

Operační systémy 1

4. Cvičení

Radek Janošík



**KATEDRA
INFORMATIKY**

UNIVERZITA PALACKÉHO V OLOMOUCI

Úkoly z minula – jak to šlo?



- Kolik se vám toho podařilo udělat?
- Dalo něco zabrat?
- Nějaké dotazy?

Úkoly z minula – jak to šlo?



- Kolik se vám toho podařilo udělat?
- Dalo něco zabrat?
- Nějaké dotazy?
- Prezenčka (až přijdou všichni :))

- Dosud se vykonávaly instrukce „jedna za druhou“
- Neměli jsme možnost řídit toky programu
- Registr `rip` ukazuje na adresu instrukce, která se má provést jako další
 - Nemůžeme jej měnit přímo (`mov`)
 - Speciální instrukce – *skoky* – s ním manipulují
- Díky skokům můžeme realizovat
 - Podmíněné příkazy
 - Cykly

- instrukce `jmp r/m/i` – operandem je cílová adresa
- často s konstantou – adresa zadána pomocí *návěští (label)*
 - assembler doplní konkrétní adresu při překladu
 - u AMD64 jsou adresy relativní

```
1      mov eax, 0x42
2  foo:
3      inc eax
4      jmp foo
```

- Jde skočit „kamkoliv“ – silný nástroj
 - Je potřeba dobře logicky členit kód

- Provedou se pouze za určitých okolností, jinak program plyne dál
- Skok určen příznaky v registru `rf`:
 - `ZF` (zero flag) – výsledek byla nula
 - `SF` (sign flag) – výsledek je nezáporný (0) nebo záporný (1)
 - `CF` (carry flag) – výsledek je větší/menší než největší/nejmenší možné číslo
 - `OF` (overflow flag) – přetečení znaménkové hodnoty mimo daný rozsah
- Pro každý příznak dvě instrukce skoku (pokud nastaven, pokud nenastaven)
- Příklad: `ZF` – instrukce `jz`, `jnz`

```
1     sub eax, 1
2     jz  foo
3     inc ebx
4     foo:
```

- Podobně: pro `SF` (`js`, `jns`), `CF` (`jc`, `jnc`), `OF` (`jo`, `jno`).

- Předchozí lehce neintuitivní \Rightarrow instrukce `cmp`
- Porovná dvě hodnoty a nastaví příznaky + instrukce skoků
 - „vysokoúrovňovější“

■ Znaménkové:

| instrukce | alt. jméno | příznaky | podmínka |
|------------------|-------------------|------------------------------|------------|
| <code>jg</code> | <code>jnle</code> | $(SF = OF) \ \& \ ZF = 0$ | $A > B$ |
| <code>jge</code> | <code>jnl</code> | $(SF = OF)$ | $A \geq B$ |
| <code>jl</code> | <code>jnge</code> | $SF \neq OF$ | $A < B$ |
| <code>jle</code> | <code>jng</code> | $(SF \neq OF)$ nebo $ZF = 1$ | $A \leq B$ |

■ Neznaménkové:

| instrukce | alt. jméno | příznaky | podmínka |
|------------------|-------------------|---------------------------|------------|
| <code>ja</code> | <code>jnbe</code> | $(CF \text{ or } ZF) = 0$ | $A > B$ |
| <code>jae</code> | <code>jnb</code> | $(CF = 0)$ | $A \geq B$ |
| <code>jb</code> | <code>jnae</code> | $CF = 1$ | $A < B$ |
| <code>jbe</code> | <code>jna</code> | $(CF \text{ or } ZF) = 1$ | $A \leq B$ |



```
1  ;;
2  ;; funkce pro vypocet absolutni hodnoty
3  ;; nazev 'absi' zvolen z duvodu kolize s klicovym slovem 'abs'
4  ;;
5  ;; int absi(int n);
6  ;;
7  absi:
8      mov eax, edi ; precteni argumentu n
9      cmp eax, 0 ; porovnani s 0
10     jge konec ; skok na konec funkce
11     neg eax     ; otoceni znamena
12  konec:
13     ret         ; navrat z funkce
```



```
1 ;;
2 ;; funkce pro vypocet faktorialu
3 ;; int fact(int n);
4 ;;
5 fact:
6     mov ecx, edi ; ecx -- vstupni argument (n)
7     mov eax, 0x1 ; eax -- strada vyslednou hodnotu tj. n * (n -
           1) * (n - 2) * ... * 1
8 fact_loop:
9     cmp ecx, 0x0
10    jle konec    ; narazili jsme na konec, ukoncime funkci
11    imul ecx     ; vynasob eax hodnotou n
12    sub ecx, 0x1 ; opakuj pro n - 1
13    jmp fact_loop
14 konec:
15    ret
```



- 1 Napište funkci `int sgn(int i)`, která vrací hodnoty -1, 0, 1 v závislosti na tom, zda-li je hodnota `i` záporná, nulová nebo kladná.
- 2 Napište funkci `char max2c(char a, char b)`, která vrací největší hodnotu. Vyzkoušejte, že funkce funguje správně pro kladné i záporné argumenty, i jejich kombinaci.
- 3 Napište funkci `unsigned short min3us(unsigned short a, unsigned short b, unsigned short c)`, která vrací nejmenší hodnotu ze zadaných parametrů. Vyzkoušejte, že funkce funguje správně i pro hodnoty větší než 32768.



- 4 Napište funkci `int kladne(int a, int b, int c)`, která vrací 1, pokud jsou všechny argumenty kladné, jinak 0.
- 5 Napište funkci `int mocnina(int n, unsigned int m)` vracející mocninu n^m
- 6 Do registrů `al`, `bl` vložte vhodné hodnoty, proveďte s nimi operace `add` a `sub` a pomocí instrukcí `jz`, `js`, `jc` a `jo` ověřte, zda byl nastavený příznak, nebo ne.