

3 Programming Manual

MATH-MATIC

MATH-MATIC is the marketing name for the AT-3 (Algebraic Translator 3) compiler, an early programming language for the UNIVAC I and UNIVAC II. MATH-MATIC

MATH-MATIC is the marketing name for the AT-3 (Algebraic Translator 3) compiler, an early programming language for the UNIVAC I and UNIVAC II.

MATH-MATIC was written beginning around 1955 by a team led by Charles Katz under the direction of Grace Hopper. A preliminary manual was produced in 1957 and a final manual the following year.

Syntactically, MATH-MATIC was similar to Univac's contemporaneous business-oriented language, FLOW-MATIC, differing in providing algebraic-style expressions and floating-point arithmetic, and arrays rather than record structures.

Glob (programming)

written in the B programming language. It was the first piece of mainline Unix software to be developed in a high-level programming language. Later, this

glob() () is a libc function for globbing, which is the archetypal use of pattern matching against the names in a filesystem directory such that a name pattern is expanded into a list of names matching that pattern. Although globbing may now refer to glob()-style pattern matching of any string, not just expansion into a list of filesystem names, the original meaning of the term is still widespread.

The glob() function and the underlying gmatch() function originated at Bell Labs in the early 1970s alongside the original AT&T UNIX itself and had a formative influence on the syntax of UNIX command line utilities and therefore also on the present-day reimplementations thereof.

In their original form, glob() and gmatch() derived from code used in Bell Labs in-house utilities that developed alongside the original Unix in the early 1970s. Among those utilities were also two command line tools called glob and find; each could be used to pass a list of matching filenames to other command line tools, and they shared the backend code subsequently formalized as glob() and gmatch(). Shell-statement-level globbing by default became commonplace following the "builtin"-integration of globbing-functionality into the 7th edition of the Unix shell in 1978. The Unix shell's -f option to disable globbing — i.e. revert to literal "file" mode — appeared in the same version.

The glob pattern quantifiers now standardized by POSIX.2 (IEEE Std 1003.2) fall into two groups, and can be applied to any character sequence ("string"), not just to directory entries.

"Metacharacters" (also called "Wildcards"):

? (not in brackets) matches any character exactly once.

* (not in brackets) matches a string of zero or more characters.

"Ranges/sets":

[...], where the first character within the brackets is not '!', matches any single character among the characters specified in the brackets. If the first character within brackets is '!', then the [!...] matches any single character

that is not among the characters specified in the brackets.

The characters in the brackets may be a list ([abc]) or a range ([a-c]) or denote a character class (like [[:space:]] where the inner brackets are part of the classname). POSIX does not mandate multi-range ([a-c0-3]) support, which derive originally from regular expressions.

As reimplementations of Bell Labs' UNIX proliferated, so did reimplementations of its Bell Labs' libc and shell, and with them glob() and globbing. Today, glob() and globbing are standardized by the POSIX.2 specification and are integral part of every Unix-like libc ecosystem and shell, including AT&T Bourne shell-compatible Korn shell (ksh), Z shell (zsh), Almquist shell (ash) and its derivatives and reimplementations such as busybox, toybox, GNU bash, Debian dash.

ARITH-MATIC

extension of Grace Hopper's A-2 programming language, developed around 1955. ARITH-MATIC was originally known as A-3, but was renamed by the marketing

ARITH-MATIC is an extension of Grace Hopper's A-2 programming language, developed around 1955. ARITH-MATIC was originally known as A-3, but was renamed by the marketing department of Remington Rand UNIVAC.

Comparison of multi-paradigm programming languages

Programming languages can be grouped by the number and types of paradigms supported. A concise reference for the programming paradigms listed in this article

Programming languages can be grouped by the number and types of paradigms supported.

Occam (programming language)

platforms are available. The most widely known version is occam 2; its programming manual was written by Steven Ericsson-Zenith and others at Inmos. In the

occam is a programming language which is concurrent and builds on the communicating sequential processes (CSP) process algebra, and shares many of its features. It is named after philosopher William of Ockham after whom Occam's razor is named.

Occam is an imperative procedural language (such as Pascal). It was developed by David May and others at Inmos (trademark INMOS), advised by Tony Hoare, as the native programming language for their transputer microprocessors, but implementations for other platforms are available. The most widely known version is occam 2; its programming manual was written by Steven Ericsson-Zenith and others at Inmos.

The C Programming Language

The C Programming Language (sometimes termed K&R, after its authors' initials) is a computer programming book written by Brian Kernighan and Dennis Ritchie

The C Programming Language (sometimes termed K&R, after its authors' initials) is a computer programming book written by Brian Kernighan and Dennis Ritchie, the latter of whom originally designed and implemented the C programming language, as well as co-designed the Unix operating system with which development of the language was closely intertwined. The book was central to the development and popularization of C and is still widely read and used today. Because the book was co-authored by the original language designer, and because the first edition of the book served for many years as the de facto standard for the language, the book was regarded by many to be the authoritative reference on C.

Ada (programming language)

numerical, financial, and object-oriented programming (OOP). Features of Ada include: strong typing, modular programming mechanisms (packages), run-time checking

Ada is a structured, statically typed, imperative, and object-oriented high-level programming language, inspired by Pascal and other languages. It has built-in language support for design by contract (DbC), extremely strong typing, explicit concurrency, tasks, synchronous message passing, protected objects, and non-determinism. Ada improves code safety and maintainability by using the compiler to find errors in favor of runtime errors. Ada is an international technical standard, jointly defined by the International Organization for Standardization (ISO), and the International Electrotechnical Commission (IEC). As of May 2023, the standard, ISO/IEC 8652:2023, is called Ada 2022 informally.

Ada was originally designed by a team led by French computer scientist Jean Ichbiah of Honeywell under contract to the United States Department of Defense (DoD) from 1977 to 1983 to supersede over 450 programming languages then used by the DoD. Ada was named after Ada Lovelace (1815–1852), who has been credited as the first computer programmer.

Man page

(short for manual page) is a form of software documentation found on Unix and Unix-like operating systems. Topics covered include programs, system libraries

A man page (short for manual page) is a form of software documentation found on Unix and Unix-like operating systems. Topics covered include programs, system libraries, system calls, and sometimes local system details. The local host administrators can create and install manual pages associated with the specific host. A manual end user may invoke a documentation page by issuing the man command followed by the name of the item for which they want the documentation. These manual pages are typically requested by end users, programmers and administrators doing real time work but can also be formatted for printing.

By default, man typically uses a formatting program such as nroff with a macro package or mandoc, and also a terminal pager program such as more or less to display its output on the user's screen.

Man pages are often referred to as an online form of software documentation, even though the man command does not require internet access. The environment variable MANPATH often specifies a list of directory paths to search for the various documentation pages. Manual pages date back to the times when printed documentation was the norm.

Mojo (programming language)

opensourceforu.com/2024/04/programming-languages-for-ai-applications-and-why-mojo-is-among-the-best/ "Mojo programming manual"; docs.modular.com. Modular

Mojo is a programming language in the Python family that is currently under development. It is available both in browsers via Jupyter notebooks, and locally on Linux and macOS. Mojo aims to combine the usability of a high-level programming language, specifically Python, with the performance of a system programming language such as C++, Rust, and Zig. As of February 2025, the Mojo compiler is closed source with an open source standard library. Modular, the company behind Mojo, has stated an intent to eventually open source the Mojo language, as it matures.

Mojo builds on the Multi-Level Intermediate Representation (MLIR) compiler software framework, instead of directly on the lower level LLVM compiler framework like many languages such as Julia, Swift, C++, and Rust. MLIR is a newer compiler framework that allows Mojo to exploit higher level compiler passes unavailable in LLVM alone, and allows Mojo to compile down and target more than only central processing

units (CPUs), including producing code that can run on graphics processing units (GPUs), Tensor Processing Units (TPUs), application-specific integrated circuits (ASICs) and other accelerators. It can also often more effectively use certain types of CPU optimizations directly, like single instruction, multiple data (SIMD) with minor intervention by a developer, as occurs in many other languages. According to Jeremy Howard of fast.ai, Mojo can be seen as "syntax sugar for MLIR" and for that reason Mojo is well optimized for applications like artificial intelligence (AI).

Fortran

programming, array programming, modular programming, generic programming (Fortran 90), parallel computing (Fortran 95), object-oriented programming (Fortran

Fortran (; formerly FORTRAN) is a third-generation, compiled, imperative programming language that is especially suited to numeric computation and scientific computing.

Fortran was originally developed by IBM with a reference manual being released in 1956; however, the first compilers only began to produce accurate code two years later. Fortran computer programs have been written to support scientific and engineering applications, such as numerical weather prediction, finite element analysis, computational fluid dynamics, plasma physics, geophysics, computational physics, crystallography and computational chemistry. It is a popular language for high-performance computing and is used for programs that benchmark and rank the world's fastest supercomputers.

Fortran has evolved through numerous versions and dialects. In 1966, the American National Standards Institute (ANSI) developed a standard for Fortran to limit proliferation of compilers using slightly different syntax. Successive versions have added support for a character data type (Fortran 77), structured programming, array programming, modular programming, generic programming (Fortran 90), parallel computing (Fortran 95), object-oriented programming (Fortran 2003), and concurrent programming (Fortran 2008).

Since April 2024, Fortran has ranked among the top ten languages in the TIOBE index, a measure of the popularity of programming languages.

[https://heritagefarmmuseum.com/-](https://heritagefarmmuseum.com/-90269058/lcompensatef/torganizeh/kpurchasea/citroen+berlingo+1996+2008+petrol+diesel+repair+srv+manual.pdf)

[90269058/lcompensatef/torganizeh/kpurchasea/citroen+berlingo+1996+2008+petrol+diesel+repair+srv+manual.pdf](https://heritagefarmmuseum.com/_82004814/gcirculatec/pcontinuem/bpurchasel/campbell+biologia+primo+biennio)

https://heritagefarmmuseum.com/_82004814/gcirculatec/pcontinuem/bpurchasel/campbell+biologia+primo+biennio

[https://heritagefarmmuseum.com/\\$69415697/mcompensatex/jdescribeu/lreinforcee/solution+manual+of+books.pdf](https://heritagefarmmuseum.com/$69415697/mcompensatex/jdescribeu/lreinforcee/solution+manual+of+books.pdf)

[https://heritagefarmmuseum.com/\\$58064630/hcirculatef/bperceiveq/restimatec/empire+of+liberty+a+history+the+ea](https://heritagefarmmuseum.com/$58064630/hcirculatef/bperceiveq/restimatec/empire+of+liberty+a+history+the+ea)

<https://heritagefarmmuseum.com/^38123283/dschedulem/zperceiver/bcommissionn/manual+of+ocular+diagnosis+a>

https://heritagefarmmuseum.com/_36310035/zscheduleg/kfacilitater/xreinforceb/whirlpool+cabrio+washer+wtw564

[https://heritagefarmmuseum.com/-](https://heritagefarmmuseum.com/-16895229/ccirculatev/sparticipatea/nanticipatek/otter+creek+mastering+math+fact+families.pdf)

[16895229/ccirculatev/sparticipatea/nanticipatek/otter+creek+mastering+math+fact+families.pdf](https://heritagefarmmuseum.com/-16895229/ccirculatev/sparticipatea/nanticipatek/otter+creek+mastering+math+fact+families.pdf)

<https://heritagefarmmuseum.com/@82007610/jregulatek/edscribeh/wdiscoverp/ptc+dental+ana.pdf>

<https://heritagefarmmuseum.com/+82597729/bconvincev/zdescribea/cestatimet/2000+kia+spectra+gs+owners+manu>

<https://heritagefarmmuseum.com/~38735351/swithdrawz/nperceiveu/jcommissionm/the+education+of+a+gardener+>