ASSEMBLY LANGUAGE

A.I.Titenko – Sumy State University group IN-02
S.H. Zolotova E L Adviser

An assembly language is a low-level programming language for computers, microprocessors, microcontrollers, and other programmable devices in which each statement corresponds to a single machine language instruction. An assembly language is specific to a certain computer architecture, in contrast to most high-level programming languages, which may be more portable.

Assembly language allows the programmer to use symbolic representation for machine operation codes (usually called mnemonics), memory locations, registers and other parts of an instruction.

A utility program called an assembler is used to translate assembly language statements into the target computer's machine code.

In normal professional usage, the term assembler is used to refer both to an assembly language, and to software which assembles an assembly-language program. Thus: "CP/CMS was written in S/360 assembler" as well as "ASM-H was a widely-used S/370 assembler."

Many assemblers offer additional mechanisms to facilitate program development, control the assembly process, and aid debugging. Assemblers often include a macro facility (described below), and are called macro assemblers.

An assembler creates object code by translating assembly instruction mnemonics into opcodes, and by resolving symbolic names for memory locations and other entities. The use of symbolic references is a key feature of assemblers, saving tedious calculations and manual address updates after program modifications. Most assemblers also include macro facilities for performing textual substitution—e.g., to generate common short sequences of instructions as inline, instead of called subroutines.

Assembly languages date to the introduction of the stored-program computer. The EDSAC computer (1949) had an assembler called initial orders featuring one-letter mnemonics. Nathaniel Rochester wrote an assembler for an IBM 701 (1954). SOAP (Symbolic Optimal Assembly Program) (1955) was an assembly language for the IBM 650 computer written by Stan Poley.

Assembly languages eliminated much of the error-prone and time-consuming first-generation programming needed with the earliest computers, freeing programmers from tedium such as remembering numeric codes and calculating addresses. They were once widely used for all sorts of programming. However, by the 1980s (1990s on microcomputers), their use had largely been supplanted by high-level languages, in the search for improved programming productivity. Today assembly language is still used for direct hardware manipulation, access to specialized processor instructions, or to address critical performance issues. Typical uses are device drivers, low-level embedded systems, and real-time systems.

Historically, a large number of programs have been written entirely in assembly language. Operating systems were entirely written in assembly language until the introduction of the Burroughs MCP (1961), which was written in ESPOL, an Algol dialect. Many commercial applications were written in assembly language as well, including a large amount of the IBM mainframe software written by large corporations. COBOL and FORTRAN eventually displaced much of this work, although a number of large organizations retained assembly-language application infrastructures well into the 90s.