AT&T

386 UNIX® System V Release 3.1

Product Overview

NOTICE

The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

AT&T

386 UNIX® System V Release 3.1

Product Overview

AT&T Products and Services

To order documents from the Customer Information Center:

- Within the continental United States, call 1-800-432-6600
- Outside the continental United States, call 1-317-352-8556
- Send mail orders to:

  AT&T Customer Information Center
  Customer Service Representative
  P.O. Box 19901
  Indianapolis, Indiana 46219

To sign up for UNIX system or AT&T computer courses:

- Within the continental United States, call 1-800-221-1647
- Outside the continental United States, call 1-609-639-4458

To contact marketing representatives about AT&T computer hardware products and UNIX software products:

- Within the continental United States, call 1-800-372-2447
- Outside the continental United States, call collect 1-215-266-2973 or 1-215-266-2975

To find out about UNIX system source licenses:

- Within the continental United States, except North Carolina, call 1-800-828-UNIX
- In North Carolina and outside the continental United States, call 1-919-279-3666
- Or write to:

  Software Licensing
  Guilford Center
  P.O. Box 25000
  Greensboro, NC 27420

Table of Contents

Chapter 1: PRODUCT OVERVIEW

List of Tables

Introduction

Purpose and Scope

The *Product Overview* describes the software and documentation available for 386 UNIX System V Release 3.1 Version 1.0. Topics covered include the following:

- Overview of UNIX System V
- Structure and components of the UNIX operating system: kernel, shell, and file system
- Benefits of the UNIX operating system: power, portability, compatibility, flexibility, and standardization
- Description of the features for 386 UNIX System V Release 3.1
- Content and description of each 386 UNIX System V Software Package
- Documentation provided with 386 UNIX System V.

Audience

The audience for this document includes current and potential users of 386 UNIX System V Release 3.1. In particular, this document is directed toward new users who want an overview of what UNIX System V offers them.

Terms you must know are explained as they are introduced. If you do not understand other terms, see the Glossary or a computer dictionary. Most features are described in general terms; however, some features are aimed at a more sophisticated audience. These features require understanding concepts in communications, networking, or programming for which explanations are beyond the scope of this document.

Overview of UNIX System V

The UNIX operating system originated at AT&T Bell Laboratories. Subsequently, AT&T enhanced both its software and documentation, and is now licensing it commercially as UNIX System V.

The UNIX System V product consists of both software and documentation. It provides a base on which many additional products (called add-on products) are built. Some of these additional products address the following areas:

- Program development (for example, compilers, interpreters, run-time libraries, and support tools for C, Cobol, Fortran, etc.)
- Networking (e.g., STARLAN NETWORK, INTEL, ETHERNET, and several products that provide forms of IBM SNA connectivity)
- Data base management (e.g., INFORMIX)
- Text and word processing (e.g., CRYSTALWRITER Plus)
- Other applications (e.g., 20/20).

Many products available from AT&T and other vendors are described in the *AT&T Computer Software Catalog*.

The remainder of this overview gives the key elements of the UNIX system, the benefits of using the UNIX system, a description of the features provided with the 386 UNIX System V Release 3.1, the packaging of the software and documentation for 386 UNIX System V products.

Key Elements of the UNIX System

The UNIX system includes the following key elements:

- UNIX operating system kernel
- Shell command interpreter and programming language
- File system
- Various user and system commands.

The kernel, shell, and file system are discussed in more detail in the following sections. The various user and system commands are contained in the Utilities packages described in Appendix A.

The Kernel

The UNIX operating system kernel (comprising from 5 to 10 percent of the software provided with the system) is the basic resident software on which the entire system relies. It is the only part of the system permanently resident in memory. The job of the operating system kernel is to control user processes and manage system resources.

The kernel executes the shell, overseeing the execution of user programs (or commands) while controlling access to the file system. These programs seem to execute simultaneously because of the system's ability to share the processor among all the programs. Actually, each program is scheduled to use the processor for a short period of time to the exclusion of all other programs. The capability to run several programs simultaneously is called multitasking or multiprogramming.

The kernel, using this multiprogramming ability, provides a multiuser capability that lets many people use the system simultaneously. And each of these users can be running several programs at once.

The Shell

The shell command interpreter allows the user to communicate with the UNIX system. Besides providing the user interface to the kernel and interpreting operating system commands, the shell can be used as a programming language. The user can quickly write custom "shell procedures" to do simple or complex tasks. Because shell procedures are easy to create and use, much of the drudgery associated with programming is eliminated. The shell allows users to enhance and build on UNIX system capabilities and to adapt the operating system to many user applications without using a compiler or link editor.

The File System

The file system of UNIX System V consists of a set of directories and files arranged in a tree-like structure. The file system is built up in a hierarchical way from the root (/) directory which contains certain standard subdirectories including: **bin**, **dev**, **etc**, **install**, **lib**, **lost+found**, **mnt**, **tmp**, and **usr**. These standard directories are necessary for the correct operation of the system. The root directory may contain other directories, including those that users structure to suit their needs.

Some of the features of the file system are:

- Simple and consistent naming conventions. File and directory names are formed and referenced using the same conventions. References to files and directories may be expressed with either the full pathname (e.g., */usr/bin/program1*) or with a name (e.g., *bin/program1*) relative to the current directory in which a user is working (e.g., */usr*). This allows frequent references to logically related files to be made more simply if this is preferable.
- Automatic file space allocation and deallocation that is invisible to users.
- Facilities for creating, accessing, copying, moving, and processing files, directories, or sets of these in a simple, uniform, and natural way. File and record-locking capabilities are also provided. This is especially useful for programs reading or updating data bases.
- Flexible directory and file protection modes that allow all combinations of read, write, and execute access. You can control these access modes independently for the owner of each file or directory, for a group of users (such as all members of a project), and for all other users. The file protection modes are set automatically when a file is created, but can be changed from the shell or under program control by the owner of the file.
- Locking files or records within files. This capability is especially useful for programs reading or updating data bases.
- Sharing files, transparently, among computers. Each computer on a network has control over what local resources other computers can access and what remote resources its users can access.

- Input/output routines treat each physical input/output device the same, from interactive terminals to main memory. This allows uniform file and device input and output.
- Mountable and unmountable file systems and volumes allow users easy access to additional disk space.
- File linking across directories of the same file system. This allows one file to have many names, which means a file can "exist" in the various directories where it might be needed without actually existing in multiple copies.

Benefits of the UNIX System

This section describes how the UNIX system differs from most operating systems.

The job of an operating system is to allocate the resources of the computer. Some operating systems process only one request at a time, thereby restricting the resources of the computer to one user (or one task or process) at a time. UNIX System V allows many processes to appear to be running simultaneously. For example, it can be sending output to several printers, running a spreadsheet program, updating a data base, sending electronic mail to another computer, and accepting input from several terminals, all at the same time. This is known as multitasking or multiprogramming. This multitasking ability also allows many users to interact with the system simultaneously. This is called multiuser mode or time-sharing. UNIX System V multitasking also allows each user to have several processes running simultaneously.

The design of the UNIX system makes it easy to manipulate files and the input and output of commands. The hierarchical file system structure allows for easy adding, deleting, and moving of files within a structure set up by the user. Commands can take their input from either a terminal or a file, and the output of commands can be directed to a file or to a peripheral device as well as to the terminal. In addition, the output of one command can be directed (piped) to be the input of another command, so users can build their own specialized functions easily.

The basic user interface—the shell—of the UNIX system is a command interpreter that recognizes and interprets commands entered by the user. In addition, the shell is a programming language that can be used to create custom applications and functions.

Some systems offer some of these features, but the UNIX system offers them all. These benefits are organized under the topics power, portability, compatibility, flexibility, and standardization.

Power

- UNIX systems are multitasking (described earlier); this enables many tasks to be accomplished simultaneously.
- UNIX systems are multiuser systems; this enables many users to accomplish their work simultaneously, as well as share programs and data.

- Each user can run several tasks simultaneously.
- UNIX System V is designed so applications can be added to it easily without modifying the source code. Particular examples of this include the addition of new devices (terminals, printers, plotters, etc.) and new communications protocols.
- Files can be shared transparently among computers, yet access can be carefully controlled by system administrators.

Portability

- The UNIX system runs on computers ranging from microprocessors to the largest mainframes. This is because it can be changed to work on new computer hardware more easily than other operating systems. This portability has been made possible because most of the operating system is written in the high-level language C developed at AT&T Bell Laboratories.
- The UNIX system environment is portable. Once you have learned to use the UNIX system, you can use different computer hardware that runs the UNIX operating system without learning a lot of new things. The applications software, training, documentation, and your knowledge and expertise with the UNIX system can be applied immediately to make you productive on your new hardware.
- Working together, the combination of commands, languages, and libraries of programs makes it easy to develop and maintain new application programs or systems of programs. (Examples include text processing and data base management systems, and applications for medical suppliers, retailers, or other markets.) Once these applications are developed, they are easily ported to new versions of the UNIX system, or to UNIX systems running on a variety of computer hardware.
- The Extended Terminal Interface utilities allows user programs to produce output on many different types of terminals. This package provides the programmer with features such as window control, highlighting, scrolling, access to special keys, color, and menus. This package works with a data base describing hardware features and control sequences used by different terminals. Thus, the programmer can write terminal-independent applications that use the terminal-dependent specifications found in a common data base.

Compatibility

- Source code written to run on a given release of the UNIX system is generally compatible across the AT&T 6386 and 6386E Work Group Systems, the UNIX PC, the 6300 PLUS, and the 3B Computer families. (See the section titled "Standardization" for efforts to broaden source-code compatibility.)
- The 386 UNIX System V operating system is engineered to support 6300 PLUS UNIX System V applications at a binary level.
- When a new release of the UNIX system is issued, every effort is made to ensure that object code for application products will continue to work on the new software release. However, you may want to upgrade your software to new releases to take advantage of new features of the UNIX system.

Flexibility

- The flexible shell command language lets you tailor the user interface to meet the needs of specific users, and each user can have a different interface.
- The UNIX system treats all files alike, unlike other systems that have many types of files. This means you access files identically from any program or command.
- A practical (hierarchical) file system structure lets you organize files in a way that makes it easy for you to find them again.
- Input/output routines treat each physical input/output device the same, from interactive terminals to main memory. This allows uniform file and device input and output.
- Providing new networking interfaces is simplified by using the STREAMS facilities (see "STREAMS" under "Features for Version 1.0").

Standardization

- AT&T is cooperating with other vendors and industry standards groups such as /usr/group, ANSI (American National Standards Institute) X3J11, and IEEE (Institute of Electrical and Electronic Engineers) P1003 in developing standards for system and subroutine calls and for defining a base system and extensions. These standards will make it easier to develop applications that can be run on many different computers that run the UNIX operating system.

- The UNIX system supports a Transport Interface based on the the Transport Service Definition (Level 4) of the International Standards Organization (ISO) Reference Model for Open Systems Interconnection (OSI). This interface defines how to access the services of a transport protocol. (For more information, see the section "AT&T Transport Interface" under "Features for Version 1.0.")

Features for Version 1.0

This section lists and then describes the major features that are provided with Version 1.0 of the 386 UNIX System V Software Products.

- Remote File Sharing
- STREAMS Mechanism and Tools
- AT&T Transport Interface
- Media-Independent **uucp**
- Listener
- Executable Shared Libraries
- Shared Libraries Generation
- Command Syntax Standard
- Signal Mechanism Enhancements
- Improved Terminal Support Facilities
    - Remote Terminal Support
    - AT&T Windowing Utilities
    - Extended Terminal Interface (ETI) Utilities
- **crash** Command Changes
- New Header Files and Definitions
- Internationalization
    - Support for 8-bit Code Sets
    - Support for Alternate Date and Time Formats
    - Support for Alternate Character Classification and Conversion Rules
- New **awk (nawk)**
- Performance improvements
    - Paging the User-Area

  - Demand Paging
  - 2-Kilobyte File System Utility
  - Remote File Sharing Client Caching
  - Tunable Parameters
- Additional Features
  - Improved Recovery of Files from **cpio** Archives
  - Incremental Backup of Nested File Systems
  - Mandatory File and Record Locking
  - Installable Device Drivers
  - Forms and Menus Language Interpreter (FMLI)
  - Administrative Interface.

Remote File Sharing

The Remote File Sharing feature lets you share files, directories, devices, and named pipes transparently among computers that are linked by a network. The administrator of each computer on the network controls which local resources are available to other computers and which remote resources local users can access.

Sharing is done at the directory level. When you share a directory, you are sharing its entire contents: files, subdirectories, named pipes, and special devices (like printers and typesetters). With Remote File Sharing, you can share data files of interest to several departments of your business, a letter-quality printer or typesetter that no one department could fully utilize by itself, and more.

This feature is provided by the Networking-Remote File Sharing Package within the 386 UNIX System V Foundation Set Product. Since Remote File Sharing was designed to be media- and protocol-independent, you will be able to share files transparently across a wide variety of computer systems and networks that use Release 3.1 and conform to the AT&T Transport Interface.

STREAMS Mechanism and Tools

STREAMS is a general, flexible facility and a set of tools for development of communication and networking services within the UNIX system. It provides a uniform mechanism for program development, ranging from networking applications to individual device drivers.

STREAMS defines standard interfaces for character input/output within the UNIX operating system kernel, and between the kernel and the rest of the UNIX system. This standard interface and mechanism enables modular, portable program development and easy integration of network services. These interfaces have been used in the development of protocol modules and device drivers for Release 3.1. STREAMS provides a broad framework that does not impose any specific network architecture. Its user interface is consistent and compatible with the existing character input/output mechanism, so both STREAMS and character input/output mechanisms are available to the programmer. This modularity allows the following functionality:

— User-level programs (commands such as **uucp**) can be independent of underlying protocols and communications media so the programs need not be changed when new media or protocols between systems become available.

— Network architectures and higher-level protocols can be independent of underlying protocols, drivers, and media.

— Higher-level services can be created by selecting and connecting lower-level services and protocols.

In addition to the standard interfaces, STREAMS provides a set of software tools that help you build modules and drivers.

This feature is provided by the Networking-Network Support Utilities Package within the 386 UNIX System V Foundation Set Product.

AT&T Transport Interface

In Release 3.1, the UNIX system supports a Transport Interface based on the Transport Service Definition (Level 4) of the International Organization for Standardization (ISO) Reference Model for Open Systems Interconnection (OSI). The transport service supports two modes of transfer. The first mode is circuit-oriented and transports data over an established connection in a reliable, sequenced manner. The second mode is message-oriented and supports data transfer in self-contained units with no logical relationship required among units.

The AT&T Transport Interface defines how a user accesses the services of a transport protocol. The protocol module is called a Transport Provider.

Applications programs access the Transport Provider by using the Transport Interface routines in the new Network Services Library. These routines support access to a Transport Provider in a media- and protocol-independent manner. The Transport Provider uses kernel-level routines to send the information to the desired physical device. By using the AT&T Transport Interface, applications programs will be able to access other Transport Providers which may be available in the future.

Two applications that currently use the Transport Interface are Remote File Sharing and the **uucp** command. Remote File Sharing is implemented in the kernel, so it accesses the STREAMS-based Transport Provider protocol modules directly; however, because the **uucp** command is an application program, it calls routines in the Network Services Library. In turn, these routines access the Transport Provider.

This feature is provided by the Networking-Network Support Utilities Package within the 386 UNIX System V Foundation Set Product.

Media-Independent uucp

The version of **uucp** that is supplied with the Base System of the 386 UNIX System V Foundation Set Product will provide queued file transfer services to other UNIX systems via direct-connect and dial-up methods. In addition, if you install the Networking-Network Support Utilities Package, because **uucp** has been enhanced to work with networks that are compatible with the AT&T Transport Interface, **uucp** can send files to other systems over any Transport Provider that conforms to this interface.

STREAMS-based modules have been provided so that remote login services via **cu** can also be used over a network.

Other new capabilities include: the ability to define, within a **Sysfiles** file, different or multiple **Systems**, **Devices**, and **Dialers** files; and the ability to define, within a **Devconfig** file, STREAMS-based transport providers for **cu** and **uucp** services.

While you are installing the Base System software, you can indicate which, if any, Transport Providers are available. If new Transport Providers become available, no changes are needed to the Base System software to accommodate the underlying media or protocols.

Listener

For each Transport Provider on a system, there is an active user-level program called a Listener. The purpose of the Listener is to receive requests for network services from another system, interpret which network service is needed, and initiate a process that has been designated to provide the requested network service. The Listener then drops out of the communications path and continues to listen for new service requests.

This feature is provided by the Networking-Network Support Utilities Package within the 386 UNIX System V Foundation Set Product.

Executable Shared Libraries

A Shared Library is a library of routines that is accessed dynamically at runtime rather than having those routines combined with an application program at load time. The end user of an application that was built like this benefits in several ways:

— The application program may occupy less space on disk.

— When routines in a shared library are changed, the new, improved versions are accessible without recompilation of the programs that access them. For example, by improving the performance of one routine, the performance of every application that uses that routine will be improved immediately.

A shared library that contains most commonly used routines from the C Library is part of all Release 3.1 systems. Most UNIX system commands use routines from this library, and so do any applications that were built with them. When you use the commands and applications built using these shared libraries, these routines will be accessed without any special action on your part. In addition, if the Networking-Network Support Utilities Package is installed, the shared Networking Services Library will be used.

Shared Libraries Generation

The 386 UNIX System V Software Development Set Product provides utilities which enable a programmer to build a library of routines that is accessed dynamically at runtime rather than having those routines combined with an application program at load time. The end user of the application can benefit in several ways (see the previous section, "Executable Shared Libraries").

Command Syntax Standard

A new shell function, **getopts** (an enhanced version of the **getopt** command), is consistent with and supports the applicable rules (Rules 3-10) of the UNIX system command syntax standard. [The standard is described on the **intro**(1) manual page of the *UNIX System V User's/System Administrator's Reference Manual*.] Use of **getopts** in place of the command **getopt** is strongly encouraged because, beginning with the next major UNIX system release, the **getopt** command will no longer be supported. Therefore, to assist in the conversion of affected shell scripts, a conversion command, **getoptcvt,** is provided.

Signal Mechanism Enhancements

A new set of system calls provides an improved mechanism to manage signals. These new calls allow a programmer to establish critical sections of code that will not be interrupted by a set of signals. These signals are not discarded, but are held until released at the end of the critical section. These signal-handling system calls are compatible in name and calling sequences with the BSD 4.1 (Berkeley) version of the UNIX system.

Improved Facilities for Supporting Terminals

These improvements are in three areas.

- **Remote Terminal Support**

  The remote terminal support provided with the 386 UNIX System V Foundation Set Product (often called **curses/terminfo**) has the following new features:

  — support for filters, soft labels, and new AT&T terminals and printers: 510, 513, 4410/5410, 4425/5425, 605, 610, 615, 620, and 630.

  — new commands: **captoinfo** converts **termcap** entries to **terminfo** entries; **infocmp** compares two **terminfo** entries or prints entries in several formats (for example, **infocmp -I** *termtype* prints out the description of the terminal *termtype* from the **terminfo** data base)

- — new options to the **tput** command to initialize and reset a terminal
- — a version of the **terminfo** compiler, **tic**, that is from 50 to 100 times faster and does much better error checking than the previous version.

- **AT&T Windowing Utilities**

  The Base System of the 386 UNIX System V Foundation Set Product provides the basic software commonly required by AT&T windowing terminals. It is compatible with the AT&T 630 BCT terminal. Routines included perform the following functions: create, delete, and manipulate terminal windows, query terminal window status, and provide statistics about usage of windowing routines.

- **Extended Terminal Interface (ETI)**

  C language subroutine libraries that provides character mode screen management functions, color/curses enhancements, and menus. The Extended Terminal Interface provides a TAM transition library enabling character-based, multi-user applications written to the UNIX PC to be ported with minimal modification. The Extended Terminal Interface Libraries are provided in the 386 UNIX System V Software Development Set's Graphics Programming Utilities Package.

Crash Command Changes

In addition to providing debugging support for the new operating system features included in Release 3.1, the user interface to **crash** has been changed extensively to make it easier to use. The syntax of all the functions has been standardized so that similar functions share similar syntax. There is a **help** function within **crash,** a number base converter, a memory search function, and a disassembler capability.

New Header Files and Definitions

New header files and new definitions were added to **/usr/include**:

unistd.h This new header file contains definitions for symbolic constants introduced and used throughout the *1984 /usr/group Standard* publication.

limits.h This new header file contains definitions for commonly used values that vary for different implementations of the UNIX system.

sys/stat.h New definitions were added to this header file to make it easier for programmers to write portable code.

Internationalization

The following features are directed to customers writing application software for use in an international environment.

- **Support for 8-bit Code Sets**

  The **cat**, **ed**, **egrep**, **expr**, **find**, **grep**, **ls**, **pg**, **sed**, **sort**, and **vi** commands and the **curses** library were changed so they would properly handle code sets where all 8 bits are used. Because ASCII only uses seven of the available 8 bits in a byte, some commands made special use of this eighth bit; other commands assumed that if the bit was set, the byte was invalid. Other commonly-used commands, such as **sh** (the shell), already supported 8-bit code sets.

  In addition, 8-bit characters can be sent between a terminal and the system.

- **Support for Alternate Date and Time Formats**

  The **cpio**, **date**, **ls**, **mount**, **pr**, and **sort** commands were changed to provide the date and time in the language and format determined by the value of the **LANGUAGE** environment variable. While the United States conventions remain the default, other languages can be supported by creating and installing a file for the language desired in the **/lib/cftime** directory. The content of that file includes: month and weekday name (full and abbreviated), default local time and date formats, and the default output of the **date** command if the **CPTIME** environment variable is not set. In addition, time zones and alternate time zones (such as daylight time) now can be defined in terms of hours and minutes using the **TZ** environment variable.

- **Support for Alternate Character Classification and Conversion Rules**

cat, **ed**, **egrep**, **grep**, **ls**, **pg**, **sed**, **sort**, and **vi** (commands that convert characters from upper to lower case or classify characters as alphabetic, printable, upper or lower case, etc.), were changed to support code sets or classification rules according to the value of the **CHRCLASS** environment variable. While ASCII remains the default for these operations, other conversion and classification rules are supported by creating and installing a file describing these rules in the **/lib/chrclass** directory. A new administrative command, **chrtbl**, is used to create this file.

New awk (nawk)

awk is a programming language for information retrieval and data manipulation that is often used by people with no programming background. Customers using **awk** in an international environment must use this new version (**nawk**), because the old version does not support 8-bit code sets. Some of the other enhancements to this version of **awk** include:

— The ability to define functions

— New keywords: **delete**, **do**, **func**, **function**, **return**

— New built-in functions: **atan2**, **cos**, **sin**, **rand**, **srand**, **gsub**, **sub**, **match**, **close**, and **system**

— New pre-defined variables: **FNR**, **ARGC**, **ARGV**, **RSTART**, and **RLENGTH**

— The input field separator variable, **FS**, and the third argument to **split** are treated as regular expressions

— The precedence of operations now matches C language precedence.

Because some of these enhancements may not be compatible with some existing **awk** programs, you will get the old version when you type **awk**. To take advantage of these new features, you must use the **nawk** command.

Performance Improvements

Performance has been improved in several key areas:

- **Paged User Area**

  The user area for each process is now eligible to be paged, like the rest of each process. This enhancement significantly improves the performance of memory-limited systems, such as 2 mega-byte configurations, by making more memory available.

- **Demand Paging**

  A memory management technique enabling the system to divide the memory space of a process into small, fixed-size units which are swapped in and out of main memory only as needed. With demand paging, a process size may exceed the system's available memory space. Demand paging may also improve performance when many users are simultaneously active.

- **2-Kilobyte File System Utility**

  The 2-Kilobyte File System Utility package within the 386 UNIX System V Foundation Set provides an optional method of file system organization employing larger (2K) block sizes to improve disk I/O performance.

- **Remote File Sharing Client Caching**

  This feature enables a client system (one that is accessing data from another system) in a Remote File Sharing (RFS) arrangement to maintain a local copy of the data it needs. This means that when a block of data is read from, or written to, a remote system, it is placed in a local buffer where it can be accessed by subsequent requests for data by local processes. This can reduce significantly the amount of data that needs to be sent across the network, resulting in significant performance improvements for many patterns of remote file use. For example, applications programs that access data from a small set of files, such as data base systems, will benefit from this feature.

  Mechanisms are built into the client caching feature to ensure file consistency across the network when multiple machines have opened a file and when a process re-opens a previously closed cached file.

  By default, client caching is enabled; however, caching can be turned off for a particular resource or for an entire system.

- **Tunable Parameters**

  The 386 UNIX System V Foundation Set provides the capability to change system parameters such that performance may be optimized to various configurations and system requirements.

Additional Features

Other enhancements provided with the 386 UNIX System V Foundation Set are described below.

- **Remote File Sharing Loop Back**

  Loop back, also called self-mount, enables you to simulate the higher levels of Remote File Sharing (RFS) functionality on a single system. Application programs designed to use RFS can be tested partially and demonstrations given, without actually communicating with a remote computer.

- **Improved Recovery of Files from cpio Archives**

  If errors are encountered while restoring a file from floppy disk using **cpio**, you can now skip over the bad blocks and continue the restore with the next file.

- **Incremental Backup of Nested File Systems**

  A procedure has been implemented to permit incremental backup of nested file systems. (**/a1** is a file system; **a1/b1** is a nested file system.)

- **Mandatory File and Record Locking**

  This feature permits a process to lock a file or records within a file for exclusive use. This is especially useful for programs reading or updating data bases.

- **Installable Device Drivers**

  The 386 UNIX System V is designed so that device driver software can easily be added.

- **Forms and Menu Language Interpreter (FMLI)**

  These are shell level tools assisting the program developer with the creation of character-based user interfaces to application programs.

- **Administrative Interface**

  An FMLI-based interface to system administration functions available to both console and remote users.

Software Packaging

The 386 UNIX System Software Products available with Version 1.0 consist of the standard 386 UNIX System V Foundation Set and optional products listed below. See Appendix A for description of the software packages distributed with each of these products.

386 UNIX System V Foundation Set

The Foundation Set is the fundamental 386 UNIX System Software product. It provides the UNIX Operating System kernel and a basic set of utilities. This product consists of six separately installable packages. They are:

- Base System Package
- Editing Package
- Remote Terminal Package
- Security Administration Package
- 2-Kilobyte File System Utility Package
- Networking Package.

Base System Package

The **Base System Package** is the minimal required UNIX System V runtime system. All other packages need never be installed if the utilities they provide are not required by the user or by an application the user intends to install. The base system consists of the following utilities:

- Essential Utilities
- System Administration Utilities
- Basic Networking Utilities
- Line Printer Spooler Utilities
- Tunable Parameters
- Installable Drivers
- Inter-Process Communication Utilities

- User Environment Utilities
- User Interface
- Process Accounting Utilities.

Editing Package

The **Editing Package** consists of the following:

- Editing Utilities such as the "vi" screen editor.

NOTE The "ed" and "sed" utilities are provided in the Base System.

- Directory and File Management Utilities
- Spell Utilities.

Remote Terminal Package

The **Remote Terminal Package** consists of:

- Terminfo Data Base

NOTE The Terminal Filter Utilities and Termcap Data Base are provided with the base system package.

Security Administration Package

The **Security Administration Package** provides encryption mechanisms for encoding sensitive data files stored in the computer.

NOTE

The Security Administration Package is distributed only within the United States.

2-Kilobyte File System Utility Package

The **2-Kilobyte File System Utility Package** provides an optional method of file system organization employing large (2K) block sizes.

Networking Package

The **Networking Package** consists of the following two separately installable subpackages:

- Network Support Utilities (1.1)
- Remote File Sharing (1.1).

386 UNIX System V Software Development Set

The Software Development Set is purchased separately and provides the following separately installable packages:

- C Software Development Set (4.1)
  - — The C Programming Language Utilities (CPLU) and System Generation Utilities (SGU)
  - — The Advance Programming Utilities (APU)
  - — The C Programmer's Productivity Tools (CPPT)
- Graphics Programming Utilities Package
  - — The Extended Terminal Interface Libraries and Header Files, (Includes the Curses/Terminfo Low Level Function Library and the High Level Function Libraries.)
  - — TAM Transition Library

386 UNIX System V Cartridge Tape Utilities

The Cartridge Tape Utilities provides the device drivers and system control utilities for the optional cartridge tape drive. The Cartridge Tape Utilities are included with the purchase of the optional Streaming Tape Kit.

Documentation for UNIX System V Release 3.1

A set of documents is delivered with your computer and a revised set is delivered when you upgrade to a new release of UNIX System V. Additional documentation is supplied with any software product add-on that you purchase. In addition, a number of helpful documents intended for use by any general user of the UNIX system may be ordered as needed.

The tables in this section list documents supplied with each 386 UNIX System V product. These document sets may also be orderd separately from your AT&T Account Representative or Authorized Dealer. The *Documentation Roadmap* contains a brief description of each document and describes how documents are related to one another. It will also help you to identify any optional document you need for a particular task. The *Documentation Roadmap* may be ordered with select code (999-300-427).

Table 1-1: 386 UNIX System V Foundation Set

Document Title
Documentation Roadmap
Operations/System Administration Guide
Product Overview
Release Notes
STREAMS Primer
User's/System Administrator's Reference Manual

Table 1-2: 386 UNIX System V Software Development Set

Document Title
Network Programmer's Guide
Programmer Reference Manual
Programmer's Guide
Streams Programmer's Guide
Software Development Set Release Notes

Table 1-3: 386 UNIX System V ISV/VAR Document Development Set

Document Title
Intergrated Software Development Guide

Chapter A: APPENDIX A

Appendix A: Description of 386 UNIX System V Utilities Packages

This appendix describes the contents of each Utilities package available for 386 UNIX System V Release 3.1. A package is standard and delivered with the Foundation Set unless its description notes that it is part of an optional product (available at extra cost).

In addition to the other references given at the end of each package's description; see the manual pages that describe the commands in the package. For these commands see the *User's/System Administration Reference Manual*. Commands for programming are described in the *Programmer's Reference Manual*.

For details of how to install these packages, see your *Operations/System Administrator's Guide*.

- Base System Package
- Editing Package
- Remote Terminal Package
- Security Administration Package
- 2-Kilobyte File System Utility Package
- Networking Package
- C Software Development Set Package
- Graphics Programming Utilities Package
- Cartridge Tape Utilities Package

Base System Package

The Base System Package is the minimal required UNIX runtime system. All other packages need never be installed if the utilities they provide are not required by user or by an application the user intendeds to install. It consists of the following utilities:

- Essential Utilities providing the UNIX operating system kernel, device drivers, and 90 of the most essential UNIX utilities.

- System Administration Utilities providing the commands for managing the hardware and software resources used on the system.
- Basic Networking Utilities providing the **uucp** queued file transfer services. With these utilities you can transfer files and send electronic mail to other UNIX systems, interactively communicate with others using UNIX systems or non-UNIX systems, execute a restricted set of commands on a remote system without logging in, call and log in to a remote system, or call a remote terminal and allow the user of the terminal to log in on your UNIX system.
- Line Printer (LP) Spooler Utilities enabling printer sharing through the temporary storage of data and queuing of print requests. With LP spooling, print requests can be entered at the terminal or console without tying up these devices. Also, spooling enables many users to share a printer or several printers, efficiently.
- Tunable Parameters which enable the system parameters to be changed to maximize performance under various load conditions, system configurations, and applications.
- Installable Drivers which permit the modification of the UNIX system to incorporate additional device drivers.
- Inter-Process Communication (IPC) Utilities permitting cooperating processes to share data and communicate with each other. It contains system calls for obtaining, controlling, and performing operations on the three types of IPC facilities: messages, semaphores, and shared memory.

- The User Environment Utilities providing the commands used to perform mathematical calculations, check or change the executing environment, schedule commands to be run at a later time, and permit interaction with more than one shell layer or switch between layers.
- The Administrative Interface and Forms and Menu Language Interpreter.
- The Process Accounting Utilities.

Editing Package

The Editing Package is optionally installable and consists of the following:

- Editing Utilities such as the "vi" screen editor.

NOTE The "ed" and "sed" utilities are provided in the base system.

- Directory and File Management Utilities permitting comparison and manipulation of the contents of the files and directories. With these commands you can search directories or files, compare the contents of directories or files, and manipulate the contents of files.
- Spell Utilities that enable checking for misspelled words in a file.

Remote Terminal Package

The Remote Terminal Package is optionally installable and consists of:

- Curses Routines and the Terminfo Data Base containing the descriptions and operating capabilities of over 150 popular terminal devices.

NOTE The Terminal Filter Utilities and Termcap Data Base are provided with the base system package.

Security Administration Package

The Security Administration Package is optionally installable and consists of an encryption mechanism for encoding sensitive data files.

NOTE

The Security Administration Package is distributed only within the United States.

2-Kilobyte File System Utility Package

The 2-Kilobyte File System Utility Package within the 386 UNIX System V Foundation Set provides an optional method of file system organization employing larger (2K) block sizes to improve disk I/O performance.

Networking Package

The Networking Package is optionally installable and consists on the following subpackages:

- Network Support Utilities (1.1) extending the system's capabilities to support advanced networking applications such as remote file sharing. The product includes standard STREAMS protocol modules, a Network Utility that monitors Network Service Requests (the Listener), and the shared Network Services Library.
- Remote File Sharing Utilities (1.1) providing the facilities needed to share files transparently among computers. It requires the Network Support Utilities Package and a Transport Provider (such as the STARLAN NETWORK).

C Software Development Set Package

The C Software Development Set Package consists of the following:

- The C Programming Language Utilities (CPLU) and System Generation Utilities (SGU) consisting of a C compiler and a set of programming tools that enable programmers to develop C language programs. It includes a C preprocessor, optimizer, assembler, link editor, and tools for manipulating object files and libraries.
- The Advanced Programming Utilities (APU) which are a set of tools useful to C programmers who need to do advanced programming and symbolic debugging. These tools are useful for working with shared libraries and for assisting you in maintaining different versions of files and programs. It includes **yacc, sdb, lex, mkshlib, make,** and source code control system (SCCS).
- The C Programmer's Productivity Tools (CPPT) consist of the **cscope** browser and **lprof** profiler which help an experienced programmer enhance the efficiency of a C program written in a UNIX system environment.

Graphics Programming Utilities Package

The Graphics Programming Utilities Package provides the Extended Terminal Interface (1.1) libraries that promote fast development of screen management applications. They provide software tools that enable you to incorporate screen management and data entry capabilities into a program.

The Extended Terminal Interface (ETI) libraries consist of the:

- Curses/Terminfo Low Level Function Library providing basic routines for writing to a screen.
- High Level Function Libraries built on top of curses. It consists of functions that create, manipulate, and display panels, forms, and menus.

Also included is the TAM Transition Library that enables character mode applications developed for the UNIX PC using the Terminal Access Method to run on the 386 UNIX System with minimal modification.

Cartridge Tape Utilities Package

The Cartridge Tape Utilities Package provides the device drivers and system control utilities for the optional cartridge tape device.

NOTE

This product will be distributed with the optional Streaming Tape Kit.

Chapter G: GLOSSARY

Glossary

This Glossary defines some of the words and phrases used in this document. The emphasis is on words that have a special meaning with respect to the UNIX system.

-

- **add-on product**

  products that AT&T markets for use with the UNIX operating system.

- **boot**

  to start the operating system, so called because the kernel must bootstrap itself from secondary storage into memory.

- **command**

  1. an instruction to the shell, usually to run a program as a child process (commands are usually found in **/bin** or **/usr/bin**). 2. by extension, any executable file, especially a utility program (commands may be found anywhere; they are searched for according to the value of the shell environment variable **PATH**).

- **device**

  1. a file that is not a plain file or a directory. For example, a tape drive is a device and so is the null device. 2. an input-output unit, either physical or virtual, that appears in the file system as a special file.

- **directory**

  a file that comprises a catalog of file and directory names; the organizing principle of the file system, a directory consists of entries that specify further files (including directories), and constitutes a node of the directory tree.

- **execute**

  informally, to run a program.

- **file**

  1. in general, a potential source of input or destination for output. 2. a directory entry; several directories may have files with the same name.

- **file system**

  1. a collection of files that is accessible via some path from the root directory of the file system. 2. the collection of all files on a computer.

3. the part of the kernel that deals with file systems.

- **filter**

  a program that reads data from standard input, transforms it in some way, and writes the results to standard output.

- **kernel**

  the UNIX system proper; code resident in memory that implements the system calls.

- **library**

  an archive of object files from which the link editor may select functions and data as needed.

- **operating system**

  the program for managing the resources of the computer. It takes care of such things as input/output procedures, process scheduling, and the file system, removing this burden from user programs.

- **program**

  1. an executable file. 2. a process.

- **redirection**

  the feature that allows you to reassign standard input and standard output to files or other devices.

- **shell**

  1. the program (called **sh**) that acts as the interface between the user and the UNIX operating system; it causes other programs to be executed on command; the shell is usually started on a user's behalf when the user logs in. 2. by analogy, any program started upon logging in.

- **standard input**

  the place from which a program expects to receive its input, usually a terminal; however, see "redirection."

- **standard output**

  the place to which a program writes its results, usually a terminal; however, see "redirection."

- **system calls**

  1. the set of system primitive functions through which all system operations are allocated, initiated, monitored, manipulated, and terminated. 2. the system primitives invoked by user processes for system-dependent functions, such as I/O, process creation, etc.

- **UNIX system**

  the name of a family of operating systems (for example, UNIX System V), not an acronym for anything; a trademark of AT&T, the word UNIX should be used as an adjective, for example, ''UNIX system,'' ''UNIX software.''

- **Utilities package**

  a group of programs that performs related functions; examples of packages are Directory and File Management, Performance Measurement, and Line Printer Spooling.

Index

NOTES

NOTES