

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ІКНІ  
Кафедра ПЗ

**ЗВІТ**

до лабораторної роботи № 7

на тему: *“Опрацювання рядка символів засобами асемблера мікропроцесорів x86. Робота з файлами”*  
з дисципліни: “Архітектура комп’ютера”

**Лектор:**

доцент кафедри ПЗ  
Крук О.Г.

**Виконав:**

студент групи ПЗ-22  
Коваленко Д.М.

**Прийняв:**

доцент кафедри ПЗ  
Крук О.Г.

«\_\_\_\_\_» \_\_\_\_\_ 2022 р.  
 $\Sigma$  = .....

**Тема.** Опрацювання рядка символів засобами асемблера мікропроцесорів x86. Робота з файлами.

**Мета.** Освоїти команди асемблера для роботи з рядками символів; опанувати функції Win32 для роботи з файлами; розвинути навички складання програми для опрацювання рядка символів та програми для створення, записування і читання текстового файла; відтранслювати і виконати в режимі відлагодження програми, складені відповідно до свого індивідуального завдання.

## Індивідуальне завдання

ПЗ-21		ПЗ-22		ПЗ-23	
№	Послідовність	№	Послідовність	№	Послідовність
1	П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub>	1	П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>	1	П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub>
2	П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	2	П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	2	П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>
3	П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub>	3	П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub>	3	П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub>
4	П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub>	4	П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub>	4	П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub>
5	П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub>	5	П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub>	5	П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub>
6	П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub>	6	П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub>	6	П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub>
7	П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub>	7	П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>	7	П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub>
8	П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	8	П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	8	П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>

## Хід роботи

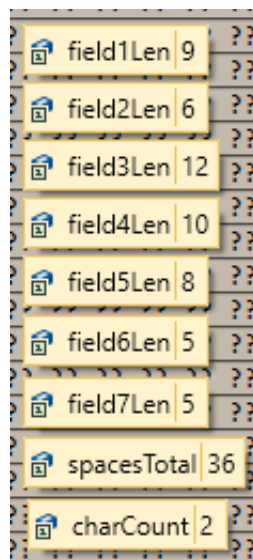


Рис. 1: Значення змінних

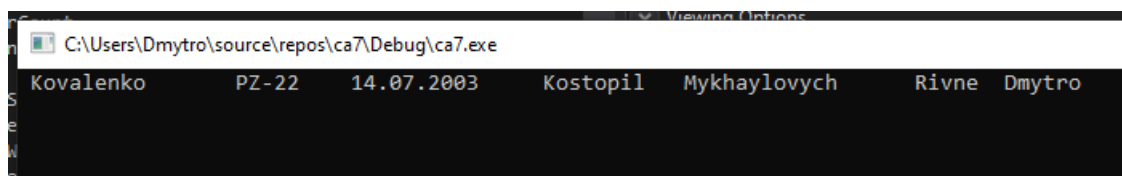


Рис. 2: Вивід у консоль програми

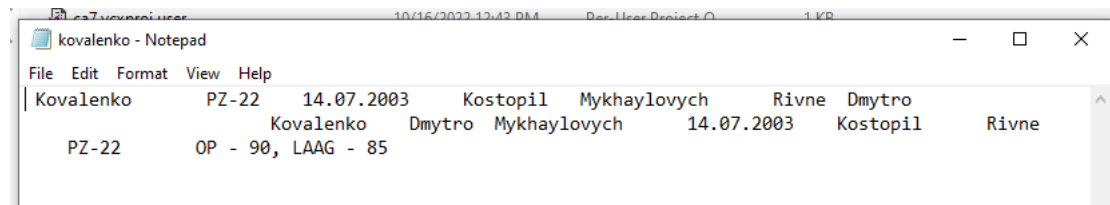


Рис. 3: Текстовый файл створений програмою

## Код програми

```
includelib C:/Irvine32.Lib
```

```
INCLUDE C:/Irvine32.inc
```

### **.DATA**

```
text          BYTE "Kovalenko      Dmytro  Mykhaylovych      14.07.2003      Kostopil
              Rivne          PZ-22      ", 0, 0
```

```
text2         BYTE 123 DUP (?)
```

```
text3         BYTE 93 DUP (?)
```

```
textLen       DWORD 93
```

```
text2Len      DWORD 123
```

```
text3Len      DWORD 93
```

```
field1        BYTE 22 DUP (?)
```

```
field2        BYTE 22 DUP (?)
```

```
field3        BYTE 22 DUP (?)
```

```
field4        BYTE 22 DUP (?)
```

```
field5        BYTE 22 DUP (?)
```

```
field6        BYTE 22 DUP (?)
```

```
field7        BYTE 22 DUP (?)
```

```
field1Len     DWORD ?
```

```
field2Len     DWORD ?
```

```
field3Len     DWORD ?
```

```
field4Len     DWORD ?
```

```
field5Len     DWORD ?
```

```
field6Len     DWORD ?
```

```
field7Len     DWORD ?
```

```
spacesTotal   DWORD 0
```

```
wordStart     DWORD 0
```

```
filename      BYTE "kovalenko.txt", 0
```

```
fileHandle    DWORD ?
```

```
bytesWritten  DWORD ?
```

```
byteCount     DWORD ?
```

```
charCount     DWORD 0
```

```
strToWrite    BYTE "OP - 90, LAAG - 85"
```

```
strToWriteLen DWORD 18
```

```
nil           BYTE 0, 0
```

### **.CODE**

```
spacesLen PROC start DWORD
```

```
    mov EDI, OFFSET text
```

```
    add EDI, start
```

```
    mov ESI, EDI
```

```
    mov AL, ' '
```

```
    mov ECX, textLen
```

```
    repe scasb
```

```
    dec EDI
```

```

    sub EDI, ESI
    mov EAX, EDI
    mov EBX, EAX
    add EBX, start
    ret
spacesLen ENDP

fieldLen PROC start :DWORD
    mov EAX, start
    mov wordStart, EAX
    mov EDI, OFFSET text
    add EDI, start
    mov ESI, EDI
    mov AL, ' '
    mov ECX, textLen
    repne scasb
    dec EDI
    sub EDI, ESI
    mov EAX, EDI
    mov EBX, EAX
    add EBX, start
    ret
fieldLen ENDP

main PROC
    invoke fieldLen, 0
    mov field1Len, EAX

    mov ECX, field1len      ; 1
    mov ESI, OFFSET text
    mov EDI, OFFSET field1
    rep movsb

    invoke spacesLen, EBX
    add spacesTotal, EAX

    invoke fieldLen, EBX    ; 2
    mov field2Len, EAX

    mov ECX, field2len
    mov ESI, OFFSET text
    add ESI, wordStart
    mov EDI, OFFSET field2
    rep movsb

    invoke spacesLen, EBX
    add spacesTotal, EAX

    invoke fieldLen, EBX    ; 3
    mov field3Len, EAX

    mov ECX, field3len
    mov ESI, OFFSET text
    add ESI, wordStart
    mov EDI, OFFSET field3
    rep movsb

    invoke spacesLen, EBX
    add spacesTotal, EAX

    invoke fieldLen, EBX    ; 4
    mov field4Len, EAX

```

```

mov ECX, field4len
mov ESI, OFFSET text
add ESI, wordStart
mov EDI, OFFSET field4
rep movsb

invoke spacesLen, EBX
add spacesTotal, EAX

invoke fieldLen, EBX      ; 5
mov field5Len, EAX

mov ECX, field5len
mov ESI, OFFSET text
add ESI, wordStart
mov EDI, OFFSET field5
rep movsb

invoke spacesLen, EBX
add spacesTotal, EAX

invoke fieldLen, EBX      ; 6
mov field6Len, EAX

mov ECX, field6len
mov ESI, OFFSET text
add ESI, wordStart
mov EDI, OFFSET field6
rep movsb

invoke spacesLen, EBX
add spacesTotal, EAX

invoke fieldLen, EBX      ; 7
mov field7Len, EAX

mov ESI, OFFSET text
add ESI, wordStart
mov EDI, OFFSET field7
mov ECX, field7len
rep movsb

invoke spacesLen, EBX
add spacesTotal, EAX

; 1, 7, 4, 5, 3, 6, 2

mov AL, ' '
mov EDI, OFFSET text2

mov ECX, 1                  ; 1
rep stosb
mov ESI, OFFSET field1
mov ECX, field1len
rep movsb

mov ECX, 7                  ; 7
rep stosb
mov ESI, OFFSET field7
mov ECX, field7len
rep movsb

mov ECX, 4                  ; 4

```

```

rep stosb
mov ESI, OFFSET field4
mov ECX, field4len
rep movsb

```

```

mov ECX, 5 ; 5
rep stosb
mov ESI, OFFSET field5
mov ECX, field5len
rep movsb

```

```

mov ECX, 3 ; 3
rep stosb
mov ESI, OFFSET field3
mov ECX, field3len
rep movsb

```

```

mov ECX, 6 ; 6
rep stosb
mov ESI, OFFSET field6
mov ECX, field6len
rep movsb

```

```

mov ECX, 2 ; 2
rep stosb
mov ESI, OFFSET field2
mov ECX, field2len
rep movsb

```

```

mov ESI, OFFSET nil
mov ECX, 2
rep movsb

```

```

mov EDX, OFFSET text2
call WriteString

```

```

INVOKE CreateFile ,
ADDR filename , GENERIC_WRITE, DO_NOT_SHARE, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, 0
mov fileHandle , EAX
INVOKE WriteFile ,
fileHandle , ADDR text2 , text2Len , ADDR bytesWritten , 0 ; Write text2
INVOKE WriteFile ,
fileHandle , ADDR text , textLen , ADDR bytesWritten , 0 ; Write text1
INVOKE CloseHandle ,
fileHandle

```

```

INVOKE CreateFile ,
ADDR filename , GENERIC_READ or GENERIC_WRITE, DO_NOT_SHARE, NULL,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0
mov fileHandle , EAX
INVOKE SetFilePointer ,
fileHandle , text2Len , 0, FILE_BEGIN
INVOKE ReadFile ,
fileHandle , ADDR text3 , text3Len , ADDR byteCount , 0

```

```

mov EDI, OFFSET text3
mov AL, 'K'
mov ECX, text3Len
cld
count:
repne scasb
jz countInc

```

```

    jnz countDone
countInc:
    inc charCount
    jmp count
countDone:
    INVOKE SetFilePointer ,
        fileHandle , 0, 0, FILE_END
    INVOKE WriteFile ,
        fileHandle , ADDR strToWrite , strToWriteLen , ADDR bytesWritten , 0

    INVOKE CloseHandle , fileHandle
    ret

main ENDP

END main

```

## Висновки

Під час виконання лабораторної роботи я освоїв команди асемблера для роботи з рядками символів; опанував функції Win32 для роботи з файлами; розвинув навички складання програми для опрацювання рядка символів та програми для створення, записування і читання текстового файла; відтранлював і виконати в режимі відлагодження програми, складені відповідно до свого індивідуального завдання.