ACKNOWLEDGEMENTS

The form of this manual follows that of the *UNIX Programmer's Manual—Seventh Edition, Volume 1* developed by M. D. McIlroy. A large part of the present manual's contents is descended from the *UNIX Programmer's Manual—Sixth Edition* by K. Thompson and D. M. Ritchie (Bell Telephone Laboratories, May 1975) and the *PWB/UNIX User's Manual* by T. A. Dolotta, R. C. Haight, and E. M. Piskorik, eds. (Bell Telephone Laboratories, May 1977). A special credit should also be given to the UNIX support effort of department 3624; their support of UNIX helped make this manual possible. The number of our colleagues who have contributed to UNIX and CB-UNIX software and documentation is, by now, too large to list here, but the usefulness and acceptance of UNIX and of CB-UNIX is a true measure of their collective success.

Columbus, Ohio J.D.D.

INTRODUCTION

This manual describes the features of CB-UNIX. It provides neither a general overview of UNIX (for that, see "The UNIX Time-Sharing System," *BSTJ*, Vol. 57, No. 6, Part 2, pp. 1905-29, by D. M. Ritchie and K. Thompson), nor details of the implementation of the system (see "UNIX Implementation," *BSTJ*, same issue, pp. 1931-46). The manual is organized as follows:

1. Title Page
2. Acknowledgements
3. Introduction
4. How To Get Started
5. Table of Contents (including a permuted index)
6. Manual pages organized into 8 sections

The table of contents is organized by section and alphabetized within each section. The permuted index is derived from the table of contents. In the permuted index (and in general throughout the manual), each manual page name is followed by the manual section to which it refers (e.g. *ed*(1)); this convention is necessary because of the duplication of names between the various sections.

The manual pages are divided into eight sections; each section is further sub-divided into 2 sub-sections. Each section starts off with an introduction to the kinds of things which are documented in the section. Following this introduction, in the sub-section labeled "basic" is the documentation for those things which are considered to be basic to the section and are in the "prime" support class of the CB-UNIX support group. The second sub-section in each section is used to document routines added by the local application group, i.e. not supported by the CB-UNIX support group. The sections of the manual are:

1. Commands and Application Programs:
   - 1. General-Purpose Commands.
   - 1C. Communications Commands.
   - 1G. Graphics Commands.
   - 1M. System Maintenance Commands.
   - 1S. SCCS Commands.
   - 1X. Games.

1. Local Commands and Application Programs.
2. System Calls.
2. Local System Calls.
3. Subroutines:
   - 3C. C and Assembler Library Routines.
   - 3M. Mathematical Library Routines.
   - 3S. Standard I/O Library Routines.
   - 3X. Miscellaneous Routines.

3. Local Subroutines.
4. Device Interfaces and Special Files.
4. Local Device Interfaces and Special Files.
5. File Formats, Tables and Macros.
5. Local File Formats, Tables and Macros.
6. UNIX System Explantions.
6. Local UNIX System Explantions.
7. Kinks and Conventions.
7. Local Kinks and Conventions.
8. Stand-Alone Utilities.
8. Local Stand-Alone Utilities.

Section 1 (*Commands and Application Programs*) describes programs intended to be invoked directly by the user or by command language procedures, in contradistinction to subroutines, which are intended to be called by the user's programs. Commands generally reside in the directory **/bin** (for **bin**ary programs). Some programs also reside in **/usr/bin**, to save space in **/bin**. These directories are searched automatically by the command interpreter called the *shell*. Sub-class 1C contains communication

programs such as *fget*, *dpr*, *cu*, etc. These entries may differ from system to system. Sub-class 1G contains graphics commands that involve graphics output on some device. Some examples of 1G commands are *gex*, *plot*, *graph*, etc. which mostly reside in the directory **/usr/bin**. Sub-class 1M contains system maintenance programs such as *fsck*, *mkfs*, etc., which generally reside in the directory **/etc**. These commands are not intended for use by the ordinary user due to their privileged nature. Sub-class 1S contains Source Code Control System commands such as *admin*, *get*, *delta*, etc. These entries mostly reside in the directory **/usr/bin**. Sub-class 1X are Games such as *ttt*, *wump*, *startrek*, etc., which reside in the directory **/usr/games**.

Section 2 (*System Calls*) describes the entries into the UNIX supervisor, including the Assembler and C language interfaces. In the Assembler, these system calls are invoked by the *sys* operation code, which is a synonym for the *trap* instruction.

Section 3 (*Subroutines*) describes the available subroutines. Their binary versions reside in various system libraries in directories **/lib** and **/usr/lib**. Sub-class 3C contains C and Assembler Library routines, which reside in **/lib/libc.a**. Sub-class 3M contains mathematical routines (*pow*, *log*, *sin*, etc.), which reside in **/lib/libm.a**. Sub-class 3S contains Standard I/O Library routines, also found in **/lib/libc.a**.

Section 4 (*Device Interfaces and Special Files*) discusses the characteristics of each special "file" that actually refers to an input/output device. The names in **Section 4** refer to the name of the device (i.e. RP04, RP03, TU16) rather than the names of the special file itself. Pseudo devices, like multiplexors and shared memory are also documented here.

Section 5 (*File Formats, Tables and Macros*) documents the structure of particular kinds of files; for example, the form of the output of the Assembler and the Loader is given. Excluded are files used by only one command, for example, the assembler's intermediate files. Some of the macro packages available, particularly for the text formatting programs *nroff*(1) and *troff*(1), are documented in **Section 5**.

Section 6 (*UNIX System Explanations*) includes boot procedures, system error messages, changable system parameters, etc.,.

Section 7 (*Kinks and Conventions*) documents proscribed conventions for file naming, C programming, etc.,.

Section 8 (*Stand-alone Utilities*) tells the story of programs which stand apart from UNIX in the sense that they run on the "bare" machine. For instance, *sacopy*, a device to device copy program is documented here.

Each section consists of a number of independent entries of a page or so each. The name of the entry appears in the upper corners of its pages. Entries within each section are alphabetized, with the exception of the introductory entry that begins each section. The page numbers of each entry start at 1. Some entries may describe several routines, commands, etc. In such cases, the entry appears only once, alphabetized under its "major" name.

All entries are based on a common format, not all of whose parts always appear:

The **NAME** part repeats the name of the entry and states (very briefly) its purpose.

The **SYNOPSIS** part summarizes the use of the program being described. A few conventions are used, particularly in Section 1 (*Commands*):

Boldface strings are literals and are to be typed just as they appear.

Italic strings usually represent substitutable argument prototypes and program names found elsewhere in the manual (they are underlined in the typed version of the entries).

Square brackets [] around an argument prototype indicate that the argument is optional. When an argument prototype is given as "name" or "file", it always refers to a *file* name.

Ellipses ... are used to show that the previous argument prototype may be repeated.

A final convention is used by the commands themselves. An argument beginning with a minus −, plus +, or equal sign = is often taken to be some sort of flag argument, even if it appears in a position where a file name could appear. Therefore, it is unwise to have files

whose names begin with −, +, or =.

The DESCRIPTION part discusses the subject at hand.

The EXAMPLE(S) part gives example(s) of usage, where appropriate.

The FILES part gives the file names that are built into the program.

The SEE ALSO part gives pointers to related information.

The DIAGNOSTICS part discusses the diagnostic indications that may be produced. Messages that are intended to be self-explanatory are not listed.

The BUGS part gives known bugs and sometimes deficiencies. Occasionally, the suggested fix is also described.

The ASSEMBLER part summarizes the procedure for invoking system calls from PDP-11 Assembler source programs. This part appears only in Section 2 and in the Introduction to Section 3.

All entries are available on-line via the *man*(1) command.