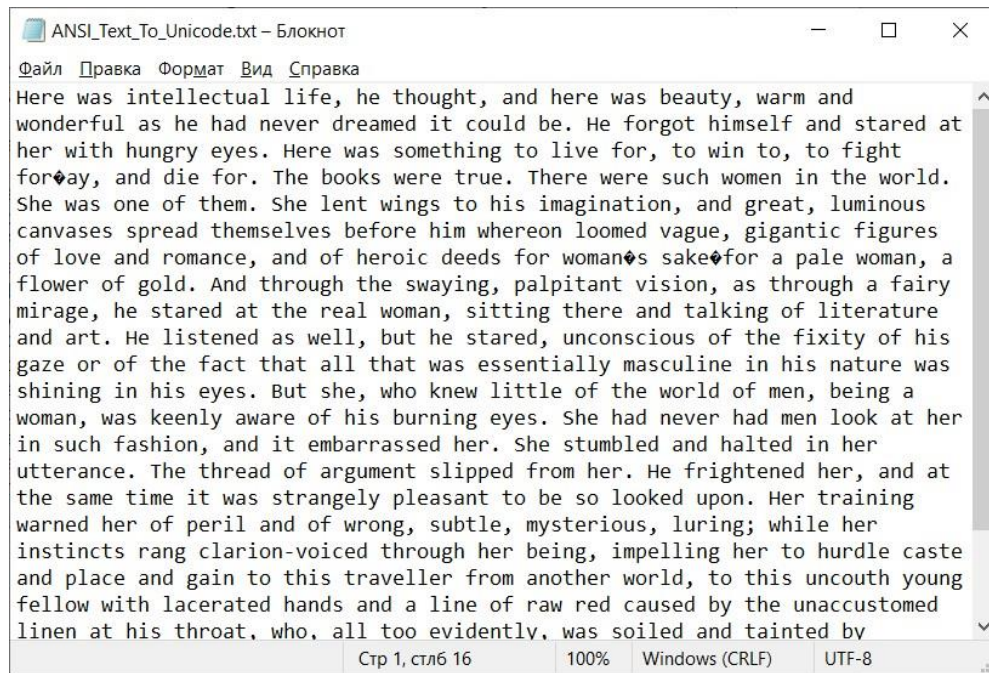
```

C:\Users\Acer>cd /d C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\x64\Debug
C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\x64\Debug>ConsoleApplication2.exe -a 'C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\ANSI_Text.txt'
Введите ключ необходимой операции:
* ключи исходной кодировки файла:
  -a - ANSI файл;
  -u - Unicode файл.
* Полный путь к файлу с занесенными данными.
Пример: -x 'C:\path\to\file'

Результат можно увидеть -> C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\ANSI_Text_To_Unicode.txt

```

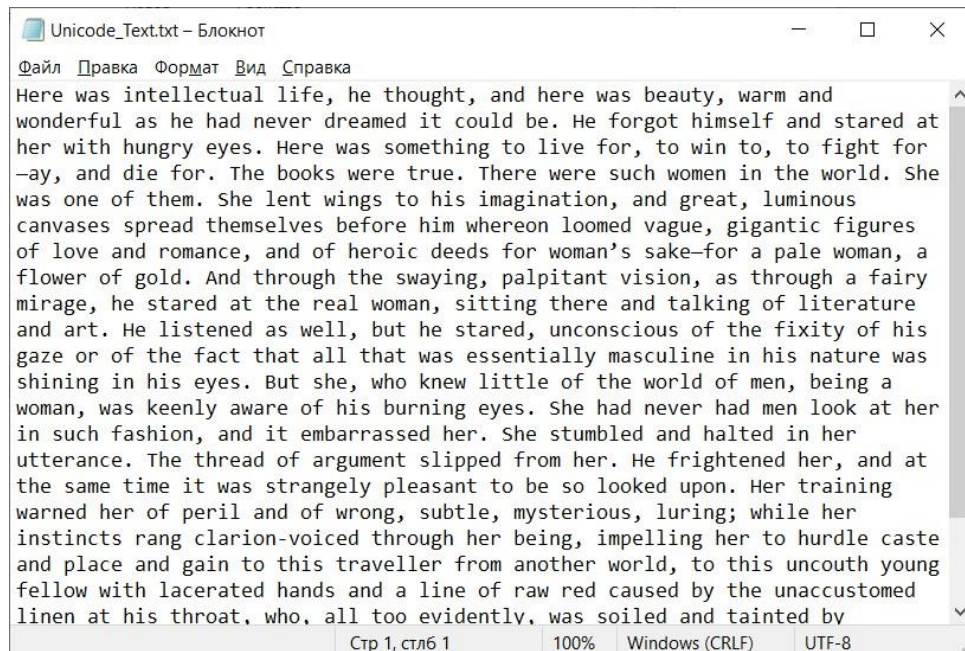


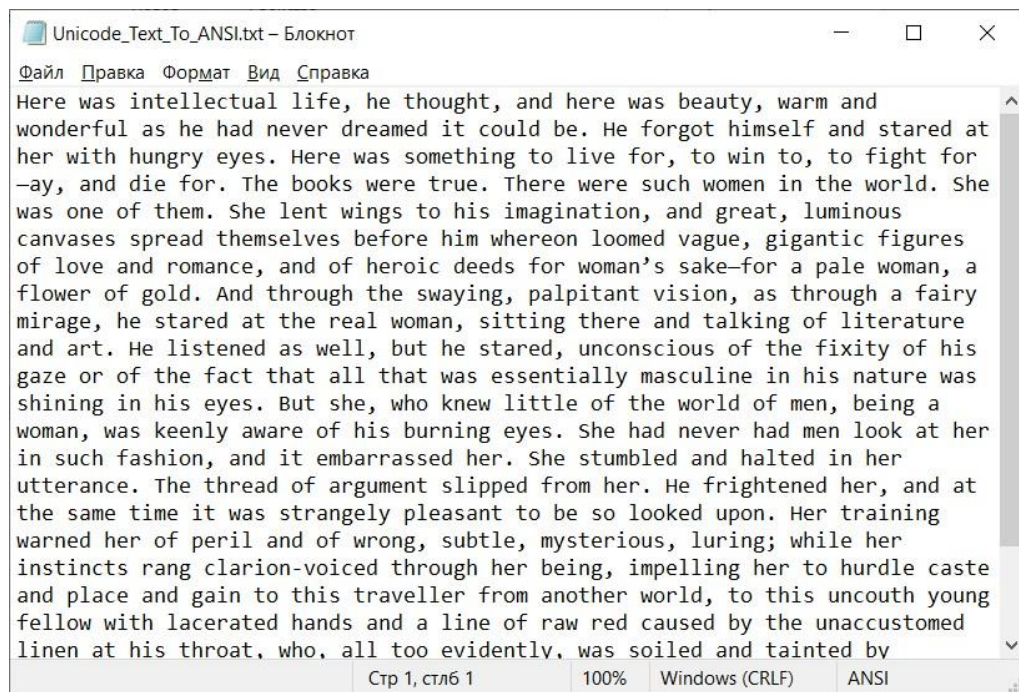


Unicode to ANSI

```
C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\x64\Debug>ConsoleApplication2.exe -u 'C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\Unicode_Text.txt'
Введите ключ необходимой операции:
* ключи исходной кодировки файла:
    -a - ANSI файл;
    -u - Unicode файл.
* Полный путь к файлу с занесенными данными.
Пример: -x 'C:\path\to\file'

Результат можно увидеть -> C:\Users\Acer\Desktop\3.2\СП\SP-Course\SysProgLab1\ConsoleApplication2\Unicode_Text_To_ANSI.txt
```





Unicode_Text_To_ANSI.txt – Блокнот

Файл Правка Формат Вид Справка

Here was intellectual life, he thought, and here was beauty, warm and wonderful as he had never dreamed it could be. He forgot himself and stared at her with hungry eyes. Here was something to live for, to win to, to fight for –ay, and die for. The books were true. There were such women in the world. She was one of them. She lent wings to his imagination, and great, luminous canvases spread themselves before him whereon loomed vague, gigantic figures of love and romance, and of heroic deeds for woman's sake—for a pale woman, a flower of gold. And through the swaying, palpitant vision, as through a fairy mirage, he stared at the real woman, sitting there and talking of literature and art. He listened as well, but he stared, unconscious of the fixity of his gaze or of the fact that all that was essentially masculine in his nature was shining in his eyes. But she, who knew little of the world of men, being a woman, was keenly aware of his burning eyes. She had never had men look at her in such fashion, and it embarrassed her. She stumbled and halted in her utterance. The thread of argument slipped from her. He frightened her, and at the same time it was strangely pleasant to be so looked upon. Her training warned her of peril and of wrong, subtle, mysterious, luring; while her instincts rang clarion-voiced through her being, impelling her to hurdle caste and place and gain to this traveller from another world, to this uncouth young fellow with lacerated hands and a line of raw red caused by the unaccustomed linen at his throat, who, all too evidently, was soiled and tainted by

Стр 1, столб 1 100% Windows (CRLF) ANSI

Выводы

В результате выполнения лабораторной работы были изучены системные вызовы Win32 API, позволяющие получить информацию об ошибке, изучены функции сбора информации о системе, перекодировки текста из ASCII в Unicode и наоборот.