FLAGS

Un flag este un indicator reprezentat pe un bit. O configuraţie a registrului de flaguri indică un rezumat sintetic a execuţiei fiecărei instrucţiuni. Pentru x86 registrul EFLAGS (the status register) are 32 biţi dintre care sunt folosiţi uzual numai 9.

PF (Parity Flag) - Valoarea lui se stabileşte a.î. împreună cu numărul de biţi 1 din octetul cel mai puțin semnificativ al reprezentarii rezultatului UOE să rezulte un număr impar de cifre 1.

O depăşire este o condiţie/situaţie matematică ce exprimă faptul că rezultatul UOE nu a încăput în spaţiul rezervat acestuia SAU acest rezultat nu aparţine intervalului de reprezentare admisibil SAU că operaţia efectuată este un nonsens matematic în respectiva interpretare (cu semn sau fără semn) şi nu poate fi astfel acceptată drept o operaţie matematică corectă

Depasire la impartire -> runtime error

COMPLEMENT FATA DE 2

Suma valorilor absolute a celor două valori complementare este cardinalul mulţimii reprezentabile pe acea dimensiune.

- Pe 8 biţi se pot reprezenta 2^8 valori = 256 valori ([0..255] sau [-128..+127])

- Pe 16 biţi se pot reprezenta 2^16 valori = 65536 valori ([0..65535] sau [-32768,+32767])

- Pe 32 biţi se pot reprezenta 2^32 valori = 4.294.967.296 valori (…)

Ex.: Deci pe 8 biti , complementul fata de 2 a lui 1001 0011 (= 93h = 147) este 256 – 147 = 109, ca urmare valoarea in interpretarea CU SEMN pentru 1001 0011 este -109

Doua valori complementare nu vor fi NICIODATA parte a aceluiasi interval de reprezentare admisibil !!! -ex.:128, 128; 147, -109; -1, 255;

LOCATII

Adresa unei locaţii – nr. de octeţi consecutivi dintre începutul memoriei RAM şi începutul locaţiei respective.

O succesiune continuă de locaţii de memorie, menite să deservească scopuri similare în timpul execuţiei unui program, formează un segment. În consecinţă, un segment reprezintă o diviziune logică a memoriei unui program, caracterizată prin adresa de bază (început), limita (dimensiune) şi tipul acesteia. Atât adresa de bază cât şi dimensiunea unui segment au valori reprezentate pe 32 biţi.

Vom numi offset sau deplasament adresa unei locaţii faţă de începutul unui segment, sau, cu alte cuvinte, numărul de octeţi aflaţi între începutul segmentului şi locaţia în cauză.(val sa numerica nu depaseste dimensiunea segm)

Vom numi specificare de adresă sau adresă logică o pereche formată dintr-un selector de segment şi un offset. Un selector de segment este o valoare numerică de 16 biţi care identifică (indică/selectează) în mod unic segmentul accesat şi caracteristicile acestuia. Un selector de segment este definit si furnizat de catre sistemul de operare !!

a7a6a5a4a3a2a1a0 := b7b6b5b4b3b2b1b0 + o7o6o5o4o3o2o1o0 -> adresa liniara(de segmentare)- >formata din adresa de baza si offset (rezultatul calculului de adresă)

Adresă fizică efectivă - rezultatul final al segmentării plus, eventual, paginării. Adresa finală obţinută de către BIU, indicând în memoria fizică (hardware)

O specificare de adresă mai poartă şi numele de adresă FAR (îndepărtată) – ex 8:1000h (8 e selector de segment si de acolo extrage adresa de baza si limita, apoi verifica daca offset ul adica 1000h nu depaseste limita). Atunci când o adresă se precizează doar prin offset, spunem ca este o adresă NEAR (apropiată).

model de memorie flat = cand segmentele încep la adresa 0, segmentarea nu contribuie efectiv in calculul adreselor

Procesoarele x86 suportă şi un mecanism de control al accesului la memorie numit paginare, independent de adresarea segmentată. Paginarea implică împărţirea memoriei virtuale în pagini, care sunt asociate (translatate) memoriei fizice disponibile (1 page = 4 KB = 2^12 bytes = 4096 bytes).

Noi lucram in mod protejat pe 32 biti

Avem 4 tipuri de segmente: de date, de cod, de stiva si segment suplimentar de date.

Regiştrii CS, DS, SS şi ES determină adresele de început şi dimensiunile segmentelor active: de cod, de date, de stivă şi suplimentar. Regiştrii FS şi GS pot reţine selectori indicând către segmente suplimentare, fără însă a avea roluri predeterminate. – registrii de segment(sau selectori). Registrul EIP conţine offsetul instrucţiunii curente în cadrul segmentului de cod curent, el fiind manipulat exclusiv de către BIU.

ADRESA FAR SI NEAR

Prin definiţie, o adresă în care se specifică doar offsetul, urmând ca segmentul să fie preluat implicit dintr-un registru de segment poartă numele de adresă NEAR (adresă apropiată). O adresă NEAR se află întotdeauna în interiorul unuia din cele patru segmente active.

O adresă în care programatorul indică explicit un selector de segment poartă numele de adresă FAR (adresă îndepărtată). O adresă FAR este deci o SPECIFICARE COMPLETA DE ADRESA si ea se poate exprima în trei moduri: - s3s2s1s0 : specificare\_offset unde s3s2s1s0 este o constantă; - registru\_segment : specificare\_offset, registru segment fiind CS, DS, SS, ES, FS sau GS; - FAR [variabilă], unde variabilă este de tip QWORD şi conţine cei 6 octeţi constituind adresa FAR. (ceea ce numim variabila pointer in limbajele de nivel inalt). Formatul intern al unei adrese FAR este: la adresa mai mică se află offsetul, iar la adresa mai mare cu 4 (cuvântul care urmează după dublucuvântul curent) se află cuvântul ce conţine selectorul care indică segmentul. (respecta little endian)

INSTRUCTIUNI MASINA

O instrucţiune maşină x86 reprezintă o secvenţă de 1 până la 15 octeţi, care prin valorile lor specifică o operaţie de executat, operanzii asupra cărora va fi aplicată, precum şi modificatori suplimentari care controlează modul în care aceasta va fi executată.

Maxim 2 operanzi (sursa si destinatie),maxim unul din memorie (celalalt registru sau constanta) -> numeinstrucţiune destinaţie, sursă

Formatul intern al unei instructiuni: [prefixe] + cod + [ModR/M] + [SIB] + [deplasament] + [imediat]. Prefixele controlează modul în care o instrucţiune se execută. Acestea sunt opţionale (0 până la maximum 4) şi ocupă câte un octet fiecare. (de ex repmovsb). Octetul ModR/M (mod registru/memorie) specifică pentru unele dintre instrucţiuni natura şi locul operanzilor (registru, memorie). Pentru cazuri mai complexe de adresare decât cele codificabile direct prin ModR/M, combinarea acestuia cu octetul SIB (Scale – Index – Base) permite următoarea formulă generală de definire a unui offset: offset = [bază] +[index × scală] (SIB)+ [constantă] (deplasament + imediat). bază şi index vor fi folosite valorile a doi regiştri iar scală este 1, 2, 4 sau 8. Regiştrii permişi ca bază sau / şi index sunt: EAX, EBX, ECX, EDX, EBP, ESI, EDI. Registrul ESP este disponibil ca bază însă nu poate fi folosit cu rol de index. Valoare imediată oferă posibilitatea definirii unui operand ca fiind o constantă numerică pe 1, 2 sau 4 octeţi.

MODURI DE ADRESARE

În cadrul unei instrucţiuni există 3 moduri de a specifica un operand pe care aceasta îl solicită:

- modul registru, dacă pe post de operand se află un registru al maşinii; mov **eax**, 17

- modul imediat, atunci când în instrucţiune se află chiar valoarea operandului (nu adresa lui şi nici un registru în care să fie conţinut); mov eax, **17**

- modul adresare la memorie, dacă operandul se află efectiv undeva în memorie. În acest caz, offsetul lui se calculează după următoarea formulă: adresa\_offset = [ bază ] + [ index × scală ] + [ constanta ]

Deci adresa\_offset se obţine din următoarele (maxim) patru elemente:

- conţinutul unuia dintre regiştrii EAX, EBX, ECX, EDX, EBP, ESI, EDI sau ESP ca bază;

- conţinutul unuia dintre regiştrii EAX, EBX, ECX, EDX, EBP, ESI sau EDI drept index;

- factor numeric (scală) pentru a înmulţi valoarea registrului index cu 1, 2, 4 sau 8

- valoarea unei constante numerice, pe octet, cuvant sau dublucuvânt.

De aici rezultă următoarele moduri de adresare la memorie:

- directă, atunci când apare numai constanta;

- bazată, dacă în calcul apare unul dintre regiştrii bază;

- scalat-indexată, dacă în calcul apare unul dintre regiştrii index;

Cele trei moduri de adresare a memoriei pot fi combinate. De exemplu, poate să apară adresare directă bazată, adresare bazată şi scalat-indexată etc

-Adresa relativă indică poziţia următoarei instrucţiuni de executat, în raport cu poziţia curentă ()adresare relativa pentru salturi