Fila 1

MOVFUSCATOR

Ivan Miruna Maria - Grupa 141
Parapeanu Maria - Grupa 131
Popa David-Gabriel - Grupa 134
Teodorescu Luca Nicolae - Grupa 151

Obiectiv:

Traducerea a cat mai multor instructiunilor dintr-un program Assembly in instructiuni de tip “mov”.

Functionalitate:

Programul citeste un fisier assembly de intrare, identifica main-ul, iar apoi in functie de instructiune, realizeaza transformari specifice pentru a o rescrie.

- Instructiunea add:

Este simulata prin operatii pe bytes si tabele de look-up si foloseste strict mutari de date(mov).

Operanzii sunt impartiti in 4 bytes. Adunarea se realizeaza byte cu byte. Stocam registrii intr-o zona de memorie special alocata pentru a putea extrage byte cu byte. Pentru fiecare pereche de bytes, se utilizeaza un tabel de suma(look-up table) pe 256x256 (cu indecsi de la 0 la 255). Conceptul acestui tabel are la baza inlocuirea timpului de procesare cu spatiu de memorie. Practic vom avea precalculate toate combinatiile de sume posibile(mod 256), astfel incat la intersectia rand(%ah) coloana(%al) se va afla rezultatul sumei celor 2 bytes. Indexul il avem in %esi. Punand byte-ul primului operand in %ah si pe celalalt in %al, vom avea in %ax pozitia corecta: (rand*256)+coloana. Este necesar si un tabel de carry care care indica, (tot conform conceptului prezentat la sum_table), prezenta carry-ului. De exemplu: daca avem in %ah 255 si in %al 1, la intersectia celor 2 vom gasi valoarea 256. Dar aceasta valoare produce un transport, asa ca verificam in tabelul de carry, unde vom gasi valoarea 1. Carry-ul este transportat intre bytes.. Rezultatul presupune reconstruirea de bytes calculati.

- Instructiunea sub:

Este in esenta logica add-ului, dar in complement fata de 2.

Se inverseaza bitii, se adauga 1, pentru a obtine -(operand). Se apeleaza add cu noul operand.

- Instructiunea inc:

Este un add cu operand $1.

- Instructiunea dec:

Este un sub cu operand $-1.

- Instructiunea mul:

- Instructiunea shift left:

Este un mul repetat cu operand 2. Instructiunea se realizeaza repetat pe baza marimii operandului.

- Instructiunea and:

- Instructiunea xor:

- Instructiunile jump si cmp:

- Instructiunea lea:

In cazul nostru, o putem substitui cu mov $operand1, operand2, din moment ce este folosita doar pentru accesarea adresei inceputului unui array.

Limitari:

Instructiunea de div pare foarte dificil de implementat folosind doar tabele si mutari byte cu byte. Din acelasi motiv, atat ea, cat si shift right vor fi ignorate si pastrate asa cum sunt.

Fila 2