Plexus
Computers,
Inc.

UNIX Sys5.2 Operating System Model 4300

Sys5.2 is the operating system of all Plexus computer systems. It is the Plexus implementation of System V release 2 of the UNIX™ operating system developed by Bell Laboratories. Unix is a multiprogramming timesharing operating system. It features a hierarchical tree-structured file system, device-independent input/output, interprocess communication, and an extensible user command language.

To ensure compatibility with utilities and applications programs written to operate under Unix, Sys5.2 was transported rather than reimplemented. Any program running under another implementation of Unix System V release 2 can be similarly transported to run under Sys5.2. This compatibility includes 32-bit systems.

Unix is coded almost entirely in the structured higher-level language C. All the utilities and compilers supplied with Unix are also coded in C. Unix, the C language, and the large number of Unix utilities, provide a highly productive programming environment.

FILE SYSTEM

Unix stores information in a hierarchical file system. Individual files are referred to by name and are grouped together into directories. Directories are arranged in a tree structure. A file is referred to by its full pathname; a list of the directories to which it belongs separated by slashes. A full directory pathname, for example, might look like this:

```
/usr/michael/accounting/source/base.c
```

A file can have several names or aliases and can appear in several directories. The system keeps track and deletes the file only when the last alias is deleted.

Unix is device independent. Peripherals such as magnetic tape and terminals are treated as files. They are all collected in the directory /dev. For example, /dev/tty2 is a terminal.

Each file has access control fields in its directory entry. The file owner can permit read, write, and/or execute access to himself, a group or project, or all users independently.

THE SHELL 'sh'

Shell Commands

The shell is the primary user interface with a CPU running Sys5.2. It is both a command language interpreter and a programming language.

As a command language interpreter it reads lines typed by the user and interprets them as requests to execute other programs, inquire about system status, and controls files and peripheral devices.

As a programming language, the shell can be driven by previously typed commands kept within files called shell scripts, and includes such features as control-flow primitives, parameter passing, variable and string substitution, language constructs such as "while," "if then else," "case" and "for," and two-way communication between shell commands.

All shell command lines have the form:

```
command [switches] [argument] [more arguments...]
```

Each field is separated by one or more spaces or tabs. The shell searches for a file with the name 'command' in the search list. It then formats the arguments and executes the file 'command' with the appropriate arguments.

File Arguments

The command line interpreter or shell accesses files in its argument list that match a filename template. The template can include pattern matching characters, such as: * (0 or more characters), ? (exactly one character), and [...] (exactly one character in the sequence contained within the brackets). The shell determines those files which match the specified template and passes them on as arguments to 'command' for execution. Large numbers of files can be processed conveniently using this arrangement.

Redirection

Each program starts out with a standard input (normally the console keyboard), a standard output (normally the console screen or printer), and a standard error output (normally the console screen or printer). The user can 'redirect' any of these to any other file or peripheral by specifying >[word] to redirect the standard output to file 'word,' or <[word] to redirect the standard input from file contents of an already existing file; specifying <<[word] causes the shell to read its input up to a line the same as 'word' or end-of-file.

Pipelines

A pipeline or pipe is a mechanism by which the output of one program can be directed to the input of another program without the use of a temporary file. The required syntax is separation of the commands by a vertical bar '|'. For example:

```
a data |b|c>processed.data
```

The pipeline in the example will execute program 'a' using file 'data.' The output of 'a' will be the input to program 'b.' The output of 'b' will be the input to 'c' whose output is redirected to the file 'processed.data.'

Filters

A program whose normal mode of operation is to accept input only from its standard input, perform some operation on it, and output it only to its standard output is called a filter. Many Unix utilities are filters. One simple example is 'pr' which divides its input into pages and places a header with the filename, the date, and the page number on each page.

One characteristic of Unix programs is that they should be small and simple, doing one thing well.

They then tend to be combined in a pipeline. Thus, for example, every program need not be concerned with the number of lines on a page or a screen and need not include code to handle headings and footings. This is better left to a simple program which handles only headers and footings.

Curses

Curses is a subroutine package for managing terminal screens. Terminfo is a database describing the hardware features and escape sequences used by different terminals. The curses/terminfo package allows programmers to write terminal independent applications that use the terminal dependent features specified in terminfo.

Shell Scripts

The shell has the ability to take its input from a command file called a shell script. This saves typing repetitive commands and avoids typographical errors. This feature can also be used to provide novice users with powerful macro commands which hide much of the complexity associated with a given task.

VIRTUAL PROTOCOL MACHINE

VPM allows the generation, compilation, and downloading of protocol scripts. These scripts are used to communicate to IBM mainframe computers or to other non-Unix systems. Standard scripts are currently available for batch protocols such as 2780/3780 bisync and HASP and interactive protocols such as 3270 bisync.

SOURCE CODE CONTROL SYSTEM

One of the features designed to satisfy the needs of developers of computer programs, the Source Code Control System (SCCS) is a collection of commands for controlling changes to files of text (typically the source code of programs or the text of documents). SCCS limits file update access and maintains a hierarchical tree of versions of files so that they may be restored to any previous state. This system provides a powerful facility for managing software development projects.

SOFTWARE GENERATION SYSTEM

The Software Generation System, SGS, is a collection of programming tools that facilitates both high-level program coding and source-level testing and debugging. High-level programming is done with the C language. A C compiler translates C programs into assembly language programs that are in turn translated into object files by the assembler, as. The link editor, ld, organizes object files into executable load modules. A utility package is also included in SGS for symbolic testing at the C-language source level.

DOCUMENTERS WORKBENCH

The Unix Documenters Workbench is a package of tools used to create papers and reports. These tools include control macros to control the format of the final result, pre-processors to work with tabular data and mathematical equations, and the processors that produce the final hardcopy results.

UNIX UTILITIES

In addition to the shell described above, more than 200 utiliies are distributed with Unix Sys5.2. Many of them are simple filters that can be pipelined together to perform complex functions. The facilities of Unix Sys5.2 cover several categories including: user access control, terminal handing, file and directory management, system management, program execution, status retrieval, accounting, communication, program development, compilation, text processing, text formatting, source code control, games, and system maintenance.

ORDERING INFORMATION

The Sys5.2 operating system, its associated utilities, the Plexus C compiler, and the Plexus assembler are available in versions for 1–8 users, 9–16 users, 17–32 users, and 33 to 64 users.

Model	# Users	Available On
4300Dx	1–8	P/15, P/20
4300Ax	9–16	P/20, P/35, P/60
4300Bx	17–32	P/60
4300Cx	33–64	P/60

UNIX is a trademark of AT&T Bell Laboratories.
Plexus is a trademark of Plexus Computers, Inc.
Specifications subject to change.

PLEXUS

Plexus
Computers,
Inc.

3833 North First Street
San Jose, CA 95134
408 • 943 9433
TWX/TELEX 910 338 2223