

Program

- opakování
- zásobník
- podprogramy
- jednoduché programy v assembleru II
- další služby operačního systému

Zásobník

- datová struktura v paměti
- přístup LIFO
- instrukce PUSH a POP
- „úschovný“ prostor pro hodnoty z registrů
- na vrchol ukazuje dvojice SS:ESP nebo SS:SP
- roste od vyšších adres k nižším
- ukládá se na něj návratová adresa z podprogramu
- Příklad: Jak se bude měnit hodnota zásobníku a co bude na něm uloženo (adresy 10 bajtů od SS:SP)
- `mov eax,2`
- `push eax`
- `mov ax,3`
- `push ax`
- `mov ax,0x4C00`
- `int 0x21`

Podprogramy

- podprogram slouží k vykonávání často používané části programu
- matematický ekvivalent (stejná věc) jako funkce
- podprogram má parametry (argumenty)
- podprogram může nebo nemusí mít návratovou hodnotu
- parametry můžeme předávat na:
 - ve smluvených registrech procesoru
 - na zásobníku (viz poslední kapitola knihy)
- podprogram sečti (AX a BX výsledek v CX)
-
- secti:
- `mov cx,ax`
- `add cx,bx`
- `ret`
-
-

Instrukce RET a CALL

- RET vezme ze zásobníku poslední hodnotu a uloží EIP nebo IP
- strana 70
- provede tedy POP EIP nebo POP IP (zase závisí na režimu procesoru)
-
- Instrukce CALL pracuje jako JMP ale na zásobník se uloží následující instrukce, která po instrukci CALL následuje.
- CALL kam
-
-
- PUSH EIP+N
- JMP kam
-

Podprogramy II

- zavoláme např:
- ...
- `mov ax,4`
- `mov bx,2`
- `call secti`
- ;v CX bude výsledek
-
- Použili jsme instrukce CALL a RET

Práce pro vás

- Vytvořte podprogram pro násobení dvou čísel pomocí sčítání
- nazvěte ho sečti
- v AX první číslo
- v BX druhé
- v CX výsledek

Práce pro vás

```
SECTION .text
ORG 0x100
jmp start
secti:
....
ret
start:
mov ax,3
mov bx,4
call secti
;cx výsledek
mov AX,0x4C00
int 0x21
```

Tiskneme znaky

- v assembleru nasm můžeme:
-
- `mov al,'A'` ; kolik bude AL?
- `mov CH,'a'` ; kolik bude CH ?
-
- Služba DOSu 0x2
- `mov ah,2`
- `mov dl,'A'`
- `int 0x21`
- Vytiskne A
-
- Úkol: vytiskněte abecedu velkých a malých písem

Tiskneme na obrazovku z klávesnice

- Přečti znak funkce 0x1
- `mov ah, 0x1`
- `int 0x21`
- ; v AL je přečtený znak
- `mov ax, 0x4C00`
- `int 0x21`
-

Přehled funkcí DOSu

- Viz kniha

Hrátky s klávesnicí

- Úkol: vyrobte program co tiskne znaky o jeden v abecedě dál
- tj např: a -> b e->f atd...
-
-
- Úkol: vyrobte program co bude tisknout znaky o 13 míst dopředu,
- stím že znaky z se změní na $A+12$... (neboli modulo 26 počítání)

Jak tisknout čísla???

- v assembleru to není zrovna jednoduché
- postup stejný jako při převádění čísel
- dělíme postupně a sepisujeme zbytky
- na zásobník je ukládáme
- vytiskneme zase ze zásobníku
-
- jak na to:
- vývojový diagram
- instrukce DIV
- push zbytek
- porovnej výsledek s nulou
- pak znovu
- postupně vyber ze zásobníku a vytiskni jako znaky

Samostatná práce

- V registru AX je uložen rok, v registru BH den a v BL měsíc narození osoby
- Pro jednoduchost předpokládejme že v CX je aktuální rok, DH den, DL měsíc
- Vytvořte program který vloží do AX jedničku pokud je dotyčný plnoletý (už mu bylo 18)
- nakreslete vývojový diagram
- napište program