

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота 1

Архітектура комп'ютера

Технологія розробки програм в архітектурі (x86)

Виконав:
Студент групи IT-01
Тимошенко Олексій

Київ 2021

Тема: технологія розробки програм в архітектурі (x86)

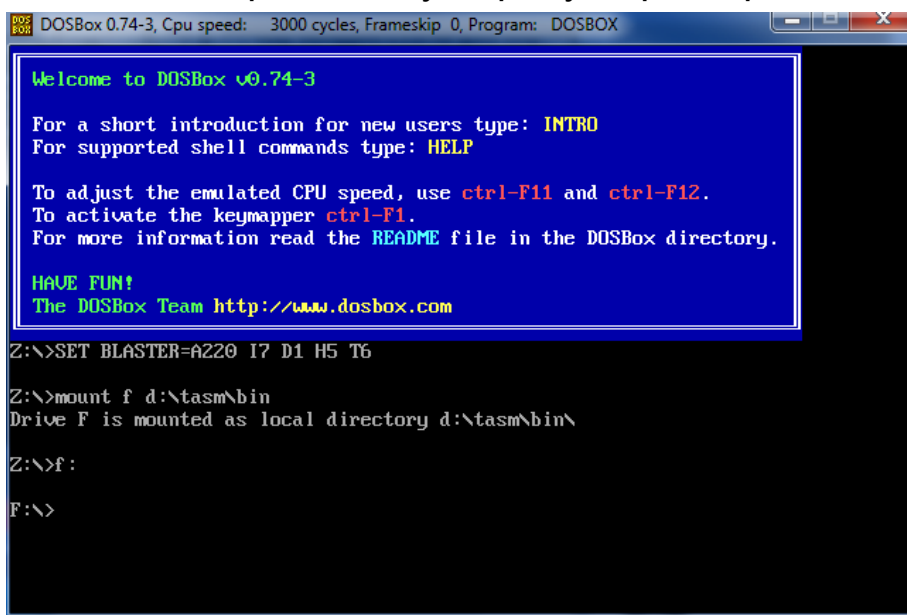
Мета: набути твердих навичок і знань технологічної основи розробки ПЗ на асемблері, в ході якої використовуються знання архітектури комп'ютера.

Файли доступні за посиланням:

<https://github.com/OlexiiT/assembler/tree/main/L1>

Хід роботи:

Відкриваємо завантажений DOSBox. Створимо віртуальний диск f за директорією, де знаходиться наш файл з розширенням .asm. Далі перейдемо у обрану директорію.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Welcome to DOSBox v0.74-3

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

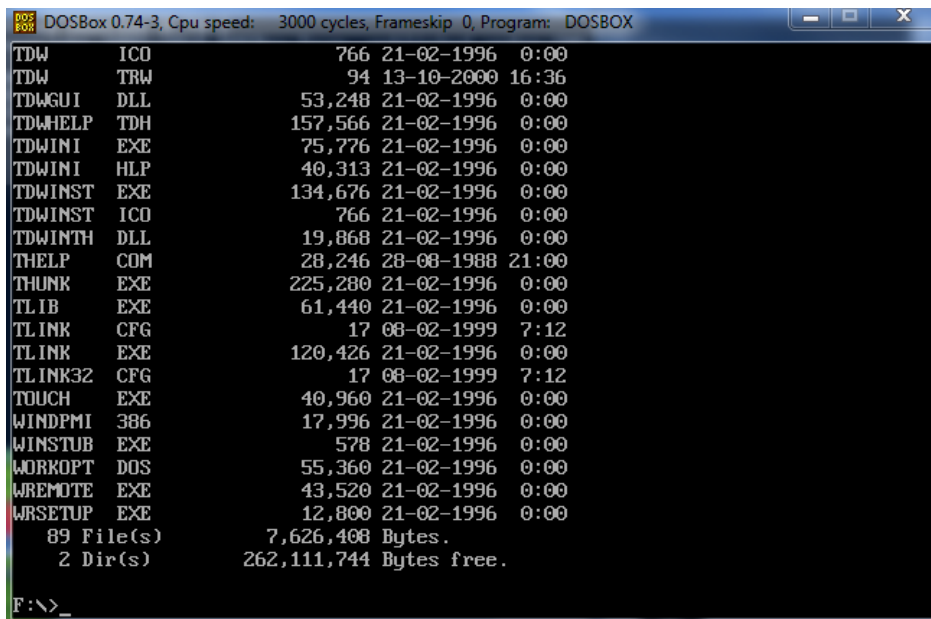
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount f d:\tasm\bin
Drive F is mounted as local directory d:\tasm\bin\

Z:\>f:

F:\>
```

Команда dir виведе усі файли, з даної директорії.



Змінюємо наш застосунок

```

1  TITLE Tymoshenko_L_1
2
3  IDEAL                ;Директива - тип Асемблера
4  MODEL small          ;Директива - тип моделі пам'яті
5  STACK 256           ;Директива - розмір стеку
6
7                      ;Початок сегменту даних
8  DATA SEG
9  exCode db 0
10 message db "Tymoshenko", 10, 13, '$' ;Рядки символів для виводу на екран
11 message2 db "Oleksii", 10, 13, '$'
12
13                      ;Початок сегменту коду
14 CODE SEG
15 Start:
16
17 ;-----Ініціалізація DS і ES-----
18 mov ax,@data          ;@data ідентифікатор, що створюється директивою model
19 mov ds,ax              ;Завантаження початку сегменту даних в регістр ds
20 mov es,ax              ;Завантаження початку сегменту даних в регістр es
21
22 ;-----Операції виводу на консоль-----
23 mov dx,offset message  ;Пересилання адреси рядка даних в регістр dx
24 mov ah,09h            ;Завантаження числа 09h до регістру ah (функція DOS 9h - команда виводу рядка на консоль)
25 int 21h               ;Виклик функції DOS 9h
26
27 mov dx,offset message2
28 mov ah,09h
29 int 21h
30
31 ;-----Операція зупинки програми й очікування натискання клавіші-----
32 mov ah,01h            ;Завантаження числа 1_ до регістру ah (функція DOS 1h - команда очікування натискання клавіші)
33 int 21h               ;Виклик функції DOS 1h
34
35 ;-----Вихід з програми-----
36 mov ah,4ch            ;Завантаження числа 4_ до регістру ah (функція DOS 4ch - команда виходу з програми)
37 mov al,[exCode]       ;Отримання коду виходу
38 int 21h               ;Виклик функції DOS 4ch
39 end Start

```

Асемблюємо застосунок за допомогою команди “`tasm hw1.asm`” (де `hw1.asm` - назва файлу з кодом)

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
THUNK   EXE           225,280 21-02-1996 0:00
TLIB    EXE           61,440 21-02-1996 0:00
TLINK   CFG           17      08-02-1999 7:12
TLINK   EXE          120,426 21-02-1996 0:00
TLINK32 CFG           17      08-02-1999 7:12
TOUCH   EXE           40,960 21-02-1996 0:00
WINDPMI 386           17,996 21-02-1996 0:00
WINSTUB EXE            578    21-02-1996 0:00
WORKOPT DOS           55,360 21-02-1996 0:00
WREMOTE EXE           43,520 21-02-1996 0:00
WRSETUP EXE           12,800 21-02-1996 0:00
  89 File(s)      7,626,408 Bytes.
   2 Dir(s)      262,111,744 Bytes free.

F:\>tasm hw1.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:   hw1.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  467k

F:\>
```

Після асемблювання утворюється файл з розширенням .obj (hw1.obj). Його необхідно перетворити у файл з розширенням .exe за допомогою команди "tlink hw1.obj".

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
TLINK   EXE          120,426 21-02-1996 0:00
TLINK32 CFG           17      08-02-1999 7:12
TOUCH   EXE           40,960 21-02-1996 0:00
WINDPMI 386           17,996 21-02-1996 0:00
WINSTUB EXE            578    21-02-1996 0:00
WORKOPT DOS           55,360 21-02-1996 0:00
WREMOTE EXE           43,520 21-02-1996 0:00
WRSETUP EXE           12,800 21-02-1996 0:00
  89 File(s)      7,626,408 Bytes.
   2 Dir(s)      262,111,744 Bytes free.

F:\>tasm hw1.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:   hw1.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  467k

F:\>tlink hw1.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

F:\>_
```

Якщо спробувати відкрити утворений файл (hw1.exe) отримаємо помилку, тому відкриємо його за допомогою DOSBox

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: HW1
WINDPMI 386 17,996 21-02-1996 0:00
WINSTUB EXE 578 21-02-1996 0:00
WORKOPT DOS 55,360 21-02-1996 0:00
WREMOTE EXE 43,520 21-02-1996 0:00
WRSETUP EXE 12,800 21-02-1996 0:00
88 File(s) 7,627,800 Bytes.
2 Dir(s) 262,111,744 Bytes free.

F:\>tasm hw1.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: hw1.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 468k

F:\>tlink hw1.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

F:\>hw1.exe
Tymoshenko
Oleksii

```

Використаємо турбо дебагер для покрокового дебагінгу коду

```

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[CPU 80486] 1=[↑][↓]
cs:0000 B87B08 mov ax,087B ax 0000 c=0
cs:0003 8ED8 mov ds,ax bx 0000 z=0
cs:0005 8EC0 mov es,ax cx 0000 s=0
cs:0007 BA0100 mov dx,0001 dx 0000 o=0
cs:000A B409 mov ah,09 si 0000 p=0
cs:000C CD21 int 21 di 0000 a=0
cs:000E BA0E00 mov dx,000E bp 0000 i=1
cs:0011 B409 mov ah,09 sp 0100 d=0
cs:0013 CD21 int 21 ds 0869
cs:0015 B401 mov ah,01 es 0869
cs:0017 CD21 int 21 ss 087D
cs:0019 B44C mov ah,4C cs 0879
cs:001B A00000 mov al,[0000] ip 0000

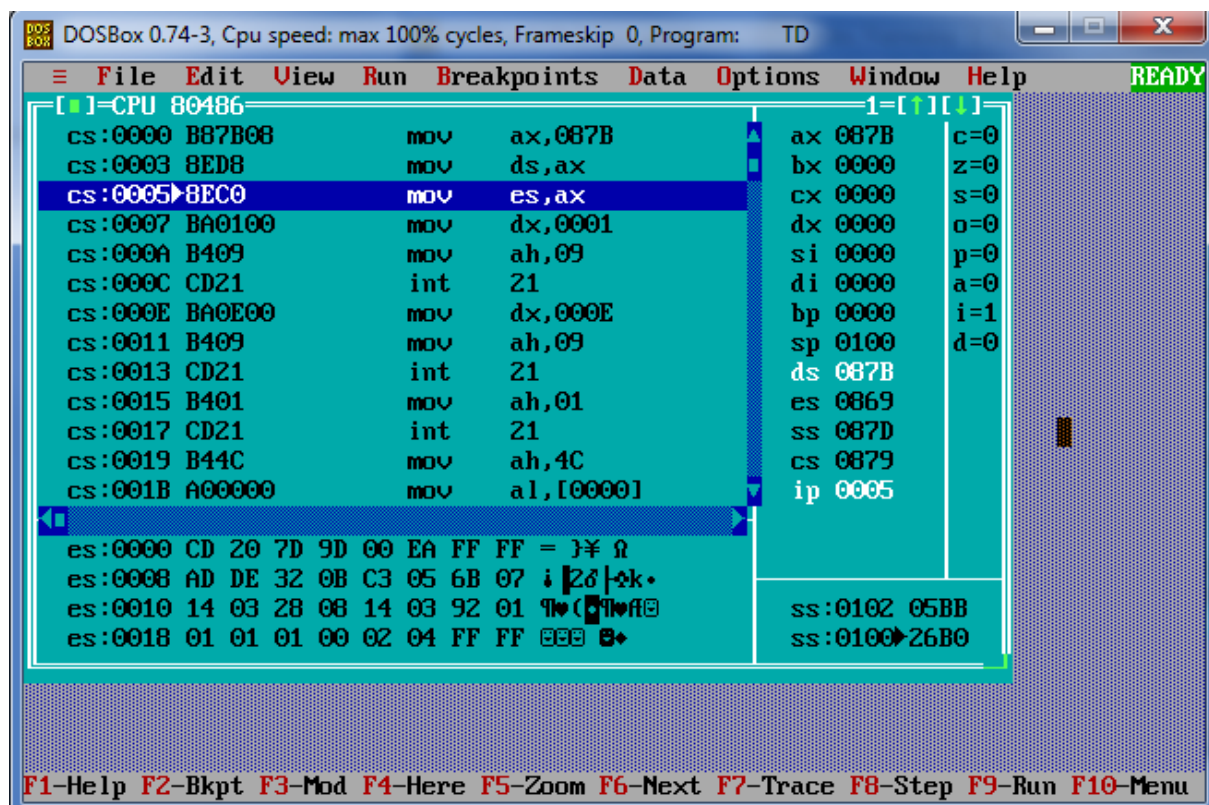
ds:0000 CD 20 7D 9D 00 EA FF FF = }¥ Ω
ds:0008 AD DE 32 0B C3 05 6B 07 i 26 |ek•
ds:0010 14 03 28 08 14 03 92 01 70 (P10ff
ds:0018 01 01 01 00 02 04 FF FF 000 0+

ss:0102 05BB
ss:0100 26B0

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

Висновки:



Білим кольором справа виділено регістри, що змінили своє значення після попереднього кроку

ax - призначений для операції вводу-виводу

bx - може зберігати адресу покажчика на масив

cx - лічильник

dx - призначений для вводу-виводу на зовнішні пристрої

si - індексна адресація

di - операції пересилання символічних рядків

bp, *sp* та *ss* - керують стеком

ds - містить початкову адресу сегменту даних

es - тимчасове значення адрес

cs - визначає адресу команду, що виконуватиметься

ip - зберігає значення ефективної адреси початку функції обробки переривання