

## **Moduri de Adresare și Acces la Memorie**

**Q: Care este regula fundamentală pentru a accesa adresa vs. conținutul unei variabile în NASM?**

**A:** Folosirea numelui variabilei (de ex. p) se referă la adresa (offset-ul) variabilei. Folosirea numelui între paranteze drepte (de ex. [p]) se referă la conținutul (valoarea) de la acea adresă.

**Q: Care este diferența dintre mov EAX, et și mov EAX, [et]?**

**A:** mov EAX, et încarcă în EAX adresa etichetei et. mov EAX, [et] încarcă în EAX conținutul (valoarea) de la adresa et.

**Q: Ce face instrucțiunea lea eax, [v]?**

**A:** lea (Load Effective Address) încarcă adresa (offset-ul) variabilei v în registrul eax.

**Q: Ce înseamnă folosirea parantezelor pătrate []?**

**A:** Indică întotdeauna accesarea unui operand din memorie. De exemplu, mov EAX, [EBX] copiază în EAX conținutul de la adresa de memorie aflată în EBX.

**Q: Care sunt cele trei tipuri de operanzi?**

**A:** Operanzi imediați (constante), operanzi registru (de ex. eax) și operanzi în memorie.

**Q: Când sunt calculate valorile operanzilor?**

**A:** \* Operanzii imediați: La momentul asamblării (assembly time).

- Operanzii registru și cei adresați indirect: La momentul execuției (execution time).
- Adresarea directă (adresa FAR): La momentul încărcării (loading time).

**Q: Ce regulă specială se aplică numerelor hexazecimale care folosesc sufîxul H? A:** Trebuie să înceapă obligatoriu cu o cifră (0-9). De exemplu, 0ABCH este un număr valid, dar ABCH este interpretat ca un simbol (etichetă).

**Q: Care sunt pașii pentru calcularea adresei fizice finale a unei variabile?**

**A:** 1. Assembly time: Se calculează deplasamentul (offset-ul) față de începutul segmentului. 2. Linking time: Se stabilește segmentul la care se raportează deplasamentul. 3. Loading time: Se calculează adresa fizică efectivă (proces numit Relocarea Adreselor).

**Q: Ce registru de segment este folosit implicit pentru accesul la date?**

**A: Implicit se folosește DS (Data Segment) pentru majoritatea accesărilor de date. SS (Stack Segment) este folosit implicit pentru adresări care folosesc EBP sau ESP ca bază. CS (Code Segment) este folosit pentru etichete de cod (salturi).**

**Q: Cum se poate suprascrie (override) explicit registrul de segment implicit? A:**  
**Folosind operatorul de specificare a segmentului, : (două puncte). De exemplu: mov eax, [ES:var].**