

## Instrukce pro práci s řetězci

Jedná se o instrukce, které slouží pro práci s řetězci. Každá instrukce po svém vykonání **upravuje index registry**, s kterými pracuje. Index registry se **inkrementují nebo dekrementují** (rozhoduje příznak **DF**) o hodnotu, která závisí, jestli pracujeme s byte, word, dword.

**Popisy instrukcí se vážou k B (byte)! V případě, že chcete použít W nebo D, musíte mít na paměti, že se mění inkrementační/dekrementační krok. V případě, že instrukce pracuje s registry, je zapotřebí použít registry s jinou velikostí. Myslete i na alokaci paměti.**

**MOVS(B|W|D)** – Přesune byte na adrese danou zdrojovým index registrem (ESI) na adresu danou cílovým index registrem (EDI).

Zjednodušeně: **Přesune** znak z jednoho řetězce do druhého.

Příklad:

```
string1 db "Ahoj"
string2 db "xxxx"
.....
mov edi, string2
mov esi, string1
movsb
;Výsledek: string2 „Axxx“
```

**CMPS(B|W|D)** – Porovná byte na adrese danou zdrojovým index registrem (ESI) s bytem na adrese danou cílovým index registrem (EDI).

Zjednodušeně: **Porovná** znaky ze dvou různých řetězců.

Příklad:

```
string1 db "Ahoj"
string2 db "xxxx"
.....
mov edi, string2
mov esi, string1
cmpsb
;Výsledek: vygenerované příznaky SF, AF, PF, CF
```

**SCAS(B|W|D)** – Porovná znak v registru AL s místem v řetězci, kam ukazuje registr (EDI).

Zjednodušeně: Vezme hodnotu z registru AL a **porovná** ji se znakem v řetězci.

Příklad:

```
string1 db "Ahoj"
```

```
string2 db "xxxx"
```

```
.....
```

```
mov al, 'a'
```

```
mov esi, string1
```

```
scasb
```

```
;Výsledek: vygenerované příznaky AF, CF
```

**STOS(B|W|D)** – Store byte to string – **Uloží** hodnotu z registru AL (znak) na místo, kam ukazuje registr EDI.

Zjednodušeně: Vezme hodnotu z registru AL a nahraje ji do řetězce.

Příklad:

```
string2 db "xxxx"
```

```
.....
```

```
mov edi, string2
```

```
mov al, 'A'
```

```
stosb
```

```
;Výsledek: string2 „Axxx“
```

**LODS(B|W|D)** – Load string byte – **Načte** hodnotu z místa, kam ukazuje registr ESI a uloží hodnotu do registru AL.

Zjednodušeně: Načítáme symbol z řetězce.

Příklad:

```
string2 db "Xxxx"
```

```
.....
```

```
mov esi, string2
```

```
lods b
```

```
;Výsledek: v registru AL bude znak 'X'
```

## Prefixové řetězové instrukce

Prefixové řetězové instrukce slouží k opakování řetězových instrukcí. Veškeré prefixové instrukce pracují registrem **ECX**. V každé iteraci se provádí **dekrementace** tohoto registru.

S jednou podmínkou:

**REP** – počet kroků je dán hodnotou v registru **ECX**. **Opakuje dokud ECX=0.**

Se dvěma podmínkami:

**REPZ/REPE** – opakuj tak dlouhou, **dokud jsou znaky stejné** (generuje se ZF) nebo dokud nebude **ECX=0.**

**REPNZ/REPNE** – opakuj tak dlouhou, **dokud jsou znaky různé** (ZF=0) nebo dokud nebude **ECX=0.**