SOFTWARE

Real Time UNIX System

■ D-NIX

D-NIX is an operating system designed with real-time capabilities and UNIX compatibility. D-NIX is written from scratch with several conceptual differences from standard UNIX while maintaining full compatibilty with UNIX System V.

D-NIX handles real-time

Real-time tasks require predictable and fast responses from the operating system as well as the ability to manage several simultaneous events and prioritize them. D-NIX meets the speed and predictability requirements with a preemptive scheduler, a small and efficient kernel, a high performance file system and memory resident processes.

D-NIX handles simultaneous events and their relative priority with asynchronous system calls and user definable process priorities.

Typical D-NIX applications are:

- **Process control**
- **Factory automation**
- **Transaction processing**
- **Data acquisition**

D-NIX features

- **Small efficient kernel**
- **Preemptive event driven scheduler**
- **Preemption latency up to 1000 times shorter than most UNIX systems**
- **Asynchronous system calls with NO_WAIT kernel, REQUEST and EVENT queues**
- **Handlers for efficient and flexible inter process communication based on the NO_WAIT kernel**
- **Memory resident processes for fast response times**
- **Contiguous files for fast disk I/O**
- **Bit-mapped disk allocation for better speed and reduced disk fragmentation**
- **File FLUSH system call with error diagnostics to guarantee write to physical disk**
- **Mirror disk facility for increased system reliability**
- **Conforms to SVID at Base System and Kernel Extension levels**
- **Extensive network and data communication support**
- **User mapped VME access**

On line transaction processing

OLTP requires high data safety and integrity as well as reliable data communication. D-NIX has a safe file system and extensive data communication support.

The scheduler

The scheduler is preemptive and has 10 static, user settable BASE priority levels. Each BASE level has 4 SUB levels used for time slicing. A process with a higher base level priority will preempt a process with a lower priority.

Asynchronous system calls

Asynchronous system calls, necessary in an event driven environment, require an efficient delivery mechanism. In D-NIX , a NO_WAIT kernel with REQUEST and EVENT QUEUES is employed. A process opens a REQUEST QUEUE and gets a file descriptor in return. This file descriptor is used for future references to the REQUEST QUEUE.

In a real-time environment it is often necessary to wait for the occurence of several events. In D-NIX this is handled by a system mechanism called EVENT QUEUE. The EVENT QUEUE appears like a normal I/O channel where the user can read stored data.

In order to use the EVENT QUEUE facility, the user first opens a EVENT QUEUE. Subsequently he can make system calls, either to read and write or to wait for an external event. The user returns directly from the system call and can make additional calls independent of the completion of previous calls.

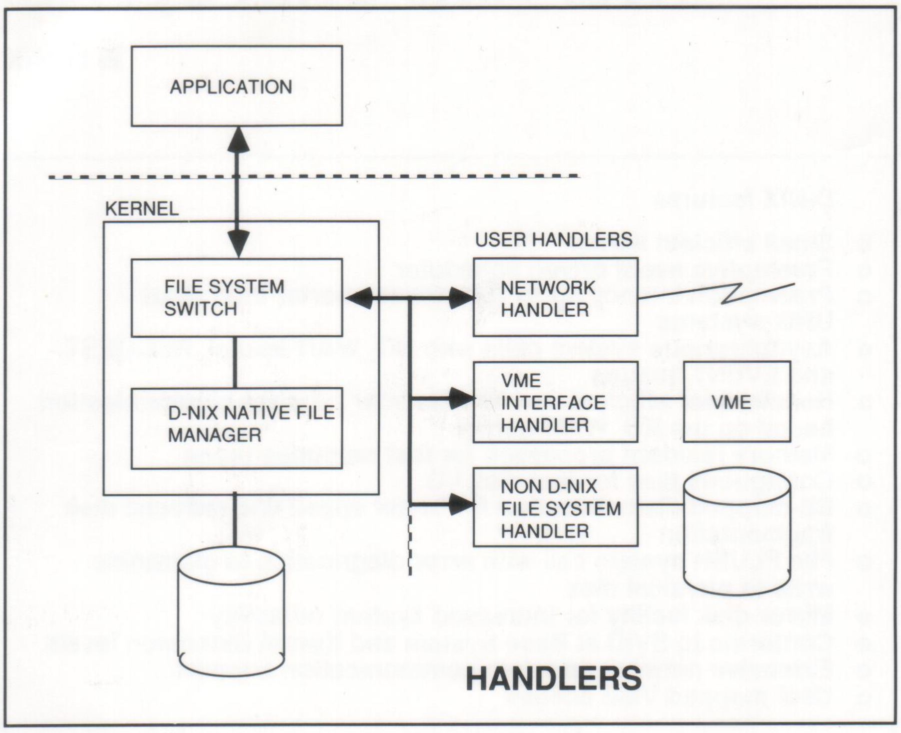
When all 'simultaneous' system calls have been placed, the user issues a read from the EVENT QUEUE and waits until one of the system calls have terminated. The system automatically calls the user from the wait state to the EVENT QUEUE when the system call is complete. Data read from the EVENT QUEUE identifies the terminated system call and the user can thereby take actions accordingly.

The file system

A bit-map is used for disk file allocation in D-NIX rather than the free list used in standard UNIX. Bit-mapping facilitates the identification of large contiguous memory blocks for the allocation of contiguous files. This is very useful in real-time applications when large data volumes are collected as this method is faster and safer than the free list.

Another benefit is that bit-map oriented file system does not degenerate like a free list file system and it is equivalent to a free list sorted at all timed. If the disc is heavily fragmented, the free list may be spread over the entire disk, with a consequential increase in access times.

DIAB DATA



Data integrity
With the file FLUSH system call, buffers for a specific file can be written to physical disk. This means high data integrity, a necessity in transaction processing. The mirror disk facility means that two physical disks represent one logical disk for higher system reliability.

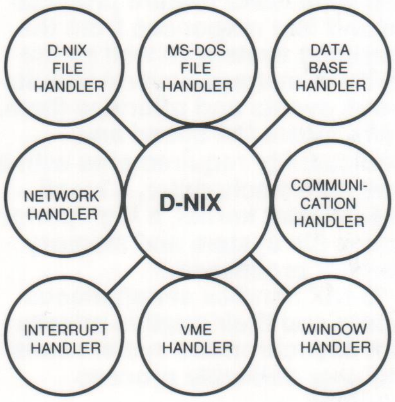
Handlers
A handler is an extension of the UNIX concept that manages system resources, local or remote, as a server to applications. Handlers normally run as user processes. With handlers, the user can add functionality without re-linking or re-configuring the operating system. Normally, operating system extensions are put in the kernel but the handler approach gives the user a possibility to extend the operating system with user defined functions without changing anything in the kernel itself.
An application program can issue standard UNIX requests such as OPEN, CLOSE, READ and WRITE to the handler without considering OS internal resource managing. The handler mounts itself on a directory and all the subsequent requests made to this mount location are passed on to the handler. Parameters are easily passed to the handler for internal use or propagation to other handlers, local or remote, by extending the pathname for the mount location with additional information.

Hardware interface handling and user specific drivers
D-NIX provides a facility called VME macros. It enables the user to access VME boards directly since the VME address space is mapped into the user process. Direct access to the hardware such as read/writes to different ports is essential in many real-time applications and the use of the VME macros makes this faster than accessing the boards through a driver in the kernel.

Virtual demand paging
Virtual demand paging has been employed for a long time on mainframe systems and becomes more and more used in small and medium size computer system. Virtual demand paging makes it possible to create programs that in size are larger than the available physical memory. To accomplish this memory strategy, a secondary storage device is required, such as the standard hard disk, as an adjunct to the main memory under the control of the operating system. As far as the user is concerned, the main memory and secondary storage becomes one contiguous memory resource.

The virtual demand paging mechanism gives the user the full advantage of extremely large system and application software without worrying about the actual memory limitations of the system.

System V compatibility
D-NIX is completely compatible with UNIX System V. D-NIX fulfills all the requirements defined in the UNIX System V Interface Definition at Base and Kernel Extensions levels. Today, D-NIX is implemented on computers based upon processors from the NS32000 family and the Motorola 68000 family. All programs written for computers running under UNIX, Xenix and other look-alikes, and programs using UNIX system calls, may be run under D-NIX.



Installation requirements
Any DS90 system

Ordering information
D-NIX	072-8701-XX

XX is machine dependent.

Please note that the DS90 systems are delivered with D-NIX.

UNIX is a trade mark of AT&T Bell Labs. MSDOS is a trade mark of Microsoft, Inc.

Diab Data/848/1500/Promotive