İkinci Bölüm

Programlama Dilleri – Kısa Kısa

1. Plankalkül -1945

Hiç gerçekleştirilmemiştir. Gelişmiş veri yapilari desteği mevcuttur.
Notasyon:

A[7] = 5 * B[6]
| 5 * B => A
V | 6
7

(indeks)

S | 1.n 1.n (veri tipi)

2 Pseudocodes -1949

1949' da geliştirilmiştir; BINAC; Mauchly
Bazi operasyonlar:

1n => (n+2)nd power
2n => (n+2)nd root
07 => addition

pseudocode: sözde kod. Makine dili ile yazilmamiş, kullanilmadan önce çevrilmesi gereken program

3 IBM 704 ve FORTRAN (FORmula TRANSlation)

(FORTRAN 0 -1954 -gerçekleştirilmedi)
FORTRAN I -1957'de ilk gerçekleştirilen FORTRAN' programlama dilidir. Yeni IBM 704 için gerçekleştirilmiştir. Bazı özellikleri şunlardır:

- Dizin yazmaci (index registers) ve kayan noktali aritmetik donanimi
- 6 karaktere kadar isim desteği
- DO loop döngüsü
- Formatted i/o
- Alt programlar
- Arithmetic IF: if(aritmetik ifade) N1,N2,N3
- veri tipi yok

400 satirdan uzun program nadiren derlenmiştir, bunun nedeni IBM 704'ün güvenilmezliğiydi. Kod hizliydi. Hızla kullanilmaya başladi.

3 IBM 704 ve FORTRAN (devam)

-FORTRAN II -1958

-FORTRAN IV -1960-62

-veri tipi deklerasyonu
-Mantiksal if,

FORTRAN 77 -1978

-karakter dizgisi
-mantiksal döngü kontrolü
-IF-THEN-ELSE deyimi
-Fortran 90 -1990
-Moduller, dinamik array, pointer
-Fortran 95 –1995
-Fortran 2003
-FORTRAN değerlendirme
-Çok büyük ölçüde değişti ve hala kullaniliyor.

4 LISP -1959
-LIStProcessing language
(Designed at MIT by McCarthy)

-İki veri tipi var: atom ve list
-Sözdizim lambda calculus'a dayanir
-Fonksiyonel programlamada öncü
-Değişkenlere gerek yok.
-Özyineleme (recursion) ve koşullu ifadeler ile kontrol.
-Yapay zeka için hala dominant.
-COMMON LISP ve Scheme çağdaş lehçeleri.
-ML, Miranda, ve Haskell ilgili diller.

4 LISP – 1959 (devam)
list (A B C D) ve (A (B C) D (E (F G)))'nin gösterimi
5 ALGOL 58 ve 60

-ACM and GAMM 4 günlük toplantida kararlaştirildi.
-ALGOL 58 Language Features:
-type kavrami
-isim boyu serbest
-Array indeksleri serbest
-Compound statements (begin ... end)
-noktali virgül komut ayiraci.
-atama operatorü :=
-if else-if
-Başta IBM desteği vardi orta-1959 da kestiler.

5 ALGOL 58 and 60 (devam)
-ALGOL 60

-Pariste bir toplantida 6 günde geliştirildi.

Yeni özelliler:

- Block yapisi (local scope)
- İki parametreli fonksiyon çağrıları
- Altprogram özyinelemesi (Subprogram recursion)

Başarı:

- 20 yıldan daha uzun süre algoritmaları sunmada standart yol oldu
- Imperative diller için temel dil oldu
- İlk makine bağımsız dildir
- Sözdizimi formal olarak (BNF) tanımlanmış ilk dildir
- Bir komite tarafından tasarlanmış ilk dildir

Başarısızlık:

- Başta US olmak üzere Hiçbir zaman geniş bir kullanıma sahip olmadı

Sebepler:

1. i/o yoktu ve karakter seti programları taşınamaz hale getirmiştir
2. Çok esnekti – gerçekleştirimi zordu
3. FORTRAN'ın siperiydi
4. Resmi sözdizim açıklaması vardı
5. IBM desteğinden yoksundu

6 COBOL -1960

-FLOW-MATIC tabanlıdır

FLOW-MATIC özellikleri:

- Names up to 12 characters, with embedded hyphens
- English names for arithmetic operators (no arithmetic expressions)
- Data and code were completely separate
- Verbs were first word in every statement

-First Design Meeting (Pentagon) -May 1959

-Design committee members were all from computer manufacturers and DoD branches
-Design Problems: arithmetic expressions? subscripts? Fights among manufacturers

-Contributions:
-First macro facility in a high-level language
-Hierarchical data structures (records)
-Nested selection statements
-Long names (up to 30 characters), with hyphens
-Separate data division

7 BASIC -1964

-Designed by Kemeny & Kurtz at Dartmouth
-Current popular dialect: Visual BASIC.NET
-First widely used language with time sharing

8 PL/I -1965

-Designed by IBM and SHARE
-Computing situation in 1964 (IBM's point of view)

1. Scientific computing
-IBM 1620 and 7090 computers
-FORTRAN
-SHARE user group

2. Business computing
-IBM 1401, 7080 computers
-COBOL
-GUIDE user group
-By 1963:
-Scientific users needed more i/o-Business users needed fl. pt. and arrays (MIS)

-The obvious solution:

1. Build a new computer to do both kinds of applications
2. Design a new language to do both kinds of applications

8 PL/I (continued)
-Designed in five months by the 3 X 3 Committee
-PL/I contributions:

1. First unit-level concurrency
2. First exception handling
3. Switch-selectable recursion
4. First pointer data type
5. First array cross sections

-Comments:
-Many new features were poorly designed
-Too large and too complex
-Was (and still is) actually used for both

- scientific and business applications

9 APL and SNOBOL

-Characterized by dynamic typing and dynamic storage allocation

-APL (A Programming Language) 1962

-Designed as a hardware description language (at IBM by Ken Iverson)
-Highly expressive (many operators, for both scalars and arrays of various dimensions)
-Programs are very difficult to read

-SNOBOL(1964)
-Designed as a string manipulation language (at Bell Labs by Farber, Griswold, and Polensky)
-Powerful operators for string pattern matching

10 SIMULA 67 -1967

-Designed primarily for system simulation

(in Norway by Nygaard and Dahl)
-Based on ALGOL 60 and SIMULA I
-Primary Contribution:

Classes, objects, and inheritance

11 ALGOL 68 -1968

-From the continued development of ALGOL 60, but it is not a superset of that language
-Design is based on the concept of orthogonality

2.12 Important ALGOL Descendants
-Pascal – 1971 -Wirth

-Designed for teaching structured programming
-Small, simple, nothing really new
-From mid-1970s until the late 1990s, it was the

most widely used language for teaching programming in colleges

-C -1972

-Designed for systems programming -Richie
-Evolved primarily from B, but also ALGOL 68
-Powerful set of operators, but poor type checking
-Initially spread through UNIX

-Perl – 1987 – Larry Wall
-Related to ALGOL only through C
-Some consider it a scripting language
-Perl variables are statically typed and implicitly declared
-Three distinctive namespaces, denoted by the first character of a variable's name
-Powerful but somewhat dangerous-Widely used as a general purpose language and

for CGI programming on the Web

2.13 Prolog – 1972

-Developed at the University of Aix-Marseille, by Comerauer and Roussel, with some help from Kowalski at the University of Edinburgh

-Based on formal logic
-Non-procedural
-Can be summarized as being an intelligent database system that uses an inferencing process to infer the truth of given queries

4 Ada -1983 (began in mid-1970s)

-Huge design effort, involving hundreds of people, much money, and about eight years

-Environment: More than 450 different languages being used for DOD embedded systems (no software reuse and no development tools)

-Contributions:

1. Packages -support for data abstraction
2. Exception handling -elaborate
3. Generic program units
4. Concurrency -through the tasking model

-Comments:

-Competitive design

-Included all that was then known **about** software engineering and language design
-First compilers were very difficult;
the first really usable compiler came nearly five years after the language design was completed

-Ada 95 (began in 1988)

-Support for OOP through type derivation
-Better control mechanisms for shared data(new concurrency features)
-More flexible libraries

15 Smalltalk -1972-1980

-Developed at Xerox PARC, initially by Alan Kay, later by Adele Goldberg
-First full implementation of an object-oriented language (data abstraction, inheritance, and dynamic binding)
-Pioneered the graphical user interface everyone now uses

16 C++ -1985

-Developed at Bell Labs by Stroustrup
-Evolved from C and SIMULA 67
-Facilities for object-oriented programming, taken partially from SIMULA 67, were added to C
-Also has exception handling
-A large and complex language
-Rapidly grew in popularity, along with OOP
-ANSI standard approved in November, 1997!!

-Eiffel -a related language that supports OOP
-(Designed by Bertrand Meyer -1992)
-Not directly derived from any other language
-Smaller and simpler than C++, but still has most of the power

16 C++ -1985 (continued)
-Delphi – another related language -Designed by Anders Hejlsberg (Turbo Pascal and C#)
-A hybrid language based on Pascal

17 Java (1995)

-Developed at Sun in the early 1990s
-Based on C++
-Significantly simplified

(does not include struct, union, enum, and half of the assignment coercions of C++)
-Supports only OOP
-Has references, but not pointers
-Includes support for applets and a form of

concurrency

18 Scripting Languages

-JavaScript (1985)
-Began as LiveScript at Netscape
-Mostly a client-side HTML-resident scripting language
-Widely used for designing client-side dynamic

Web documents and for validating form data before it is sent to the server for processing
-Purely interpreted

-PHP – 1994 -Rasmus Lerdorf

-Server-side HTML-resident scripting language

-Widely used for form processing and database access through the Web

-Purely interpreted

-Has a new data structure

-Python – early 1990s – Guido Van Rossum
-Systems administration, CGI programming
-Dynamically typed, but type checked
-In place of arrays, it has lists, immutable lists (tuples), and hashes (dictionaries)
-Ruby – middle 1990s – Yukihiro Matsumoto
-A pure OOPL – everything is an object
-A scripting language
-Dynamically typed – variables are not typed
-Both classes and objects are dynamic

2.19 C# -2000 -Microsoft

-The primary language of the .NET platform

-A successor to both Java and C++

-Includes most of the features of Java, with some modifications, as well as some of the features of C++ that were left out of Java

-Can be used for both .NET applications for the Web, as well as for general-purposes

2.20 Markup/Programming HybridLanguages

-XSLT
-Used to transform the contents of an XML document for display
-An XSLT processor takes an XML document andan XSLT document as input and produces an output document
-The XSLT processor calls subprograms when patterns in the XML document are found
-Includes control structures in the form of tags, such as <for-each>

-JSP
-A Java Server Page can be a mixture of XHTML and Java
-Pages are processed by a JSP processor into servlets
-The JSTL defines XML action elements that control the processing of the JSP document
-Example action elements are: <if>, <forEach>, etc.