

De la sursa la executabil

Fisierul pe care il creez eu (de obicei cu extensie .c sau .cpp sau .py) se numeste fisier sursa. Acest fisier sursa nu poate fi inteles de procesor, este lizibil doar pentru noi ca programatori, de asta avem nevoie de un intermediar, mai exact sa-l compilam.

Ce presupune compilarea unui fisier sursa(atunci cand dai Run in IDE-ul preferat)? Ei bine mai intai se verifica sintaxa programului tau sa fie corecta, precum si alte erori, dupa care este transformat in cod assembly iar dupa in cod binar(care este inteles de masina ta). Daca a ajuns pana in punctul asta atunci se creaza un fisier de tip obiect.

Acest fisier obiect contine codul masina insa nu poate fi executat inca. In acest fisier obiect sunt "goluri" sau placeholders, mai exact, atunci cand se face apelul unei functii in codul tau, in fisierul obiect va fi un placeholder precum 0x0000..

De asta avem nevoie de linkare= umple acele goluri cu adresa din memorie a functiilor apelate de tine(aceste functii pot fi biblioteci speciale precum math) iar dupa asta se creeaza executabilul.

Executabilul poate fi static(linker-ul ia o copie a tot codului necesar din biblioteci și îl copiază fizic în interiorul executabilului final) sau dinamic(pune în executabil doar o mică referință)

Legat de executabile dinamice exista 2 modalitati de a folosi bibliotecile partajate: load-time(implicit) run time(explicit)

load time=E ca și cum ai pleca într-o excursie și verifici rucsacul la ieșirea din casă. Dacă îți lipsește periuța de dinți, nu pleci deloc.

Mecanism: În header-ul executabilului există o listă clară: "Am nevoie de libA.dll și libB.dll".

Riscul :dacă o singură bibliotecă lipsește (chiar și una mică, neimportantă), programul nu pornește deloc. Primești direct eroarea de sistem ("Missing DLL").

Avantaj: Dacă programul a pornit, ai garanția că toate funcțiile necesare sunt acolo. Nu vei avea surprize la mijlocul rulării.

run time=Aceasta este metoda folosită pentru plugin-uri sau aplicații complexe care vor să pornească rapid. Programatorul scrie cod special pentru a încărca biblioteca manual.

- Cum funcționează: Pleci în excursie doar cu hainele de pe tine. Când îți se face sete, te oprești la un magazin să iei apă. Dacă magazinul e închis, nu mori, doar nu bei apă și îți continui drumul.
- Mecanism: Programul nu are lista de biblioteci în header. În schimb, în codul sursă apar comenzi explicite către OS (ex: LoadLibrary pe Windows sau dlopen pe Linux) exact în momentul când e nevoie de acea funcționalitate.
- Avantaj major:
 - Viteză: Programul pornește instantaneu (nu încarcă nimic inutil).
 - Flexibilitate: Dacă biblioteca lipsește, programul NU crapă obligatoriu. Programatorul poate scrie un cod de genul: "Dacă nu găsești biblioteca de sunet, afișează un mesaj de eroare, dar lasă utilizatorul să folosească restul aplicației (partea de text)".

- Dezavantaj: Este mult mai greu de programat. Trebuie să gestionezi manual pointeri și erori.