

# The First Compiler: Correcting the Historical Record

A persistent myth in computing history credits Grace Hopper with inventing the first compiler via her A-0 system in 1952. While Hopper's contributions to programming languages are undeniable, the claim that A-0 was a compiler in the modern sense requires significant qualification.

The A-0 system was in practice a linker and loader: it retrieved pre-written subroutines from a library and assembled them into an executable program. It did not translate human-readable source code into machine instructions, which is the defining characteristic of a compiler as the term is used today.

The first true compiler was the IBM FORTRAN compiler, completed in April 1957 by John Backus and his team of thirteen engineers at IBM. FORTRAN (Formula Translation) accepted algebraic expressions as input and produced efficient machine code - tasks the A-0 system never attempted.

The confusion is compounded by Hopper's own later accounts, which retrospectively described A-0 as a compiler to emphasise its conceptual novelty. The IEEE Annals of the History of Computing (Vol. 25, No. 1, 2003) concludes that FORTRAN deserves priority for the compiler title.

None of this diminishes Hopper's genuine achievements. Her FLOW-MATIC language (1955) directly influenced COBOL, and her advocacy for human-readable programming languages transformed the industry. But historical accuracy requires distinguishing between a subroutine library tool and a language compiler.