Contents

1 Archi segédlet

1.1 Fejlesztőkörnyezet

- Visual Studio Code
  - MASM/TASM extension (feltöltő: clcxsrolau)

1.2 Regiszterek

1.2.1 Alapszabály

- ha `X`-re végződik, 16 bit
  - pl `AX`
- Ha `A`-ra vagy `B`-re végződik (**H**igh, **L**ow), 8 bit
  - pl `AH`, az `AX` felső 8 bitje

1.2.2 Általános célú regiszterek

- `AX` Accumulator (általános regiszter)
  - `AH`, `AL`
- `BX` Base - Memóriacímzésnél használjuk
  - `BH`, `BL`
- `CX` Cilusnál használjuk
  - A `loop` utasítás minden ciklusa 1-gyel csökkenti
- `DX`

1.2.3 Index regiszterek

- `SI`
- `DI`

1.2.4 Mutató regiszterek

- `SP` (**S**tack **P**ointer)
  - Verem tetejére mutat
- `BP`
  - A verem egy elemét jelöli

1.2.5 Státuszregiszter

- `SR`
- Flagek
  - `CS`

  Work in progress...

<!– TODO befejezni –>

1.3 Hello world

```
.model small
.data
    msg db "Hello$"
.code
.startup
    mov ah, 09h
    mov dx, offset msg
    int 21h
.exit
end
```

1.4 Ugrások

1.4.1 `jmp` Feltétel nélküli ugrás

```
hello:
  mov ah, 09h
  mov dx, offset msg
  int 21h

jmp hello
```

1.4.2 ~ jz~ Ugrás ha a ZERO flag 1

1.5 Regiszter nullázás

```
xor ax, ax
```

Ugyanaz, mint `mov ax, 0`, de gyorsabb

1.6 `int 10h` BIOS interrupt

https://en.wikipedia.org/wiki/INT_10H*List_of_supported_functions

1.6.1 Video mode / text mode `AH = 0`

1. 80x25 Text mode ("képernyő törlése") `AL = 03h`

```
mov ax, 0003h    ; AH = 0, AL = 03h
int 10h
```

2. 320x200 Grafikus mód `AL = 13h`

```
mov ax, 0013h    ; AH = 0, AL = 13h
int 10h;
```

1.6.2 Kurzor pozíció `AH = 02h`

```
mov ah, 02h
mov bh, 0   ; Oldal
mov dh, 5   ; Sor
mov dl, 28  ; Oszlop
int 10h
```

1.7 `int 16h` BIOS interrupt

Billentyűzet funkciók
https://en.wikipedia.org/wiki/INT_16H*List_of_services_of_the_INT_16_h

1.7.1 Gombnyomás kérése `AH = 0`

`AH`-ba menti a [scan code](https://www.millisecond.com/support/docs/current/html/language/scancodes.htm)-ot,
`AL`-be az [ASCII](https://www.ascii-code.com/) kódot

1. Beolvasás

```
xor ax, ax
int 16h
```

2. Tesztelés

```
cmp al, 27 ; ESC
jz Program_vege
```

1.8 `int 21h` DOS interrupt

http://bbc.nvg.org/doc/Master%20512%20Technical%20Guide/m512techb_int21.htm

1.8.1 String kiírás `AH = 09h`

```
.data
    msg db "Hello$"

mov ah, 09h
mov dx, offset msg
int 21h
```

1.9 Loop

A `loop` utasítás 1-gyel csökenti `CX`-et és ha ezután ez nem 0, ugrik a megjelölt címkére

```
mov cx, 5   ; Hányszor szeretnénk a cilust lefuttatni
loop_start:
    ; Kiírjuk, hogy "Hello "
    mov ah, 09h
    mov dx, offset msg    ; msg db "Hello $"
    int 21h

    ; Csökkenti ~CX~-et. Ha nem 0, ugrás loop_start-ra
    loop loop_start
```