Lecture 07

Assembly Language to ALGOL-60

Mr. Mubashir Ali
Lecturer (Dept. of Computer Science)
dr.mubashirali1@gmail.com

Summary of Previous Lecture

1. Von Neumann Architecture
2. Influences on Language Design
3. Application Domains

Outline

1. Zuse's Plankalkul – 1945
2. Assemblers and Assembly Languages
3. FORTRAN I
4. FORTRAN IV
5. FORTRAN 77 & FORTRAN 90
6. LISP
7. ALGOL 58
8. ALGOL 60

Zuse's Plankalkül – 1945

- **Never implemented**
  - **Had advanced data types and structures**
  - **Floating point, arrays, records, nesting in records**
  - **No explicit goto statements**
  - **Iteration**
  - **Selection without else part**
  - **Invariants and assertions**
- **Problems Zuse Solved**
  - **Sorting**
  - **Graph connectivity**
  - **Integer and floating point arithmetic**
  - **Expressions with operator precedence**
  - **Chess playing**
- **Terse Notation**

Assemblers and Assembly Language

- **Problem with machine code?**
  - **Poor readability**
  - **Poor modifiability – addition and deletion of instructions was difficult**
  - **Expression coding was tedious**
  - **Machine deficiencies--no indexing or fl. pt. operation available**

- **Speedcoding; 1954; IBM 701, Backus**
  - **Pseudo ops for arithmetic and math functions**
  - **Conditional and unconditional branching**
  - **Auto-increment registers for array access**
  - **Interpreted – Slow!**
  - **After loading the interpreter, only 700 words left for user program**

The First Compiler
Laning and Zierler System - 1953

- **Implemented on the MIT Whirlwind computer**
  - **First "algebraic" compiler system**
  - **Subscripted variables, function calls, expression translation**
  - **Never ported to any other machine**

FORTRAN I – John Backus 1957

- First implemented version of FORTRAN
- (FORTRAN 0 - 1954 - not implemented)
- FORmula TRANslating system
- First compiled high-level language
- Designed for the new IBM 704, which had index registers
- Environment of development:
  - Computers were small and unreliable
  - Applications were scientific
  - No programming methodology or tools
  - Machine efficiency was most important
  - No need for dynamic storage
  - Need for good array handling and counting loops
  - No string handling and decimal arithmetic

Fortran I (continued)

- **Names could have up to six characters**
- **Formatted I/O**
- **User-defined subprograms**
- **No data typing statements**
- **No separate compilation**
- **Compiler released in April 1957, after 18 man months of effort**
- **Programs larger than 400 lines rarely compiled correctly**
- **Code was very fast**
- **Within one year, 50% of the code written for IBM 704 machine was being developed in FORTRAN.**

Fortran I – 1957
Influence of HW on Language Design

- **All statements of FORTRAN I were based on 704's instruction set**
- **3 way branch**
  - **computed if → If (EXPRESSION) L1, L2, L3**
- **Posttest counting loop**
  - **DO L1 I = N, M**

Fortran IV (1960)

- **Fortran IV**
  - **Explicit type declarations**
  - **Logical IF statement**
  - **Passing subprogram as parameters**
  - **ANSI standard in 1966**

Fortran 77 and 90

- **FORTRAN 77 – 1977**
  - **Structured Programming**
  - **Character string handling**
  - **Logical loop control statement**
  - **IF-THEN-ELSE statement**
- **FORTRAN 90 – 1990**
  - **Modules**
  - **Dynamic arrays**
  - **Pointers**
  - **Recursion**
  - **`CASE` statement**
  - **Parameter type checking**

Functional Programming
LISP – McCarthy 1959

- LIST Processing language
- AI research lab at MIT
  - Linguistic, psychology, and mathematics
- Needed a language that:
  - Process data in dynamically growing lists (rather than arrays)
  - Symbolic computation (rather than numeric)
  - Only two data types: atoms and lists
  - Syntax is based on lambda calculus
- Pioneered functional programming
  - No need for assignment
  - Control via recursion and conditional expressions
- It is still the dominant language for AI
- COMMON LISP and Scheme are contemporary dialects of LISP
- ML, Miranda, and Haskell are related languages

ALGOL 58 – 1958

- **ALGOrithmic Language**
- **Search for a "Universal Language"**
- **Environment of development:**
  - **FORTRAN had (barely) arrived for IBM 70x and was owned by IBM**
  - **Many other languages were being developed, all for specific machines**
  - **No portable language; all were machine-dependent**
  - **No universal language for communicating algorithms**

- **Goals of the language:**
  - **Close to mathematical notation**
  - **Good for describing algorithms**
  - **Machine independent, algorithmic language for use on all kinds of computers.**

Algol 58 (Continued)

- **Language Features:**
  - **Concept of type was formalized**
  - **Names could have any length**
  - **Arrays could have any number of subscripts**
  - **Lower bound could be defined**
  - **Parameters were separated by mode (in & out)**
  - **Subscripts were placed in brackets**
  - **Compound statements (begin ... end)**
  - **Semicolon as a statement separator**
  - **Assignment operator was :=**
  - **‘if’ had an ‘else-if’ clause**

- **Never implemented**
- **IBM was initially enthusiastic but vested interest in FORTRAN resulted in taking back all support by mid-1959**

Algol 60 - 1960

- **New Features*:***
  - ***Block structure (local scope)***
  - ***Two parameter passing methods – value and name***
  - ***Subprogram recursion***
  - ***Stack-dynamic arrays – run time size definition and space allocation***
  - ***No I/O***
  - ***Syntax was defined in BNF***

Algol 60 (Continued)

- **Successes:**
  - **It was the standard way to publish algorithms for over 20 years**
  - **All subsequent imperative languages are based on it**
  - **First machine-independent language**
  - **First language whose syntax was formally defined in BNF**
  - **Had impact on hardware design**
- **Failure:**
  - **Never widely used, especially in U.S.**
  - **Reasons:**
    - **No I/O and the character set made programs non-portable**
    - **Too flexible, so hard to understand and implement**
    - **IBM's interest in FORTRAN**
    - **Formal syntax description – at that time BNF was considered strange and complicated!**

Summary

- ✓ Zuse's Plankalkul – 1945
- ✓ Assemblers and Assembly Languages
- ✓ FORTRAN I
- ✓ FORTRAN IV
- ✓ FORTRAN 77 & FORTRAN 90
- ✓ LISP
- ✓ ALGOL 58
- ✓ ALGOL 60