

Instrukce pro práci s řetězcí (poli)

Ing. Dominika Regéciová
Výzkumná skupina formálních modelů

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 1/2, 602 00 Brno - Královo Pole

iregeciova@fit.vutbr.cz



- **Převeďte $(-114)_{10}$ do šestnáctkové soustavy na 8 bitů:**

- **Převeďte $(-114)_{10}$ do šestnáctkové soustavy na 8 bitů:**
 - $(8E)_{16}$
 - přes binární soustavu $(1000\ 1110)_2$
 - s využitím vlastnosti doplňkového kódu: $-114 + 256 = 142$

- MOVS, CMPS, SCAS, LODS, STOS, INS, OUTS
- Ukazatele na data v registrech DS:ESI a ES:EDI
- Zdrojová data jsou uložena na adrese DS:ESI ("SI-Source Index")
- Cílová data jsou uložena na adrese ES:EDI ("DI-Destination Index")
- Instrukce pracují s:
 - Slabikami (LODSB, SCASB, ...)
 - Slovy (LODSW, MOVSW, ...)
 - Dvojslovy (LODSD, MOVSD, ...)
- Instrukce automaticky zvyšuje/snižuje (podle DF) indexové registry ESI a EDI
 - Nastavení DF: instrukce CLD, STD
- CMPS, SCAS - ovlivňují příznaky v EFLAGS

- MOVS a CMPS pracují pouze s pamětí
- SCAS, LODS, STOS pracují s pamětí i registrem EAX
- INS a OUTS pracují s registrem EAX a vstupně/výstupními porty
- POZOR! SCAS čte data z ES:EDI
- Instrukce automaticky zvyšuje/snižuje indexové registry ESI a EDI dle DF (Direction Flag)
 - Zvyšuje - DF == 0 (DF nastavíme na 0 instrukcí CLD)
 - Snižuje - DF == 1 (DF nastavíme na 1 instrukcí STD)

- Slouží k opakování řetězových instrukcí
- Pracují s registrem ECX
- V každé iteraci se provádí dekrementace tohoto registru
- **REP** Počet kroků je dán hodnotou v registru ECX. Opakuje dokud není $ECX = 0$
- **REPZ/REPE** Opakuj tak dlouho, dokud jsou znaky stejné (generuje ZF) nebo dokud nebude $ECX = 0$
- **REPNZ/REPNE** Opakuj tak dlouho, dokud jsou znaky různé ($ZF = 0$) nebo dokud nebude $ECX = 0$

Přesune byte na adrese danou zdrojovým registrem (ESI) na adresu danou cílovým registrem (EDI).

Zjednodušeně: Přesune znak z jednoho řetězce do druhého

```
string1 db "Ahoj", 0  
string2 db "xxxx", 0  
mov edi, string2  
mov esi, string1  
movsb
```

Výsledek:

```
string2 "Axxx"
```

Porovnává byte na adrese danou zdrojovým registrem (ESI) s bytem na adrese danou cílovým registrem (EDI).

Zjednodušeně: Porovnává znaky ze dvou různých řetězců

```
string1 db "Ahoj", 0  
string2 db "xxxx", 0  
mov edi, string2  
mov esi, string1  
cmpsb
```

Výsledek:

vygenerované příznaky SF, AF, PF, CF

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

```
string1 db "Ahoj", 0  
string2 db "xxxx", 0  
mov al, 'A'  
mov edi, string1  
scasb
```

Výsledek:

vygenerované příznaky PF, ZF

Uloží hodnotu z registru AL (znak) na místo, kam ukazuje registr EDI.

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

```
string1 db "Ahoj", 0  
string2 db "xxxx", 0  
mov al, 'A'  
mov edi, string2  
stosb
```

Výsledek:

```
string2 "Axxx"
```

Načte hodnotu z místa, kam ukazuje registr ESI a uloží hodnotu do registru AL.

Zjednodušeně: Načítáme symbol z řetězce

```
string1 db "Xxxx", 0  
mov esi, string1  
lodsb
```

Výsledek:

v registru AL bude znak 'X'

- Můžete získat až 12 bodů
- Programování jednoduchých funkcí (práce s poli a řetězci)
- Můžete používat vlastní poznámky, přehled instrukcí a materiály k přednáškám
- Bude mne zastupovat kolega Ing. Jakub Husa

