Instrucţiuni de apel de procedură şi de salt

Forma generală pentru definirea unei proceduri este:

```
nume_proc PROC [FAR | NEAR]
        ……
        RET
nume_proc ENDP
```

unde *nume_proc* este numele procedurii, iar parametrii opţionali FAR sau NEAR indică tipul procedurii. Procedurile sunt de două tipuri: FAR și NEAR. O procedură FAR poate fi apelată și din alte segmente de cod decât cel în care este definită, pe când o procedură NEAR poate fi apelată numai din segmentul de cod în care este definită.

Dacă se omit parametrii FAR sau NEAR, tipul procedurii este dedus din directivele simplificate de definire a segmentelor (modelul de memorie folosit).

```
1 nume PROC
2     <corpul_procedurii>
3     ret [<dimensiune_parametri>]
4 nume ENDP
```

În mod corespunzător, există apeluri de tip FAR, respectiv NEAR, precum și instrucţiuni de revenire de tip FAR, respectiv NEAR. Instrucţiunea RET (Return) provoacă revenirea în programul apelant; tipul instrucţiunii este dedus din tipul procedurii (NEAR sau FAR). Putem folosi o instrucţiune de revenire explicită: RETN (Return Near) sau RETF (Return Far).

1. Apelul procedurilor și revenirea din proceduri

Instrucţiunea CALL (Apel de procedură)

```
CALL nume_proc
CALL FAR PTR nume_proc
CALL NEAR PTR nume_proc
```

În primul caz, tipul apelului este dedus din tipul procedurii, iar în celelalte este specificat explicit (FAR sau NEAR). Tipul apelului trebuie să coincidă cu tipul procedurii și cu tipul instrucţiunilor Return din interiorul procedurii, altfel se ajunge la funcţionări defectuoase ale programului.

În cazul unui apel de procedură de tip NEAR, se salvează în stivă conţinutul registrului IP, care reprezintă adresa de revenire, iar apoi în IP se încarcă adresa primei instrucţiuni din procedură.

În cazul unui apel de tip FAR, se salvează în stivă CS:IP, adresa completă de revenire (pe 32 de biţi), iar apoi în CS:IP se încarcă adresa primei instrucţiuni din procedură.

Instrucţiunea RET (Return - Revenire din procedură)

```
RET
RETF
RETN
```

În primul caz, tipul tipul instrucţiunii este dedus din tipul procedurii. În cazul unei reveniri de tip NEAR, se reface registrul IP din stivă, astfel se transferă controlul la instrucţiunea care urmează instrucţiunii CALL care a provocat apelul procedurii. În cazul unei reveniri de tip FAR, se reface din stivă perechea de registre CS:IP.

Instrucţiunea JMP (Jump - Salt)

```
JMP     tinta
JMP     SHORT  PTR     tinta
JMP     NEAR  PTR      tinta
JMP     FAR  PTR       tinta
```

În primul caz, tipul saltului este dedus din atributele expresiei care precizează ţinta. Ţinta specifică adresa de salt poate fi o etichetă sau o expresie. Există trei tipuri de instrucţiuni de salt:

- SHORT - adresa ţintă se află la o faţă de adresa instrucţiunii de salt;
- NEAR - adresa ţintă este în același segment de cod cu instrucţiunea de salt;
- FAR - adresa ţintă poate fi în alt segment de cod faţă de instrucţiunea de salt.

2. Tipuri de salt/apel

JMP/CALL direct

Operandul care se află în formatul instrucţiunii este o etichetă care identifică adresa ţintă. Poate fi de două tipuri:

- salt/apel direct intrasegment (NEAR) - eticheta este în același segment de cod cu instrucţiunea JMP/CALL;
- salt/apel direct intersegment (FAR) - eticheta poate fi definită ți în alt segment de cod decât cel care conţine instrucţiunea JMP/CALL.

Adresare imediată

```
1 mov EAX, 1234h
2 add CX, 30
3 and BH, 01111b
```

JMP/CALL indirect

Operandul care apare în formatul instrucţiunii reprezintă o adresă de memorie. Poate fi de două tipuri:

- salt/apel indirect intrasegment (NEAR), cu forma generală
  JMP/CALL expr
  în care *expr* precizează adresa efectivă a ţintei și poate fi un registru, o variabilă de tip WORD, sau un cuvânt din memorie;

- salt/apel indirect intersegment (FAR), cu forma generală
  JMP/CALL expr
  în care *expr* precizează adresa completă a ţintei și poate fi o variabilă de tip DWORD, sau un dublu-cuvânt din memorie.

3. Instrucţiuni de salt condiţionat

Instrucţiunile din această categorie implementează salturi condiţionate de valoarea unor bistabili (FLAGS). Dacă condiţia nu este îndeplinită, saltul nu are loc, deci execuţia continuă cu instrucţiunea următoare. Toate instrucţiunile de salt condiţionat sunt de tip SHORT, ceea ce înseamnă că adresa ţintă trebuie să fie la o distanţă cuprinsă între -127 i +127 de octeţi faţă de instrucţiunea de salt. În tabelul următor se prezintă instrucţiunile de salt condiţionat:

Instrucţiune	Condiţie de salt	Interpretare
JE, JZ	ZF=1	Zero, Equal
JL, JNGE	SF≠OF	Less, Not Greater or Equal
JLE, JNG	SF ≠ OF sau ZF = 1	Less or Equal, Not Greater
JB, JNAE, JC	CF=1	Below, Not Above or Equal, Carry
JBE, JNA	CF = 1 sau ZF = 1	Below or Equal, Not Above
JP, JPE	PF=1	Parity, Parity Even
JO	OF=1	Overflow
JS	SF=1	Sign
JNE, JNZ	ZF=0	Not Zero, Not Equal
JNL, JGE	SF=OF	Not Less, Greater or Equal
JNLE, JG	SF = OF i ZF = 0	Not Less or Equal, Greater
JNB, JAE, JNC	CF=0	Not Below, Above or Equal, Not Carry
JNBE, JA	CF = 0 i ZF = 0	Not Below or Equal, Above
JNP, JPO	PF=0	Not Parity, Parity Odd
JNO	OF=0	Not Overflow
JNS	SF=0	Not Sign

La comparaţiile cu semn folosim GREATER și LESS, iar la comparaţiile fără semn folosim ABOVE și BELOW.

Uneori este necesar să folosim instrucţiuni de salt condiţionat la etichete care ies în afara domeniului faţă de instrucţiunea curentă. În această situaţie înlocuim saltul pe o condiţie directă “departe” cu un salt pe condiţia negată “aproape” cu un salt necondiţionat “departe”. În următorul exemplu, eticheta *et1* se află în afara domeniului, astfel instrucţiunea:

```
        JE  et1
```

se înlocuie te cu:

```
        JNE  et2
        JMP et1
    et2:
```

4. Instrucţiuni pentru controlul buclelor de program

Instrucţiunea JCXZ (Jump if CX is Zero - Salt dacă CX este zero)

JCXZ eticheta

Eticheta trebuie să se afle în domeniul [-127, +127] faţă de instrucţiunea curentă. Se face salt la eticheta specificată dacă CX conţine valoarea 0.

Instrucţiunea LOOP (Ciclare)

LOOP eticheta

Această instrucţiune este, de fapt, un salt condiţionat de valoarea registrului CX. Cu alte cuvinte, se decrementează CX , dacă acesta este diferit de zero, se sare la eticheta specificată. Eticheta trebuie să se afle în domeniul [-127, +127] faţă de instrucţiunea curentă.

Instrucţiunea LOOPZ/LOOPE (Loop While Zero/Equal - Ciclează cât timp este zero/egal)

LOOPZ eticheta

LOOPE eticheta

Se decrementează CX , dacă acesta este diferit de zero ZF este 1 (rezultatul ultimei operaţii a fost zero), se sare la eticheta specificată.

Instrucţiunea LOOPNZ/LOOPNE (Loop While Not Zero/Equal - Ciclează cât timp nu este zero/egal)

LOOPNZ eticheta

LOOPNE eticheta

Se decrementează CX , dacă acesta este diferit de zero ZF este 0 (rezultatul ultimei operaţii a fost zero), se sare la eticheta specificată.