Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра автоматики та управління в технічних системах

Лабораторна робота №2

з дисципліни «Архітектура комп'ютера» Варіант-3

Виконали: Студенти групи IT-01
Гончаренко А.А.
Чорній В.І.
Захищено з балом

Мета: набутті впевнених знань і навичок з розробки ПЗ на Асемблері для управління відеопам'яттю з урахуванням знань архітектури IA-32 y real address mode.

Хід роботи

1. Виконати повний цикл розробки, тестування і налагодження програмного забезпечення.

Для вирішення завдання був написаний код на Асемблері. (посилання на код <u>Чорнія</u> та <u>Гончаренка</u>).

На початку сегменту коду у регістрі bx задається початок прямокутника, у регістр еs задається початок сегменту відеопам'яті через регістр ах та задаємо кількість повторень циклу, що виводить перший рядок.

```
CODESEG
Start:

mov bx, 1604; задаємо початок прямокутника (80*2*10+2*2)
mov ах, 0B800h; задаємо початок сегменту відеопам'яті
mov еs, ах
mov сх, 20; задаємо кількість символів у 1 рядку
```

Результат виконання у турбодебагері:

```
■ 1=CPU 80486=
                           es:0644 = 3430<del>-</del>
cs:0000 BB4406
                                           ax B800
                     MOV
                            bx.0644
                            ax, B800
cs:0003 B800B8
                                           bx 0644
                                                      z=0
                     MOV
cs:0006 8EC0
                                           cx 0014
                     MOV
                            es,ax
                                                      s=0
                            cx,0014
cs:0008 B91400
                     MOV
                                           dx 0000
                                                      p=0
cs:000B>26C7073034
                            es:word ptr [
                                           si 0000
                    MOV
                            bx,0002
                     add
                                           di 0000
cs:0010 83C302
                                                      a=0
                                                      i=1
cs:0013 E2F6
                     loop
                            000B
                                           bp 0000
                                           sp 0100
cs:0015 83C378
                                                      d=0
                     add
                            bx,0078
                            cx,0014
cs:0018 B91400
                     MOV
                                           ds 48DD
cs:001B 26C7073034
                            es:word ptr [
                                           es B800
                     MOV
cs:0020 83C302
                     add
                            bx,0002
                                           ss 48F8
cs:0023 E2F6
                     loop
                            001B
                                           cs 48ED
cs:0025 83C378
                     add
                            bx,0078
                                           ip 000B
ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0010 C9 15 80 02 24 10 92 01 [§Ç©$▶ff©
                                           ss:0102 0000
ds:0018 01 01 01 00 02 FF FF FF 🖼 🛢
                                           ss:0100>0000
```

Наступним кроком виконується цикл виводу символів у рядок. У регістр ез записується символ. Після цього зміщується адреса у відеопам'яті для запису

наступного символа. Після виконання циклу адреса у відеопам'яті зміщується на 120 для переносу на наступний рядок та задається кількість повторень наступного циклу.

```
my_ptr:;створення циклу для 1 рядка
mov [word es:bx], 3430h; запис симола у відео пам'ять
add bx,2; зміщення адреси у відеопа'мяті
Loop my_ptr; виклик циклу
add bx, 120; переведення рядка(80*2-20*2)
mov cx,20; задаємо кількість символів у 1 рядку
```

Результат виконання у турбодебагері:

```
es:06E4 = 345C=
[ 1=CPIL 80486==
                          bx,0644
cs:0000 BB4406
                                        ax B800
                                                  c=0
                    MOU
cs:0003 B800B8
                                                  z=0
                                       bx 06E4
                          ax, B800
                    MOV
                  MOV
cs:0006 8EC0
                                        cx 0014
                          es,ax
cs:0008 B91400
                          cx,0014
                  MOV
                                       dx 0000
                                                  0=0
cs:000B 26C7073034
                          es:word ptr [
                                       si 0000
                                                  p=1
                  MOV
cs:0010 83C302
                   add b×,0002
                                       di 0000
                                                  a=1
cs:0013 E2F6
                   loop 000B
                                       bp 0000
                                                  i=1
                add
mov
                          bx,0078
cx,0014
cs:0015 83C378
                         bx,0078
                                       sp 0100
                                                  d=0
cs:0018 B91400
                                        ds 48DD
cs:001B>26C7073034 mov
                          es:word ptr [
                                       es B800
                    add
cs:0020 83C302
                          bx,0002
                                        ss 48F8
cs:0023 E2F6
                          001B
                                        cs 48ED
                    loop
cs:0025 83C378
                    add
                          bx,0078
                                       ip 001B
ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0010 C9 15 80 02 24 10 92 01 [89$ ft®
                                        ss:0102 0000
ds:0018 01 01 01 00 02 FF FF FF 🕮 🛢
                                        ss:0100>0000
```

Аналогічно написано наступні 9 циклів для виводу усіх рядків.

Завершується програма записом у регістр аһ функції виходу з програми, повернення вихідного коду та викликом функції завершення програми.

```
mov ah,04Ch; функція виходу з програми mov al,0; повернення вихідного коду int 21h; виклик функції 04Ch
```

Асемблюємо код командою tasm:

```
F:\>tasm laba2.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International
Serial No: Tester:
Assembling file: laba2.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 470k
```

Лінкуємо .obj файл командою tlink:

```
F:\>tlink laba2.obj
Turbo Link Version 5.1 Copyright (c) 1992 Borland International
```

Запускаємо .exe файл:

Висновки.

В ході лабораторної було набуто твердих навичок і знань технологічної основи розробки ПЗ на Асемблері для управління відеопам'яттю з урахуванням знань архітектури IA-32 y real address mode.