Memória címzési módok

Egy program futása során (legyen szó a program vezérléséről vagy adatkezelésről) a program utasításai illetve egy utasítás argumentumai a memóriában találhatók. A memória-szervezési modell mondja meg azt, hogy miként és mekkora területhez férhetünk hozzá a memóriában tárolt adathoz. A legkisebb memória-egység, amelyet meg tudunk címezni 8 bit, vagyis 1 bájt. Gyakran azonban *szavas* címeket használunk. Egy szót (word) 2 vagy 4 bájton tudunk ábrázolni, dupla szavakat (double word vagy dword) pedig 4 vagy 8 bájton ábrázolunk.

A 8086-os architektúrájú számítógépek memóriája szegmensekre van osztva. A szegmensek a memória adott méretű, összefüggő területei. A memória-szegmensek *báziscímei* (kezdőcímei) általában ismertek, és mivel a szegmensek összefüggőek, szegmensen belüli bájtokat a báziscíméhez képest tudjuk megadni. A szegmensen belüli bájtok távolságát a báziscímhez képest *offszet*nek hívjuk. Egy memória-terület címének fizikai címét a báziscímmel és az offszettel tudjuk megadni.

BÁZISCÍM:OFFSZET

A 8086/8088-as architektúrájú számítógépek regiszterei 16-bitesek, a címbusz pedig 20 bites[1]. Ebből kifolyólag két regiszter szükséges egy cím előállításához. A fizikai cím kiszámítását úgy valósították meg, hogy szegmens címek mindig oszthatóak 16-tal (paragrafushatár). A szegmens regiszter a szegmens kezdő paragrafusának a sorszámát tartalmazza (valós mód). Valós módban a fizikai cím úgy számítható ki, hogy a megfelelő szegmens regiszter tartalmát megszorozzuk 16-tal (vagyis egy helyiértékkel balra toljuk és a legalacsonyabb helyi értéken 0 lesz) és hozzáadujuk az offszet címet. Az eredmény alsó 20 bitjén kapott érték a fizikai cím függetlenül attól, hogy a kapott érték elfér-e 20 biten vagy sem[2].

[1] Andrew S. Tanenbaum: Számítógép architektúrák, PANEM, 2001
[2] Máté Eörs: Assembly programozás, NOVODAT

Példa:

Tegyük fel, hogy a processzor valós módban van. Legyen egy szegmens regiszter értéke 04F2h. Határozzuk meg a fizikai címet, ha az offszet 00B2h!

1. A szegmens regiszter tartalmát megszorozzuk 16-tal (egy helyiértékkel balra toljuk): 4F20h
2. Hozzáadjuk az offszet címet:

```
   4F20h
+  00B2h
-------
   4FD2h
```

3. Mivel az így kapott érték elfér 20 biten, ezért továbbítható a címbuszon.

A 8086/8088 architektúra regiszterei

Szegmens regiszterek (16 bites regiszterek)

- CS (Code Segment): utasítások címzéséhez
- SS (Stack Segment): verem címzéséhez
- DS (Data Segment): adat terület címzéshez
- ES (Extra Segment): másodlagos adatterület címzéshez

Vezérlő regiszterek (16 bites regiszterek)

- IP (Instruction Pointer): az éppen végrehajtandó utasítás logikai címét tartalmazza a CS által mutatott szegmensben
- SP (Stack Pointer): a verem tetejére beírt adat logikai címére mutat az SS által mutatott szegmensben
- STATUS (vagy SR, vagy FLAGS): a processzor állapotát jelző regiszter
- BP (Base Pointer): a verem indexelt címzéséhez használatos
- SI (Source Index): a kiindulási adat terület indexelt címzéséhez használatos
- DI (Destination Index): a cél adat terület indexelt címzéséhez használatos

Általános regiszterek 16-bites regiszterek.
Az általános regiszterek felső 8 bitje és alsó 8 bitje külön is címezhető

regiszter	felső bájt	alsó bájt	
AX	AH	AL	Accumulator (szorzás, osztás)
BX	BH	BL	Base Register (címző regiszter)
CX	CH	CL	Counter Regiszter (számláló)
DX	DH	DL	Data Register (szorzás, osztás, I/O)

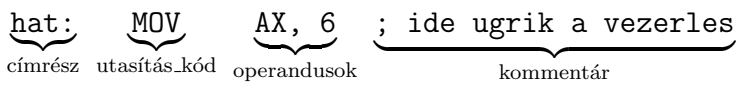
Címzési módok assembly-ben

Assembly programozási nyelvben egy utasítás a következő sémát követi:

`címrész operációs_kód operandusok kommentár`

- címrész: egyes adatok illetve utasítások szimbólikus jelölése
- operációs_kód: mnemonic, az utasítás, művelet megnevezésére szolgál
- operandusok: az utasítás paraméterei
- kommentár: A program jobb olvashatóságát és érthetőségét teszi lehetővé, de nincs hatása a program működésére

Példa:



Adat terület címzés

Kódba épített adat Az adatot közvetlenül a regiszterbe írjuk. Ennél a címzési módnál csak konstansokat tudunk megadni második operandusként.

Formátum: `regiszter, konstans`
Például:

- `AX, 6`
- `AX, 06F2h`

Direkt memória címzés A címrészen az operandus logikai (offszet) címét adjuk meg, nem az adatot.

Formátum: `regiszter, memóriacím`
Például:

- `AX, SZO` *, ahol a SZO egy word értéket tartalmaz*
- `AL, KAR` *, ahol a KAR egy byte értéket tartalmaz*

Ügyelni kell arra, hogy a két operandus mérete összhangban legyen egymással. 16 bites regiszterhez csak 16 bites (word) címeket használhatunk, 8 bites regiszterhez csak 8 bites címeket használhatunk!

Direkt címzésnél megadhatunk második operandusként egy regisztert is. Akár a egy regisztert, akár egy logikai címet adunk meg, mindkét esetben a memóriacímen tárolt adat változhat, viszont a cím azonos marad.

Indexelt címzés Az operandusban megadott 8 vagy 16 bites számot (eltolás) hozzáadjuk az index-regiszter (SI vagy DI) tartalmához, így alakul ki a logikai cím.

Formátum: `regiszter, szám[index-regiszter]`
Például:

- `AX, [SI]` *, itt csak az SI-ben tárolt értéket számítjuk*
- `AX, 10h[SI]` *, SI tartalmához hozzáadjuk a 10h konstanst*
- `AX, -10h[SI]` *, SI tartalmához hozzáadjuk a -10h konstanst*

A 8 biten megadott eltolás érték előjel helyesen 16 bitre bővül a cím kiszámításakor.

Regiszter indirekt címzés Az operandusban a címet egy regiszter tartalmazza. Az ilyen módon megadott címet mutatónak hívjuk. Nem adható meg eltolás. Ebben a címzési módnál csak az SI, DI és BX regiszter használható.

Formátum: `regiszter, [regiszter]`
Például:

- `AX, [SI]` , *az SI-ben tárolt értéket használjuk*
- `AX, [BX]` , *a BX-ben tárolt értéket használjuk*
- `AX, [DI]` , *a DI-ben tárolt értéket használjuk*

Bázis relatív címzés Az operandusban megadott logikai címet úgy kapjuk meg, hogy az eltolás értékét, az SI vagy DI valamelyikének értékét, valamint a BX regiszterben tárolt értéket összeadjuk.

Formátum: `regiszter, szám[regiszter][regiszter]`
Például:

- `AX, 10h[SI][BX]`
- `AX, [BX][DI]`
- `AX, [BX + SI + 10h]`

Verem terület címzés

A verem (stack) terület címzése bázis relatív címzéssel történik, azzal a különbséggel, hogy BX helyett BP-t használjuk. A fizikai cím meghatározásához nem DS-t, hanem SS lesz a szegmens regiszter.

Program terület címzés

Egy-egy utasítás során az IP értéke az utasítás hosszával növekszik és a soron következő utasításra mutat a CS-ben. Bizonyos vezérlési szerkezetekben (pl. ciklusok, feltételek) az IP értékének megadásával a kívánt helyen folytathatjuk a program futását.

IP relatív címzés Az IP (már módosult) pillanatnyi értékéhez hozzáadódik a 8 bites előjeles operandus. A programkódokban bizonyos kódrészletek kezdő utasítását címkével láthatjuk el, ha azt szeretnénk, hogy egy adott feltétel esetén ott folytatódjon a program futása. A feltételes vezérlés átadás és a ciklus utasítás mindig ilyen címzési móddal valósul meg.

Például:
`JMP CIKLUS` , *ahol CIKLUS egy címke az assembly forráskódban*

Direkt utasítás címzés Ez a címzési mód közeli (NEAR) vagy távoli (FAR) vezérlés átadásoknál játszik szerepet. A vezérlés átadás közeli, ha a programkód ugyanabban a szegmensben folytatódik tovább és távoli szegmensváltás esetén. Közeli vezérlésátadás esetén a 16 bites operandus lesz az IP új tartalma, távoli vezérlésátadásnál pedig az IP és a CS tartalma is megváltozik. Ilyen vezérlés lehet például az eljáráshívás.

Például:
`CALL ELJARAS` , *az ELJARAS nevű eljárás hívása*

Indirekt utasítás címzés Bármilyen címzési móddal megadott szóban vagy dupla szóban tárolt címre történő vezérlés átadás.

Például:

- `JMP AX`
- `JMP [BX]`

Feladatok

1. Melyik címzés helyes és melyik helytelen?

   ```
   [12h+56h]
   ES:[09D3h]
   [123456789ABCh]
   [SI+DI]
   [IP]
   [SI-500d]
   [AX]
   SS:[BX+DI+1999d]
   [CX+1234h]
   [SP+BP]
   DS:[BP+SI]
   IP:[0000h]
   ```

2. Melyik milyen típusú címzés?

```
AX, BX
BX, [SI]
AX, [SI + 0Fh]
```

3. Oldjuk meg a következő feladatot!

   Egy programban a következő adat szegmens részlet szerepel:

```
ADAT SEGMENT PARA PUBLIC 'DATA'
; ...
A DB 10h
SZOVEG DB 'Egy szoveg', 0
B DW 100h
SZAMSOR DW 0000h, 0001h, 0010h, 0100h, 1000h
; ...
ADAT ENDS
```

   Számítsuk ki

   - az „A", és „B" változók,
   - A „SZOVEG" változó 2., 7. elemeinek („g", és „o" betű)
   - a „SZAMSOR" változó 1., 5. elemeinek (0000h, és 1000h értékek)

   fizikai memóriacímét feltéve, hogy a processzor valós módban van, DS regiszter az ADAT szegmensre van állítva, értéke 0F7Dh. Valamint a változók szegmensen belüli eltolás címei:

   - A: 0064h,
   - SZOVEG: 0065h,
   - B: 0070h,
   - SZAMSOR: 0072h