Documents for UNIX

VOLUME 1

T. A. Dolotta
S. B. Olsson
A. G. Petruccelli
Editors

January 1981

Not for use or disclosure outside the Bell System except under written agreement

Laboratory 4517
Bell Telephone Laboratories, Incorporated
Murray Hill, NJ 07974

Copyright © 1981 Bell Telephone Laboratories, Inc.

UNIX is a trademark of Bell Telephone Laboratories, Inc.

These documents were set on an AUTOLOGIC, Inc. APS-5 phototypesetter driven by the TROFF formatter operating under the UNIX system.

ANNOTATED TABLE OF CONTENTS

NOTES: All the documents included here are supplements to the ***UNIX User's Manual*** (see G.1 below); the reader's attention is also drawn to documents G.2, G.3, and G.4.

Each document listed in Sections A through F below applies to UNIX Release 4.0, unless otherwise indicated after its title.

The number of pages in each document is given after the name(s) of its author(s).

VOLUME 1

A. OVERVIEWS

1. Overview and Synopsis
   1. ***UNIX—Overview and Synopsis of Facilities***
      T. A. Dolotta, R. C. Haight, and A. G. Petruccelli (p. 17)
      A concise outline of the features and facilities of UNIX.
2. The UNIX Time-Sharing System
   1. ***The UNIX Time-Sharing System***
      D. M. Ritchie and K. Thompson (p. 16)
      The original, prize-winning UNIX paper, reprinted from G.5 below.

B. GETTING STARTED

1. Road Map
   1. ***UNIX Documentation Road Map***
      G. A. Snyder and J. R. Mashey (p. 8)
      A structured list of UNIX documents and information sources.
      ☞ *A local section should be added to this document at each installation.*
2. Editors
   1. ***A Tutorial Introduction to the UNIX Text Editor***
      B. W. Kernighan (p. 11)
      An easy way to get started with the text editor.
   2. ***Advanced Editing on UNIX***
      B. W. Kernighan (p. 16)
      A guide to the more advanced features of the text editor.
   3. ***SED—A Non-Interactive Text Editor***
      L. E. McMahon (p. 10)
      A variant of the text editor for stream editing.
3. UNIX for Beginners
   1. ***UNIX for Beginners (Second Edition)***
      B. W. Kernighan (p. 13)
      An introduction to some of the basic uses of UNIX.
4. Shell
   1. ***UNIX Shell Tutorial***
      G. A. Snyder and J. R. Mashey (p. 36+ii)
      An introduction to the various uses and facilities of the UNIX command language interpreter, with many examples.
   2. ***An Introduction to the UNIX Shell***
      S. R. Bourne (p. 24)
      Description of the UNIX command language interpreter.

C. DOCUMENT PREPARATION

1. NROFF/TROFF
    1. ***A TROFF Tutorial***
        B. W. Kernighan (p. 14)
        A beginner's guide to phototypesetting with TROFF.
    2. ***NROFF/TROFF User's Manual***
        J. F. Ossanna (p. 37)
        Reference manual for the UNIX text formatters.
2. Macros for NROFF/TROFF
    1. ***MM—Memorandum Macros***
        D. W. Smith and J. R. Mashey (p. 69+iv)
        Reference manual for MM, the standard BTL text-formatting macros.
    2. ***Typing Documents with MM***
        D. W. Smith and E. M. Piskorik (p. 16)
        A fold-out card that summarizes the MM macros; furnished separately.
    3. ***A Macro Package for View Graphs and Slides***
        T. A. Dolotta and D. W. Smith (p. 23)
        A guide to making visual aids with TROFF.
3. TBL and EQN
    1. ***TBL—A Program to Format Tables***
        M. E. Lesk (p. 18)
        An NROFF/TROFF preprocessor that permits easy formatting of tabular matter.
    2. ***Typesetting Mathematics—User's Guide (Second Edition)***
        B. W. Kernighan and L. L. Cherry (p. 11)
        Manual for the EQN and NEQN preprocessors for TROFF and NROFF, respectively; these preprocessors allow one to specify, in an easy-to-learn language, how to typeset complex mathematical expressions.
    3. ***A System for Typesetting Mathematics***
        B. W. Kernighan and L. L. Cherry (p. 8)
        A revision of the original EQN paper (*CACM* **18**, March 1975), describing the principles behind the design of its input language and internal structure.

D. PROGRAMMING

1. C and LINT
    1. ***The C Programming Language—Reference Manual***
        D. M. Ritchie (p. 31)
        Official statement of the syntax and semantics of C; supplemented by G.9 below.
    2. ***A Guide to the C Library for UNIX Users***
        C. D. Perez (p. 20)
        An explanation of how to use the C library.
    3. ***LINT, a C Program Checker***
        S. C. Johnson (p. 11)
        A program that checks C code for syntax errors, type violations, portability problems, and a variety of potential errors.

2. FORTRAN, RATFOR, and EFL
    1. ***A Portable FORTRAN 77 Compiler***
        S. I. Feldman and P. J. Weinberger (p. 19)
        The FORTRAN 77 language and its interfaces with the operating system.
    2. ***RATFOR—A Preprocessor for a Rational FORTRAN***
        B. W. Kernighan (p. 12)
        A preprocessor that endows FORTRAN with C-like control structures and input format.
    3. ***The Programming Language EFL***
        S. I. Feldman (p. 36)
        A general-purpose computer language intended to encourage portable programming, while making use of the good features and facilities of FORTRAN.

3. UNIX Programming
    1. ***UNIX Programming (Second Edition)***
        B. W. Kernighan and D. M. Ritchie (p. 22)
        A guide to writing programs that interface to the UNIX operating system, either directly or through the Standard I/O Library.

4. MAKE
    1. ***MAKE—A Program for Maintaining Computer Programs***
        S. I. Feldman (p. 9)
        A tool for automating the recompilation of large programs.
    2. ***An Augmented Version of MAKE***
        E. G. Bradford (p. 16)
        A discussion of how to use MAKE to its fullest advantage.

5. Debuggers
    1. ***SDB—A Symbolic Debugger***
        H. P. Katseff (p. 9)
        A debugger that allows one to examine the "core image" of an aborted program.
    2. ***A Tutorial Introduction to ADB***
        J. F. Maranzano and S. R. Bourne (p. 27)
        A guide to debugging crashed systems and programs; ADB is used mostly by system programmers.

VOLUME 2

E. SUPPORTING TOOLS AND LANGUAGES

1. LEX and YACC
   1. ***LEX—A Lexical Analyzer Generator***
      M. E. Lesk and E. Schmidt (p. 19)
      A program that generates recognizers of sets of regular expressions; each regular expression can be followed by arbitrary C code that is executed when the regular expression is found.
   2. ***YACC—Yet Another Compiler-Compiler***
      S. C. Johnson (p. 33)
      A converter from a BNF specification of a language and semantic actions written in C into a compiler for that language.
2. M4 Macro Processor
   1. ***The M4 Macro Processor***
      B. W. Kernighan and D. M. Ritchie (p. 6)
      A macro processor, also useful as a front end for languages such as C and RATFOR.
3. AWK
   1. ***AWK—A Pattern Scanning and Processing Language (Second Edition)***
      A. V. Aho, B. W. Kernighan, and P. J. Weinberger (p. 8)
      A language that makes it easy to specify many data selection and transformation operations.
4. SCCS
   1. ***Source Code Control System User's Guide***
      L. E. Bonanni and C. A. Salemi (p. 27)
      A package for controlling access and changes to (possibly multiple versions of) source programs and text files.
   2. ***Function and Use of an SCCS Interface Program***
      L. E. Bonanni and A. Guyton (p. 3)
      A discussion of how to control concurrent updates to SCCS files.
5. Calculators
   1. ***BC—An Arbitrary Precision Desk-Calculator Language***
      L. L. Cherry and R. Morris (p. 14)
      A front end for DC (see below) that provides infix notation, flow control, and built-in functions.
   2. ***DC—An Interactive Desk Calculator***
      R. Morris and L. L. Cherry (p. 8)
      An interactive desk calculator program that implements arbitrary-precision integer arithmetic.
6. Graphics
   1. ***UNIX Graphics Overview***
      A. R. Feuer (p. 7)
      An introduction to the UNIX graphics facility.
   2. ***A Tutorial Introduction to the Graphics Editor***
      A. R. Feuer (p. 17)
      A guide to making graphs, drawings, and pictures on Tektronix series 4010 terminals.

3. ***STAT—A Tool for Analyzing Data***
   A. R. Feuer and A. Guyton (p. 20)
   A collection of programs that can be interconnected via the shell to analyze statistical data and display the results in graphical form.
4. ***Administrative Information for the UNIX Graphics Package***
   R. L. Chen, D. E. Pinkston, and A. Guyton (p. 6)
   A reference guide for administrators of UNIX graphics facilities.

7. RJE and Networking
   1. ***UNIX Remote Job Entry User's Guide***
      A. L. Sabsevitz and K. A. Kelleman (p. 7)
      A guide to submitting jobs to an IBM system via the UNIX Remote Job Entry (RJE) facility.
   2. ***UNIX Remote Job Entry Administrator's Guide***
      M. J. Fitton (p. 20)
      A guide to setting up RJE on both UNIX and IBM systems, and to trouble-shooting when things go wrong.
   3. ***Release 1.0 of the UNIX Virtual Protocol Machine*** (UNIX 3.0)
      P. F. Long and C. Mee, III (p. 7)
      A description of the first version of VPM; good background reading.
   4. ***Release 2.0 of the UNIX Virtual Protocol Machine*** (UNIX 3.0)
      P. F. Long and C. Mee, III (p. 20)
      A newer release of VPM; supports bit-oriented, full-duplex protocols.

8. UUCP
   1. ***A Dial-up Network of UNIX Systems***
      D. A. Nowitz and M. E. Lesk (p. 10)
      Description of the design of a dial-up UNIX network called UUCP and used for transmission and distribution of programs and text files.
   2. ***UUCP Implementation Description***
      D. A. Nowitz (p. 15)
      A detailed description of UUCP for use by administrators of UNIX systems.

9. Printer Spooler
   1. ***The Implementation of the LP Spooling System***
      J. R. Kliegman (p. 13)
      Explanation of how the LP spooler works and how it can be used as a general-purpose spooler, as well as a line-printer spooler.
   2. ***LP Administrator's Guide***
      J. R. Kliegman (p. 12)
      A guide for those who oversee the operation of LP spoolers.

F. ADMINISTRATION, MAINTENANCE, AND IMPLEMENTATION

1. Operations and FSCK
   1. ***UNIX Operations Manual***
      A. G. Petruccelli (p. 24+ii)
      Duties of a UNIX operator.
   2. ***FSCK—The UNIX File System Check Program***
      T. J. Kowalski (p. 20)
      A guide to checking and fixing UNIX file systems.

2. Accounting and System Activity
    1. ***The UNIX Accounting System***
        H. S. McCreary and A. G. Petruccelli (p. 19)
        A guide to the use and management of the UNIX accounting system.
    2. ***The UNIX System Activity Package***
        T. W. Pao (p. 8)
        A package that reports on processor utilization, terminal activity, disk and tape I/O, swapping, system calls, etc.

3. Stand-Alone I/O
    1. ***A Stand-Alone Input/Output Library***
        S. R. Eisen (p. 11)
        A guide to the stand-alone library and the stand-alone shell (SASH).

4. ETP
    1. ***The UNIX Equipment Test Package: Operational Procedures*** (UNIX 3.0)
        A. L. Chellis and T. J. Kowalski (p. 24)
        The Equipment Test Package, a collection of UNIX hardware exercisers.

5. UNIX Internals
    1. ***UNIX Implementation***
        K. Thompson (p. 10)
        An explanation of how UNIX works; reprinted from G.5 below.
    2. ***The UNIX I/O System***
        D. M. Ritchie (p. 7)
        Guide for writers of UNIX device drivers.
    3. ***UNIX on the PDP-11/23 and 11/34 Computers*** (UNIX 3.0)
        T. J. Kowalski (p. 7)
        Description of what had to be done to UNIX to make it run on the PDP-11/23 and the PDP-11/34.
    4. ***UNIX Assembler Reference Manual***
        D. M. Ritchie (p. 12)
        Describes the UNIX PDP-11 assembler; a tool of last resort.

6. C Internals
    1. ***A Tour Through the Portable C Compiler***
        S. C. Johnson (p. 25)
        A description of how the portable C compiler works.
    2. ***A Tour Through the UNIX C Compiler***
        D. M. Ritchie (p. 15)
        A description of how the PDP-11 C compiler works.

7. Security
    1. ***On the Security of UNIX***
        D. M. Ritchie (p. 3)
        Hints on how to break UNIX and how to prevent it.
    2. ***Password Security—A Case History***
        R. Morris and K. Thompson (p. 6)
        The story of how the bad guys used to be able to break the password algorithm and why they can't now, at least not so easily.

G. RECOMMENDED READING (*not included*)

1. ***UNIX User's Manual***—Release 3.0
   T. A. Dolotta, S. B. Olsson, and A. G. Petruccelli (eds.)
   Bell Laboratories (June 1980).
   The basic document for every UNIX user.
2. ***UNIX Reference Guide***
   J. C. White (compiler) and P. V. Guidi (ed.)
   Bell Laboratories (April 1981).
   A pocket-size summary of UNIX commands, macro packages, etc.
3. ***Setting up UNIX***
   R. C. Haight, M. J. Petrella, and L. A. Wehr
   Bell Laboratories.
   Procedures for installing UNIX; must reading for anyone who wants to configure and/or generate a UNIX system. (Because this document changes with each release of UNIX, it is not included here; it is distributed with each copy of the UNIX system itself.)
4. ***Administrative Advice for UNIX***
   R. C. Haight
   Bell Laboratories.
   Hints for getting UNIX up, getting it going, and keeping it going, plus some information about hardware; must reading for UNIX system administrators. (This document is distributed just like G.3 above.)
5. ***The Bell System Technical Journal***
   Vol. 57, No. 6, Part 2 (July-August 1978).
   Special issue devoted to UNIX.
6. ***Using a Command Language as the Primary Programming Tool***
   T. A. Dolotta and J. R. Mashey
   In: Beech, D. (ed.), *Command Language Directions* (Proc. Second IFIP Working Conf. on Command Languages). Amsterdam: North Holland (1980), pp. 35-55.
   A discussion of how to get the most out of the UNIX shell.
7. ***The UNIX Programming Environment***
   B. W. Kernighan and J. R. Mashey
   COMPUTER, Vol. 14, No. 4, pp. 12-24 (April 1981); an earlier version of this paper was published in *Software—Practice & Experience*, Vol. 9, No. 1, pp. 1-15 (Jan. 1979).
   A discussion of what's good about UNIX.
8. ***Software Tools***
   B. W. Kernighan and P. J. Plauger
   Reading, MA: Addison-Wesley (1976).
   A textbook for building good software tools similar to those available in UNIX.
9. ***The C Programming Language***
   B. W. Kernighan and D. M. Ritchie
   Englewood Cliffs, NJ: Prentice-Hall (1978).
   The basic book for every C programmer; contains a tutorial and many examples.
10. ***Experiences with the UNIX Time-sharing System***
    J. Lions
    Software—Practice & Experience, Vol. 9, No. 9, pp. 701-709 (September 1979).
    An enjoyable article that tells why they like UNIX in New South Wales.

11. ***The Evolution of the UNIX Time-sharing System***

D. M. Ritchie

Proc. Symposium on Language Design and Programming Methodology, Sydney, Australia (September 1979).

Ten years later, one of the creators of UNIX looks back.

12. ***The Source Code Control System***

M. J. Rochkind

IEEE Trans. Software Eng., Vol. SE-1, No. 4, pp. 364-370 (December 1975).

The motivation for, and the underlying design of, SCCS.