# UNIX—Overview and Synopsis of Facilities

*T. A. Dolotta*
*R. C. Haight*
*A. G. Petruccelli*

Bell Laboratories
Murray Hill, New Jersey 07974

## OVERVIEW

### 1. UNIX TIME-SHARING SYSTEM

The UNIX† Time-Sharing System is a general-purpose, multi-user, interactive operating system specifically engineered to make the designer's, programmer's, and documenter's computing environment simple, efficient, flexible, and productive. UNIX contains features such as:

- A hierarchical file system.
- A flexible, easy-to-use command language (can be "tailored" to meet specific user needs).
- Ability to execute sequential, asynchronous, and background processes.
- A powerful context editor.
- Very flexible document preparation and text processing systems.
- A high-level programming language conducive to structured programming (C).
- Other languages, including *FORTRAN 77*, *EFL*, and variants of *SNOBOL* and *BASIC*.
- Symbolic debugging systems.
- A variety of system programming tools (i.e., lexical analyzers, compiler-compilers, etc.).
- Sophisticated "desk-calculator" packages.
- Inter-machine communication by both hard-wired and dial-up facilities.
- A system designed to help control changes to source code and files of text (SCCS).
- A graphical plotting package.

Currently, UNIX runs on the Western Electric Co. 3B-20; Digital Equipment Corporation's (DEC) PDP-11/23, /34, /45, /70, VAX-11/780, and VAX-11/750; and IBM System/370 and equivalent. The cost per user-hour of UNIX is significantly lower than that of most other interactive computer systems; UNIX typically runs unattended.

The UNIX file system consists of a highly-uniform set of directories and files arranged in a tree-like hierarchical structure. Some of its features are:

- Simple and consistent naming conventions; names can be absolute, or relative to any directory in the file system hierarchy.
- Mountable and de-mountable file systems and volumes.
- File linking across directories.
- Automatic file space allocation and de-allocation that is invisible to users.
- A complete set of flexible directory and file protection modes, allowing all combinations of *read*, *write*, and *execute* access, independently for the owner of each file or directory, for a group of users (e.g., all members of a project), and for all other users; protection modes can be set dynamically.
- Facilities for creating, accessing, moving, and processing files, directories, or sets of these in a simple, uniform, and natural way.
- Each physical I/O device, from interactive terminals to main memory, is treated like a file, allowing uniform file and device I/O.

† UNIX is a trademark of Bell Laboratories.

## 2. UNIX COMMAND LANGUAGE

Unlike other interactive command languages, the UNIX shell is a full programming language. The shell provides variables, conditional and iterative constructs, and a user environment that can be tailored to an individual's or group's needs. Any user can to create new commands simply by writing shell scripts.

## 3. DOCUMENT PREPARATION AND TEXT PROCESSING

In a software development project of any appreciable size, the production of usable, accurate documentation may well consume more effort than the production of the software itself. Several years of experience with many projects that use UNIX have shown that document preparation should not be separated from software development, and that the combination of a flexible operating system, a powerful command language, and good text processing facilities permit quick and convenient production of many kinds of documentation that might be otherwise unobtainable, impractical, or very expensive.

In UNIX, one also obtains a very useful "word processing" system—an editing system, text formatting systems, a typesetting system, and spelling and typographical error-detection facilities. The document preparation and text processing facilities of UNIX include commands that automatically control pagination, style of paragraphs, line justification, hyphenation, multi-column pages, footnote placement, generation of marginal revision bars, generation of tables of contents, etc., for specialized documents such as program run books, or for general documents such as letters, memoranda, legal briefs, etc. There are also excellent facilities for formatting and typesetting complex tables and equations. This document was produced in its entirety by these facilities.

## 4. REMOTE JOB ENTRY

The RJE facility provides for the submission and retrieval of jobs from an IBM host system (e.g., a System/360 or System/370 computer using HASP, ASP, JES2, or JES3). To the host system, RJE appears to be a System/360 work station.

At the request of a UNIX user, RJE gathers the job control statements and source code from files created and stored on UNIX, sends them to the host IBM system and, subsequently, retrieves from the host the resulting output, either placing it in a convenient UNIX file for later perusal, or using that output as the standard input to a specified shell procedure. Automatic notification of the output's arrival is also available.

## 5. SOURCE CODE CONTROL SYSTEM

The UNIX Source Code Control System (SCCS) is an integrated set of commands designed to help software development projects control changes to source code and to files of text (e.g., manuals). It provides facilities for storing, updating, and retrieving, by version number or date, all versions of source code modules or of documents, and for recording who made each software change, when it was made, and why. SCCS is designed to solve most of the source code and documentation control problems that software development projects encounter when customer support, system testing, and development are all proceeding simultaneously. Some of the main characteristics of SCCS are:

- The exact source code or text, as it existed at any point of development or maintenance, can be recreated at any later time.
- All releases and versions of a source code module or document are stored together, so the common code or text is stored only once.
- Releases in production or system test status can be protected from unauthorized changes.
- Enough identifying information can be automatically inserted into source code modules to enable one to identify the exact version and release of any such module, given only the corresponding load module or its memory dump.

## SOFTWARE, FACILITIES, AND DOCUMENTATION

Often-used UNIX commands are listed below. Every command, including all its options, is issued as a single line, unless specifically described below as being "interactive." Interactive programs can be made to run from a prepared "script" simply by redirecting their input. All commands are fully described in the *UNIX User's Manual* (see Section 6.1 below). Commands for which additional manuals and tutorials are provided are marked with [m] and [t], respectively. All indicated manuals and tutorials are listed in Section 6.2 below.

File processing commands that go from standard input to standard output are called "filters" and are marked with [f]. The "pipe" facility of the shell may be used to connect filters directly to the input or output of other filters and programs thus creating a "pipeline."

Almost all of UNIX is written in C. UNIX is totally self-supporting: it contains all the software that is needed to generate it, maintain it, and modify it. Source code is included except as noted below.

## 1. BASIC SOFTWARE

Included are the operating system with utilities, an assembler, and a compiler for the programming language C—enough software to regenerate, maintain, and modify UNIX itself, and to write and run new applications. Due to hardware constraints, not all the commands listed below will work on all the supported hardware configurations.

### 1.1. Operating System

■ UNIX [m] This is the basic resident code, also known as the kernel, on which everything else depends. It executes the system calls, maintains the file system, and manages the system's resources; it contains device drivers, I/O buffers, and other system information. A general description of UNIX design philosophy and system facilities appeared in the *Communications of the ACM*. A more extensive survey is in the *Bell System Technical Journal* for July-August 1978. Further capabilities include:

- Automatically-supported reentrant code.
- Separation of instruction and data spaces (machine dependent).
- Timer-interrupt sampling and interprocess monitoring for debugging and measurement.

■ Devices [m] All I/O is logically synchronous. Normally, automatic buffering by the system makes the physical record structure invisible and exploits the hardware's ability to do overlapped I/O. Unbuffered physical record I/O is available for unusual applications. Software drivers are provided for many devices; others can be easily written.

### 1.2. User Access Control

■ LOGIN Signs on a new user:

- Adapts to characteristics of terminal.
- Verifies password and establishes user's individual and group (project) identity.
- Establishes working directory.
- Publishes message of the day.
- Announces presence of mail.
- Lists unseen news items.
- Executes an optional user-specified profile.
- Starts command interpreter (shell) or other user-specified program.

■ PASSWD Changes a password:

- User can change own password.
- Passwords are kept encrypted for security.

■ SU Assume the permissions and privileges of another user or root (super-user) provided that the proper password is supplied.

■ NEWGRP Changes working group (project ID). This provides access with protection for groups of related users.

■ STTY Sets up options for optimal control of a terminal. In so far as they are deducible from the input, these options are set automatically by LOGIN:

- Speed.
- Parity.
- Mapping of upper-case characters to lower case.
- Carriage-return plus line-feed versus new-line.
- Interpretation of tab characters.
- Delays for tab, new-line, and carriage-return characters.
- Raw versus edited input.

■ TABS Sets terminal's tab stops. Knows several "standard" formats.

### 1.3. Manipulation of Files and Directories

■ ED [m,t] Interactive line-oriented context editor. Random access to all lines of a file. It can:

- Find lines by number or pattern (regular expressions). Patterns can include: specified characters, "don't care" characters, choices among characters, (specified numbers of) repetitions of these constructs, beginning of line, end of line.
- Add, delete, change, copy, or move lines.
- Permute contents of a line.
- Replace one or more instances of a pattern within a line.
- Combine or split lines.
- Combine or split files.
- Do any of above operations on every line (in a given range) that matches a pattern.
- Escape to the shell (UNIX command language) during editing.

■ SED [f,m] A stream (one-pass) editor with facilities similar to those of ED.

■ CAT [f] Concatenates one or more files onto standard output. Mostly used for unadorned printing, for inserting data into a "pipe," and for buffering output that comes in dribs and drabs.

■ PR [f] Prints files with title, date, and page number on every page:

- Multi-column output.
- Parallel column merge of several files.

■ SPLIT Splits a large file into more manageable pieces.

■ CSPLIT Like SPLIT, with the splitting controlled by context.

■ SUM Computes the check sum of a file.

■ DD [f] Physical file format translator, for exchanging data with non-UNIX systems, especially OS/360, VS1, MVS, etc.

■ CP Copies one file to another or many files to a directory. Works on any file regardless of its contents.

■ LN Links another name (alias) to an existing file.

■ MV Moves one or more files. Usually used for renaming files or directories.

■ RM — Removes one or more files. If any names are linked to the file, only the name being removed goes away.

■ CHMOD — Changes access permissions on a file(s). Executable by the owner of the file(s), or by the super-user.

■ CHOWN — Changes owner of a file(s).

■ MKDIR — Makes one or more new directories.

■ RMDIR — Removes one or more (empty) directories.

■ CD — Changes working (i.e., current) directory.

■ FIND — Searches the directory hierarchy for, and performs specified commands on, every file that meets given criteria:

- File name matches a given pattern.
- Modified date in given range.
- Date of last use in given range.
- Given permissions.
- Given owner.
- Given special file characteristics.
- Any logical combination of the above.
- Any directory can be the starting "node."

■ CPIO [f] — Copies a sub-tree of the file system (directories, links, and all) to another place in the file system. Can also copy a sub-tree onto a tape, and later recreate it from tape. Often used with the FIND command.

■ SCCS [m] — SCCS (Source Code Control System) is a collections of UNIX commands (some interactive) for controlling changes to files of text (typically the source code of programs or the text of documents). It provides facilities for:

- Storing, updating, and retrieving any version of any source or text file.
- Controlling updating privileges.
- Identifying both source and object (or load) modules by version number.
- Recording who made each change, when it was made, and why.

### 1.4. Execution of Programs

■ SH [f,m,t] — The shell, or command language interpreter, understands a set of constructs that constitute a full programming language; it allows a user or a command procedure to:

- Supply arguments to and run any executable program.
- Redirect standard input, standard output, and standard error files.
- Pipes: simultaneous execution with output of one process connected to the input of another.
- Compose compound commands using:  
  *if ... then ... else*,  
  *case* switches,  
  *while* loops,  
  *for* loops over lists,  
  *break*, *continue*, and *exit*,  
  parentheses for grouping.
- Initiate background processes.
- Perform shell procedures (i.e., command scripts with substitutable arguments).
- Construct argument lists from all file names matching specified patterns.
- Take user-specified action on traps and interrupts.

- Specify a search path for finding commands.
- Upon login, automatically create a user-specifiable environment.
- Optionally announce presence of mail as it arrives.
- Provide variables and parameters with default settings.

■ TEST — Tests argument values in shell conditional constructs:

- String comparison.
- File nature and accessibility.
- Boolean combinations of the above.

■ EXPR — String computations for calculating command arguments:

- Integer arithmetic
- Pattern matching
- Like TEST above, EXPR can be used for conditional side-effect.

■ ECHO — Prints its arguments on the standard output. Useful for diagnostics or prompts in shell procedures, or for inserting data into a ''pipe.''

■ RSH — Restricted shell; restricts a user to a subset of UNIX commands. The system administrator may construct different levels of restriction.

■ SLEEP — Suspends execution for a specified time.

■ WAIT — Waits for termination of a specific or all processes that are running in the background.

■ NOHUP — Runs a command immune to interruption from ''hanging up'' the terminal.

■ NICE — Runs a command at low (or high) priority.

■ KILL — Terminates named process(es).

■ CRON — Performs actions at specified times:

- Actions are arbitrary shell procedures or executable programs.
- Times are conjunctions of month, day of month, day of week, hour, and minute. Ranges are specifiable for each.

■ TEE [f] — Passes data between processes (like a ''pipe''), but also diverts copies into one or more files.

■ HELP — Explains error messages from certain other programs.

### 1.5. Status Inquiries

■ LS — Lists the names of one, several, or all files in one or more directories:

- Alphabetic or chronological sorting, up or down.
- Optional information: size, owner, group, date last modified, date last accessed, permissions.

■ FILE — Tries to determine what kind of information is in a file by consulting the file system index and by reading the file itself.

■ DATE — Print current date and time. Has considerable knowledge of calendrical and horologic peculiarities; can be used to set UNIX's idea of date and time. (As yet, cannot cope with Daylight Saving Time in the Southern Hemisphere.)

■ DF — Reports amount of free space in file system.

■ DU — Prints a summary of total space occupied by all files in a hierarchy.

■ TTY — Prints the ''name'' of your terminal (i.e., the name of the port to which your terminal is connected).

- WHO Tells who is logged onto the system:
  - Lists logged-in users, their ports, and time they logged in.
  - Optional history of all logins and logouts.
  - Tells you who you are logged in as.
- PS Reports on active processes:
  - Lists your own or everybody's processes.
  - Tells what commands are being executed at the moment.
  - Optional status information: state and scheduling information, priority, attached terminal, what the process is waiting for, its size, etc.
- ACCTCOM [f] Reports a chronological history of all process that have terminated. Information includes:
  - User and system times and sizes.
  - Start and end real times.
  - Owner and terminal line associated with process.
  - System exit status.
- PWD Prints name of your working (i.e., current) directory.
- RJESTAT Reports on the status of the Remote Job Entry (RJE) interface(s) to an IBM host.
- WHAT Prints informational lines found in files usually inserted by SCCS.

### 1.6. Inter-User Communication

- MAIL Mails a message to one or more users. Also used to read and dispose of incoming mail. The presence of mail is announced by LOGIN.
- NEWS Prints out current general information and announcement files.
- CALENDAR An automatic reminder service.
- WRITE Establishes direct, interactive terminal-to-terminal communication with another user.
- WALL Broadcasts a message to all users who are logged in.
- MESG Inhibits or permits receipt of messages from WRITE and WALL.

### 1.7. Inter-Machine Communication

- UUCP [m] Sends files back and forth between UNIX machines.
- SEND [m] Collects files together to be sent as a "job" to an IBM host.
- FSEND Sends files to the HONEYWELL 6000.
- FGET Retrieves files from the HONEYWELL 6000.
- CU Dials a phone number and attempts to make an interactive connection with another machine.
- CT Dials the phone number of a modem that is attached to a terminal, and spawns a LOGIN process to that terminal.
- VPM [m] A software package for implementing communications protocols. It consists of a protocol script interpreter that runs in a front-end microprocessor, allowing a variety of different protocols to be implemented with the same hardware.
- BX.25 A superset of the international X.25 communications protocol; it is implemented using VPM.

### 1.8. Program Development Package

A kit of fundamental programming tools. Some of these are used as integral parts of the higher-level languages described in Section 2 below.

■ AR Maintains library archives, especially useful with LD. Combines several files into one for housekeeping efficiency:

- Creates new archive.
- Updates archive by date.
- Replaces or deletes files.
- Prints table of contents.
- Retrieves from archive.

■ Libraries [m] Basic run-time libraries. They are used freely by all system software:

- Number conversions.
- Time conversions.
- Mathematical functions: *sin, cos, log, exp, atan, sqrt, gamma.*
- Buffered character-by-character I/O.
- Random number generator.
- An elaborate library for formatted I/O.
- Password encryption.

■ ADB [t] Interactive debugger:

- Postmortem dumping.
- Examination of arbitrary files, with no limit on size.
- Interactive breakpoint debugging; the debugger is a separate process.
- Symbolic reference to local and global variables.
- Stack trace for C programs.
- Output formats:
  1-, 2-, or 4-byte integers in octal, decimal, or hex
  single and double floating point
  character and string
  disassembled machine instructions
- Patching.
- Searching for integer, character, or floating patterns.
- Handles separated instruction and data space.

■ OD [f] Dumps any file:

- Output options include: octal or decimal by words, octal by bytes, ASCII, operation codes, hexadecimal, or any combination thereof.
- Range of dumping is controllable.

■ SDB [m] Symbolic debugger for C and F77 programs.

■ LD Linkage editor. Combines relocatable object files. Inserts required routines from specified libraries; resulting code:

- Can be made sharable.
- Can be made to have separate instruction and data spaces.

■ NM Prints the *namelist* (symbol table) of an object program. Provides control over the style and order of names that are printed.

■ SIZE Reports the main memory requirements of one or more object files.

■ STRIP Removes the relocation and symbol table information from an object file to save file space.

■ PROF Constructs a profile of time spent in each routine from data gathered by time-sampling the execution of a program; gives subroutine call frequencies and average times for C programs.

■ MAKE [m] — Controls creation of large programs. Uses a control file specifying source file dependencies to make new version; uses time last changed to deduce minimum amount of work necessary. Knows about SCCS, CC, YACC, LEX, etc.

### 1.9. Utilities

■ CXREF — Makes cross-reference listings of a set of C source files. The listing contains all symbols in each file separately or, optionally, in combination. An asterisk appears before a symbol's declaration.

■ SORT [f] — Merges and/or sorts ASCII files line-by-line:

- In ascending or descending order.
- Lexicographically or on numeric key.
- On multiple keys located by delimiters or by position.
- Can fold upper-case characters together with lower-case into dictionary order.

■ UNIQ [f] — Deletes successive duplicate lines in a file:

- Prints lines that were originally unique, duplicated, or both.
- Can give redundancy count for each line.

■ TR [f] — Does character translation according to an arbitrary code:

- Can "squeeze out" repetitions of selected characters.
- Can delete selected characters.

■ DIFF [f] — Reports line changes, additions, and deletions necessary to bring two files into agreement; can produce an editor script to convert one file into another.

■ COMM [f] — Identifies common lines in two sorted files. Output in up to 3 columns shows lines present in first file only, present in second file only, and/or present in both.

■ CMP — Compares two files and reports disagreeing bytes.

■ GREP [f] — Prints all lines in one or more files that match a pattern of the kind used by ED (the editor):

- Can print all lines that fail to match.
- Can print count of "hits."

■ WC [f] — Counts lines and "words" (strings separated by blanks or tab characters) in a file.

■ TIME — Runs a command and reports timing information about it.

## 2. PROGRAMMING LANGUAGES

### 2.1. The Programming Language C

■ CC [m,t] — Compiles and/or link-edits programs in the C language. The UNIX operating system, almost all of its subsystems, and C itself are written in C:

- General-purpose language designed for structured programming.
- Data types:
  - — Character.
  - — Short.
  - — Integer.
  - — Long integer.
  - — Floating-point.
  - — Double.
  - — Pointers to all types.

— Functions returning all types.<br>
— Arrays of any type.<br>
— Structures containing various types.

- Provides machine-independent control of all machine facilities, including to-memory operations and pointer arithmetic.
- Macro-preprocessor for parameterized code and for the inclusion of other files.
- All procedures recursive, with parameters passed by value.
- Run-time library gives access to all system facilities.

■ PCC [m] Portable version of CC for a variety of computers.

■ CB [f] C beautifier: gives a C program that well-groomed, structured, indented look.

### 2.2. FORTRAN

■ F77 [m] A full compiler for ANSI Standard *FORTRAN 77:*

- Compatible with C and supporting tools at object level.
- Optional source compatibility with *FORTRAN 66*.
- Free format source.
- Optional subscript-range checking, detection of uninitialized variables.
- All widths of arithmetic: 2- and 4-byte integer; 4- and 8-byte real; 8- and 16-byte complex.

■ RATFOR [m] Ratfor adds rational control structure à la C to *FORTRAN:*

- Compound statements.
- *If-else*, *do*, *for*, *while*, *repeat-until*, *break*, *next* statements.
- Symbolic constants.
- File insertion.
- Free format source
- Translation of relationals like >, >=, etc.
- Produces genuine *FORTRAN* to carry away.
- May be used with F77.

■ EFL [m] Compiles a program written in the *EFL* Language into clean *FORTRAN* on the standard output. It provides the C-like control constructs of RATFOR.

### 2.3. Other Algorithmic Languages

■ AWK [m] Pattern scanning and processing language. Searches input for patterns, and performs actions on each line of input that satisfies the pattern:

- Patterns include regular expressions, arithmetic and lexicographic conditions, boolean combinations and ranges of these.
- Data treated as string or numeric as appropriate.
- Can break input into fields; fields are variables.
- Variables and arrays (with non-numeric subscripts).
- Full set of arithmetic operators and control flow.
- Multiple output streams to files and pipes.
- Output can be formatted as desired.
- Multi-line capabilities.

■ BS An interactive interpreter, containing features of both *BASIC* and *SNOBOL4:*

- Statements include:

— *for/while . . . next*<br>
— *goto*<br>
— *if . . . else . . . fi*<br>
— trace<br>
— symbolic dump

- All numeric calculations in double precision.
- Recursive function defining and calling.
- Built-in functions include *log, exp, sin, cos, atan, ceil, floor, sqrt, abs, rand.*
- String operations include regular expression pattern matching.
- Very general I/O (including pipes to commands) is provided.

■ DC [m] Interactive programmable desk calculator. Has named storage locations, as well as conventional stack for holding integers and programs:

- Arbitrary-precision decimal arithmetic.
- Appropriate treatment of decimal fractions.
- Arbitrary input and output radices, in particular binary, octal, decimal, and hexadecimal.
- Postfix ("Reverse Polish") operators:
  + − * /
  remainder, power, square root
  load, store, duplicate, clear
  print, enter program text, execute

■ BC [m] A C-like interactive interface to the desk calculator DC:

- All the capabilities of DC with a high-level syntax.
- Arrays and recursive functions.
- Immediate evaluation of expressions and evaluation of functions upon call.
- Arbitrary-precision elementary functions: *exp, sin, cos, atan.*
- Goto-less programming.

■ SNO An interpreter very similar to *SNOBOL 3*; its limitations are:

- Function definitions are static.
- Pattern matches are always anchored.
- No built-in functions.

### 2.4. Macro-Processors and Compiler-Compilers

■ M4 [f,m] A general-purpose macro-processor:

- Stream-oriented, recognizes macros anywhere in text.
- Integer arithmetic.
- String and substring capabilities.
- Condition testing, file manipulation, arguments.

■ YACC [m] An LALR(1)-based compiler-writing system. During execution of resulting parsers, arbitrary C functions can be called to do code generation or take semantic actions:

- BNF syntax specifications.
- Precedence relations.
- Accepts formally ambiguous grammars with non-BNF resolution rules.

■ LEX [m] LEX helps write programs whose control flow is directed by instances of regular expressions in the input stream. It is well suited for editor-script type transformations and for segmenting input in preparation for a parsing routine.

## 3. TEXT PROCESSING

### 3.1. Formatters

High-level formatting macros have been developed to ease the preparation of documents with NROFF and TROFF, as well as to exploit their more complex formatting capabilities.

■ NROFF [f,m,t] Advanced formatter for terminals. Capable of many elaborate feats:

- Justification of either or both margins.
- Automatic hyphenation.
- Generalized page headers and footers, automatic page numbering, with even-odd page differentiation capability, etc.
- Hanging indents and one-line indents.
- Absolute and relative parameter settings.
- Optional legal-style numbering of output lines.
- Nested or chained input files.
- Complete page format control, keyed to dynamically-planted "traps" at specified lines.
- Several separately-definable formatting environments (e.g., one for regular text, one for footnotes, and one for "floating" tables and displays).
- Macros with substitutable arguments.
- Conditional execution of macros.
- Conditional insertion or deletion of text.
- String variables that can be invoked in mid-line.
- Computation and printing of numerical quantities.
- String-width computations for unusually-difficult layout problems.
- Positions and distances expressible in inches, centimeters, ems, ens, line spaces, points, picas, machine units, and arithmetic combinations thereof.
- Dynamic (relative or absolute) positioning.
- Horizontal and vertical line drawing.
- Multi-column output on terminals capable of reverse line-feed, or through the postprocessor COL.

■ TROFF [f,m,t] This formatter generates output on a phototypesetter. It provides facilities that are upward-compatible with NROFF, but with the following additions:

- Vocabulary of several 102-character fonts (any 4 simultaneously) in 15 different point sizes.
- Character-width and string-width computations for unusually difficult layout problems.
- Overstrikes and built-up brackets.
- Dynamic (relative or absolute) point size selection, globally or at the character level.
- Terminal output for rough sampling of the product.

☞ *This entire document was typeset by TROFF, assisted by MM, TBL, and EQN.* ☜

■ EQN [f,m] A mathematical preprocessor for TROFF. Translates in-line or displayed formulae from a very easy-to-type form into detailed typesetting instructions. For example:

sigma sup 2 = 1 over N sum from j=1 to N ( x sub j − x bar ) sup 2

produces:

$$\sigma^2 = \frac{1}{N}\sum_{j=1}^{N}(x_j - \bar{x})^2$$

- Automatic calculation of point size changes for subscripts, superscripts,
- Full vocabulary of Greek letters, such as $\gamma$, $\Pi$, $\Gamma$, $\alpha$.
- Automatic calculation of the size of large brackets.

- Vertical "piling" of formulae for matrices, conditional alternatives, etc.
- Integrals, sums, etc., with arbitrarily complex limits.
- Diacriticals: dots, double dots, hats, bars, etc.

Formulae can appear within tables to be formatted by TBL (see below).

■ NEQN [f,m] A mathematical preprocessor for NROFF with the same facilities as EQN, except for the limitations imposed by the graphic capabilities of the terminal being used. Prepares formulae for display on various Diablo-mechanism terminals, etc.

■ MM [m] A standardized manuscript layout macro package for use with NROFF/TROFF. Provides a flexible, user-oriented interface to these two formatters; designed to be:

- Robust in face of user errors.
- Adaptable to a wide range of output styles.
- Can be extended by users familiar with the formatter.
- Compatible with both NROFF and TROFF.

Some of its features are:

- Page numbers and draft dates.
- Cover sheets and title pages.
- Automatically-numbered or "lettered" headings.
- Automatically-numbered or "lettered" lists.
- Automatically-numbered figure and table captions.
- Automatically-numbered and positioned footnotes.
- Single- or double-column text.
- Paragraphing, displays, and indentation.
- Automatic table of contents.

■ MV [m] A TROFF macro package that makes it easy to typeset professional-looking projection foils and slides.

■ TBL [f,m] A preprocessor for NROFF that translates simple descriptions of table layouts and contents into detailed formatting instructions:

- Computes appropriate column widths.
- Handles left- and right-justified columns, centered columns, and decimal-point aligned columns.
- Places column titles; spans these titles, as appropriate.

For example:

| **Composition of Foods** | | | |
|---|---|---|---|
| *Food* | *Percent by Weight* | | |
| | Protein | Fat | Carbo-hydrate |
| Apples | .4 | .5 | 13.0 |
| Halibut | 18.4 | 5.2 | ... |
| Lima beans | 7.5 | .8 | 22.0 |
| Milk | 3.3 | 4.0 | 5.0 |
| Mushrooms | 3.5 | .4 | 6.0 |
| Rye bread | 9.0 | .6 | 52.7 |

■ CW [f] A preprocessor for TROFF that prepares text to be displayed in a special "constant-width" typeface; this typeface is very useful for printing examples of computer output in, e.g., programming manuals.

### 3.2. Other Text Processing Tools

■ SPELL [f] Finds spelling errors by looking up all uncommon words from a document in a large spelling list. Knows about prefixes and suffixes and can cope with such rotten spellings as "roted."

■ PTX Generates a permuted index, like the one in the *UNIX User's Manual.*

■ GRAPH [f] Given the coordinates of the points to be plotted, draws the corresponding graph; has many options for scaling, axes, grids, labeling, etc.

■ TPLOT [f] Makes the output of GRAPH suitable for plotting on a Diablo-mechanism terminal.

■ 300, 450 [f] Exploits the hardware facilities of GSI 300, DASI 450, and other Diablo-mechanism terminals:

- Implements reverse line-feeds and forward and reverse fractional-line motions.
- Allows any combination of 10- or 12-pitch printing with 6 or 8 lines/inch spacing.
- Approximates Greek letters and other special characters by overstriking in plot mode.

■ HP [f] Like 300, but for the Hewlett-Packard 2640 family of terminals.

■ COL [f] Reformats files with reverse line-feeds so that they can be correctly printed on terminals that cannot reverse line-feed.

■ Graphics [m,t] Graphics is the name of a collection of commands for manipulating and plotting statistical and graphical data on a Tektronix series 4010 terminal or a Hewlett-Packard 7221A Graphics Plotter. Its facilities include:

- A sophisticated graphical editor.
- Pie and bar chart generators.
- Built-in mathematical functions such as powers, roots, logarithms, and slope and intercept generation.
- Histograms.
- Additive sequence, prime number, and random sequence generators.
- Table of contents generators.

## 4. SYSTEM ADMINISTRATION

### 4.1. Normal Day-to-Day Administration and Maintenance

■ MOUNT Attaches a device containing a file system to the tree of directories. Protects against nonsense arrangements.

■ UMOUNT Removes the file system contained on a device from the tree of directories. Protects against removing a busy device.

■ MKFS Makes a new file system on a device.

■ MKNOD Makes a file system entry for a special file. Special files are physical devices, virtual devices, physical memory, etc.

■ VOLCOPY File system backup/recovery system for disk/disk or disk/tape. Protective labeling of disks and tapes is included.

■ FSCK [m] Used to check the consistency of file systems and directories and make interactive repairs:

- Print statistics: number of files, space used, free space.
- Report duplicate use of space.
- Retrieve lost space.

- Report inaccessible files.
- Check consistency of directories.
- Reorganize free disk space for maximum operating efficiency.

■ SYNC Forces all outstanding I/O on the system to completion. Used to shut down the system gracefully.

■ CONFIG Tailors device-dependent system code to a specific hardware configuration. As distributed, UNIX can be brought up directly on any supported computer equipped with an acceptable tape drive and disk, sufficient amount of main memory, a console terminal, and a clock.

■ CRASH Prints out tables and structures in the operating system. May be used on a running system, but more useful for examining operating system core dumps after a "crash."

### 4.2. System Monitoring Facilities

■ Accounting [m]

The process accounting package covers connect time accounting, command usage, command frequency, disk utilization, and line usage. All of these are summarized by user and by command on a daily, monthly, and fiscal basis. The system lends itself to local needs and modification.

■ Error Logging The UNIX operating system incorporates continuous hardware error detection and reporting.

■ Equipment Test Package [m]

The Equipment Test Package (ETP—available on a separate tape) is a useful addition to a hardware supplier's diagnostic software. It is essentially a UNIX-based hardware exerciser and verifier.

■ System Activity Report [m]

The System Activity Report (SAR) package is a body of programs for sampling the behavior of the operating system. The sampling consists of several time counters, I/O activity counters, context-switching counters, system-call counters, and file-access counters. Reports can be generated on a daily basis, or as desired.

■ Profiler The Profiler is another group of commands for studying the activity of the operating system. It reports the percentage of time that the operating system spends on user tasks, on system functions, and in being idle.

### 4.3. Installation, Administration, and Operation

■ Installation [m] The *Setting up UNIX* document contains the procedures and advice for the first-time installation and for the periodic upgrading of the operating system.

■ Administration [m]

The *Administrative Advice for UNIX* document describes various problems that can occasionally arise during normal operation, and suggests possible solutions. Included are tips on data-set options, specifications for phototypesetter fonts and chemicals, for system tuning, security, troubleshooting, as well as other useful information.

■ Operation [m] The *UNIX Operations Manual* contains a description of console operations, step-by-step operator functions, and operating system error messages and their meanings.

## 5. DEMONSTRATION AND TRAINING PROGRAMS

Unless otherwise indicated, source code for the following interactive programs is *not* included:

■ QUIZ Tests your knowledge of Shakespeare, presidents, capitals, etc. Source code included.

■ BJ A blackjack dealer.

■ MOO A fascinating number-guessing game, rather like Mastermind®.

■ CAL Prints a calendar of specified month or year between A.D. 1 and 9999. Source code included.

■ UNITS Converts quantities between different scales of measurement. Knows hundreds of units; for example, how many kilometers/second (or furlongs/fortnight) is a parsec/megayear? Source code included.

■ TTT A traditional 3×3 tic-tac-toe program that learns. It never makes the same mistake twice, unless you make it forget what it has learned.

■ BACK The game of Backgammon.

■ HANGMAN Children's "guess the word" game.

■ WUMP Thrilling hunt for the mighty wumpus in a dangerous cave.

## 6. USER DOCUMENTATION

### 6.1. UNIX User's Manual

■ MAN [m] On-line and hard-copy versions are provided. The manual contains:

- A system overview.
- Commands.
- System calls.
- Subroutines in the C, math, standard I/O, and specialized libraries.
- File formats for most files known to the system software.
- etc.

### 6.2. Documents For UNIX

This two-volume collection contains documents that supplement the information in the *UNIX User's Manual.* It contains:

■ OVERVIEWS

- UNIX—Overview and Synopsis of Facilities
- The UNIX Time-Sharing System

■ GETTING STARTED

- UNIX Documentation Road Map
- A Tutorial Introduction to the UNIX Text Editor
- Advanced Editing on UNIX
- SED—A Non-Interactive Text Editor
- UNIX for Beginners (Second Edition)
- UNIX Shell Tutorial
- An Introduction to the UNIX Shell

■ DOCUMENT PREPARATION

- A TROFF Tutorial
- NROFF/TROFF User's Manual
- MM—Memorandum Macros
- Typing Documents with MM
- A Macro Package for View Graphs and Slides
- TBL—A Program to Format Tables

- Typesetting Mathematics—User's Guide (Second Edition)
- A System for Typesetting Mathematics

■ PROGRAMMING

- The C Programming Language—Reference Manual
- A Guide to the C Library for UNIX Users
- LINT, a C Program Checker
- A Portable FORTRAN 77 Compiler
- RATFOR—A Preprocessor for a Rational FORTRAN
- The Programming Language EFL
- UNIX Programming (Second Edition)
- MAKE—A Program for Maintaining Computer Programs
- An Augmented Version of MAKE
- SDB—A Symbolic Debugger
- A Tutorial Introduction to ADB

■ SUPPORTING TOOLS AND LANGUAGES

- LEX—A Lexical Analyzer Generator
- YACC—Yet Another Compiler-Compiler
- The M4 Macro Processor
- AWK—A Pattern Scanning and Processing Language (Second Edition)
- Source Code Control System User's Guide
- Function and Use of an SCCS Interface Program
- BC—An Arbitrary Precision Desk-Calculator Language
- DC—An Interactive Desk Calculator
- UNIX Graphics Overview
- A Tutorial Introduction to the Graphics Editor
- STAT—A Tool for Analyzing Data
- Administrative Information for the UNIX Graphics Package
- UNIX Remote Job Entry User's Guide
- UNIX Remote Job Entry Administrator's Guide
- Release 1.0 of the UNIX Virtual Protocol Machine
- Release 2.0 of the UNIX Virtual Protocol Machine
- A Dial-up Network of UNIX Systems
- UUCP Implementation Description
- The Implementation of the LP Spooling System
- LP Administrator's Guide

■ ADMINISTRATION, MAINTENANCE, AND IMPLEMENTATION

- UNIX Operations Manual
- FSCK—The UNIX File System Check Program
- The UNIX Accounting System
- The UNIX System Activity Package
- A Stand-Alone Input/Output Library
- The UNIX Equipment Test Package: Operational Procedures
- UNIX Implementation
- The UNIX I/O System
- UNIX on the PDP-11/23 and 11/34 Computers
- UNIX Assembler Reference Manual
- A Tour Through the Portable C Compiler
- A Tour Through the UNIX C Compiler
- On the Security of UNIX
- Password Security—A Case History