

## **Modelul de Memorie "Flat" (Flat Memory Model)**

**Q:** Pot defini date (de ex., a db 17) în interiorul segmentului de cod? **A:** Da, puteți plasa directive de date în segmentul de cod, deși buna practică este să separați datele în segmentul de date.

**Q:** Care este diferența dintre Mov eax, c și Mov edx, [c]? **A:** Mov eax, c încarcă în EAX adresa (offset-ul) variabilei c. Mov edx, [c] încarcă în EDX valoarea stocată la adresa variabilei c.

**Q:** De ce instrucțiunile mov edx, [CS:c], mov edx, [DS:c], mov edx, [SS:c] și mov edx, [ES:c] au toate același efect? **A:** Acest lucru se datorează modelului de memorie flat (flat memory model). În acest model, toți selectorii de segment (CS, DS, SS, ES) descriu în realitate aceeași zonă de memorie: întreaga memorie de 4GiB, începând de la adresa 0.

**Q:** Ce înseamnă că "mecanismul de segmentare este transparent" în modelul flat? **A:** Înseamnă că programatorul nu trebuie să își facă griji cu privire la gestionarea diferitelor segmente de memorie. Putem accesa orice locație de memorie fără a schimba explicit registrele de segment (CS, DS, etc.), deoarece toate indică aceeași zonă.

**Q:** Selectorii FS și GS respectă și ei modelul flat? **A:** Nu întotdeauna. FS și GS sunt segmente speciale, care nu respectă neapărat modelul flat. Ele sunt adesea rezervate pentru interacțiunea cu sistemul de operare (de exemplu, pentru a accesa date specifice thread-ului - Thread Local Storage). [FS:c] nu garantează accesarea aceleiași locații ca [CS:c].