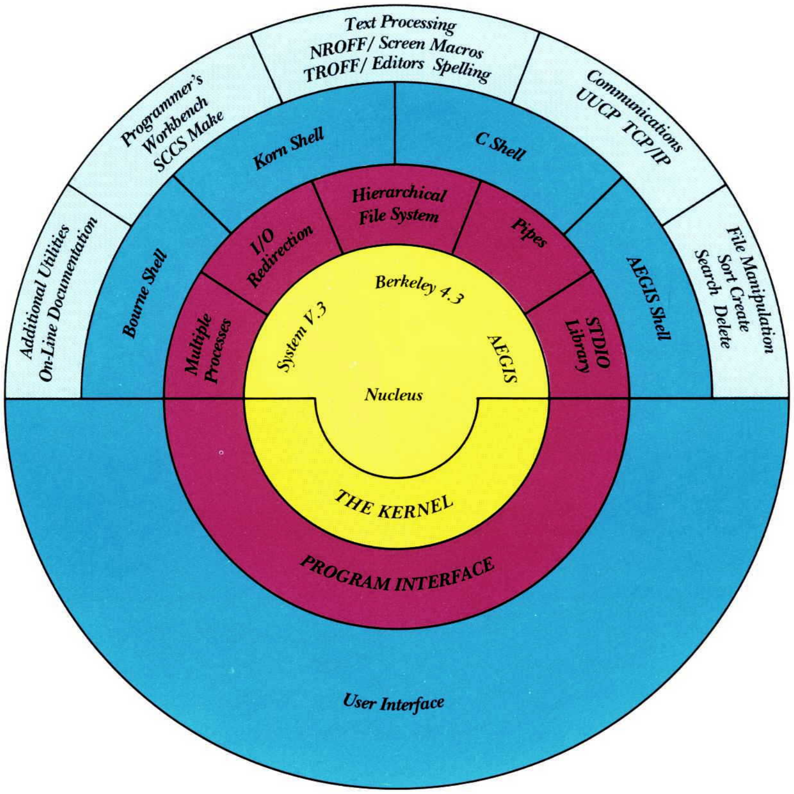
Domain/OS

Software Environment



Description

Apollo's Domain/OS™ is the industry's first operating system that integrates the Network Computing System™ (NCS) architecture to provide a true distributed UNIX® operating system. Built on advanced object-oriented technology, integrating the principles of abstraction and modularity, Domain/OS represents a generational leap forward in application portability, operating system functionality, and performance. It delivers industry standards plus compatible extensions—all on Apollo's advanced workstation platform.

Software support represents a substantial investment for end users, applications developers, and computer vendors alike. The growing importance of standards is tied directly to the industry's need to control software life-cycle costs and provide program and user portability in a heterogeneous computing environment. Domain/OS is the direct result of Apollo's commitment to provide an operating system that meets industry standards. Domain/OS complies with AT&T's System V Interface Definition (SVID), the *de facto* UNIX standard. Domain/OS is also on a migration path to become compliant with IEEE's POSIX, as soon as that standard is finalized. In addition, Apollo is equally committed to providing compatible extensions to those standards to meet users' advanced technological needs.

Domain/OS is a single operating system that provides users a choice of three operating environments—UNIX System V Release 3, 4.3BSD, and Aegis.™ With Domain/OS, users are able to configure nodes and networks to meet their specific functionality, portability and performance needs.

Highlights

- Provides a choice of operating environments for greater flexibility and configurability
- Implements the latest advanced functionality from Berkeley and AT&T
- Meets the SVID standard and provides a migration path to the POSIX standard
- Provides dynamic linking of global shared libraries to reduce object module size, improve paging behavior, reduce disk space requirements, and provide the flexibility to change the implementation of a routine without having to relink programs using that routine
- Supports, through demand paging across the network, the execution of programs on both disked and diskless workstations
- Integrates the Network Computing System architecture to provide a distributed UNIX operating system

- Extends the standard UNIX protection model with compatible Access Control Lists for improved granularity and security
- Enables users, through improved system administration capabilities, to reduce the effort required to maintain a large, integrated network of independent systems
- Provides support in several key areas for internationalization, or native language extensions
- Supports a complete communication product set that promotes the sharing of programs and data in a heterogeneous computing environment
- Incorporates a superior programming environment, consisting of programming languages and CASE tools (including a new object-oriented, user-extensible distributed debugger), that shortens development life-cycle while improving software quality
- Provides compiler support for the UNIX Common Object File Format (COFF) standard, offering a high degree of compatibility and portability for software tools and compilers
- Supports an online, interactive help facility

Flexible Architecture

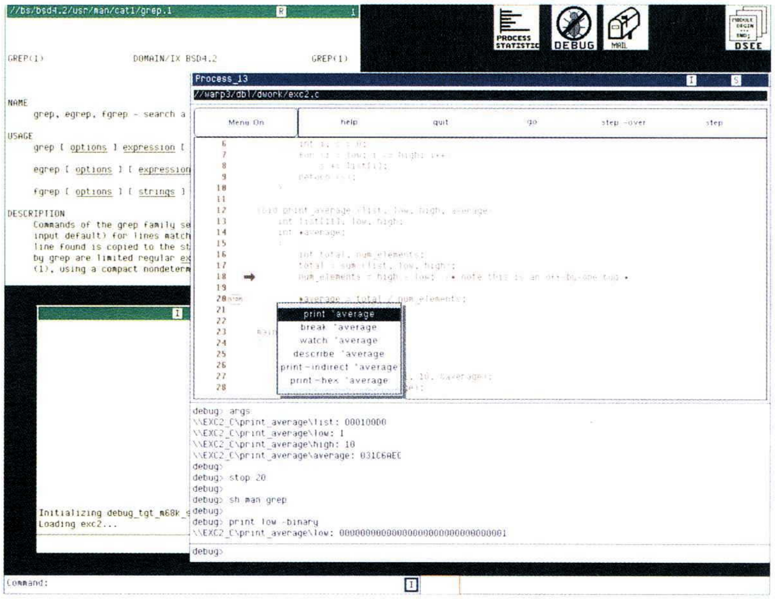
Apollo's object-oriented operating system, Domain/OS, is designed to be easy to maintain, easy to customize, and easy to modify. Object orientation, based on the concepts of data abstraction and modularity, lets users define conceptual objects as software objects. Objects can be devices, special data structures, or standard ASCII files. A software object includes all the software associated with manipulating the device or data (for example, operations such as open, close, read, write) plus the device or data itself. The flexibility and modularity of object orientation allow the implementation of objects to be changed or replaced without affecting other parts of the code. Under Domain/OS, users can redefine or extend traditional kernel functionality without modifying the kernel source code.

The UNIX System V Release 3 and 4.3BSD kernels were originally built on the design center, or computing concept, of a departmental timesharing system. Force fitting these kernels into a workstation architecture would have resulted in a network of sovereign, isolated timesharing systems costly to administer and lacking in important file and data sharing capabilities. Since Apollo started out with a design center that stressed the importance of allowing a large number of distributed workstations (e.g., up to 10,000 nodes or more at a site) to be able to function and be managed as though they were a single system, Apollo decided to build an operating system kernel that supported this design center while faithfully adhering to the UNIX kernel-level interfaces. The result is Domain/OS, a distributed UNIX operating system that is unequalled in the industry for network computing.

The kernel of Domain/OS (the Domain Nucleus) is smaller than most modern UNIX kernels. It is designed so that only absolutely necessary functions are in the kernel. All other functions usually found in the kernel, such as the device-independent I/O subsystem, user-developed device drivers, and Berkeley sockets, as well as system subroutines, are stored in global shared libraries to which programs are dynamically linked, automatically, at execution time. These libraries store most of the program-level interfaces of Domain/OS. This results in smaller object modules, more usable disk space, better paging behavior, and shorter compile/test cycles. This approach also helps produce a more robust and reliable kernel by minimizing the amount of code that can cause a serious system failure, and provides all of the advantages of openness. For example, because device drivers (represented as objects in the operating system) run outside the nucleus, users can easily write and debug device drivers using standard language debuggers. With portions of the I/O system outside the nucleus, users can make major extensions to the I/O subsystem using Apollo's Open System Toolkit.™ There is no need to rebuild and debug the kernel. A similar architectural approach is being taken in modern operating system research efforts at universities like Carnegie Mellon

The new Domain/OS Debugger is a powerful distributed computing object-oriented, user-extensible debugger built on the NCS architecture. It enhances Apollo's environment through its support of multiprocess applications.

With Domain/OS, there is also a new release and installation tool set which allows the system administrator wide latitude in configuring software both on the network and on individual nodes. The administrator can tailor installation procedures easily and set up network defaults, including pre-selecting and/or limiting options available for installation on the network. Software can be configured for both large and small disks; and heavily used files can be installed locally while others can be installed via links to file servers. In addition, software installation can be performed at an individual node by either pulling from the authorized install area or by pushing to multiple target nodes simultaneously.



```
GREP(1)                  DOMAIN/IX BSD4.2                  GREP(1)

NAME
     grep, egrep, fgrep - search a

USAGE
     grep [ options ] expression [

     egrep [ options ] [ expression

     fgrep [ options ] [ strings ]

DESCRIPTION
     Commands of the grep family se
     input default) for lines match
     line found is copied to the st
     by grep are limited regular ex
     (1), using a compact nondeterm
```

```
         int i, s = 0;
         for (i = low; i <= high; i++)
             s += list[i];
         return (s);
     }

void print_average (list, low, high, average)
     int list[11], low, high;
     int *average;
     {
         int total, num_elements;
         total = sum(list, low, high);
         num_elements = high - low;  /* note this is an off-by-one bug */

         *average = total / num_elements;
```

```
debug> args
\\EXC2_C\print_average\list: 00010000
\\EXC2_C\print_average\low: 1
\\EXC2_C\print_average\high: 10
\\EXC2_C\print_average\average: 031C6AEC
debug>
debug> stop 20
debug>
debug> sh man grep
debug>
debug> print low -binary
\\EXC2_C\print_average\low: 00000000000000000000000000000001
debug>
```

```
Initializing debug_tgt_m68k_s
Loading exc2...
```

Native Language Support

Apollo has taken a number of steps toward providing full support for multiple international character sets. The Domain/OS kernel supports the ISO Latin-1 character set, an 8-bit character set that integrates all the popular European characters into Apollo's standard fonts. This means that users can name files and other environment objects using their own native language. The System V Bourne and Korn shells and the Aegis shell all support 8-bit characters for input and output. Apollo also provides an advanced editor, integrated into the window system, that can edit and display text in a user's native language. In addition, the print subsystem has been designed to print 8-bit characters. Native language support includes: Danish, Dutch, English, Faeroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish.

Communications Capabilities

Domain/OS supports a wide range of industry and *de facto* standard communications products for both homogeneous and heterogeneous networks. A single version of TCP/IP, accessible from all three environments, is integrated in Domain/OS. The BSD and SysV environments also access a single version of the HoneyDanBer uucp.

Optional communications products include Ethernet®; Domain/Access™ for file access to DEC VAX/VMS® systems; NFS™; VT100® terminal emulation; IBM SNA, LU 6.2, RJE and BISYNC for communications with IBM host systems; Domain/PCI™ to extend Apollo workstation capabilities to PC users; and X.25.

Domain/OS also integrates components of AT&T's System V Release 3 Network Services Extensions. These components, STREAMS and Transport Level Interface (TLI), provide standard interfaces to ensure portability of networking applications. By using these services, users can write applications that are independent of underlying protocols and, therefore, portable to any compatible network.

Programmer's Environment

Apollo offers a rich complement of languages and CASE tools to provide the most advanced software development environment in the industry. Domain/OS supports the languages Domain/C (with new features such as function prototypes and compile-time type checking, C++ reference variables, and informational messages in addition to warning and error messages), University (MACH project) and University of California-Berkeley (Sprite).

Domain/OS provides the standard UNIX I/O interfaces – open, close, read, write, seek. However, the underlying implementation of file I/O is unique in that both disk and network I/O are achieved through demand paging. The file I/O manager is simply a set of routines that map file pages into the virtual memory of a process. When a program attempts to retrieve data not in real memory, a page fault occurs. The layering of the file I/O manager on top of the paging system makes more efficient use of physical memory by allowing all of it to be available as a cache over the file system. Domain/OS also allows direct access to the mapped file interface for applications that need the maximum possible I/O throughput. In addition, the system uses the same mapping and demand paging mechanism for program execution itself. Because demand paging operates transparently across the network as well as from a local disk, the system is able to support program execution on nodes with or without disks.

The flexibility of Apollo's system architecture allows Domain/OS to support three environments: SysV, BSD, and Aegis. The SysV environment, based on AT&T's System V Release 3, is SVID-compliant and on the migration path to become POSIX- compliant. Included in the SysV environment are the UNIX Bourne, C, and Korn shells. The BSD environment, based on the latest release of 4.3BSD, provides complete Berkeley functionality through the C, Bourne, and Korn shells. The Aegis environment has its own Aegis shell.

Network Computing-based Distributed UNIX

Domain/OS is the first truly distributed UNIX operating system, providing transparent file sharing, automatic concurrency control, support for large networks, and easy additions of both disked and diskless nodes. Domain/OS also implements a distributed registry, remote debugger and network print manager all based on Apollo's Network Computing System architecture, thereby providing a distributed operating system in a heterogeneous computing environment.

The distributed registry supports both very large networks and networks of different machines. The remote debugger supports multiprocess distributed applications. The network print manager coordinates print requests with network print resources. It also supports a set of tools that lets users customize the user interface and extend the print server to support additional printers.

The capabilities of Domain/OS dramatically reduce the cost of owning a network of distributed workstations. System administration is much simpler and less time consuming because the system automatically mounts files, protects files and data from being lost or corrupted and disseminates information on users and passwords to all computers in the network. For the first time, users can harness all the capabilities of a tightly integrated network to distribute data, files and computing power. Domain/OS allows any network, whether it has 2 nodes or 2000 nodes or more, to project a single-system image to users.

UNIX Extensions

Apollo supports existing standards and provides technological advances in the form of compatible extensions to those standards. Two such extensions take the form of Access Control Lists and a new directory called /usr/apollo/bin.

Domain/OS directly implements the UNIX file protection model. The UNIX operating system provides a file protection system that divides the world into the file owner ("user"), individuals in the same group ("group"), and the rest of the world ("other"). Permissions include read, write, and execute. This security system has an advantage in that it is simple to understand and, for many environments, provides adequate security.

However, the standard UNIX protection system is inflexible and inadequate in large networks of many users and where tighter security is required. The UNIX model falls down when there is a need to give a specific individual certain file access (either more or less than others in the group or world), or when there is a need to restrict access to a file so that it can be accessed only from a specific node, or when there is a need for another level of delineation between group and other (for example, one that corresponds to a project team or an organization).

Apollo's solution to the limitations of the UNIX model was to implement an extension to it in the form of Access Control Lists (ACLs). ACLs are lists of access rights that are attached to files. To determine the rights of an individual to a file, the operating system simply searches down the list for the first entry that applies. Entries can be made that explicitly allow or disallow access rights for specific individuals. An object can be protected so that it can be accessed only from the local node, without restricting the availability across the network of other objects on the same disk. In addition, ACLs support a fourth grouping level, called organization, that provides granularity beyond the standard UNIX owner, group, and world. The availability of ACLs provides users a choice. When standard protections are important, users can choose the standard UNIX protection system. When greater granularity and security are required, users can take advantage of the extended ACLs.

A second important extension to the standard UNIX implementation is the development of over 50 utilities that enhance the productivity of UNIX users in a distributed environment. These utilities include network status and management commands, backup and other system administration commands, and commands which support the Open System Toolkit and Apollo's Access Control List extensions. They are stored in the new /usr/apollo/bin directory to distinguish them from standard UNIX utilities. Users who need the extra functionality can choose to use these facilities.

Easy, Network-wide System Administration

Because Apollo builds its systems and networks for workgroup computing, Apollo bases its designs on the concept that "the default is to share" data and files and resources across the network. Domain/OS therefore provides and supports capabilities such as completely transparent file sharing, automatic concurrency control, dynamic re-configuration of diskless nodes (no need to create disk partitions), simple addition of new nodes using one command (no need for *mount* or *umount* on each node in the network), dynamic allocation of paging space, and global shared libraries. All these capabilities simplify the life of a system administrator by eliminating routine tasks.

In addition, to make system administration easier and to reduce security risks, there is a single password system, or registry. The Domain/OS registry consists of a database of naming and account information and a server that manages changes to the data and propagates updated information. The registry, implemented using the Network Computing System, is a distributed system which allows excellent support for both very large networks and networks of different machines. The registry can be replicated; that is, there can be more than one copy of the database/server combination residing on different nodes in the network. Domains of control in the registry can be established, if desired, by assigning the ownership of groups and organizations in the registry database. This flexibility allows system administrators to separate the administration of finance accounts, for example, from the administration of R&D accounts. Registry information is manipulated by the registry owner(s) only, using a single tool, the *edrgy* command. Read-only versions of the UNIX /etc/passwd and /etc/group files, which contain user account and group information, automatically reflect the current contents of the registry. They are not directly editable because an editable ASCII file of user account information is prone to error and represents a security risk in a distributed environment. Import and export of /etc/passwd files are possible to and from non-Apollo systems, to simplify administration of heterogeneous networks.

FORTRAN 77 (with f77 support and the COMPLEX*16 data type), Pascal, Domain/CommonLISP,™ and Domain/Ada.™ By default, Domain/OS compilers generate absolute code for high performance of user applications, while retaining the ability to generate position independent code.

Apollo's CASE tools support Apollo's life cycle approach to CASE. From initial product concept and design, to coding and testing, through tuning, release and maintenance, Apollo offers powerful software tools to make software development faster and more productive.

The new Domain/OS Debugger is a powerful object-oriented, user-extensible debugger built on the NCS architecture. It enhances Apollo's distributed computing environment through its support of multiprocess applications. It provides all the features of Domain/Debug™ plus an extensive offering of new features. Furthermore, the Debugger supports all the features available in dbx, the UNIX debugger, and provides a mode that accepts dbx command syntax. In addition it supports programs written in C, FORTRAN 77, and Pascal, or any combination thereof.

The Domain Software Engineering Environment (DSEE™) offers unequaled support for complex, team-oriented projects because it was specifically designed to take full advantage of Apollo's distributed networking and computing capabilities. The Domain Performance Analysis Kit (Domain/PAK™) gives users a multilevel graphical profiler for complete performance testing and analysis.

Domain/Delphi Documentation Retrieval System

Domain/OS incorporates manual pages for the SysV, BSD, and Aegis environments. In addition, it supports the Domain/Delphi documentation retrieval system, a new optional advanced online system for Apollo documentation. Hardcopy pages are displayed online with attractive fonts and graphics, identical to the corresponding printed manual. An easy-to-use mouse-driven interface with a universal "quick-help" facility makes interacting with Domain/Delphi as simple as point and click.

With Domain/Delphi, users can perform both hierarchical browsing through a global "table of contents" and personally tailored keyword searches. Searches can be limited to cover only those books which are relevant. Domain/Delphi also lets users create personal Reference Lists of frequently used documents to provide direct access to needed information. Search constraint preferences and Reference Lists are maintained from session to session.

Domain/Delphi runs in its own process and window, which means that documents can be retrieved without interrupting ongoing work in other processes and windows. Apollo's network environment allows hundreds of users to share Domain/Delphi data residing on a single host node.

Corporate Headquarters: Apollo Computer Inc., 330 Billerica Rd., Chelmsford, MA 01824, 508-256-6600; TWX: 710-444-8017. CABLE: APOLLOCO
Canadian Headquarters: Apollo Computer Inc., 1530 Markham Road, Suite 130, Scarborough, Ontario, Canada M1B 3G4, 416-297-0700; FAX: 416-297-1020
International Headquarters: Apollo Computer, S.A., 108, Avenue Louis-Casai, P.O. Box 409, 1215 Geneva, Switzerland (41-22) 99 57 88; TWX: 286 18 ch, FAX: (41-22) 98 58 79

The materials contained herein are summary in nature, subject to change, and intended for general information only. Details and specifications concerning the use and operation of Apollo products are available in the applicable manuals, available through local sales representatives.

apollo

Open Software
FOUNDATION

Apollo is proud to be a founding sponsor of the Open Software Foundation.