PWB/UNIX—Overview and Synopsis of Facilities

T. A. Dolotta
R. C. Haight

June 1977

Bell Telephone Laboratories, Incorporated

PWB/UNIX
Overview and Synopsis of Facilities

CONTENTS

OVERVIEW 1
1. UNIX TIME-SHARING SYSTEM 1
2. PWB/UNIX COMMAND LANGUAGE 2
3. DOCUMENT PREPARATION AND TEXT PROCESSING 2
4. REMOTE JOB ENTRY 2
5. SOURCE CODE CONTROL SYSTEM 2
HARDWARE 3
SOFTWARE, FACILITIES, AND DOCUMENTATION 3
1. BASIC SOFTWARE 4
  1.1. Operating System 4
  1.2. Access Control 4
  1.3. Manipulation of Files and Directories 4
  1.4. Execution of Programs 6
  1.5. Status Inquiries 7
  1.6. Inter-User Communication 8
  1.7. Program Development Package 8
  1.8. Utilities 9
  1.9. Operations, Maintenance, Backup, and Accounting 10
2. PROGRAMMING LANGUAGES 11
  2.1. The Programming Language C 11
  2.2. FORTRAN 12
  2.3. Other Algorithmic Languages 12
  2.4. Macroprocessors and Compiler-Compilers 13
3. TEXT PROCESSING 14
  3.1. Formatters 14
  3.2. Other Text Processing Tools 15
4. REMOTE JOB ENTRY TO IBM SYSTEM/370 16
5. TYPESETTING 16
6. DEMONSTRATION AND TRAINING PROGRAMS 17
7. PWB/UNIX USER'S MANUAL 17
8. DOCUMENTS FOR PWB/UNIX 18

PWB/UNIX—Overview and Synopsis of Facilities

T. A. Dolotta
R. C. Haight

Bell Laboratories
Piscataway, New Jersey 08854

OVERVIEW

The Programmer's Workbench (PWB) is a computing facility that provides a convenient working environment and a uniform set of tools for computer program development, as well as for text processing. PWB was developed at Bell Laboratories. As of mid-1977, it supports in excess of 1,000 users. It is based on Bell Laboratories' UNIX* time-sharing system, so that it is more properly known as PWB/UNIX.

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. PWB/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.
- Access to all the facilities of other ("host" or "target") computer systems, such as the IBM System/370.
- A high-level programming language conducive to structured programming (C).
- The programming languages Basic and FORTRAN.
- Symbolic debugging systems.
- A variety of system programming tools (e.g., compiler-compilers).
- Sophisticated "desk-calculator" packages.

The central processing units (CPUs) for PWB/UNIX are the Digital Equipment Corporation's (DEC†) PDP-11/45 and /70 computers. These computers, because of their price and word size (16 bits per word), are really large minicomputers. Although PWB/UNIX can run on hardware costing as little as $60,000, a typical PWB/UNIX system costs about $120,000 and can support 24 simultaneous users with ease. Larger systems can support twice that number. The cost per user-hour of PWB/UNIX is significantly lower than that of most other interactive computer systems. PWB/UNIX typically runs unattended.

The PWB/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; allows 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.

---

* UNIX is a Trademark/Service Mark of the Bell System.

† DEC and PDP are registered Trademarks of the Digital Equipment Corporation, Maynard, MA.

- 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.

2. PWB/UNIX COMMAND LANGUAGE

The PWB/UNIX command language utilizes an extended version of the UNIX *Shell* (command language interpreter), as well as commands designed mainly for use within Shell *procedures* (command files). These PWB/UNIX extensions to the UNIX Shell have been aimed at improving its use by large programming groups, and at making it even more convenient for use as a high-level programming language. In line with the philosophy of much existing UNIX software, a very strong attempt has been made to add new features only when they were shown to be necessary by actual user experience, in order to avoid contaminating a compact, elegant system with unnecessary frills that would destroy the inherent simplicity of the design of UNIX. By utilizing the Shell as a programming language, PWB/UNIX users have been able to eliminate a great deal of the programming drudgery that often accompanies a large project. Many manual procedures have been quickly, inexpensively, and conveniently automated. Because it is so easy to create and use Shell procedures, each project that uses PWB/UNIX has tended to customize the general PWB/UNIX environment into one tailored to its own requirements, organizational structure, and terminology.

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 PWB/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 permits quick and convenient production of many kinds of documentation that might be otherwise unobtainable, impractical, or very expensive.

In PWB/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 PWB/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 PWB/UNIX 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, or JES2). To the host system, RJE appears to be a card reader/punch and line-printer station. At the request of a PWB/UNIX user, RJE gathers the job control statements to be sent to the host system and, subsequently, retrieves from the host the resulting output, places it in a convenient PWB/UNIX file for later perusal, and notifies the user of the output's arrival.

5. SOURCE CODE CONTROL SYSTEM

The PWB/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 that 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.

HARDWARE

PWB/UNIX runs on a DEC PDP-11/45 or /70 with at least the following equipment:

- 96K words of memory (2 bytes/word), with memory management.
- Disks: RP03, RP04, RP05, RP06, or equivalent.
- Console typewriter terminal.
- Clock: KW11L or KW11P.
- Tape: TU10, TU16, or equivalent.
- Floating point: FP11B or FP11C.

The following equipment is strongly recommended:

- DH11 communications controller(s) with full modem control.
- Full-duplex 96-character ASCII terminals.
- Extra disk drive for system backup.
- DQS11B communications controller(s) for RJE.

The minimum memory and disk space specified is enough to run and maintain PWB/UNIX. More will be needed to keep all source on line, or to handle a large number of users, big data bases, diversified complements of devices, or large programs. PWB/UNIX does swapping and sharing of reentrant user code in order to minimize main memory requirements. The resident PWB/UNIX operating system uses 40-48K words, depending on the configuration.

A large PWB/UNIX configuration (PDP-11/70, 256K words of main memory, fixed and moving head disks) can generally provide reasonable service to between 40 and 48 simultaneous users.

SOFTWARE, FACILITIES, AND DOCUMENTATION

The most-often-used PWB/UNIX commands are listed below. Every command, including all 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 described in the PWB/UNIX User's Manual (see Section 7 below). Commands for which additional manuals and tutorials are provided are marked with [m] and [t], respectively. All manuals and tutorials are listed in Section 8 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.

Almost all of PWB/UNIX is written in C. PWB/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.

The system is distributed on three 9-track, 800 BPI magnetic tapes. The programs listed in Section 5 below (*Typesetting*) are distributed on a separate tape at an additional cost.

1. BASIC SOFTWARE

It includes the time-sharing operating system with utilities, an assembler, and a compiler for the programming language C—enough software to regenerate, maintain, and modify PWB/UNIX itself, and to write and run new applications.

1.1. Operating System

- UNIX [m,t] — The basic resident code on which everything else depends. Executes the system calls, maintains the file system, and manages the system's resources; 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* (see Section 8 below). Further capabilities include:
  - Automatically-supported reentrant code.
  - Separation of instruction and data spaces.
  - Timer-interrupt sampling and interprocess monitoring for debugging and measurement.
- Devices — 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. Drivers for the following devices are available (others can be written easily):
  - Asynchronous interfaces: DH11, DL11; provide support for the most common interactive ASCII terminals.
  - Synchronous interfaces: DU11, DQS11B.
  - Automatic calling unit interface: DN11.
  - Line printer: LP11, or equivalent.
  - Magnetic tape drives: TU10 and TU16, or equivalent.
  - Fixed head disks: RS03 and RS04, or equivalent.
  - Removable-pack disks: RP03 (recommended only for PDP-11/45) and RP04-5-6, or equivalent, with one or more logical devices per physical device and minimum-latency seek scheduling.
  - Physical memory of PDP-11, or mapped memory in resident system.
  - Phototypesetter: Graphic Systems, Inc., System/1, connected through a DR11C.

1.2. 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.
  - Starts command interpreter (*Shell*) or other user-specified program.
- PASSWD — Changes a password:
  - User can change own password.
  - Passwords are kept in encrypted form.

1.3. Manipulation of Files and Directories

- 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.
- **ED [f,m,t]** — Interactive 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.
  - Escape to Shell (PWB/UNIX command language) during editing.
  - Do any of above operations on every line (in a given range) that matches a pattern.
- **SED [f]** — A stream (one-pass) editor with facilities similar to those of ED.
- **REFORM [f]** — Reformats a file, especially source code from another computer.
- **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; often used to verify that all of a file can be read.
- **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. Works on any file regardless of its contents.
- **RM** — Removes file(s). If any names are linked to the file, only the name being removed goes away.
- **LN** — Links another name (alias) to an existing file.
- **MV** — Moves a file. Used for renaming files or directories.
- **CHMOD** — Changes access permissions on file(s). Executable by owner of file(s), or by the super-user.
- **CHOWN** — Changes owner of file(s).
- **MKDIR** — Makes one or more new directories.
- **RMDIR** — Removes one or more (empty) directories.
- **CHDIR** — 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."

- **DSW** — Interactively steps through a directory, deleting or keeping files.
- **AR** — Maintains archives and libraries. 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.
- **CPIO** — 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.
- **SCCS [m]** — SCCS (Source Code Control System) is a collection of PWB/UNIX commands (some interactive) for controlling changes to files of text (typically, the source code of programs or 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,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.
  - Very easily redirect standard input and/or standard output.
  - Compose compound commands using the following operators:
    - `;` for sequential execution.
    - `|` for simultaneous execution with output of one process "piped" to the input of another.
    - `&` for asynchronous operation.
    - `()` parentheses for grouping.
    - `||` and `&&` for left-to-right conditional evaluation.
  - Use 26 string variables (*a-z*), and a set of string and integer operators.
  - Control the order in which directories are searched for commands.
  - Trace execution of commands for debugging.
  - Execute Shell procedures, which are command scripts with substitutable arguments.
  - Construct arguments that denote all file names that match a specified pattern (regular expression).
  - Use the Shell itself as a command (recursively).
  - Collect command usage statistics.
  - Tutorial covers most of the commands in this section.
- **IF** — Conditionally executes commands on basis of:
  - String comparison.
  - File existence, type, and accessibility.
  - IF-THEN-ELSE-ENDIF control constructs.
  - Return code of a process.
- **WHILE** — Shell procedure looping; END shows end of loop; BREAK terminates loop; CONTINUE requests next iteration; condition testing as in IF.

- SWITCH — Multi-way branch; ENDSW shows end of switch; BREAKSW ends each choice.
- SHIFT — Shifts Shell's arguments one argument to the left; allows a Shell procedure to iterate over its arguments.
- = (equals) — Assigns values to Shell variables.
- EXPR — Evaluates expressions containing Shell variables; knows about arithmetic, logical, and string operators.
- GOTO — Goes to a label in a Shell procedure.
- EXIT — Terminates a Shell procedure; useful with IF.
- NEXT — Names a new standard input for the current Shell procedure.
- PUMP [f] — Creates an instance of the current Shell to substitute Shell arguments and variables into the standard input.
- ECHO — Prints its arguments on the standard output. Useful for diagnostics or prompts in Shell procedures, or for inserting data into a "pipe."
- ONINTR — Traps terminal interrupts.
- SLEEP — Suspends execution for a specified time.
- WAIT — Waits for termination of processes that are running asynchronously.
- NOHUP — Runs a command immune to "hanging up" the terminal.
- NICE — Runs a command in 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.
- RSH — Restricted Shell; restricts a user to a subset of PWB/UNIX commands.
- HELP — Explains error messages from 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 PWB/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 into 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, and its size.
- PWD — Prints name of your working (i.e., current) directory.
- 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:
  - 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: FORTRAN, PL/1, COBOL; eases preparation of input for various RJE host systems.

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.
- WRITE — Establishes direct, interactive terminal-to-terminal communication with another user.
- WALL — Writes to all users who are logged in.
- MESG — Inhibits or permits receipt of messages from WRITE and WALL.

1.7. Program Development Package

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

- AS [m] — Assembler:
  - Creates object program consisting of:
    - Code, possibly read-only.
    - Initialized data or read-write code.
    - Uninitialized data.
  - Relocatable object code is executable without further transformation.
  - Object code normally includes a symbol table (that can be deleted).
  - Several source files can be assembled together.
  - Local labels.
  - Conditional assembly.
  - "Conditional jump" instructions become branches or branches plus jumps, depending on distance.
- Library — The basic run-time library. It is 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 [m].
- Password encryption.

■ DB — Interactive post-mortem debugger. Works on core dump files produced by program aborts, on object files, and on arbitrary files:

- Symbolic addressing within files that have symbol tables.
- Octal, decimal, or ASCII output.
- Symbolic disassembly.
- Octal or decimal patching.

■ 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.

■ 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.

1.8. Utilities

■ CREF — Makes cross-reference listings of a set of files. Each symbol is listed together with file name, line number within file, and text of each line in which it occurs:

- Assembler or C language files can be cross-referenced.
- Gathers or suppresses references to selected symbols.
- Various ways to sort output are available.
- Allows selective printing of uniquely occurring symbols.

■ 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.
- GATH [f] Gathers real and virtual files, including the output of other commands; performs keyword substitution, expansion of tabs, nested file inclusion, etc.
- MAKE [m] Provides a general mechanism for keeping up-to-date large programs that are created by performing many operations on a number of files.
- TIME Runs a command and reports timing information about it.

1.9. Operations, Maintenance, Backup, and Accounting

- 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 an i-node (file system entry) for a special file. Special files are physical devices, virtual devices, physical memory, etc.
- TP Manages file archives on magnetic tape:
  - Collects files into an archive.
  - Updates archive by date.
  - Replaces or deletes archived files.
  - Prints table of contents of an archive.
  - Retrieves from an archive.
- DUMP Dumps the file system on the specified device, selectively by date, or indiscriminately.
- RESTOR Restores a dumped file system, or selectively retrieves parts thereof.
- VOLCOPY An alternate file backup arrangement; system should have a spare disk drive. Protective labeling of disks and tapes is included.
- SU Temporarily allows one to become the super-user with all the rights and privileges thereof. Requires a (distinct) password.
- CHECK Together with DCHECK, ICHECK, and NCHECK, is used to check the consistency of file systems and directories; among them, these programs can:
  - Print statistics: number of files, number of directories, number of special files, space used, free space.
  - Report duplicate use of space.
  - Retrieve lost space.

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

■ CLRM — Peremptorily expunges a file and its space from a file system. Used to repair damaged file systems.

■ 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, PWB/UNIX can be brought up directly on any acceptable CPU with an acceptable tape drive and disk, sufficient amount of main memory, a console terminal, and either clock.

■ AC — Publishes cumulative connect-time report:

- Connect time by user or by day.
- For all users, or for selected users.

■ SA — Prints Shell accounting report. For each command executed, gives:

- Number of times used.
- Total system time, user time, and elapsed time.
- Optionally, averages and percentages.
- Output can be sorted on various fields.

■ LASTCOM — Searches Shell accounting records in reverse chronological order for commands executed by the given user(s), or from the given terminal(s).

■ DCAT — Transfers files between two PWB/UNIX systems over a medium-speed, dial-up synchronous link.

■ Operations — A number of commands and several manuals (see Section 8 below) are provided for the operations staff.

2. PROGRAMMING LANGUAGES

2.1. The Programming Language C

■ CC [m,t] — Compiles and/or link-edits programs in the C language. The PWB/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.
  - Integer.
  - Long integer.
  - Floating-point.
  - Double.
  - Pointers to all types.
  - Functions returning all types.
  - Arrays of any type.
  - 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.
- Object code uses full addressing capability of the PDP-11.
- Run-time library gives access to all system facilities.

■ CDB — An interactive debugger tailored for use with C:

- Usable in real time or post-mortem.
- The debugger is a completely separate process from the program being debugged. No debugging code is loaded with the program being debugged.
- Prints in the most appropriate notation: characters, integers (octal and decimal), floating-point numbers, double precision numbers, machine instructions (disassembled).
- Stack trace and fault identification.
- Break-point tracing.

■ ADB — A more recent, more powerful, but also more complex interactive debugger than CDB.

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

2.2. FORTRAN

■ FC — Compiles and/or link-edits FORTRAN IV programs. Object code is partially interpreted.

- Idiosyncrasies:
  - Free-form, lower-case source code.
  - No arithmetic statement functions.
  - Unformatted I/O requires that expected and actual record lengths agree.
  - No BACKSPACE statement.
  - No P format control character on input.
- Handles mixed-mode arithmetic, general subscripts, and general DO limits.
- 32-bit integer arithmetic.
- Free-format numeric input.
- Understands these nonstandard specifications:
  - LOGICAL*1, *2, *4
  - INTEGER*2, *4
  - REAL*4, *8
  - COMPLEX*8, *16
  - IMPLICIT

■ RC [m] — Compiles and/or link-edits RATFOR programs. RATFOR adds rational control structures to FORTRAN:

- ELSE, FOR, WHILE, REPEAT ... UNTIL statements.
- Symbolic constants.
- File inclusion.
- Compound statements.
- Can produce genuine FORTRAN to carry away.

2.3. Other Algorithmic Languages

■ BAS — An interactive interpreter, similar in style to Basic. Interprets unnumbered statements immediately, numbered statements when given the "run" command:

- Statements include:
  - comment
  - dump
  - for ... next
  - goto

  - if ... else ... fi
  - list
  - print
  - prompt
  - return
  - run
  - save
- All calculations in double precision.
- Recursive function defining and calling.
- Built-in functions include *log, exp, sin, cos, atn, int, sqr, abs, rnd.*
- Escape to the standard PWB/UNIX editor for complex program editing.

■ 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. Macroprocessors and Compiler-Compilers

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

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

■ YACC [m] An LR(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 task of preparing documents with NROFF, as well as to exploit its 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.

- −MM [m,t] A standardized manuscript layout for use with NROFF/TROFF. Provides a flexible, user-oriented interface to these two formatters; designed to be:
  - Robust in face of user errors.
  - Easy to use by all levels of users.
  - 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.

- NEQN [f,m] A mathematical preprocessor for NROFF with the same facilities as EQN (see Section 5 below), except for the limitations imposed by the graphic capabilities of the terminal being used. Prepares formulae for display on the Model 37 TELETYPE®, various Diablo-mechanism terminals, etc.

- 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

3.2. Other Text Processing Tools

- SPELL 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."
- TYPO Finds typographical errors. Statistically analyzes all the words in a text, weeds out several thousand familiar ones, and prints the rest sorted so that the most improbably spelled ones tend to come to the top of the list.
- DIFFMARK [f] Judiciously combines DIFF, ED, and NROFF/TROFF to produce marginal revision bars on output.
- PTX Generates a permuted index, like the one in the PWB/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.
- PLOT [f] Makes the output of GRAPH suitable for plotting on a Diablo-mechanism terminal.
- GSI [f] Exploits the hardware facilities of GSI300, DASI300, 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.
- 450 [f] Like GSI, but for the newer Diablo-mechanism terminals (e.g., DASI450).
- HP [f] Like GSI, 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.

4. REMOTE JOB ENTRY TO IBM SYSTEM/370

- SEND [m] — User interface to PWB/UNIX Remote Job Entry (RJE) facility; may be used interactively. RJE provides for submission and retrieval of jobs from a host system (e.g., an IBM System/360 or System/370 computer using HASP, ASP, or JES2). To the host system, RJE appears to be an IBM 2770 card reader/punch and line-printer station. At the request of a PWB/UNIX user, RJE gathers job control and source statements to be sent to the host system, and subsequently returns to the user output listings and object "decks" (in the form of PWB/UNIX files).
- RJESTAT — Reports interactively on the status of any job(s) on the RJE host systems, as well as on the status of the RJE links themselves.

5. TYPESETTING

This software generates output on a Graphic Systems, Inc., C/A/T System/1 phototypesetter. It is distributed separately as an (extra cost) add-on to PWB/UNIX.

TROFF is much like NROFF (see Section 3.1 above), but provides a number of different character styles, a selection of character (point) sizes, and proportional spacing; −MM, TBL, and the typesetting language for mathematics (EQN/NEQN) are compatible with both NROFF and TROFF, so that the user-level formatting instructions can be the same for both.

- TROFF [f,m,t] — 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, sub-subscripts, etc.
  - Full vocabulary of Greek letters, such as γ, Π, Γ, α.
  - 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.
  - Easily learned by non-programmers and mathematical typists.
  - Formulae can appear within tables to be formatted by TBL (see Section 3.1 above).
- −MV [m] — A set of easy-to-use TROFF macros for making view graphs and slides. Intended for use by secretaries and others not trained in "graphic arts." All facilities of TROFF, TBL, and EQN are available.

6. DEMONSTRATION AND TRAINING PROGRAMS

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

- QUIZ — Tests your knowledge of Shakespeare, presidents, capitals, etc. Source code included.
- CHESS — This chess-playing program scored 1-2-1 and 3-0-1 in the 1973 and 1974 ACM Computer Chess Championships.
- BJ — A blackjack dealer.
- REVERSI — A game of dramatic reversals.
- MOO — A fascinating number-guessing game, rather like Mastermind®.
- FACTOR — Computes prime factors of a number.
- 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, two-dimensional) tic-tac-toe program that learns. It never makes the same mistake twice, unless you make it forget what it has learned.
- CUBIC — An accomplished player of 4×4×4 tic-tac-toe.
- WUMP — Thrilling hunt for the mighty wumpus in a dangerous cave.

7. PWB/UNIX USER'S MANUAL

- MAN — Prints one or more specified PWB/UNIX User's Manual entries on your terminal. Machine-readable and hard-copy versions of that manual are provided:
  - System overview.
  - All commands.
  - All system calls.
  - Most subroutines in the assembler, C, and FORTRAN libraries.
  - All devices and other special files.
  - File formats for all files known to the system software.
  - Descriptions of various terminals commonly used with PWB/UNIX.
  - "Boot" procedures.

8. DOCUMENTS FOR PWB/UNIX

In addition to the PWB/UNIX User's Manual—Edition 1.0, which is a separate volume (see Section 7 above), the following are provided with PWB/UNIX:

General:

1. PWB/UNIX—Overview and Synopsis of Facilities (this document).
2. The UNIX Time-Sharing System (reprinted from *Comm. ACM* 17(7):365-75, July 1974).
3. The UNIX Time-sharing System—A Retrospective.
4. PWB Papers from the Second International Conference on Software Engineering.

Basic User Information:

1. PWB/UNIX Documentation Roadmap.
2. PWB/UNIX Beginner's Course.
3. A Tutorial Introduction to the UNIX Text Editor.
4. Advanced Editing on UNIX.
5. PWB/UNIX Shell Tutorial.
6. UNIX for Beginners.
7. UNIX Programming.
8. C Reference Manual.
9. Programming in C—A Tutorial.
10. A New Input-Output Package.
11. A General-Purpose Subroutine Library for PWB/UNIX.
12. Guide to IBM Remote Job Entry for PWB/UNIX Users.
13. SCCS/PWB User's Manual.

Text Processing, Formatting, and Typesetting:

1. NROFF/TROFF User's Manual.
2. PWB/MM—Programmer's Workbench Memorandum Macros.
3. Typing Documents with PWB/MM.
4. PWB/MM Tutorial.
5. Tbl—A Program to Format Tables.
6. A TROFF Tutorial.
7. Typesetting Mathematics—User's Guide (Second Edition).
8. New Graphic Symbols for EQN and NEQN.
9. PWB/UNIX View Graph and Slide Macros.

Additional Facilities:

1. BC—An Arbitrary Precision Desk Calculator Language.
2. DC—An Interactive Desk Calculator.
3. YACC—Yet Another Compiler Compiler.
4. LEX—Lexical Analyzer Generator.
5. RATFOR—A Preprocessor for a Rational FORTRAN.
6. The M4 Macro Processor.
7. Make—A Program for Maintaining Computer Programs.

Internals, Operations, and Administration:

1. Setting up PWB/UNIX.
2. Administrative Advice for PWB/UNIX.
3. PWB/UNIX Operations Manual.
4. Repairing Damaged PWB/UNIX File Systems.
5. PWB/UNIX RJE Administrator's Guide.
6. The UNIX I/O System.
7. On the Security of UNIX.
8. UNIX Assembler Reference Manual.
9. PWB/UNIX Manual Page Macros.