

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu:
Interfejsy komputerów cyfrowych

Sprawozdanie z ćwiczenia laboratoryjnego nr 4:
Programowanie klawiatury

Prowadzący:
mgr inż. Krzysztof Szajewski

Wykonał: Radosław Relidzyński

Grupa: WCY20IY4S1

Data laboratoriów: 30.05.2021 r.

Spis treści

| | | |
|----|---------------------|---|
| A. | Treść zadania | 2 |
| B. | Zadanie 1 | 2 |
| C. | Zadanie 2 | 3 |
| D. | Zadanie 3 | 5 |
| E. | Zadanie 4 | 6 |

A.Treść zadania

Napisać następujące programy realizujące następujące funkcje:

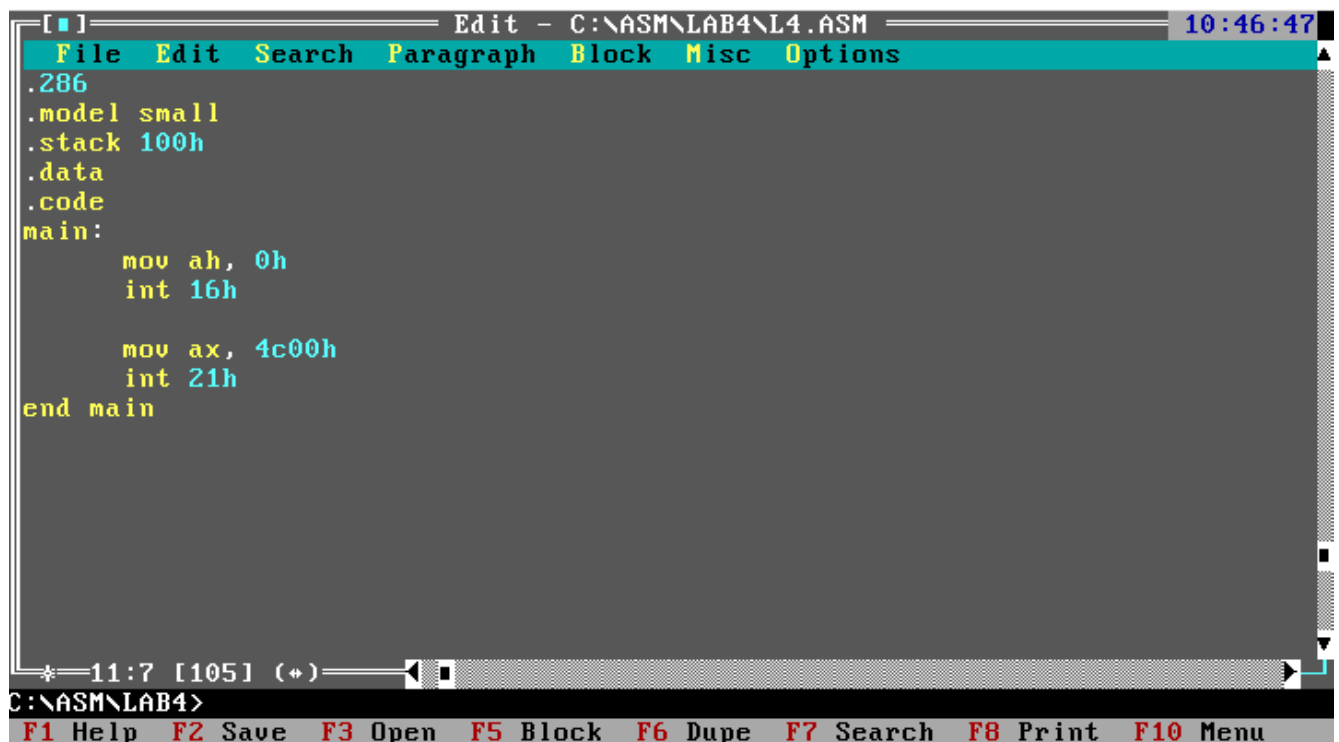
- odczytanie kodu ASCII i scan kodu wciśniętego klawisza
- odczytanie stanu klawiszy Caps Lock, Num Lock i Scroll Lock, Insert
- sterowanie diodami klawiatury (rozkaz EDH)
- ustawienie opóźnienia 500 ms i prędkość autorepetycji 5 znaków na sekundę (rozkaz F3H)

Wyświetlić obszaru danych BIOS dla klawiatury w turbo debugger – opisać strukturę tych danych.

Zobrazować wyniki działania programów – zrzuty ekranu z linii poleceń i z turbo debugger'a. Wyjaśnić działanie programów.

Opóźnienie ma być równe 1000 ms

B.Zadanie 1



```
[ ] Edit - C:\ASM\LAB4\L4.ASM 10:46:47
File Edit Search Paragraph Block Misc Options
.286
.model small
.stack 100h
.data
.code
main:
    mov ah, 0h
    int 16h

    mov ax, 4c00h
    int 21h
end main

*==11:7 [105] (*==
C:\ASM\LAB4>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu
```

MOV AH, 00H– pobranie kodu klawisza

INT 16H– wywołanie funkcji klawiatury

MOV AX, 4C00H – definiowanie funkcji zakończenia programu

AX=[AH, AL] – AL – kod klawisza, AH – scan code klawisza

INT 21H – wywołanie powyższej funkcji

Wchodzę w program turbo deggubera. Odczytuję wartości AH i AL:

Skrótem F4 uruchamiam program i wprowadzam literę „a”:

The screenshot shows the Turbo Debugger window with the following content:

- Menu Bar:** File, Edit, View, Run, Breakpoints, Data, Options, Window, Help.
- Assembly List:**
 - cs:0002 CD16 int 16
 - cs:0004 B8004C mov ax,4C00 (highlighted)
 - cs:0007 CD21 int 21
 - cs:0009 0000 add [bx+si],al
 - cs:000B 0000 add [bx+si],al
 - cs:000D 0000 add [bx+si],al
 - cs:000F 0000 add [bx+si],al
 - cs:0011 0000 add [bx+si],al
 - cs:0013 0000 add [bx+si],al
 - cs:0015 0000 add [bx+si],al
 - cs:0017 0000 add [bx+si],al
 - cs:0019 0000 add [bx+si],al
 - cs:001B 0000 add [bx+si],al
- Registers Panel:**
 - ax 1E61
 - bx 0000
 - cx 0000
 - dx 0000
 - si 0000
 - di 0000
 - bp 0000
 - sp 0100
 - ds 24D4
 - es 24D4
 - ss 24E5
 - cs 24E4
 - ip 0004
- Memory Dump:**
 - ds:0000 CD 20 20 9B 00 9A F0 FE = Ĥ Ū-■
 - ds:0008 1D F0 32 0B 6F 22 6B 07 +-28o''k•
 - ds:0010 C2 1F 28 08 C2 1F 40 1E T▼(T▼e▲
 - ds:0018 01 01 01 00 02 04 FF FF ☐☐☐☐ ☐+
- Stack Pointer:** ss:0102 0403, ss:0100 52FB

AH = 1Eh – scan code

AL. = 61h – kod ASCII dla „a”

C. Zadanie 2

Uruchamiam ponownie program. Tym razem wciskam kombinację CTRL + „a”.

```

- File Edit View Run Breakpoints Data Options Window Help READY
[ ]=CPU Pentium Pro
24E4:0000 B400 mov ah,00 ax 23B7 c=0
24E4:0002 CD16 int 16 bx 764C z=0
24E4:0004 B8004C mov ax,4C00 cx 01C7 s=0
24E4:0007 CD21 int 21 dx 01EE o=0
24E4:0009 0000 add [bx+sil],al si 23B7 p=0
24E4:000B 0000 add [bx+sil],al di 00D0 a=0
24E4:000D 0000 add [bx+sil],al bp 0314 i=0
24E4:000F 0000 add [bx+sil],al sp 031A d=0
24E4:0011 0000 add [bx+sil],al ds 0E90
24E4:0013 0000 add [bx+sil],al es 07B6
24E4:0015 0000 add [bx+sil],al ss 23B7
24E4:0017 0000 add [bx+sil],al cs 0000
24E4:0019 0000 add [bx+sil],al ip 0000
24E4:001B 0000 add [bx+sil],al
24E4:001D 0000 add [bx+sil],al
0040:0017 04 01 00 32 00 32 00 00 + 2 2 24E5:0108 0002
0040:001F 43 0D 1C E0 50 E0 48 E0 CŹ-óPÓHÓ 24E5:0106 0000
0040:0027 50 E0 48 E0 48 E0 50 00 PÓHÓHÓP 24E5:0104 000A
0040:002F 3E 06 21 E0 50 E0 48 E0 >?úPÓHÓ 24E5:0102 0403
0040:0037 50 E0 48 00 43 0D 1C 81 PÓH CŹ-ü 24E5:0100 52FB
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```

Sprawdzam adres 0040:0017. Jest tam na początku wartość 04, która binarnie wynosi 00000100.

Odpowienie miejsca rozwinięcia binarnego to aktywność danego klawisza, zgodnie z poniższym opisem:

Bajt 0040H:0017H

- bit 7 - aktywny jest tryb Insert
- bit 6 - aktywny jest tryb Caps Lock
- bit 5 - aktywny jest tryb Num Lock
- bit 4 - aktywny jest tryb Scroll Lock
- bit 3 - naciśnięto klawisz Alt (lewy lub prawy)
- bit 2 - naciśnięto klawisz Ctrl (lewy lub prawy)
- bit 1 - naciśnięto lewy klawisz Shift
- bit 0 - naciśnięto prawy klawisz Shift

Dla powyższej wartości widać, że wciśnięty był ctrl.

Sprawdzam dla innej kombinacji:

```

- File Edit View Run Breakpoints Data Options Window Help READY
[ ]=CPU Pentium Pro
24E4:0000 B400 mov ah,00 ax 23B7 c=0
24E4:0002 CD16 int 16 bx 764C z=0
24E4:0004 B8004C mov ax,4C00 cx 01C7 s=0
24E4:0007 CD21 int 21 dx 01EE o=0
24E4:0009 0000 add [bx+sil,al] si 23B7 p=0
24E4:000B 0000 add [bx+sil,al] di 00D0 a=0
24E4:000D 0000 add [bx+sil,al] bp 0314 i=0
24E4:000F 0000 add [bx+sil,al] sp 031A d=0
24E4:0011 0000 add [bx+sil,al] ds 0E90
24E4:0013 0000 add [bx+sil,al] es 07B6
24E4:0015 0000 add [bx+sil,al] ss 23B7
24E4:0017 0000 add [bx+sil,al] cs 0000
24E4:0019 0000 add [bx+sil,al] ip 0000
24E4:001B 0000 add [bx+sil,al]
24E4:001D 0000 add [bx+sil,al]

0040:0017 46 01 00 3C 00 3C 00 00 F0 < < 24E5:0108 0002
0040:001F 43 0D 1C E0 50 E0 48 E0 CF-ÓPÓHÓ 24E5:0106 0000
0040:0027 50 E0 48 E0 48 E0 50 00 PÓHÓHÓP 24E5:0104 000A
0040:002F 3E 06 21 0D 1C 00 3E 0D >?P- >P 24E5:0102 0403
0040:0037 1C 00 3E 12 13 0D 1C 81 - >!!P-ü 24E5:0100 52FB

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

```

46h = 0100 0110

Z kolei tutaj widać, że był włączony caps lock, ctrl i shift.

D. Zadanie 3

```

Edit - C:\ASM\LAB4\L4.ASM 11:37:41
File Edit Search Paragraph Block Misc Options
_286
.model small
.stack 100h
.data
.code
main:
    mov al, 0EDh
    out 60h, al

    czekaj:
        in al, 64h
        test al, 02h
        jnz czekaj

    mov al, 00000111b
    out 60h, al

    mov ax, 4c00h
    int 21h
1:1 [046] (*)
C:\ASM\LAB4>
F1 Help F2 Save F3 Open F5 Block F6 Dupe F7 Search F8 Print F10 Menu

```

Label o nazwie „czekaj” czeka na możliwość wykonania instrukcji. Oczekiwanie służy temu, aby program nie ingerował w bieżące działanie procesora.

Cykl działania programu:

1. Aktualny widok stanów capslock, scrolllock i numlock (na przykład zapalony tylko scrolllock).
2. Uruchomienie programu
3. Wszystkie 3 stany są włączone.
4. Wciśnięcie dowolnego klawisza zmieniającego stan.
5. Powrót do początkowych stanów sprzed uruchomienia programu.

E. Zadanie 4

```

main:
    mov al, 0EDh
    out 60h, al
    czekaj:
        in al, 64h
        test al, 02h
        jnz czekaj
    mov al, 00000111b
    out 60h, al
    mov al, 0f3h
    out 60h, al
    czekaj2:
        in al, 64h
        test al, 02h
        jnz czekaj2
    mov al, 0110100b
    out 60h, al
    mov ax, 4c00h
    int 21h
end main

```

Dla rozkazu MOV AL, 0110100b

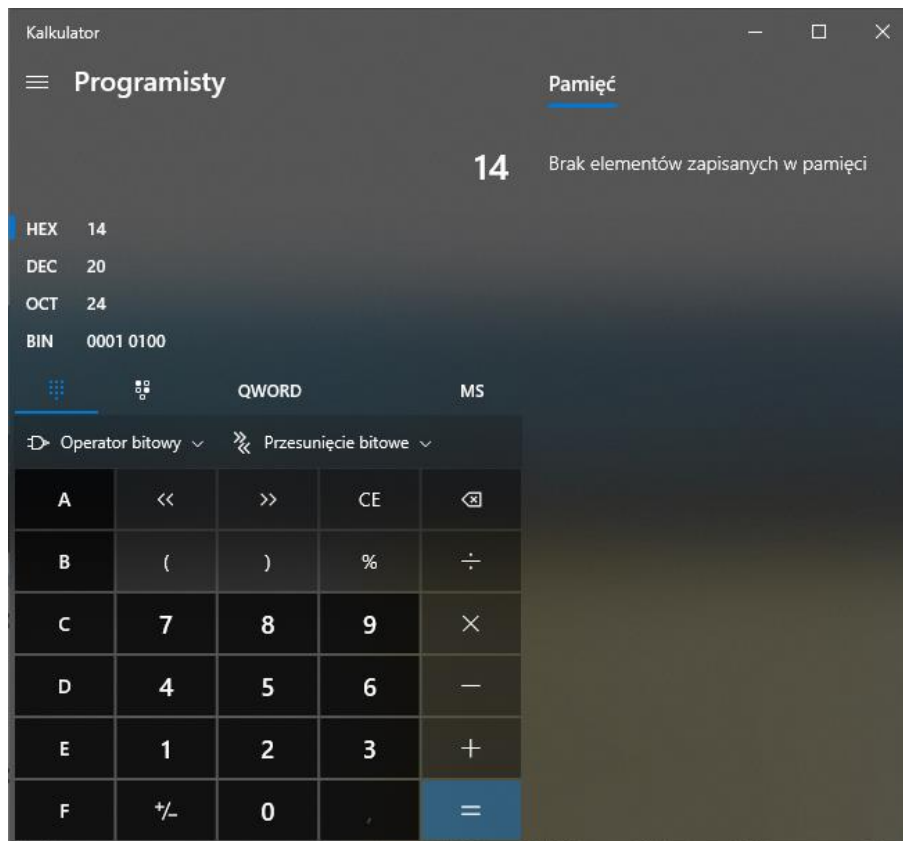
Wartość binarną wyliczam z poniższego opisu:

| | | | |
|----------|----------|---------|---------|
| 00H=30 | 08H=15 | 10H=7.5 | 18H=3.7 |
| 01H=26.7 | 09H=13.3 | 11H=6.7 | 19H=3.3 |
| 02H=24 | 0AH=12 | 12H=6 | 1AH=3 |
| 03H=21,8 | 0BH=10.9 | 13H=5.5 | 1Bh=2.7 |
| 04H=20 | 0CH=10 | 14H=5 | 1CH=2.5 |
| 04H=18.5 | 0DH=9.2 | 15H=4.6 | 1DH=2.3 |
| 06H=17.1 | 0EH=8.5 | 16H=4.3 | 1EH=2.1 |
| 07H=16 | 0FH=8 | 17H=4 | 1FH=2 |

Wartości bitów 6,5 są oznaczają następujące opóźnienia:

| | | | |
|--------|--------|--------|---------|
| 00=250 | 01=500 | 10=750 | 11=1000 |
|--------|--------|--------|---------|

Górne wartości hexadecymalne przedstawiają prędkość autorepetycji (częstotliwość znaków na sekundę). Ja potrzebuję 5, więc potrzebuję wartości 14h:



Z tego wynika, że bit 2 i 4 musi być podniesiony.

Następnie opóźnienie – warunkowane jest przez bity 5 i 6. Aby czas wynosił 1 sekundę powinny być ustawione w poniższy sposób:

$$11=1000$$

Zatem ostateczna wartość prezentuje się tak:

01110100b

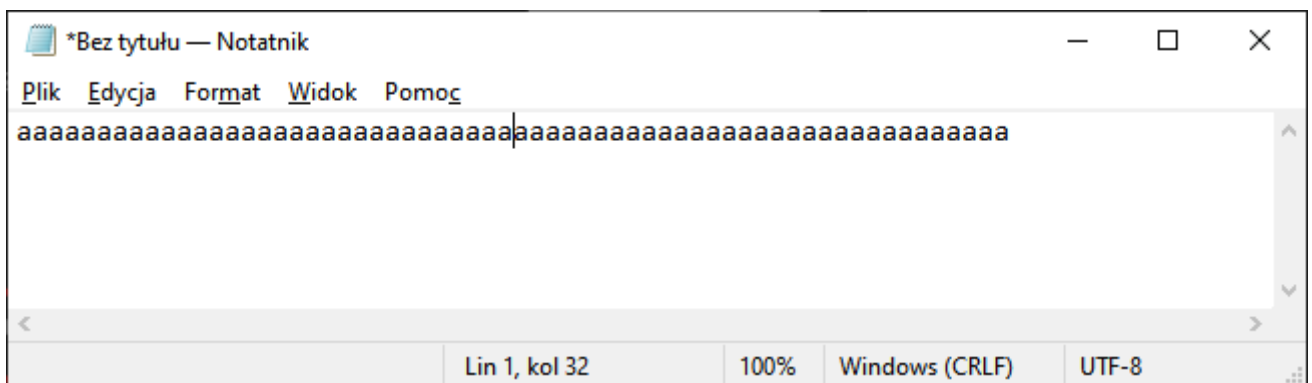
Dla prostego testu uruchomiłem program i przez około 3 sekundy trzymałem klawisz:

```

C:\ASM\LAB4>
C:\ASM\LAB4>dn
Dos Navigator Version 1.51 Copyright (C) 1991-99 RIT Research Labs
C:\ASM\LAB4>comp l4
C:\ASM\LAB4>tasm l4.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International
Assembling file: l4.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 352k
C:\ASM\LAB4>tlink /v l4.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
C:\ASM\LAB4>
C:\ASM\LAB4>l4.exe
C:\ASM\LAB4>aaaaaaaaaaaaaa_

```

To samo zrobiłem dla notatnika na komputerze



Jak widać prędkość autorepetycji jest znacznie różna, dla systemu dos w 3 sekundy zapisało kilkanaście znaków (15), a w moim systemie wpisało ich kilkardziesiąt.