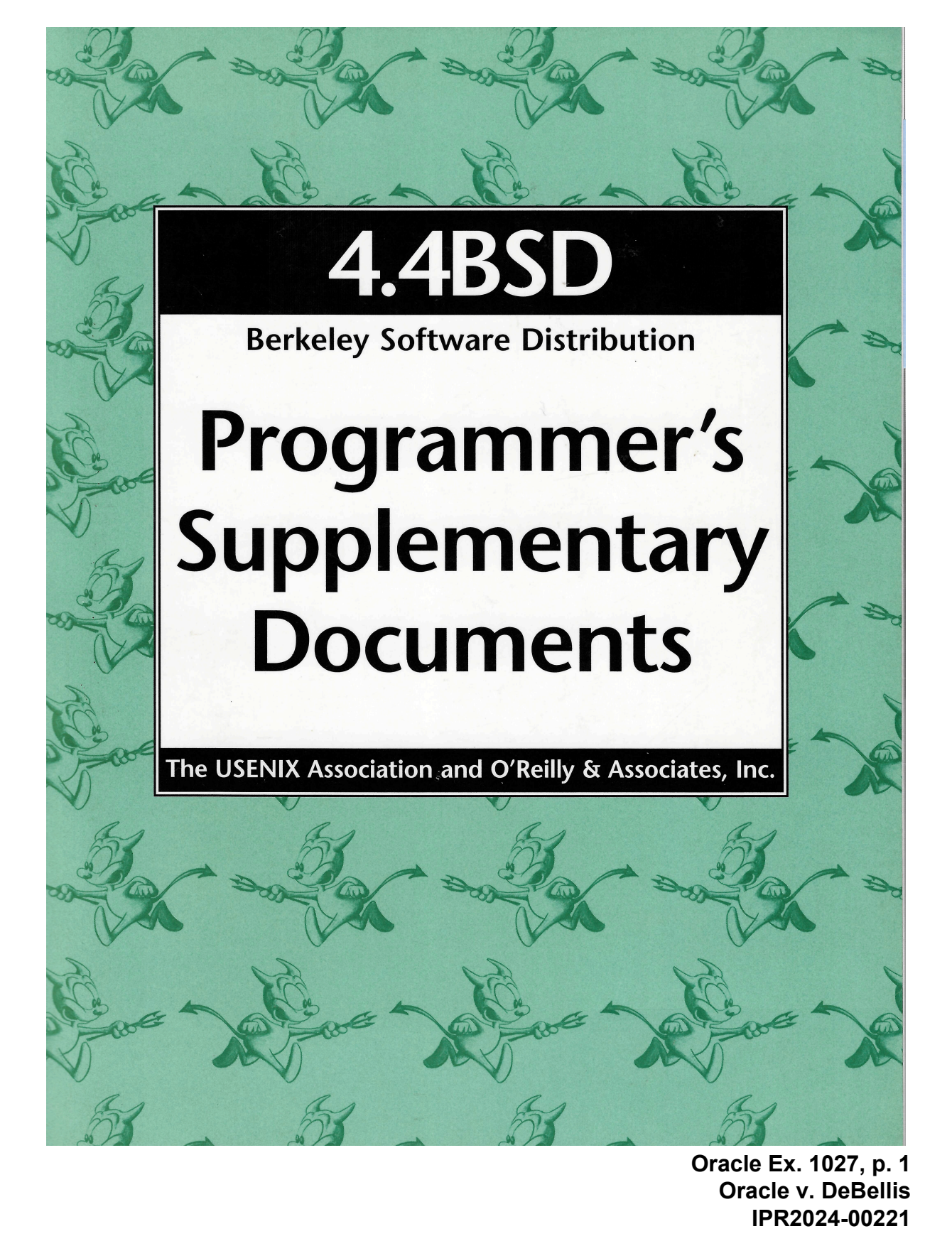




# 4.4BSD

Berkeley Software Distribution

# Programmer's Supplementary Documents

The USENIX Association and O'Reilly & Associates, Inc.

First Printing, 1994

Copyright 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994 The Regents of the University of California. All rights reserved.

Other than the specific manual pages and documents listed below as copyrighted by AT&T, redistribution and use of this manual in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1) Redistributions of this manual must retain the copyright notices on this page, this list of conditions and the following disclaimer.
- 2) Software or documentation that incorporates part of this manual must reproduce the copyright notices on this page, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3) All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors."
- 4) Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

**THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

Documents PSD:1, 2, 3, 4, 6, 11, 15, 16, and 17 are copyright 1979, AT&T Bell Laboratories, Incorporated. Document PSD:8 is a modification of an earlier document that is copyrighted 1979 by AT&T Bell Laboratories, Incorporated. Holders of UNIX<sup>TM</sup>/32V, System III, or System V software licenses are permitted to copy these documents, or any portion of them, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included.

Document PSD:10 is part of the user contributed software and is copyright 1992 by the Free Software Foundation, Inc. Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Document PSD:13 is part of the user contributed software and is copyright 1983 by Walter F. Tichy. Permission to copy the RCS documentation or any portion thereof as necessary for licensed use of the software is granted to licensees of this software, provided this copyright notice is included.

The views and conclusions contained in this manual are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Regents of the University of California.

This book was printed and bound in the United States of America.  
Distributed by O'Reilly & Associates, Inc.



This book is printed on acid-free paper with 50% recycled content, 10-13% post-consumer waste. O'Reilly & Associates is committed to using paper with the highest recycled content available consistent with high quality.

ISBN: 1-56592-079-1

The M4 Macro Processor tabbed 17  
M4 is a macro processor useful in its own right and as a front-end for C, Ratfor, and Cobol.

gprof: a Call Graph Execution Profiler tabbed 18  
A program to show the call graph and execution time of a program. Indispensable aid for improving the running time of almost everything.

### **Programming Libraries**

Screen Updating and Cursor Movement Optimization tabbed 19  
Describes the *curses* package, an aid for writing screen-oriented, terminal-independent programs.

### **General Reference**

An Introductory 4.4BSD Interprocess Communication Tutorial tabbed 20  
How to write programs that use the Interprocess Communication Facilities of 4.4BSD.

An Advanced 4.4BSD Interprocess Communication Tutorial tabbed 21  
The reference document (with some examples) for the Interprocess Communication Facilities of 4.4BSD.

List of Documents inside back cover

`inch();†`

Returns the character at the current position on the given window. This does not make any changes to the window.

`initscr();`

Initialize the screen routines. This must be called before any of the screen routines are used. It initializes the terminal-type data and such, and without it none of the routines can operate. If standard input is not a tty, it sets the specifications to the terminal whose name is pointed to by *Def\_term* (initially "dumb"). If the boolean *My\_term* is non-zero, *Def\_term* is always used. If the system supports the **TIOCGWINSZ** *ioctl*(2) call, it is used to get the number of lines and columns for the terminal, otherwise it is taken from the **termcap** description.

`insch(char c);`

Insert *c* at the current (y, x) co-ordinates. Each character after it shifts to the right, and the last character disappears. This returns ERR if it would cause the screen to scroll illegally.

`insertln();`

Insert a line above the current one. Every line below the current line will be shifted down, and the bottom line will disappear. The current line will become blank, and the current (y, x) co-ordinates will remain unchanged.

`killchar();†`

Returns the line kill character for the terminal, *i.e.*, the character used by the user to erase an entire line from the input.

`leaveok(WINDOW *win, int boolf);†`

Sets the boolean flag for leaving the cursor after the last change. If *boolf* is non-zero, the cursor will be left after the last update on the terminal, and the current (y, x) co-ordinates for *win* will be changed accordingly. If *boolf* is 0 the cursor will be moved to the current (y, x) co-ordinates. This flag (initially 0) retains its value until changed by the user.

`move(int y, int x);`

Change the current (y, x) co-ordinates of the window to (y, x). This returns ERR if it would cause the screen to scroll illegally.

`mvcur(int lasty, int lastx, int newy, int newx);`

Moves the terminal's cursor from (*lasty*, *lastx*) to (*newy*, *newx*) in an approximation of optimal fashion. This routine uses the functions borrowed from *ex* version 2.6. It is possible to use this optimization without the benefit of the screen routines. With the screen routines, this should not be called by the user. `move()` and `refresh()` should be used to move the cursor position, so that the routines know what's going on.

`mvprintw(int y, int x, const char *fmt, ...);`

Equivalent to:

```
move(y, x);
printw(fmt, ...);
```

# 4.4BSD

## Berkeley Software Distribution Programmer's Supplementary Documents

Unlike many other modern operating systems, UNIX was not written by a single monolithic development team. It started out as a research operating system, and its power came from the creativity of hundreds of brilliant programmers solving the problems they came across in their work. After writing a new tool, they'd typically write a reference document (a "man page"), and if the tool was significant, a technical paper describing its background and use.

For years, the technical papers (or "supplementary documents" as they have now come to be called) were the only tutorial documentation for many UNIX programs. Now, some of these papers have been superseded by in-depth books on individual programs.

However, for many programs, the Supplementary Documents remain the single, authoritative source for detailed documentation. This is particularly true in the programming area. In this volume, you'll find useful papers on such tools as the gdb and adb debuggers, source code control systems RCS and SCCS, lex and yacc, and the m4 macro processor. You'll also find such historical documents as Dennis Ritchie and Ken Thompson's classic first paper about UNIX and Kernighan and Ritchie's classic introduction to UNIX programming.

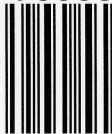
On a more up-to-date note, this book also includes a two part tutorial on interprocess communication (IPC) under 4.4BSD UNIX.

ISBN 1-56592-079-1



9 781565 920798

90000



Printed on Recycled Paper

5 Volume Set: ISBN 1-56592-082-1  
ISBN 1-56592-079-1