

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA POČÍTAČOV A INFORMATIKY

Dokumentácia ku prvému zadaniu z predmetu Assembler

Denys Kukhar

2019

1 Znenie zadania

Načítajte z klávesnice reťazec znakov ukončený znakom "nového riadku". Slová vo vstupe sú oddelené najmenej jedným znakom "medzera". Uvažujte aj prvé, resp. posledné slovo vstupu. Určte počet slov obsahujúcich len písmená malej abecedy. Počet (0-255) vytlačte osmičkovo.

2 Popis riešenia

Stratégia

Samotná stratégia spočíva v tom, že použijem niektoré registre ako indikátory. Takým spôsobom, 8-bitový register **B** použijem ako počítadlo porušených podmienok, tzn. že k jeho inkrementácii dôjde keď nachádzajúc v slove, nájdem buď čo, okrem malého písmena. Register **L** použijem ako indikátor, že pri čítaní sa nachádzam v slove (vynuluje sa po nájdení medzery).

Program ukončí svoju činnosť vtedy keď narazí na symbol ukončenia riadku (10 pre **LF**, 13 pre **CR**).

Hlavný cyklus programu sa nachádza pod lablom **in_word_loop**.

Osmičkový výpis

Pre osmičkový výpis vykonávam ANDovanie s korešpodnými hodnotami pre získanie potrebných bitov. Tak, najprv beriem horné dva bity (**AND** operácia s číslom 11000000) a potom pre výpis rotujem vpravo na šesť miest. Podobne spravím s bitmi 3-5 a 6-8. Pre použitie funkcie putchar operáciou **adi** dosiahnem ten správny offset pre výpis číslice.

3 Kód v jazyku Assembler (i8080)

```
org 1000
```

```
dcx sp  
jmp init
```

```
init:  
    mvi c,0  
    mvi b,0  
    mvi l,0  
    jmp in_word_loop
```

```
words_amount_increment:  
    inr c  
    ret
```

```
increment_broken:  
    inr b  
    jmp in_word_loop
```

```
check_new_line:  
    call check  
    jmp end
```

```
check_space:  
    call check  
    jmp in_word_loop
```

```
check:  
    mov a,l  
    cpi 0  
    jnz reset_and_check  
    ret
```

```
reset_and_check:  
    mov a,b  
    cpi 0  
    cz words_amount_increment  
    mvi l,0  
    mvi b,0  
    ret
```

```
in_word_loop:  
    call getchar  
    mov h, a
```

```
    cpi 10
        cz check_new_line
    cpi 13
        cz check_new_line
    cpi 32
        jz check_space
    mvi l,1
    mov a,h
    cpi 'a'
        jc increment_broken
    cpi 'z'+1
        jnc increment_broken
    jmp in_word_loop

end:
    call newline
    mov a,c
    ani 0C0H
    rlc
    rlc
    adi '0'
    call putchar
    mov a, c
    ani 038H
    rrc
    rrc
    rrc
    adi '0'
    call putchar
    mov a, c
    ani 07H
    adi '0'
    call putchar
    mvi a, 'o'
    call putchar
    hlt

include 'examples\8080\include\putchar.inc'
include 'examples\8080\include\getchar.inc'
include 'examples\8080\include\newline.inc'
```