Directive de Definire a Datelor (Data Definition)

Q: Unde încep de obicei segmentele CODE și DATA în OllyDbg? A: De obicei, un segment (de ex., CODE) începe la offset-ul 00401000, iar celălalt (de ex., DATA) la 00402000. Ordinea poate fi inversă, în funcție de setările linker-ului.

Q: Care este diferența dintre offset-ul unei variabile determinat de NASM (la asamblare) și cel văzut în OllyDbg (la încărcare)? A: NASM (asamblorul) calculează offset-ul variabilei față de începutul segmentului său (de exemplu, 0, 8, etc.). OllyDbg (debugger-ul) arată adresa de memorie absolută, la momentul încărcării (loading time), care include adresa de bază a segmentului (de exemplu, 00402000).

Q: Offset-ul unei variabile este determinat la asamblare sau la încărcare? A: Offset-ul variabilei *față de începutul segmentului* este o constantă determinată la momentul asamblării. Adresa finală (absolută) în memorie este determinată la momentul încărcării programului.

Q: Ce se întâmplă dacă definesc db 300? A: Veți primi un avertisment (Warning) "byte data exceeds bounds". Valoarea 300 depășește valoarea maximă a unui octet (255) și va fi trunchiată (de exemplu, la 2Ch).

Q: Cum se folosește directiva TIMES? A: Repetă directiva de definire a datelor de un număr specificat de ori. De exemplu, a2 TIMES 3 db 44h este echivalent cu db 44h, 44h, 44h.

Q: De ce o definire ca a11 db [a2] produce "expression syntax error"? A: Deoarece [a2] implică o dereferențiere (citirea valorii *de la* adresa a2). Directivele de definire a datelor (db, dw, dd) necesită valori constante, care pot fi calculate la momentul asamblării. O dereferențiere nu este o constantă.

Q: De ce o definire ca a15 dd eax produce "expression syntax error"? A: Deoarece eax este un registru. Valoarea unui registru nu este o constantă cunoscută la momentul asamblării și nu poate fi folosită pentru a defini date statice.

Q: De ce a7 db a2 poate produce o eroare de sintaxă? A: În formatul OBJ, este posibil să se genereze o eroare "can only handle 16- or 32- relocation". Adresele (pointerii) sunt de obicei stocate ca word (dw) sau double-word (dd), nu ca byte (db).

Q: Ce face directiva SEGMENT (sau SECTION)?

A: Direcționează asamblorul să emită octeții generați (cod sau date) într-un segment cu un nume și atribute specifice.

Q: Numiți câteva atribute importante ale unui segment.

A: * **tip:** code (executabil), data (citire/scriere), rdata (doar citire).

- **combinare:** PUBLIC (concatenare), COMMON (suprapunere), PRIVATE (necombinat), STACK.
- **utilizare:** use32 (indică un segment de 32 de biți).

Q: Care este diferența dintre directivele DB și RESB?

A: DB (Define Byte) **definește și inițializează** unul sau mai mulți octeți cu valori specifice. RESB (Reserve Byte) doar **rezervă** un spațiu de un număr de octeți neinițializați.

Q: Care sunt directivele pentru definirea datelor *inițializate*?

A: DB (1 octet), DW (2 octeți), DD (4 octeți), DQ (8 octeți), DT (10 octeți).

Q: Care sunt directivele pentru rezervarea spațiului *neinițializat*?

A: RESB, RESW, RESD, RESQ, REST.

Q: Cum se rezervă un word neinițializat în NASM? Se folosește DW ??

A: Nu, NASM nu suportă sintaxa DW ?. Pentru a rezerva un word neinițializat, se folosește nume_var: resw 1.

Q: Ce face directiva TIMES?

A: Permite asamblarea repetată a unei instrucțiuni sau a unei definiții de dată de un număr specificat de ori.

Q: Dați un exemplu de folosire a directivei TIMES pentru date și pentru instrucțiuni.

A: * **Date:** matrice10x10 TIMES 10*10 DD 0 (creează un tablou de 100 de dublucuvinte, toate inițializate cu 0).

- **Instrucțiune:** TIMES 32 add eax, edx (repetă instrucțiunea add eax, edx de 32 de ori).

Q: Ce face directiva EQU? A: Atribuie o valoare numerică sau un șir de caractere unei etichete la momentul asamblării. **Nu alocă spațiu** de memorie. Este similară cu definirea unei constante. Exemplu: BUFFER_SIZE EQU 1000h.

Q: De ce instrucțiunea mov [v], 0 produce o eroare de sintaxă?

A: Produce eroarea "operation size not specified". Asamblorul nu știe dacă să scrie un octet, un cuvânt sau un dublucuvânt la adresa v.

Q: Cum se corectează eroarea mov [v], 0?

A: Prin specificarea explicită a dimensiunii folosind un operator de tip: mov BYTE [v], 0 sau mov WORD [v], 0 sau mov DWORD [v], 0.

Q: Ce se întâmplă cu instrucțiunea push 15? A: Aceasta este o inconsistență în NASM. Asamblorul nu va emite o eroare, ci o va trata implicit ca push DWORD 15.