

Limbajul Fortran: istoric și caracteristici

1.1. Scurt istoric al limbajului Fortran

Denumirea limbajului **Fortran**, vine de la *FORmula TRANslation*, abreviere de la inițialul (*IBM Mathematical FORmula TRANslation System*). Fortran este unul dintre primele limbaje de „nivel înalt” dezvoltate și adoptate de comunitatea academică și științifică. Limbajul a fost dezvoltat într-o perioadă de trei ani (1954-1957) de către o echipă de la **IBM** condusă de **John W. Backus** și compusă din programatori de marcă, cum ar fi Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Richard Goldberg, Lois Haibt și David Sayre.

Noua invenție a avut succes repede și, pe bună dreptate, scrierea programelor de calcul implicând parametri legați de reactoarele de electricitate nucleară durau numai câteva ore, în loc de săptămâni și cereau mult mai puține cunoștințe de programare. Un alt avantaj al noii invenții a fost portabilitatea programelor. Fortran câștiga bătălia împotriva limbajului de asamblare, prima din seria de bătălii ce aveau să vină și a fost adoptat de comunitățile științifice și militare fiind folosit intensiv în Programul Spațial și în proiectele militare.

Limbajul a devenit atât de popular la începutul anilor '60 încât alte firme au început să producă versiuni proprii, ceea ce a dus la creșterea dialectelor (în 1963 existau 40 de compilatoare diferite). Au apărut versiunile **Fortran II(1958)**, **Fortran III(1958)** și **Fortran IV(1961)**. A fost folosit un translator Fortran II - Fortran IV pentru a se menține compatibilitatea cu programele Fortran anterioare [Cou 91]. Totuși, divergența produsă de mulțimea dialectelor reprezenta un dezavantaj atât pentru utilizatori cât și pentru producători și astfel, în mai 1962, Fortran a devenit *primul limbaj de nivel înalt standardizat oficial*. În 1966 a fost adoptat primul standard ANSI (American National Standards Institute). Standardizarea a făcut ca Fortranul să fie folosit mai mult ca oricare alt limbaj. Pe la mijlocul anilor '70, practic orice calculator, de birou sau mainframe, avea un sistem de procesare standard compatibil cu **Fortran 66**. Se făcea posibilă scrierea programelor Fortran pe orice sistem, iar acestea puteau fi mutate cu ușurință pe orice alt sistem. Acest lucru, ca și faptul că programele Fortran puteau fi procesate în mod eficient, au făcut ca Fortran să fie cel mai utilizat limbaj de programare în aplicațiile non-comerciale.

Definiția standard a Fortranului a fost actualizată la sfârșitul anilor '70, iar în 1978 Institutul American National de Standarde a publicat al doilea standard, ANSI X3.9, care permitea programarea structurată și introducea caracteristici noi pentru IF și tipul de date caracter. Acest standard a fost adoptat în 1980 de ISO (International Organisation Standardisation) ca standard internațional. Limbajul a devenit cunoscut ca **Fortran 77** și reprezenta versiunea cea mai răspândită. Compilatoarele care la început suportau un număr mic de extensii au devenit, cu trecerea anilor, foarte eficiente.

Caracterul conservator al standardului pentru Fortran 77 a lăsat limbajului un număr de facilități de modă veche care ar putea părea deficiente sau cel puțin cazuri nefericite. Multe caracteristici necesare nu erau disponibile. De exemplu, în Fortran 77 sunt greu de reprezentat structurile de date și lipsa reprezentării dinamice nu permite crearea unor matrici de dimensiune mare. Era necesară dezvoltarea unui limbaj nou, modern. În anii '80 a început munca la un limbaj

nou cunoscut ca Fortran 8x. Dezvoltarea limbajului a durat 12 ani, în parte din cauza încercării de a păstra caracterul limbajului și totuși de a îi mări eficiența. Limbaje ca Pascal, ADA și Algol sunt mai ușor de utilizat dar nu se pot compara, din punct de vedere al eficienței, cu Fortranul.

A treia standardizare a limbajului Fortran a avut loc în 1991 și s-a numit **Fortran 90**. Apariția târzie a acestei versiuni a permis apariția și dezvoltarea altor limbaje care concurează cu Fortranul. De exemplu limbajul de programare C și varianta sa evoluată C++, au devenit mai populare în rândul susținătorilor convingși ai Fortranului, în ciuda lipsei de orientare computațională a acestuia.

Versiunea Fortran 90 reprezintă o îmbunătățire majoră a versiunilor anterioare și caracteristicile sale sunt prezentate mai jos.

Fortran 90 include Fortran 77 ca pe o submulțime strictă și astfel, orice program Fortran 77 va continua să reprezinte un program Fortran 90 valid, făcându-se astfel posibilă și refolosirea tuturor programelor Fortran 77. În plus, Fortran 90 permite exprimarea programelor în moduri mult mai adecvate unui mediu de calcul modern.

În ultimii ani [**1] a fost dezvoltat limbajul bazat pe Fortran 90 și cunoscut sub numele de **High Performance Fortran (HPF)**. Acest limbaj conține întregul Fortran 90 precum și alte îmbunătățiri. **Fortran 95** conține multe dintre caracteristicile din HPF. HPF este destinat programării mașinilor cu memorie distribuită și a introdus „directive” (comentarii Fortran 90 structurate) pentru distribuirea datelor (tablouri) pe grile de procesoare (neomogene). Ideea este de a scuti programatorul de scrierea codului de transfer de mesaje: acest lucru este făcut de compilator. HPF mai introduce un mic număr de instrucțiuni executabile (atribuirii paralele, proceduri libere de efecte colaterale (PURE procedures) care au fost adoptate de Fortran 95).

Munca a continuat în 1996 și s-a încheiat cu publicarea standardului 1995, la care s-au adăugat două *rapoarte tehnice* ISO (International Organization for Standardization) care au fost publicate în *IEEE Floating Point Arithmetic and Allocatable Attributes*.

O variantă foarte bună a limbajului Fortran este și cea din 2000 [w24], [w25], [w26].

1.2. Noile caracteristici ale limbajului Fortran 90

În această formă, limbajul Fortran redevine competitiv cu alte limbaje de programare.

Printre noile caracteristici ale limbajului se numără următoarele:

- forma liberă a sursei (independentă de coloane);
- numele de obiecte pot avea lungimea de până la 31 de caractere;
- sunt permise atât literele mici cât și majusculele sau underscore-ul;
- blanc-urile sunt semnificative;
- liniile pot conține până la 132 caractere;
- sunt permise până la 39 de linii de continuare;
- ; ca separator între mai multe instrucțiuni aflate pe aceeași linie;
- ! ca simbol de comentariu; sunt permise comentarii în cadrul liniilor codului;
- include opțiuni pentru textul sursă din fișiere;
- structuri de control moderne;
- Fortran 90 prezintă o instrucțiune DO modernă cu opțiunile CYCLE (pentru abandonarea iterației curente a unui ciclu și începerea următoarei) și EXIT (pentru ieșirea

din ciclul DO), iar partea de control a instrucțiunii DO poate fi reprezentată de către o iterație convențională, WHILE, sau poate să nu existe o clauză de control;

- blocul de control SELECT CASE, care este mult mai succint, elegant și eficient decât un bloc IF ... ELSEIF ... ELSEIF;
- specificarea preciziei numerice, prin parametrul KIND, permite creșterea portabilității;
- specificarea controlului intervalului:

REAL(SELECTED_REAL_KIND(PRECISION=10,RANGE=50)) :: x, y, z;

- procesarea matricelor (secțiuni de matrice, operatori pentru matrice etc) caracteristică paralelismului; ca suport al acestei caracteristici au fost introduse mai multe funcții paralele intrinseci, inclusiv operații de reducere ca SUM (întoarce rezultatul sumei elementelor unui tablou) și MAXVALL (întoarce valoarea maximă a elementelor unui tablou). O caracteristică nouă o reprezintă și instrucțiunea de atribuire paralelă WHERE;

- este acum posibilă tratarea matricelor ca entități și scrierea

A=B*SIN(A)

unde A și B sunt matrice.

- comportament dinamic, incluzând alocarea, dealocarea, pointeri și recursivitate explicită; pot fi create tablouri temporare, pot fi implementate liste înlănțuite și arbori, cu ajutorul pointerilor;
- tipuri de date definite de utilizator;
- modulele pot fi folosite pentru definiții globale (de tipuri, operatori și proceduri) și pentru ascunderea datelor; se pot declara interfețe, programele și procedurile pot fi compilate separat, rezultând o dezvoltare mai eficientă, forma tablourilor se poate moșteni de la argumentul actual, sunt permise procedurile interne (o procedură poate conține o altă procedură cu un domeniu local - nu poate fi accesată din afara procedurii în care a fost declarată). Modulele oferă și facilități bazate pe obiecte, modulul poate fi folosit ca unitate de bibliotecă
- supraîncărcarea operatorilor pentru tipuri de date intrinseci și derivate, adică se pot face 'extensii semantice'. De exemplu, un întreg de lungime arbitrară poate fi implementat folosind o structură de listă înlănțuită (o cifră pe celulă) și apoi se pot defini toți operatorii intrinseci (supraîncărcați) astfel încât acest tip să fie tratat în același mod ca toate celelalte tipuri. Limbajul pare astfel extins. Toți operatorii intrinseci +, -, *, / și ** și atribuirea =, pot fi supraîncărcați (definiți pentru tipuri noi);
- proceduri generice;
- moștenire (controlată).

Diagrama din figura 1.1 [**1] prezintă principalele caracteristici ale limbajului Fortran 90. Mărimea fiecărei "felii" de diagramă este aproximativ proporțională cu numărul de reguli de sintaxă necesare pentru a descrie caracteristicile asociate și reprezintă o măsură a complexității structurale a acestora.