THE INSIDER'S GUIDE TO THE UNIVERSE

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 Charles River Data Systems equipment and software are available in the applicable technical manuals, available through local sales representatives.

Contents

1 Cosmology of the Universe

Charles River Data Systems has been designing and manufacturing computer system components, memory subsystems, and disk subsystems since its founding in 1973. Through this early history the company focused its talents on systems compatible with DEC computers. It developed strengths in engineering (both hardware and software), system and subsystem testing, and system integration.

In 1979, a software team of individuals with MULTICS and UNIX experience was formed and given the charter to build the next generation operating system for the next generation of microprocessors. They selected Motorola's 68000 microprocessor as the primary processor and modeled their operating system after UNIX. They designed an operating system that took advantage of the most recent engineering advances and chose as their audience the laboratory/scientific, engineering computation, industrial control, and commercial market segments. By mid-1980, UNOS, the new operating system, was running in-house, and in May of 1981 it was demonstrated at the National Computer Conference in Chicago. By September of 1981, Charles River Data Systems shipped its first UNOS systems, running on the company's newly introduced Universe computers. The Universe was the first 32-bit "supermicro" system to combine the performance capabilities of "supermini" with the price and reliability of advanced microcomputer technology. In October, 1982, the company introduced the Universe 68/05, the first 32-bit computer priced under $10,000 (OEM quantity one price), and the first commercial product incorporating the 12.5 MHz 68000 microprocessor. In January of 1983, Charles River Data Systems began shipping Universe 68 systems built around the company's single-board 32-bit processor, the CP32, which makes 32-bit data transfers to the VERSAbus. The new processor, incorporating the 12.5 MHz 68000, a second 68000 character processor, and a 4 Kbyte cache, contributes significantly to the improved system performance that enabled the Universe 68 to break the 1 MIPS (millions of instructions per second) performance barrier.

The Universe family of computers incorporates a number of significant technical and business concepts. It not only brings mainframe computing to the microcomputer world, but it incorporates bus, system, and software architecture that will accommodate future growth in processing power. In addition, Charles River Data Systems has established a business approach designed to suit the various needs of start-up companies, major computer vendors, and systems OEMS. Because of these engineering and business benefits companies with highly diverse requirements have chosen the Universe systems to meet the needs of a wide range of applications, from robotics to commercial data systems.

The purpose of this book is to present an overview of the Universe 68 systems, both hardware and software, and to explain the business side of the company.

Keys to the Universe

The Universe Computer

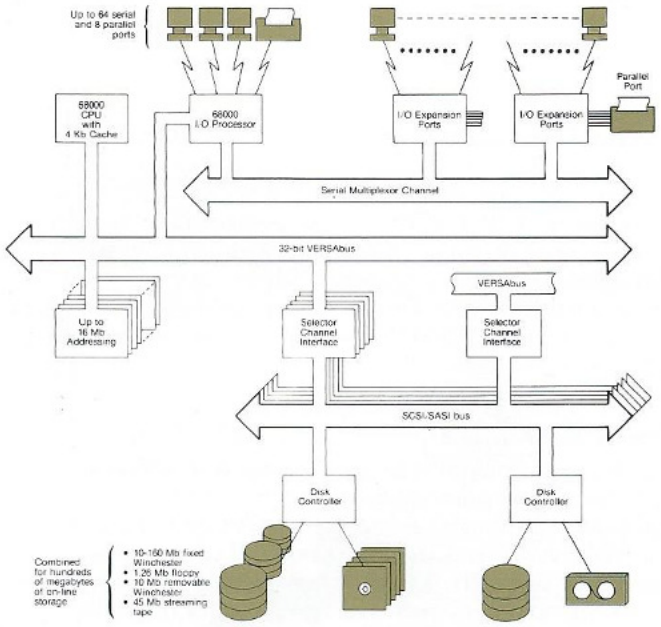
- 32-bit computer with 16 megabytes addressing and ability to handle 32-bit data in parallel
- 32-bit industry standard bus for compatible growth
- 32-bit memory organized as 256 Kbyte, 512 Kbyte, and 1 megabyte modules
- 32-bit 20-megabyte bandwidth master bus
- 32-bit 4 Kbyte cache, eliminating most processor wait states
- ANSI SCSI (SASI) bus interface to peripheral devices
- Separate 68000 character processor for terminal devices
- A wide range of disk devices and backup/interchange media
- Up to 64 serial ports
- Multi-user configurations, providing low per-user cost
- First 32-bit computer system under $10,000
- First commercial product to use the 12.5 MHz 68000 microprocessor

The UNOS Operating System

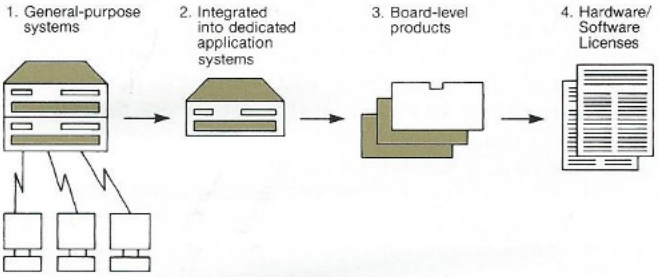
- UNIX-compatible standard systems environment
- Real-time processing extensions
- Designed for efficient and reliable operation
- Optional UNIX System III utilities with many Berkeley enhancements
- A comprehensive selection of languages
- Network facilities for electronic mail, file transfer, remote job execution, and print spooling

A Different Business Approach

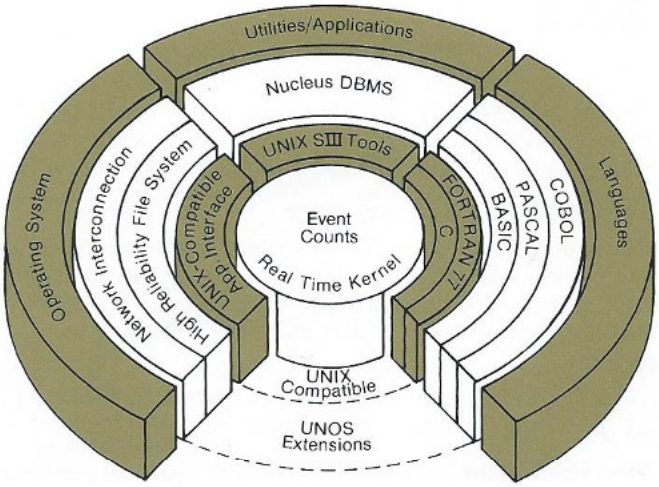
- Targeted at OEMs and end users doing applications development
- Quantity discounts for both software and systems purchases
- Dedication to the use of industry standards to maximize the availability of third party products and minimize the risks associated with proprietary architectures
- An unbundled approach that permits increasing vertical integration including hardware and software licensing
- Support programs tailored to customer needs and capabilities



The Universe 68 has three bus structures. The 32-bit VERSAbus (up to 20Mb/sec bandwidth) interfaces main processor, memory, and channel control processors. Selector channel interfaces connect burst mode or block transfer devices on the SASI bus. A second 68000 controls character-oriented devices.



The availability of a variety of system forms allows OEMs to select a level of vertical integration. Large systems integrate processor and main memory with expansion disk subsystems. Desktop configurations include 5-slot packages with integrated disks. Board-sets and hardware and software manufacturing licenses are also available.



UNOS provides a core, a UNIX-compatible shell of languages, utilities, and operating system capabilities, plus important extensions beyond UNIX in all three areas. Operating system extensions beyond UNIX include Eventcount synchronization and real-time capabilities.

1973	Company founded
1976	1st disk subsystem
1978	1st Winchester disk subsystem
1979	Universe/UNOS team formed
June, 1980	UNOS operational
May, 1981	UNOS demonstrated publicly
September, 1981	Universe/UNOS formally introduced, first shipments made to customers
October, 1982	Universe 68/05 introduced: • 1st commercial product using 12.5 MHz Motorola 68000 μprocessor • 1st 32-bit computer system priced under $10,000 (Universe 68/05)
January, 1983	First shipments of systems based on CP32 single-board central processor

In its early years, Charles River Data Systems focused on DEC-compatible systems. In 1979, a team was formed to develop a next generation operating system for the next generation of microprocessors. The Motorola 68000 was selected as the primary processor, and the operating system was modeled after UNIX.

2 Architecture of the Universe

Charles River Data Systems' Universe family of computers shares a common architecture, bus structure and peripheral modularity that provide substantial processing power.

The 68000 VLSI Processor

The keys to providing a 32-bit systems environment are large direct address space and the numeric precision of the processor. The Universe 68 incorporates the Motorola 68000 microprocessor, which offers the following features:

- up to 16 megabytes of directly addressable memory;
- 32-bit registers and operations.

The IEEE-xxx* Bus: VERSAbus

The Universe 68 is built around the 32-bit, non-proprietary VERSAbus. The VERSAbus was chosen for two reasons. First, because it is a full 32-bit bus with 32 bits of both data and addressing. And second, because it is a standard bus that does not tie either the manufacturer or the customer to any particular processor or vendor. With the next generation of processors, such as the MC68020 or iAP386, a full 32-bit bus will be a *requirement.* VERSAbus provides this capability now. VERSAbus also provides a standard interface for custom devices as well as multiple sources of common interface boards.

The VERSAbus offers these advantages:

- a large (14″ x 9.5″) board geometry well-suited to the complexity and level of functionality of a high performance 32-bit minicomputer;
- 50 user-definable pins (each with a separate ground pin) for direct connection at the backplane;
- a comprehensive priority structure: seven levels for interrupt interfaces, each with multiple devices "daisy chained" on the bus, and five levels of DMA priority, permitting a control of the memory channel not available on any other standard bus;
- 5 MHz bus utilization rate, permitting 16-bit transfers at 10 megabytes per second and 32-bit transfers at 20 megabytes per second; high speed disk units (with transfer rates between one and two megabytes per second) have negligible impact on system performance, and there is enough bandwidth for the most demanding custom devices;
- electrical compatibility with the IEEE-xxx*(VME) bus, for future compact board products.

The Peripheral Control Channels

Selector Channel Interface

A key ingredient to the flexibility of the Universe disk configurations is the Selector Channel Interface. The Selector Channel is modeled after those used in mainframe computer systems. The interface board has its own processor and, as an independent master device, uses the VERSAbus to access channel control blocks stored in the system's buffer area. The burden of disk control is no longer on the central processor. In addition, the unit can transfer 32-bit words to memory, allowing maximum bandwidth transfers with the least impact on concurrent central processor activity. Multiple Selector Channel Interfaces can provide maximum disk storage and concurrent transfer rates. Multiple channels provide an aggregate throughput previously possible only in large mainframe computers. High-performance computers, such as the IBM 370, CRAY I, and VAX 780, use similar intelligent peripheral control channels as slaves to their master bus structures.

The SCSI/SASI Bus

The bus used by the Selector Channel is the ANSI standard Small Computer Systems Interface (SCSI) bus, also known as the Shugart Associates System Interface (SASI) bus. The SASI bus is used by a number of peripheral controller suppliers. The SASI bus permits great flexibility in system configurations. The bus provides multiple masters for either shared disk operations or on-line backup. More than one type of master (multiple Selector Channel Interfaces, the diagnostic control panel, or custom interfaces) can drive the bus. In addition to this multimaster facility, multiple device controllers can be run on the SASI bus. Available disk controllers support multiple disk devices, floppy drives, streaming tapes, control panel or any combination of these devices. They all can be configured on a single SASI bus. The SASI bus is another standard point of interface for custom interfaces and common devices.

The Serial Multiplexor Channel

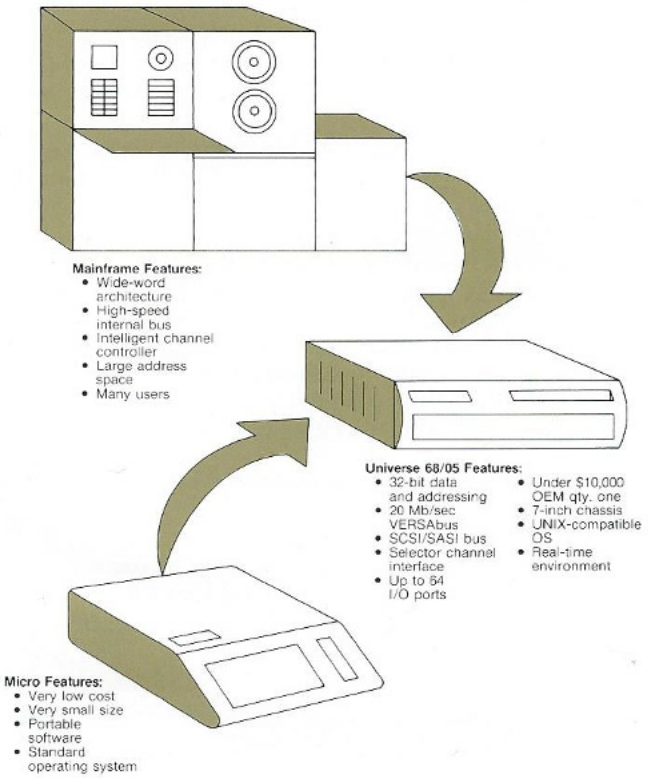
A second type of channel, the Serial Multiplexor Channel, is designed for efficient control of character-oriented devices. Output to a single terminal, at the common rate of 9600 baud, results in about 1000 interrupts per second for the controlling processor; if this data is not moved through a channel device, even a small number of terminals can have a substantial impact on system performance. On the Universe 68 systems these interrupts are handled by the Serial Multiplexor Channel, thus relieving the main processor of this burden. The Serial Multiplexor Channel supports serial and parallel ports for connections to terminals, printers, modems, local networks, and other character-oriented devices. It can support up to 64 devices and interfaces to custom devices.

Multibus and VME Adapter Cards

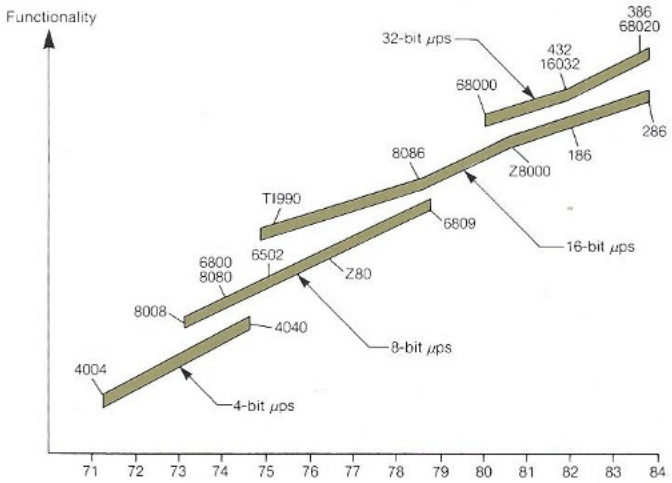
Multibus and VME adapter cards permit the smaller Multibus and VME cards to be plugged directly into the Universe card cage.

*IEEE project numbers have not been assigned for VERSAbus or VME bus at the time of this printing.

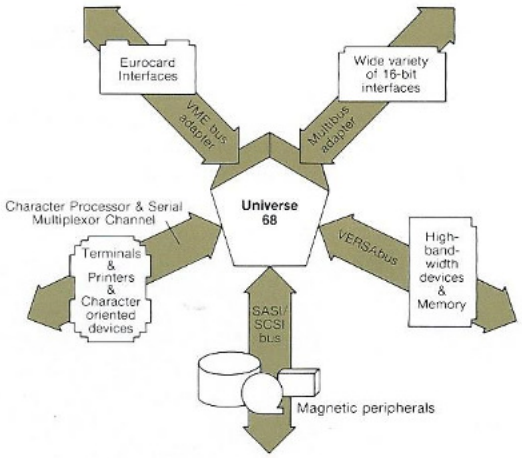
VAX is a trademark of Digital Equipment Corporation.
Multibus is a trademark of Intel Corporation.



The Universe 68 is a mainframe in micro clothing. It combined the low cost and small size of the micros with the large address space, high throughput, 32-bit architecture, and high-speed bus structures inherited from the mainframe world.



Succeeding generations of equal-word length microprocessors have offered more performance. But more performance could also be achieved by moving up to the next-longer wordlength. Microprocessors of greater wordlength have eventually offered performance beyond the upper limits possible at the shorter wordlength.



Multiple-bus architecture allows the Universe 68 to communicate with DMA devices on the VERSAbus, magnetic storage devices on the SASI/SCSI bus, character-oriented devices over the Serial Multiplexor bus, and Eurocard and Multibus interfaces using VME bus and Multibus adapters.

	Data Path	Address Bits	Max Bandwidth Mb/Sec	Single Card Size (sq inches)	User I/O Pins	Multi Master	Inter-Rupt Levels	Multiple Devices at Same Level	DMA Levels	IEEE STD or Proposed	Non-Pro-Prietary
VERSAbus	32-bit	32-bit	20	134	50	yes	7	Both DMA and Interrupt	5	yes (proposed)	yes
VME bus	32-bit	32-bit	20	53	64	yes	7	Both DMA and Interrupt	4	yes (proposed)	yes
Multibus	16-bit	24-bit	10	70	6 (w/22-bit addresses)	yes	7	DMA only	1	yes	yes
Q-Bus LSI-11	16-bit	22-bit	1-2	80	0	no	4	Both DMA and Interrupt	1	no	no
Unibus PDP-11	16-bit	22-bit	2-3	132	0	no	4	Both DMA and Interrupt	1	no	no
SBI VAX	32-bit	32-bit	13.3		0	yes	0	no	4	no	no

VERSAbus and VME bus are the only nonproprietary, 32-bit bus architectures now available. With next-generation microprocessors, such as the MC68020, all systems will require a full 32-bit bus.

3 Cache: The Center of the Universe

The Universe 68 computer line provides the power and functionality of 32-bit systems with the packaging (and price) characteristics of microcomputer systems. Engineering efforts have focused on getting the maximum performance from micro technology while keeping the size and complexity of the system at a minimum. A good example of this effort is the CP32 central processor board. This single board incorporates cache memory, memory segmentation, a separate character processor, and data communications ports.

The Cache

With the 4 Kbyte cache, the 12.5 MHz 68000 microprocessor can execute programs with no wait states for most memory references. The cache is a 45ns RAM memory that stores instructions and data on the processor board. If a request for data is found in the cache (a hit), it can be immediately accessed, thus avoiding the delay of accessing main memory.

The cache is kept filled because every 68000 request for an 8- or 16-bit object results in a 32-bit transfer from memory. This loads the cache with a 32-bit block of data, keeping the cache one step ahead of the processor. This pre-fetching, the locality of references in program loops and stack manipulation result in a cache hit rate of 70 to 95%. To support these cache memory accesses, and to provide maximum throughput for high speed peripherals, the 32-bit bandwidth bus is critical.

The cache uses a write-through technique for processor memory modifications so that the data in the cache never need to be flushed to memory. A cache validity array ensures that the cache accesses return only valid results even when other direct memory access devices are active.

With the cache implementation and the 12.5 MHz 68000 processor, the CP32 can perform at up to 1.25 MIPS. This level of performance is typically associated with mainframe and super-mini class computers.

Memory Segmentation

The Memory Allocation and Protection (MAP) hardware performs two vital tasks for reliable and efficient multiprocessing. First, it provides protection between tasks. This includes protection for a shared instruction area between processes and invalid protection for references outside the user's logical address space. Second, it allocates memory on a logical basis, thus permitting programs to load and/or swap into any area of physical memory. Processes can share instructions, common libraries, and data.

Each process has eight segments. An additional eight segments are for system operations, which reduces the overhead for switching contexts between processes. The design of the MAP is compatible with the cache environment. Again, 45ns access RAM memory ensures that the address translation does not cause delays in the memory access sequence.

The Character Processor

The CP32 contains a second 68000 microprocessor dedicated to character oriented devices (terminals, printers, and network servers). Although the main processor can support 4,000 to 8,000 interrupts per second (the case with as few as four terminals operating at 9600 baud), these interrupts would use all the compute power of the system. The separate character processor relieves the host of this burden.

The character processor is a Direct Memory Access (DMA) device whose function is to eliminate character interrupt overhead from the main processor. It controls four on-board serial ports, which are useful for initial connection to terminals and printers. In addition, the character processor provides an external bus, the Serial Multiplexor Channel, for the connection of other character-oriented device controllers. This simple interface and expansion capability for character devices minimizes the cost for efficient expansion. Up to 64 ports can be added without another VERSAbus board or DMA controller.

Economy

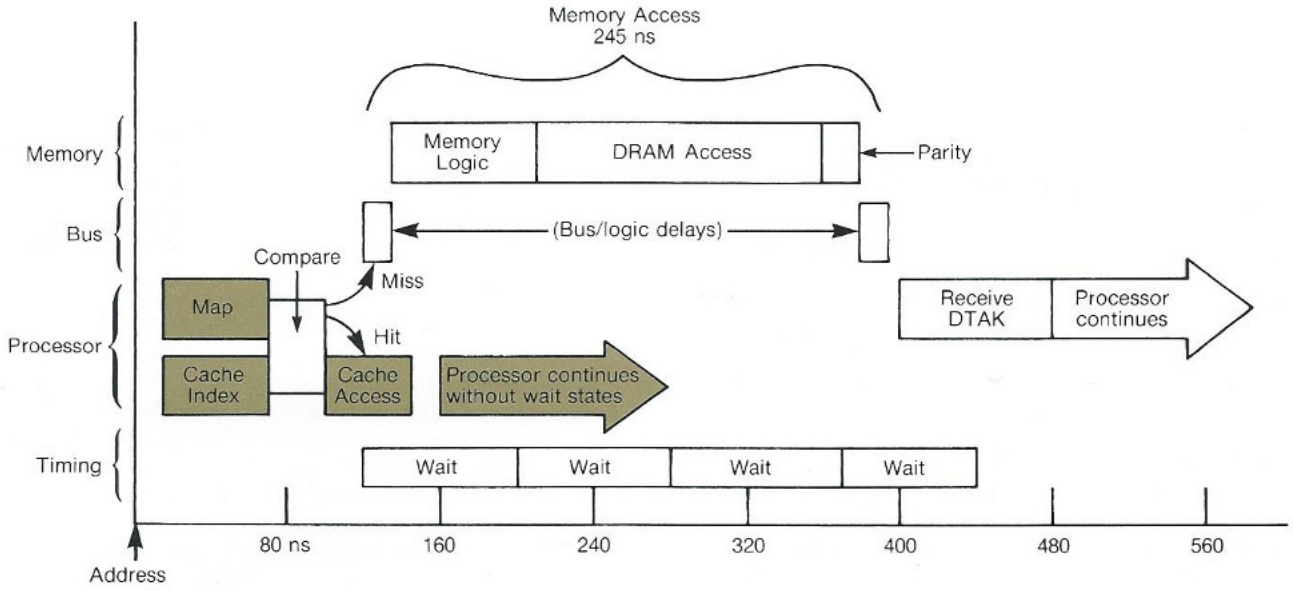
The use of VLSI technology keeps the processor on a single board, which reduces the cost of parts, manufacturing, and packaging. The resulting lower level of interconnection is also an advantage in reliability and maintenance.

	Universe 68/05	EXORmacs Processor	VAX 730	VAX 750	VAX 780
Cache Size	4 Kb	na	na	4 Kb	8 Kb
Cache Hit Rate	85%	na	na	90%	95%
Micro-cycle	80 ns	125 ns	270 ns	320 ns	200 ns
Main Memory Cycle	400 ns	425 ns	810 ns	640 ns	800 ns
Memory Width	32	16	32	32	64
MIPS	1.25	.42	.35	.7	1.5
DMA Rate (max.)	8.4 Mb	4 Mb	1.5 Mb	5 Mb	5 Mb
Bus Limit	20 Mb	20 Mb	1.5 Mb	13.5 Mb	13.5 Mb
Maximum Memory Capacity	3 Mb (5-slot chassis)	6 Mb	5 Mb	8 Mb	8 Mb
Physical Address Range	16 Mb	16 Mb	16 Mb	16 Mb	1 Billion
Number of CPU Boards required	1	2	3	3	12
Height of enclosure	7 inches	60 inches	30 inches	36 inches	60 inches
Basic Cost*	$10,000	$35,000	$40,000	$80,000	$100,000

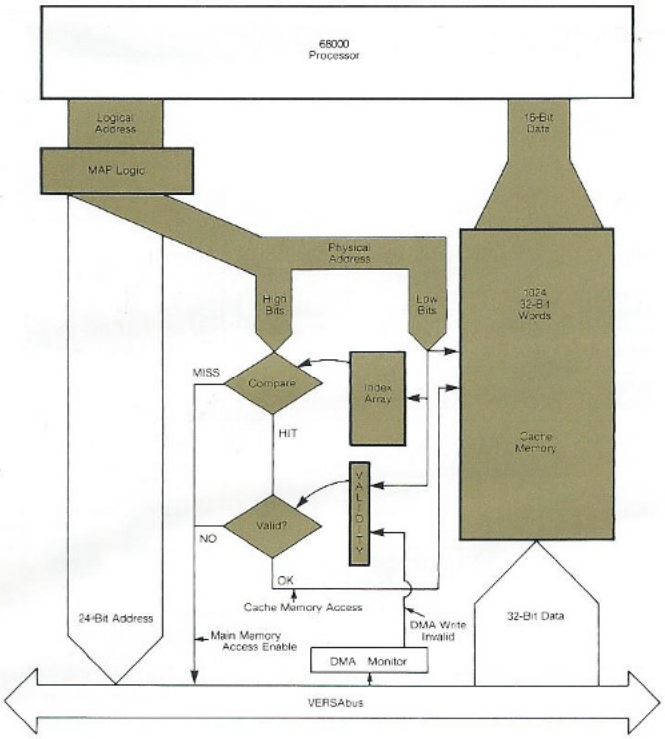
*Based on manufacturers' advertised prices.

Performance characteristics of the Universe 68/05 are compared with the DEC VAX product line and the Motorola EXORmacs processor. Cache hit rate contributes substantially to performance, and the "extra width" memory transfers of the VAX 780 and Universe 68/05 improve the hit rate.

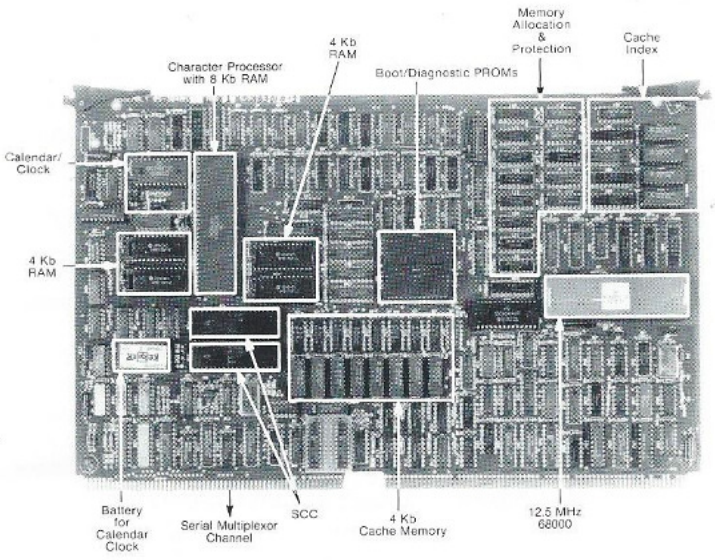
EXORmacs is a trademark of Motorola.



Memory timing is crucial to efficient processor operation. The 12.5MHz 68000 processor requires under 160ns access memory to operate at full speed. Main memory delays associated with mapping, bus access, and parity/ECC calculations prevent this. Accesses to cache on the processor board encounter only delays of mapping and accessing the cache (fast static RAM). This permits no-wait-state operation.



The CP32 cache interfaces the 32-bit VERSAbus to the 16-bit data pin structure of the 68000. Cache "hits" are detected by indexing an array of high-order physical address bits with the low-order address bits, and comparing them to the high-order bits of the address register.



The CP32 processor incorporates the 12.5MHz Motorola 68000 microprocessor. Cache memory provides data to the processor without wait states. A second 68000 controls 4 on-board serial ports, connected off the VERSAbus backplane. Memory references go through mapping logic, which translates a 24-bit logical address to a 24-bit physical address. 32-bit data is transferred between processor and main memory across the VERSAbus.

4 The Universe System

A natural outgrowth of Charles River Data Systems' ten years experience designing and manufacturing subsystems is the integration of the disk and memory for the Universe product line.

Each Universe system includes the CP32 processor, memory, main disk storage, and some form of removable media. The removable media is used for backing up the main disk, for off-line storage, and for interchanging software and data between systems. The memory boards support byte, 16-bit, and 32-bit data transfers with either byte parity or ECC protection.

Disk Packaging

The 68/05 and 68/35 are single box systems with 10- and 32-megabyte main disks. For users who require more disk storage, there are separate disk units. The 7-inch high Universe disk subsystem can accommodate a mix of mass storage devices: 8-inch Winchesters, floppy disks, 1/4-inch streaming tape units, and removable media Winchesters.

The Universe offers three types of removable media devices.

- floppy disks for small units and incremental backups;
- quarter-inch streaming cartridge tape drives for backing up larger devices;
- removable Winchester disk cartridges for quick-access off-line data storage.

The removable Winchester disk cartidge satisfies the need for off-line storage, be it for backups or for alternate data bases. All removable disks operate on their own spindles, thereby providing independence from the main disk. A cartridge can be mounted, the required update or report generated and then the cartridge removed without impacting other users on the system. This is ideal for departmental or personal data bases, security control, and reduction of overhead on the fixed storage unit.

A convenient medium for exchange of data is a common OEM concern. The floppy disk, the most common exchange medium, best satisfies this need. The Universe 68 hardware can handle both single- and double-density diskettes.

The disk subsystem package includes a front panel that offers direct control and diagnostic information. Disk formatting can be done directly from the front panel without the need for connection to the host processor. Diagnostic checks, with status codes displayed on this panel, simplify error reporting and field maintenance procedures.

Multiuser Systems versus Workstations

Universe systems are designed to operate as multiuser systems or as cluster nodes in a local network. This flexibility is possible because of the Universe's high performance. With lower performance micro-systems there is no such choice; one processor can handle only a single user and cannot keep up with memory speeds. The Universe 68 CP32 central processor can put considerable demand on memory (thanks to the cache) and still handle a substantial number of concurrent tasks. Thus these resources can be shared among several users and thereby reduce the cost of the overall facility.

Work stations connected with local area networks, such as Ethernet, have a high cost. A workstation, which by definition is a single user system, costs from $5,000 to $35,000. To that must be added the costs of network connections, software, and shared resources (such as hard disk and the network cabling). As a result the average cost per terminal in a workstation environment comes to $8,000 or more. A multiuser system, on the other hand, can support several terminals at a cost per terminal of $2,000 to $4,000, depending on the configuration. In addition, special interfaces, such as closely coupled array processors, can be accessed by all users on a multiuser system. This is another example of cost reduction through resource sharing.

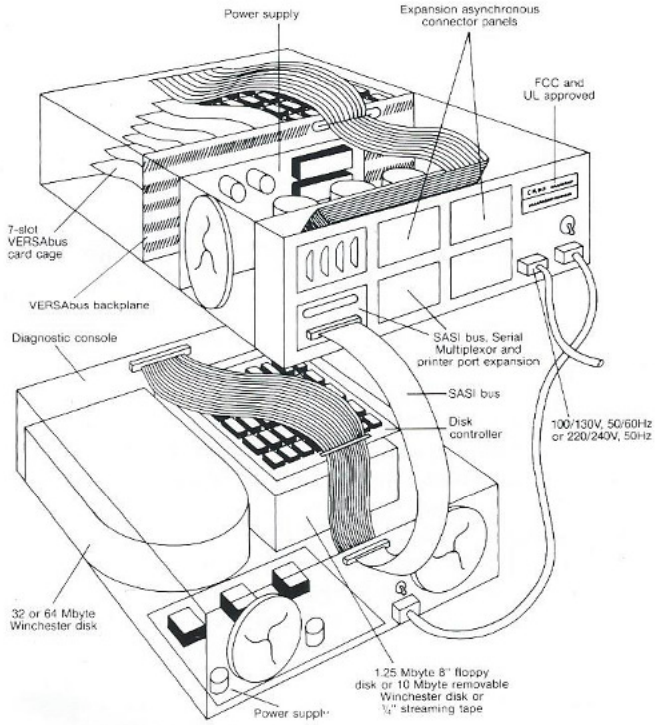
In short, multiuser systems provide lower cost per terminal and permit more users access to the system than workstations. In addition, multiuser systems can be connected in local area networks, thus reaping the advantages of both approaches.

System Configurations

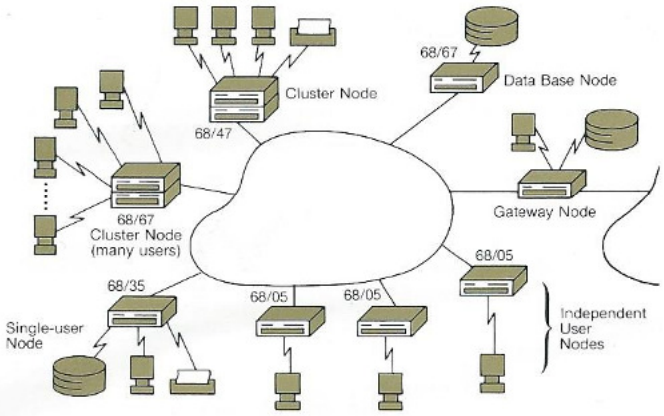
	68/05	68/35T **	68/37	68/47	68/67T **
Chassis slots	5	5	7	7	7
Max. Main Memory	3 Mb	3 Mb	5 Mb	5 Mb	5 Mb
I/O* Ports	4 to 64	4 to 64	4 to 64	4 to 64	4 to 64
CPU	CP32 Single board	CP32 Single board	CP32 Single board	CP32 Single board	CP32 Single board
Available memory increments	256 or 512 Kb or 1 Mb	256 or 512 Kb or 1 Mb	256 or 512 Kb or 1 Mb	256 or 512 Kb or 1 Mb	256 512 Kb or 1 Mb
Selector Channel	yes	yes	yes	yes	yes
SMD disk control	optional	optional	optional	optional	standard
Main disk (fixed/ formatted)	10 Mb	32 Mb	32 Mb	32 Mb	64 Mb
Backup medium	1.26 Mb floppy	45 Mb ¼ inch streaming tape	1.26 Mb floppy	10 Mb removable Winchester	45 Mb ¼ inch streaming tape
Package height	7 inches	7 inches	14 inches	14 inches	14 inches

*Expansion beyond 12 ports requires the addition of a communication chassis.
**Available Summer 1983.

Universe 68 systems are available in a variety of configurations. Principal distinguishing characteristics include chassis size, number of available chassis slots, and configuration of main and backup disks.



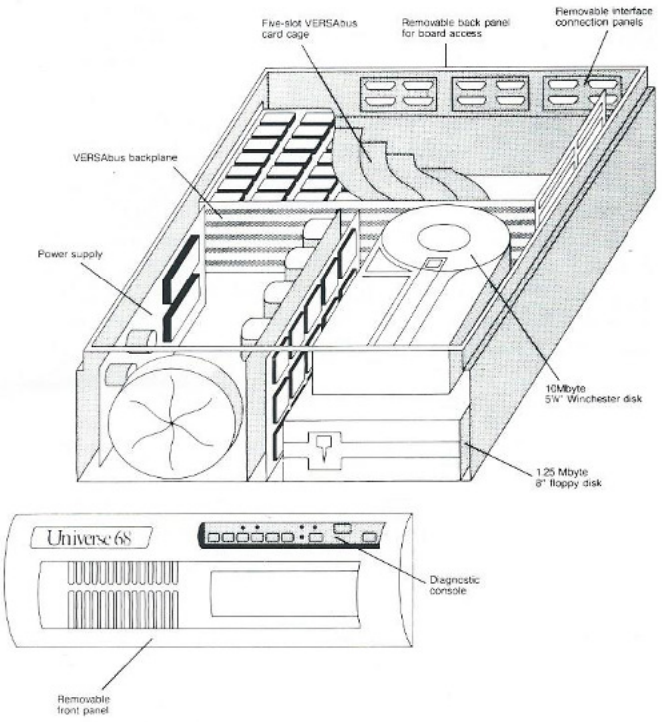
The Universe 68 double-height enclosure includes, in a 14-inch package, central processor, main disk (up to 64Mb), backup disk or streaming tape, and 7-slot VERSAbus card cage. Basic 256Kbyte memory can be expanded in 256-Kbyte, 512-Kbyte, or 1-megabyte increments (up to 5Mb). Removable interface connection panels ease configuration of application systems.



High performance Universe systems operate as multiuser systems or cluster nodes in a local network. Resources can be shared, reducing overall cost. A multiuser Universe system can support several terminals at $2,000 to $4,000 per terminal. A single workstation, including support facilities, can be 2 to 4 times more costly.

	Add-on Subsystems		Backup Drives		Integrated Systems		
	DK32R Drive	DK64T Drive	10 Mb Removable Winchester	Floppy Drive	68/05	68/35T	SMD Drives
Universe 68 Interface	Selector Channel	Selector Channel	Selector Channel	Selector Channel	Selector Channel	Selector Channel	Optional
Formatted Storage	32 Mb	64 Mb	10 Mb	1.26 Mb	10 Mb	30 Mb	80-300 Mb
Average Access	60 ms	28 ms	60 ms	150 ms	93 ms	45 ms	46 ms
Transfer Rate	540 Kb/sec	1.2 Mb/sec	874 Kb/sec	62 Kb/sec	625 Kb/sec	625 Kb/sec	1.2 Mb/sec
Expected Accesses/sec per Spindle	16 (2 spindles)	35	76	6	11	22	22
Removable Backup	10 Mb Winchester (built-in)	45 Mb ¼ inch streaming tape	yes	yes	1.26 Mb 8 inch floppy	45 Mb ¼ inch streaming tape	For comparison

Disk storage for Universe systems comes in a variety of formats. DK32R and DK46T disk subsystems are add-on units. The 10Mb removable Winchester and the floppy drive are backup drives (not available separately).Universe 68/05 and 6835T are integrated systems. SMD drives are shown for comparison.



The Universe 68/05 includes in its 7-inch enclosure a 10-Mbyte, 5 ¼-inch Winchester disk, 1.25-Mbyte, 8-inch floppy, and 5-slot VERSAbus card cage. Basic 256 Kbyte memory can be expanded in 256-Kbyte, 512-Kbyte, or 1-megabyte increments (up to 3Mb).

5 UNOS: A UNIX-Inspired System

UNIX and its predecessor, MULTICS, have become the models, since 1976, for all advanced operating systems including DEC's VMS, DG's AOS, HP's MPE, and Prime's PRIMOS. UNIX and UNIX-compatible systems, however, are more than just a source for good ideas; they comprise a set of compatible systems that encompass multiple processors from multiple vendors. UNIX's widespread use and the technical advantages of the UNIX concepts have led to the formation of standards for UNIX-compatible systems. Charles River Data Systems, having independently developed UNOS, has been involved in setting and adhering to these standards.

Common to all UNIX-like systems are powerful file system and process management facilities. Concepts such as multiprocessing, device independent I/O, extensible files, hierarchical directory structure, and I/O redirection are all part of this heritage.

Pipes

The real key to the current wave of UNIX popularity is a concept called *pipes*. Pipes allow the output of one program to be used as the input to another program without creating intermediate disk files.

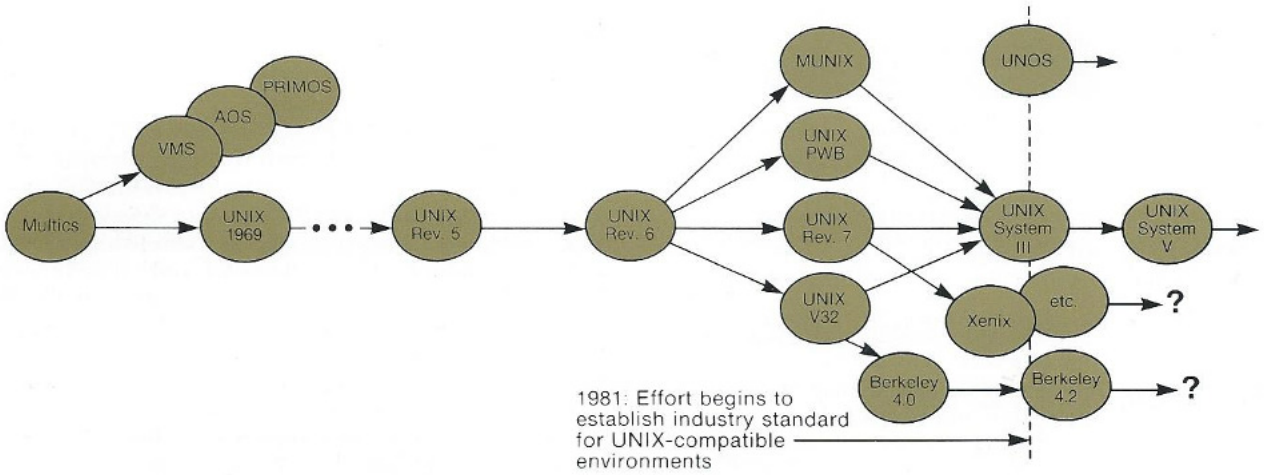
Although pipes are similar to connecting programs in a Batch Stream (or Multi-Pass Process Stream) via temporary files, pipes have two advantages. First, the processes run in parallel, producing results as they come down the pipe, thereby permitting interactive operations through pipes. Second, pipes require no temporary file storage — the system manages this channel with an implicit buffer mechanism.

Pipes permit the use of programs as filters or tool programs. Because most utility programs were written to perform general functions, when they are linked together in a pipe they serve as tools. Thus they can perform the work normally done by a complex program. With no programming effort at all, regular system programs can be combined to do many functions. And, by adding other simple filter programs, useful applications can be quickly assembled by combining unique application functions with standard tools.

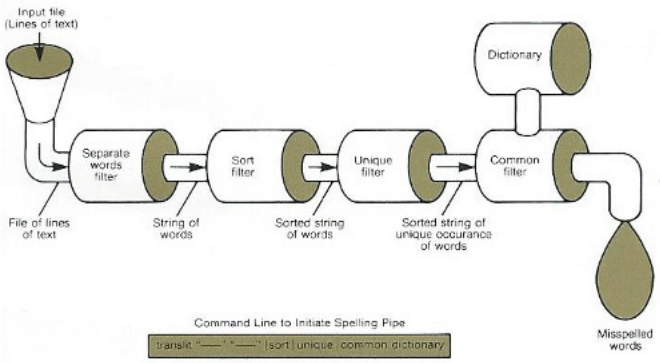
An additional benefit is software reliability — both in the tools used and the input/output data. Since many of the components of a 'pipe' application are based on existing modules, there is less actual development work and therefore fewer bugs. Since data scripts are used as input, test cases for simple questions and answers can be written. Similarly, intermediate results can be collected into files to check the program's progress at each step. Pipes immediately become a factor in developing applications. Data input processes can collect and pass data into a pipe, allowing easy connection to a diagnostic process. The analysis routines can be driven by the input routine or by stored data files. The tee command allows data to be "tapped" at one or more points to a simultaneous file with the image of the data at that point for later diagnostic evaluation, or for use as input for other tools.

Pipes have two implicit limitations. First, all processes in a pipe must be initiated by the same parent process. Second, the data streams are byte streams, with no status or identification. Thus, a server process cannot use pipe input to handle multiple processes or processes that were not connected to it at the beginning of its operation. In order to overcome these pipe limitations, UNOS provides a queue package for server-user situations (including multiple server models), eventcounts for process synchronization, and a number of extensions for production oriented environments. (See Section 7, "UNOS: A Real-Time System" for a description of how these work.)

VMS is a trademark of Digital Equipment Corporation.
AOS is a trademark of Data General Corporation.
MPE is a trademark of Hewlett-Packard.
PRIMOS is a trademark of Prime Computer.



UNIX was developed at Bell Laboratories in the late 1960s, an outgrowth of the MULTICS language developed at MIT. In the '70s and early '80s, UNIX dialects proliferated. Since 1981 there has been a movement towards standardization.



A spelling correction program illustrates the usefulness of UNOS pipes. Functions can be strung together, with the output of one serving as the input to the next function.

Tool	Function
cat	combine files into single stream
change	replaces patterns in stream
common	compare sorted stream and file on line by line basis
compare	compare stream and file at binary level
cut	dissect stream by field
diff	compare source stream and file on multi-line basis
echo	put argument data into stream
find	search directory tree for files that match a template
lpr	print stream on line printer
ls	list file directory contents to output stream
match	extract lines from stream with substrings that match pattern
paste	paste fields of stream and file together
pr	display stream as pages with header/page #
sort	sort stream (fields, numeric, etc.)
tee	output to specified file and stream at same time
translit	transliterate one character set to another
unique	remove duplicate entries (or unique ones) from stream

Some examples of tools designed for pipe operations. Pipes permit the use of programs as filters or tool programs. Because most utility programs were written to perform general functions, they can serve as tools when they are linked together in a pipe.

Features	UNOS Rev. 4	UNIX Rev. 6	UNIX Rev. 7	UNIX III	UNIX V
I/O redirection	yes	yes	yes	yes	yes
File independent I/O	yes	yes	yes	yes	yes
Pipe process management	yes	yes	yes	yes	yes
Multi-server queues	yes	no	no	no	no
Named pipes	yes	no	no	yes	yes
Signal/alarm facility	yes	ltd.	yes	yes	yes
Process synchronization	yes	no	no	no	ltd.
Multiplexed files	no	no	yes	no	no
Keyed file facilities	opt.	no	yes	no	no
DBMS facilities	opt.	no	no	no	no
Max. file size (bytes)	2^{31}	2^{24}	2^{30}	2^{30}	2^{30}
File name length (bytes)	30	14	14	14	?
File allocation	Bit map	Linked list	Linked list	Linked list	Linked list
Record/file locking	yes	no	no	no	no
Multi-level directories	yes	yes	yes	yes	yes
Bad block handling	yes	no	no	no	?
Exception handling	yes	no	no	no	no
Symbolic debugger	w/ Macros	ltd.	ltd.	yes	yes
Graphics packages	no	no	yes	yes	?
Text formatting	yes	yes	yes	yes	yes
Typesetting package	opt.	no	yes	yes	yes
Compiler writing aids	opt.	no	yes	yes	yes
Environments	yes	no	yes	yes	yes
Process locking	yes	no	yes	yes	yes
User controlled priority scheduling	yes	no	no	no	no
Heuristic timesharing scheduling	yes	yes	yes	yes	yes
Shared text	yes	yes	yes	yes	yes
Shared data	yes	no	no	no	no
User support	yes	no	no	no	ltd.
Update service	yes	no	no	no	ltd.

UNOS Rev. 4 incorporates most major features of the widely-used versions of UNIX. UNOS has real-time, transaction-oriented features not generally available in UNIX. UNOS offers support not previously available with UNIX. (UNIX V includes source code user support and updates.)

6 UNOS: UNIX Compatibility Plus UNIX System III Tools

UNOS was designed to satisfy the reliability and process management requirements for technical and commercial applications.

The key advantages of the UNOS system are:

- efficient process synchronization;
- real-time control of system scheduling, memory, devices;
- suitability for production environment as well as for development operations;
- high performance and reliable disk management through bit map allocation and bad block specification.

Compatibility

UNOS is both operationally and C application source code compatible with UNIX System III (as opposed to Rev. 6, PWB, V32, or Berkeley UNIX). Unfortunately, there are differences between all versions of UNIX, even those based on UNIX System III. Charles River Data Systems is currently participating in efforts to document a standard for portable UNIX software that will make it possible to measure the concepts of "compatibility" or "UNIX-like."

Generally, the question of portability is highly dependent on program type. The following types of programs are not easily ported to UNOS, and probably not portable to most other operating systems:

- programs that are hardware dependent (for example, a program that is dependent on the wordsize of the machine on which it runs);
- programs that access physical devices and/or device dependent characteristics;
- programs that use certain privilege and/or administrative commands (Superuser, PTRACE);
- programs that use special libraries (multiplexed files, graphics, extended-precision arithmetic, and the keyed file facility). In this regard, UNOS has its own extended precision arithmetic and DBMS packages.

UNIX System III Tools

UNOS provides a number of tools and filters comparable to those of UNIX System III. In addition, for users who want to work in the UNIX System III licensed environment (including programs such as scc, yacc, lex, lint, troff, and awk), a UNIX System III license is available as an optional package from Charles River Data Systems. In addition to providing specific software developed by AT&T, this license provides access to variations of that software developed by U.C.Berkeley and other suppliers. Customers can choose to use this option solely for development and to then distribute application packages to their customers with only the UNOS tool set, thereby minimizing their software product costs.

All UNOS commands have English names as the default convention (for example, **copy** for cp, **match** for grep, **move** for mv). In addition, an abbreviation facility enables renaming UNOS commands with UNIX names, or it can be used to specify a combination of commands. This facility requires no disk overhead to access the functions. Thus the environment can be tailored to match experience and needs of each user.

For a copy of the proposed standard for UNIX-compatible software, contact:

UNIX Standards Proposal
UNIFORUM
P.O. Box 8570
Stanford, CA 94305

UNIX Tool	Function
yacc	compiler-compiler
lex	lexographical scanner
awk	pattern matcher
lint	C type checker
troff	typesetting tool
C-shell (Berkeley)	Berkeley shell
Bourne shell	Bell UNIX shell
vi/termcaps	screen editor (Berkeley)
cu/uucp	intersystem asynchronous communications

A partial listing of UNIX tools available from Charles River Data Systems under Bell Laboratories license which can be used under UNOS. The license covers UNIX tools, including those provided by U.C. Berkeley.

UNIX Tool	UNOS Tool	Function
adb	debug	UNOS debug runs as separate process with many extensions
cat	cat	combines multiple files into one output stream
cd	cd	change your current working directory
cmp	compare	compare two files at binary level
comm	common	compare sorted lists on line by line basis
cp	copy	copies a file, or set of files
diff	diff	compare source files on multi-line basis
du	diskusage	display report on disk space utilization
echo	echo	put argument data into stream
find	find	search directory tree for files that match a template
grep	match	extract lines with substrings that match pattern
ld	link	combine separately combined modules into a load module
ln	addname	add a second name to an existing file
lpr	lpr	print stream on line printer
ls	ls	list file directory contents
mail	mail	electronic mail facility, UNOS has extended facilities
make	make	build an execution image, re-compiling modified modules
mount	mount	mount a logical file system
mv	move	move files to new directory
pr	pr	display stream on terminal, with interactive page breaks
ps	ps	display report of current active processes
pwd	pwd	display pathname to current working directory
rm	delete	files (UNOS has 'verify' mode)
roff	format	format data for printing: justification, centering, etc.
sort	sort	sort stream (fields, numeric, etc.)
sync	sync	flush cache of modified disk blocks to disk
tee	tee	output to specified file and stream at same time
tr	translit	transliterate one character set to another
umount	unmount	remove a logical file system
uniq	unique	remove duplicate entries (or unique ones) from list
who	who	display report of logged on users

UNOS tools compatible with UNIX tools. UNOS comes with a long list, part of which is included above, of tools which are analogous to tools found in UNIX. UNOS generally uses English words as tool names, while UNIX uses often obscure mnemonics.

Subroutine or System Call	Function in UNOS Rev. 4	Meets or exceeds proposed standard
ARITHMETIC Functions	Integer functions, random numbers, bessel functions, trig, hyperbolic, and exponentiation.	yes
DATA CONVERSION ROUTINES AND FORMATTING	Conversion between various data types: ASCII, floating point, integer, quad precision (UNOS extension). Also input and output formatting routines for various types.	yes
INPUT/OUTPUT FILE AND PIPE CONTROL	Control functions, file creation, deletion, changing characteristics, input, output, status, as well as control over terminal characteristics. Also, directory control, creation, deletion, and positioning.	yes
STRING AND CHARACTER PROCESSING	Routines for classifying characters, string manipulation such as indexing into string, concatenation, comparison, and sorting.	yes
PROCESS AND MEMORY MANAGEMENT	Memory allocation and information, process status & information about id's, password, names, and changes to these. Also control over process creation (fork), execution, or termination. Also, the ability to lock processes resident in memory, priority scheduling, and shared memory control are UNOS extensions.	yes
	The UNOS system eventcount mechanism and queuing facilities are not available with UNIX.	yes
TIME ACCESS AND CONVERSION ROUTINES	Routines to set the time, get the time, and convert the time into various formats, and time zones.	yes
FILE/RECORD LOCKING	Per proposed standard	yes
NAMED FIFO PIPES	Per proposed standard	yes
GROUPS	Access control mechanism	yes

UNOS Rev. 4 provides the subroutines in these areas that are considered "standard," in that all of them meet or exceed the proposed standard for UNIX-compatible environments.

6 UNOS: UNIX Compatibility Plus UNIX System III Tools

UNOS was designed to satisfy the reliability and process management requirements for technical and commercial applications.

The key advantages of the UNOS system are:

- efficient process synchronization;
- real-time control of system scheduling, memory, devices;
- suitability for production environment as well as for development operations;
- high performance and reliable disk management through bit map allocation and bad block specification.

Compatibility

UNOS is both operationally and C application source code compatible with UNIX System III (as opposed to Rev. 6, PWB, V32, or Berkeley UNIX). Unfortunately, there are differences between all versions of UNIX, even those based on UNIX System III. Charles River Data Systems is currently participating in efforts to document a standard for portable UNIX software that will make it possible to measure the concepts of "compatibility" or "UNIX-like."

Generally, the question of portability is highly dependent on program type. The following types of programs are not easily ported to UNOS, and probably not portable to most other operating systems:

- programs that are hardware dependent (for example, a program that is dependent on the wordsize of the machine on which it runs);
- programs that access physical devices and/or device dependent characteristics;
- programs that use certain privilege and/or administrative commands (Superuser, PTRACE);
- programs that use special libraries (multiplexed files, graphics, extended-precision arithmetic, and the keyed file facility). In this regard, UNOS has its own extended precision arithmetic and DBMS packages.

UNIX System III Tools

UNOS provides a number of tools and filters comparable to those of UNIX System III. In addition, for users who want to work in the UNIX System III licensed environment (including programs such as scc, yacc, lex, lint, troff, and awk), a UNIX System III license is available as an optional package from Charles River Data Systems. In addition to providing specific software developed by AT&T, this license provides access to variations of that software developed by U.C.Berkeley and other suppliers. Customers can choose to use this option solely for development and to then distribute application packages to their customers with only the UNOS tool set, thereby minimizing their software product costs.

All UNOS commands have English names as the default convention (for example, **copy** for cp, **match** for grep, **move** for mv). In addition, an abbreviation facility enables renaming UNOS commands with UNIX names, or it can be used to specify a combination of commands. This facility requires no disk overhead to access the functions. Thus the environment can be tailored to match experience and needs of each user.

For a copy of the proposed standard for UNIX-compatible software, contact:

UNIX Standards Proposal
UNIFORUM
P.O. Box 8570
Stanford, CA 94305

UNIX Tool	Function
yacc	compiler-compiler
lex	lexographical scanner
awk	pattern matcher
lint	C type checker
troff	typesetting tool
C-shell (Berkeley)	Berkeley shell
Bourne shell	Bell UNIX shell
vi/termcaps	screen editor (Berkeley)
cu/uucp	intersystem asynchronous communications

A partial listing of UNIX tools available from Charles River Data Systems under Bell Laboratories license which can be used under UNOS. The license covers UNIX tools, including those provided by U.C. Berkeley.

UNIX Tool	UNOS Tool	Function
adb	debug	UNOS debug runs as separate process with many extensions
cat	cat	combines multiple files into one output stream
cd	cd	change your current working directory
cmp	compare	compare two files at binary level
comm	common	compare sorted lists on line by line basis
cp	copy	copies a file, or set of files
diff	diff	compare source files on multi-line basis
du	diskusage	display report on disk space utilization
echo	echo	put argument data into stream
find	find	search directory tree for files that match a template
grep	match	extract lines with substrings that match pattern
ld	link	combine separately combined modules into a load module
ln	addname	add a second name to an existing file
lpr	lpr	print stream on line printer
ls	ls	list file directory contents
mail	mail	electronic mail facility, UNOS has extended facilities
make	make	build an execution image, re-compiling modified modules
mount	mount	mount a logical file system
mv	move	move files to new directory
pr	pr	display stream on terminal, with interactive page breaks
ps	ps	display report of current active processes
pwd	pwd	display pathname to current working directory
rm	delete	files (UNOS has 'verify' mode)
roff	format	format data for printing: justification, centering, etc.
sort	sort	sort stream (fields, numeric, etc.)
sync	sync	flush cache of modified disk blocks to disk
tee	tee	output to specified file and stream at same time
tr	translit	transliterate one character set to another
umount	unmount	remove a logical file system
uniq	unique	remove duplicate entries (or unique ones) from list
who	who	display report of logged on users

UNOS tools compatible with UNIX tools. UNOS comes with a long list, part of which is included above, of tools which are analogous to tools found in UNIX. UNOS generally uses English words as tool names, while UNIX uses often obscure mnemonics.

Subroutine or System Call	Function in UNOS Rev. 4	Meets or exceeds proposed standard
ARITHMETIC Functions	Integer functions, random numbers, bessel functions, trig, hyperbolic, and exponentiation.	yes
DATA CONVERSION ROUTINES AND FORMATTING	Conversion between various data types: ASCII, floating point, integer, quad precision (UNOS extension). Also input and output formatting routines for various types.	yes
INPUT/OUTPUT FILE AND PIPE CONTROL	Control functions, file creation, deletion, changing characteristics, input, output, status, as well as control over terminal characteristics. Also, directory control, creation, deletion, and positioning.	yes
STRING AND CHARACTER PROCESSING	Routines for classifying characters, string manipulation such as indexing into string, concatenation, comparison, and sorting.	yes
PROCESS AND MEMORY MANAGEMENT	Memory allocation and information, process status & information about id's, password, names, and changes to these. Also control over process creation (fork), execution, or termination. Also, the ability to lock processes resident in memory, priority scheduling, and shared memory control are UNOS extensions.	yes
	The UNOS system eventcount mechanism and queuing facilities are not available with UNIX.	yes
TIME ACCESS AND CONVERSION ROUTINES	Routines to set the time, get the time, and convert the time into various formats, and time zones.	yes
FILE/RECORD LOCKING	Per proposed standard	yes
NAMED FIFO PIPES	Per proposed standard	yes
GROUPS	Access control mechanism	yes

UNOS Rev. 4 provides the subroutines in these areas that are considered "standard," in that all of them meet or exceed the proposed standard for UNIX-compatible environments.

7 UNOS: A Real-Time System

UNOS can be configured as a real-time operating system. Although real-time operations are associated with speed, speed is relative to the base hardware capabilities. The essential requirement of real-time systems is concurrency with priority-oriented control. Data structures within the system must be kept up-to-date with the external conditions and as current as the decisions they must support. UNOS provides the tools needed for synchronization, prioritization, and allocation of critical resources.

Key real-time concepts implemented in the UNOS operating system include:

- resident locking of processes, ensuring performance and control of memory;
- preemptive priority scheduling, giving control of the CPU resource to the highest priority "ready" process;
- user-controllable priorities, including min/max with automatic heuristic adjustment if enabled;
- user-defined device drivers for custom peripherals and control over physical devices;
- disk file blocking for high performance disk I/O, permitting multi-block transfers on loading/swapping and minimizing head seek times;
- shared data segments for efficient message passing and information exchange between multiple tasks;
- context switching as low as 650 microseconds between processes on the CP32 processor;
- interrupt service entry as low as 30 microseconds on the CP32;
- generalized exception processing, permitting a process to recover or gracefully terminate from error conditions.

Eventcounts

The essence of real-time control is a scheduling and synchronization mechanism. It must be sophisticated enough to permit reliable control in a complex concurrent operations environment. In UNOS, at both the kernel and the user level, all scheduling and synchronization are controlled by eventcounts. Eventcounts are dynamically defined resources, which can be opened by unrelated processes and used concurrently by many processes. Initially developed to meet the needs of network synchronization, they are equally suited for the more traditional semaphore or message situations of single processor systems.

Eventcounts occupy single file descriptors (FDs) in the file system and benefit from such mechanisms as pathnames and protection modes. But while in use they reside in memory, providing maximum efficiency. When no longer in use, their values are retained on disk for future reference or for continuing applications use, such as data base transaction sequencing.

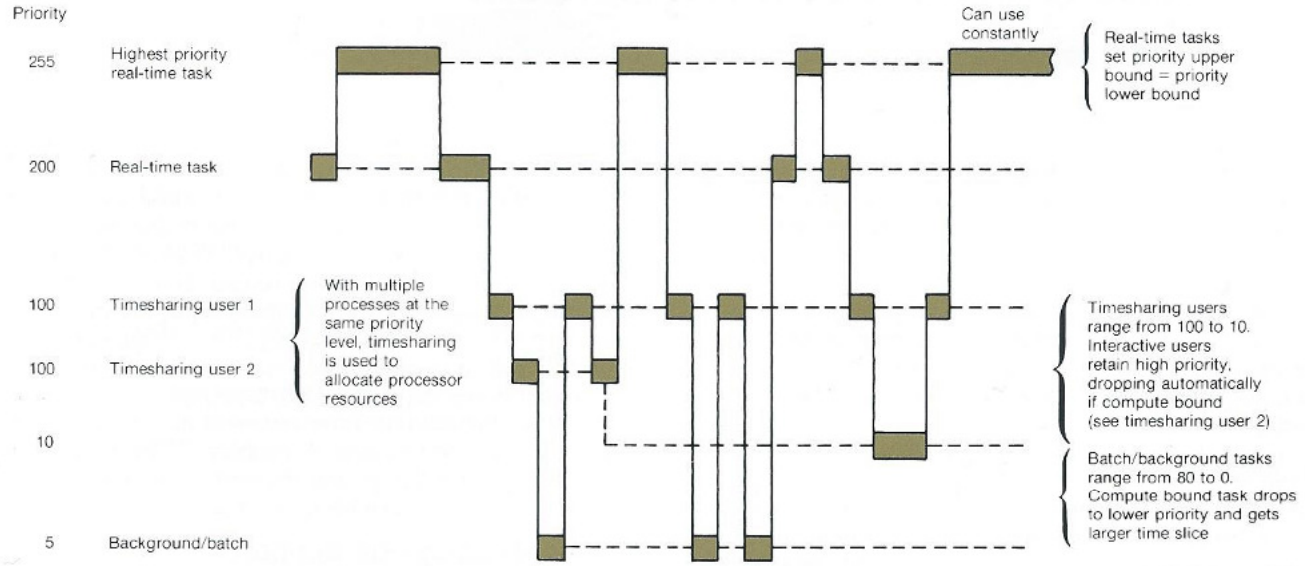
Eventcount Implementations

Eventcounts can be used as counters. Unlike semaphores, which have a value of zero or one, eventcount values monotonically increase. Therefore, they can serve as both an indication of event transitions and as a record of the number of occurrences of an event. This can be important in monitoring events associated with low priority processes, which may not get control until multiple occurrences have been registered. Eventcounts can also be used in situations where there are multiple processes monitoring a single resource. And, unlike synchronization with signals, the accuracy of eventcount implementation does not depend on a process's ability to constantly 'catch' its current value. (These operations are not possible using the standard UNIX signal mechanism.)

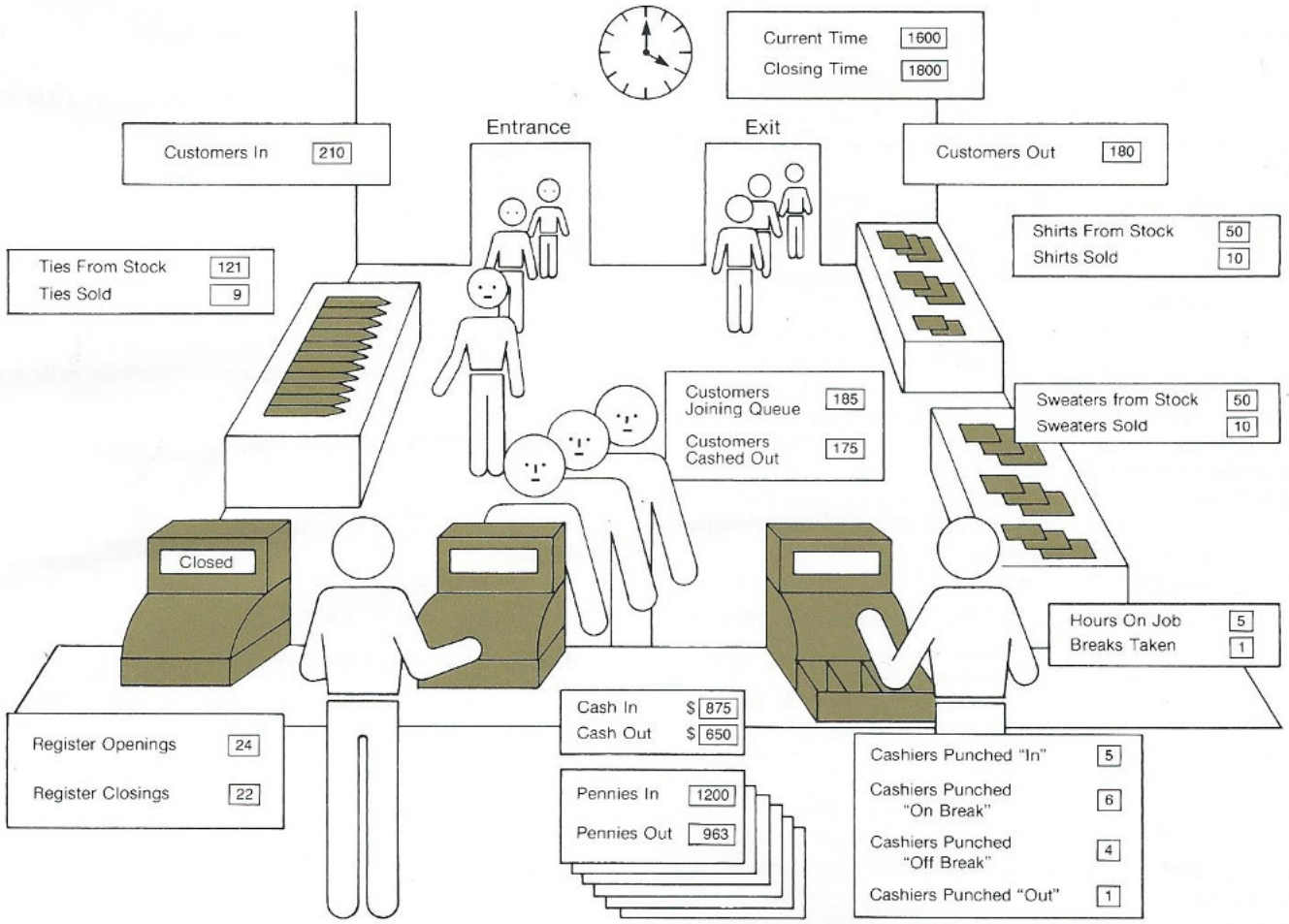
Eventcounts can be used in a sequence or server relationship to provide a sophisticated synchronization facility. Taking advantage of the fact that every user-created eventcount occurs as a pair, the first eventcount can be used to synchronize access to the critical resource, while the second eventcount can be used to give each contender a unique number. Access to the critical resource is then ordered by number. Each user takes a "ticket" and receives the current value of the sequencer (second) eventcount. The user then waits for this value to be reached in the server counter. As each user is served, the server eventcount is advanced so that the next process can have access to the critical resource. The two counters can also be used to determine the number of processes waiting for service — the difference between the first and second counter. The server counter can be used as a historical record of the number of users served, and can be monitored to determine average service rates. The ticket counter can be monitored to determine arrival rates.

Eventcounts can be used for inter-process communication. One process can wait for one or several events. Waiting on several events permits timeouts to be added easily and allows one process to serve multiple eventcounts. If the value waited for has passed, the waiting process is notified, thus processing can continue undelayed.

Since an eventcount is not owned by a single process, an independent process can monitor a range of key events, looking for conditions unique to a specific application. These conditions can be overall timing constraints, the measurement of key processing rates, profiling statistical information, or the detection of complex deadlock or error conditions. The monitor process could also act as a guide for maintenance and scheduled service activities, such as backing up a data base.



A multi-level, highly dynamic priority scheduling system distinguishes real-time UNOS from UNIX and many UNIX-like operating systems. This system allows UNOS to respond to the demands of real-time tasks as well as satisfying the needs of multiple users and background/batch jobs.



Eventcounts are an essential feature of UNOS, pervading virtually all areas of operation. An analogy might be drawn between a UNOS program and a department store. Both are complex organisms in which practically every element constantly keeps track of a number of "counts," many of which are interdependent. In the store, for example, a cashier keeps track of "eventcounts" including cash on hand, customers in queue, customers in store, hours at post, breaks taken, and clock time. Supervisors monitor these eventcounts as well as number of cashiers on duty, cashiers on break, etc.

8 UNOS: Software Development System and Language Set

For an operating system to be a good development system two sides of the development process must be addressed: software creation and on-going software management and maintenance. The operating system should facilitate easy writing and debugging of code. Once the programs are running, all subsequent changes — updating, recompiling and informing other users of changes — should be simple. UNOS fulfills these requirements by providing substantial development and management tools, and support for commonly used languages.

Creation

UNOS provides an easy-to-use interface to the system. The *command interpreter*, with its English language command set, provides a natural interface to the system. Moreover, with the abbreviation facility, commands can be renamed and/or combined to create an environment that matches personal tastes.

For entering programs, UNOS provides a terminal independent screen-based *editor*. It has a rich command set and interfaces nicely with the UNOS *text formatter* so that documentation can be written and formatted using the same tools.

UNOS supplies a complete *library of subroutines*, including both UNOS- and UNIX-compatible routines. There are also tools for creating additional libraries based on in-house routines.

For monitoring and analyzing running programs, UNOS provides an interactive *debugger*. It operates as a separate process and requires no prior special compiler switches or linkage procedures. In addition, the debugger provides a unique single-stepping facility with full symbolic debugging. It provides the capability for defining local and global symbols, tracing back through the stack, and creating macro display commands. The default exception handler causes a process to suspend when an unexpected condition occurs. The debugger can then be attached to the suspended process, and an analysis made of the program in the error context. Processes can be suspended and inspected with a debugger and resumed, simplifying the debugging of new software.

Management

For times when programs must be combined in different ways or when new features must be added, UNOS provides tools for ongoing program management.

An automatic *version* time stamping allows the user to easily determine revision levels. A source *comparison* program reveals at a glance where two programs differ. An on-line log provides a maintenance history of all changes to existing software. A *calltree* program generates a list of all references across multiple storage modules, so that the impact of changes through multiple levels of routines can be traced.

A *make* utility permits all the components and dependencies involved in creating a program to be defined. The effort needed to build modules is reduced, because the make facility can compile just the modules that have been modified and link them together into a binary image. This utility looks to the lowest level dependencies, verifies that these have been updated, or performs the specified update functions to reach this state. It then continues up the tree until the top level module is complete. Since all modules are checked for their level of update, a program is always operating at the most recent level. The program does not mix out-of-date versions, which could introduce revision inconsistency bugs.

Finally, a sophisticated *electronic mail* program encourages communication between developers, and reporting and responding to problems. The UNOS mailer provides many ways of sending and responding to messages and can also be used as a personal log.

Optional Language Support

C — The UNOS operating system language

This matches the 'C' language as described in the book: *The C Programming Language* by Dennis Ritchie and Brian Kernigan.

PASCAL — An ANSI Standard Compiler

UNOS PASCAL is implemented in accordance with the ANSI standard for PASCAL, with extensions for strings, random access to files, and separate compilation of modules following the UCSD PASCAL model.

FORTRAN — A full FORTRAN 77 compiler

UNOS FORTRAN implements the full ANSI 77 specification for the FORTRAN language, with extensions for character processing that are compatible with ANSI 66 implementations of FORTRAN.

COBOL — The popular Ryan-McFarland COBOL

This provides a COBOL with extensions for screen-oriented display and data entry, as well as compatibility with the wide range of systems and applications code that has been developed with RM/COBOL.

BASIC — An interpreter modeled on RT-11 BASIC

UNOS BASIC provides an interactive/interpreter environment ideal for quick programming tasks, and calculations.

Key Functions	UNOS/C	UNOS/ Pascal	COBOL	FORTRAN	BASIC
Free format	yes	yes	yes	no	no
Block structured	yes	yes	ltd.	no	no
Data structures	yes	yes	yes	no	no
Strict type checks	no	yes	yes	ltd.	no
Floating point #	32 & 64 bit	32 bit	NA	32 & 64 bit	32 bit
Integers	16 & 32 bit	16 & 32 bit	32 digit	16 & 32 bit	32 bit
Strings	yes	yes	yes	yes	yes
Bit manipulation	yes	(using Sets)	no	yes	no
Global data initialization	yes	no	yes	yes	yes
Separate compilation modules	yes	yes	yes	yes	yes
Random file access	yes	yes	yes	yes	yes
Exception processing	yes	ltd.	yes	no	no
Call for C subroutines	yes	*	no	*	no
Call for Pascal routines	no	yes	no	yes	no
Call ASM routines	yes	yes	no	yes	no
Can produce standalone code	yes	yes	no	yes	no

*— this can be done by connection to the UNOS C language environment using the separate compilation facility and ASM language interface.

The programming languages available under UNOS. The features listed are usually considered important for quality software engineering.

Tool	Function
help	online system assistance—users may extend as needed
ved	screen oriented editor
format	documentation formatting aid
log	software revision maintenance aid
change	change patterns in files
make	execute command sequences needed to build a program module
cc	compile modules
version	display current revision (date/time maintained by link...)
link	combine multiple modules into a single execution module
genlib	generate library entries from object program
archive	merge object modules for efficient link processing
notify	run program in background and report completion status
(^y)	suspend process to debug, kill or move to background
resume	resume execution of a suspended process
debug	interactive, symbolic debugger can be attached to any task
calltree	show C program call structure and generate cross-reference
move	move multiple files to target directory
diskusage	show current disk utilization
ps	show current job status

UNOS system tools for software management make the internal structure of a program visible and easily accessible, make the impact of changes easily traceable, and allow programmers to modify and manipulate programs in easily defined modules.

9 UNOS' Nucleus Data Base Management System

A variety of optional facilities are available with UNOS to provide the tools for effective application development. These include most common languages and the UNOS Nucleus DBMS facility.

Nucleus DBMS

UNOS NDBMS provides the essential elements for efficient data base management. Through its C-callable subroutines and commands, UNOS NDBMS has the key facilities needed to write and maintain data base oriented application programs. Nucleus DBMS is designed to meet the needs of application OEMs for a low-overhead file management facility, rather than the generalized query-language/report-generator facilities associated with end-user DBMS products.

UNOS NDBMS features a full range of database file management facilities, including data independence, multi-keyed ISAM, automatic index maintenance, atomic transaction definition, logging, backout, and audit trail, and a comprehensive backup and recovery facility.

Schemas, Records, Fields, and Keys

A database is designed by creating a schema. Schemas contain definitions of fixed length structures called records, detailing the name and characteristics of the fields in each record.

A single process uses one or more records. However, the relationship between records is not restricted or defined by the schema definition.

As part of the schema design, any field can be specified as an index key. This allows for multiple indexes into a database. Plus, it provides keyed, relative, and generic key access (match leading bytes in key). Data formatting descriptors are used to extract fields and for formatting fields for simple report generation.

UNOS NDBMS also provides a sort facility for those instances when the sequence for access (or reporting) does not match an established index path. With the sort, a temporary index can be created using any fields of the record in either ascending or descending order, and in combination with other fields. Once created, this temporary index can be used in place of any other index for either keyed or relative access.

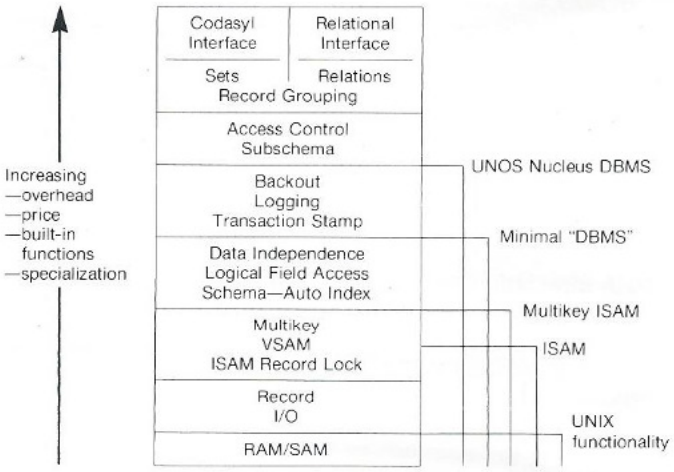
Data Independence

When it is necessary to change the data base structure, the system manager creates a new schema entry and transforms the old data base to the new. This may involve reordering fields, adding or deleting fields, providing new indexes or extracting a subset of the database. However, since UNOS NDBMS provides data independence, these changes will not require modification of programs using the database, only recompilation against the new schema.

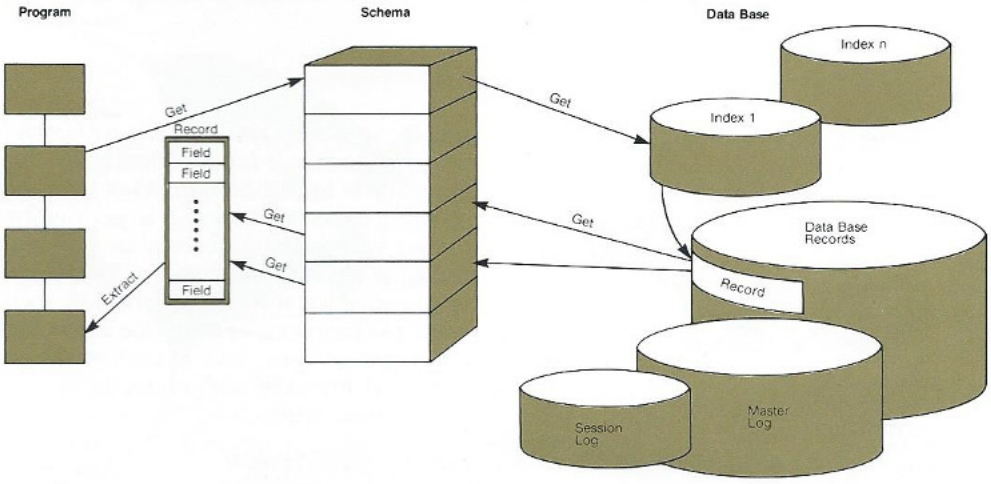
In addition, UNOS NDBMS provides automatic key maintenance, updating all index entries whenever a record is written. This is another aspect of data independence; the program does not need to know the structure of the database. The program allocates the necessary buffers dynamically using C-language subroutines. Other C subroutines provide record access and field extraction. Therefore, the size and location of the field within a record is not critical to the program. Other subroutines are built into UNOS NDBMS to format fields or records for output and to prompt on input.

Backup and Recovery

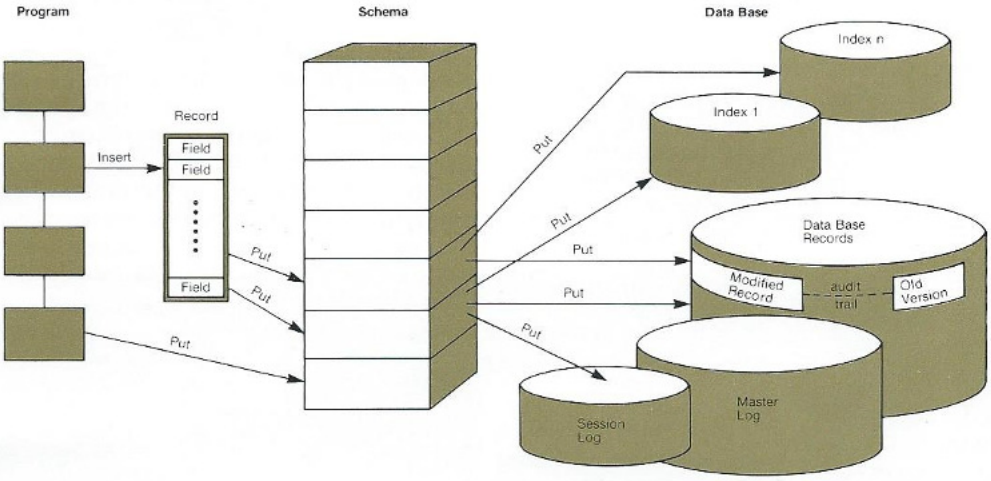
UNOS NDBMS uses atomic transactions as the basis of its backup and recovery facility. It provides the ability to lock records and, by nesting calls to initiate and commit subroutines, to combine complex transactions into atomic operations. An entire atomic transaction can be committed and the database fully updated, or it can be aborted and the database "rolled back" to its prior state. All committed transactions are logged as a part of the overall backup/recovery scheme. An audit trail is maintained, so the system manager can back out the transactions associated with abnormally terminated sessions using an administrative utility.



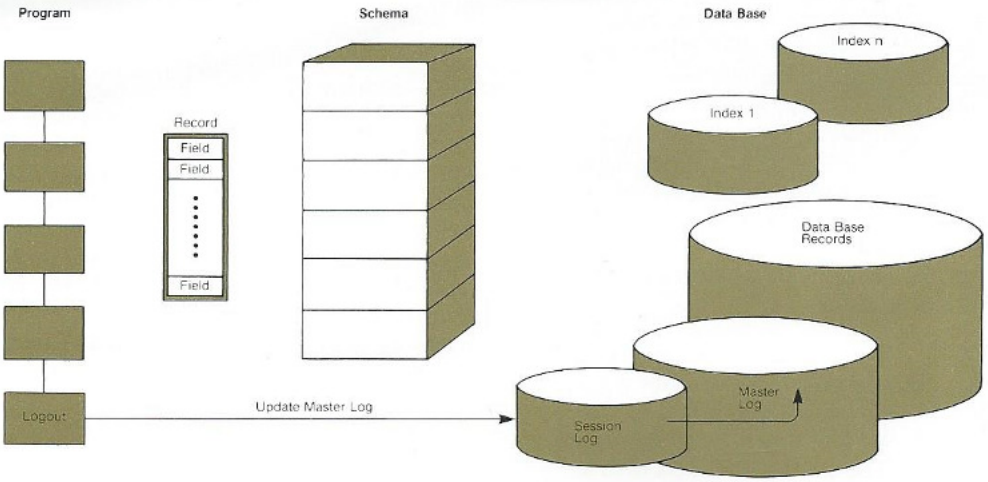
UNOS nucleus DBMS is designed for system integrators. It includes the facilities needed for efficient development and maintenance of database-oriented application programs, but does not include the generalized query-language/report-generator facilities associated with end-user DBMS products.



Database retrieval involves "getting" a record using one of the keyed fields and then extracting fields in a data-independent manner.



Updating a data record involves inserting a new field in a data-independent fashion and "putting" the record to the database with automatic index-updating and logging.



Terminating a session involves committing all the transactions entered and then merging the session log with the master log.

10 The Business Side: A Win-Win Proposition

When considering purchasing a computer system, three major questions arise:

- Are the hardware and software right?
- Is the price right?
- Is the vendor right?

If multiple systems are being purchased, the last question gains in significance. Since we have targeted the quantity purchaser as our market, we feel this last subject warrants discussion.

Charles River Data Systems' approach to doing business offers several business advantages:

- standard hardware and software products, which means multiple suppliers;
- quantity discount structures tailored to match the needs of start-up operations as well as major manufacturers;
- long-term software discounting;
- customer designed service programs.

Standard Products

Our success is based on customer confidence in our engineering development and on our ability to expand our product lines. This leads us to a unique position of offering unbundled systems. By following the industry standards in both software (C, Pascal, FORTRAN, COBOL, BASIC and a UNIX-compatible OS) and hardware (VERSAbus, SCSI/SASI bus, VME and Multibus interfaces) there are multiple sources for both main-line and peripheral products to complement the products we offer. This ability to mix and match permits our customers to utilize vertical integration where it makes sense in their businesses, and encourages a dialogue on how to best accomplish their objectives for both the short and long term.

Quantity Discounts

This same concept of matching customer needs and strengths extends to our contracts. For hardware, we have no-commitment discount arrangements that allow start-up projects to obtain attractive reductions as they buy more equipment with no bill-back risk. Customers with an established product flow find our quantity purchase agreement offers immediate discounts with adjustment to match the actual quantity purchased. This means retroactive discounts for equipment already purchased if quantities exceed those estimated. We also offer hardware licenses for situations in which it makes business sense for a customer to manufacture elements of a system. This flexibility allows an OEM to get applications into the market quickly, with minimal risk, and with immediate cash flow, and to later follow up with whatever make-or-buy strategy yields the best return.

Software Discounts

For Charles River Data Systems software, we offer a long-term license (beyond the end of this century) and discounting across that term. We feel this reflects our actual costs in software products. The costs associated with a specific customer drop license-by-license and year-by-year as experience and independence develop. Therefore, our prices drop in this fashion. Also, we charge just for the software used. We expect a larger number of our customers will produce systems for dedicated use, either in resale or internal applications. Therefore, we have a lower price for software involved in such run-time systems, as compared to the price associated with full development uses. And, of course, the software is supported by our staff, with corrections and updates offered on a regular basis.

Customer Service

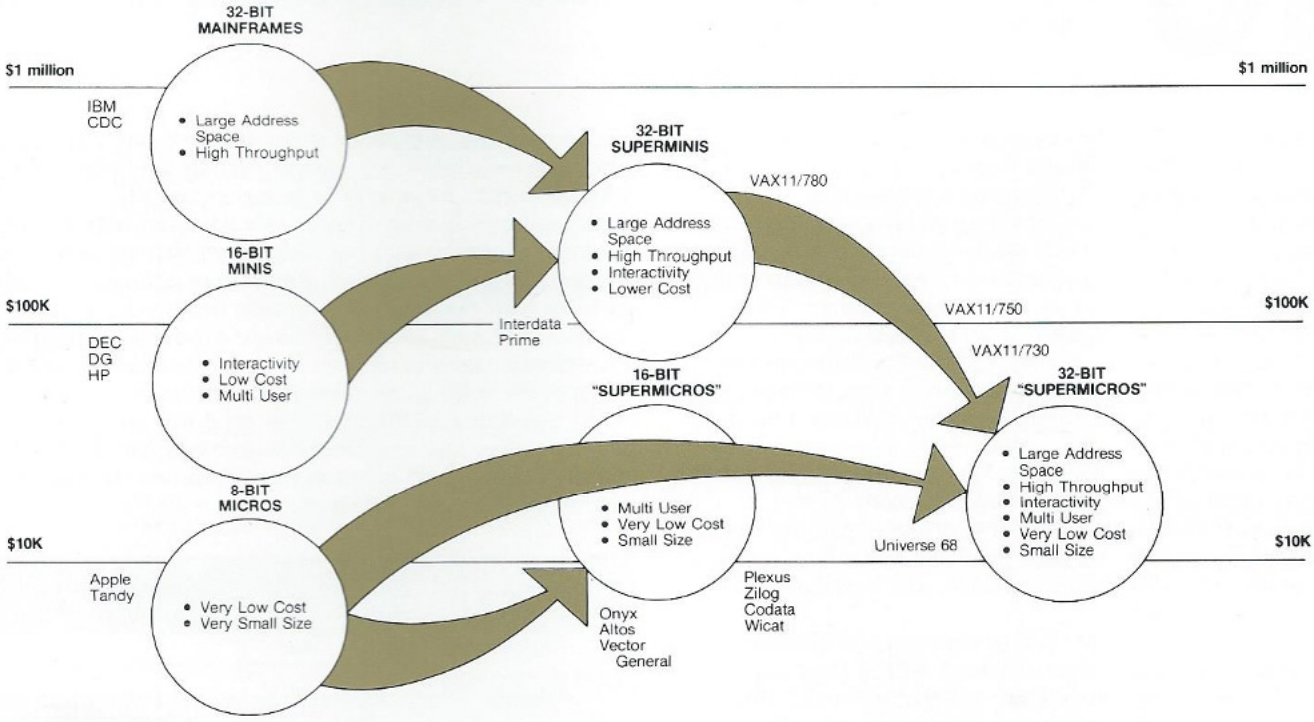
In addition to software support and update services, we have programs designed to match customer's hardware service requirements. Our equipment comes with error monitoring. Each system has built-in diagnostics, front panel display and on-board LEDs that help to quickly identify failure conditions and pin-point the necessary corrective action. Either the user, the OEM, or our third party maintenance partner can repair the problem. We work over the phone with the on-site person to quickly isolate trouble spots and start the response.

Actual repairs are done on a board level basis with exchange of boards. The approach used depends on the nature of the problem. In the most critical situations on-site spares or even a backup system may be warranted. The next level entails use of spares maintained by the OEM or third party service group in the local area. Same day air freight shipment of replacement boards from our service depot is possible on our express swap service contract. The user simply ships the failing board to us and keeps the replacement. For less critical operations, we provide a standard depot repair procedure. The user sends us the board, and we then repair and return it. Which of these options is selected depends on the type of on-site skills available, what type of hardware extensions have been made on the basic system, and how sensitive the system is to downtime.

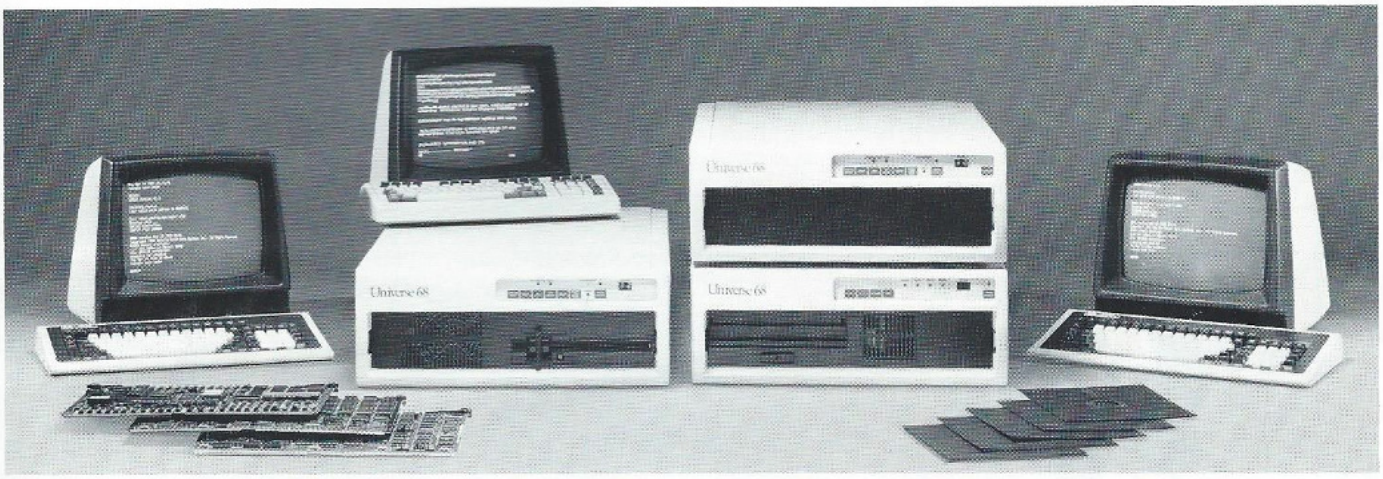
The Dilemma

If the Universe 68 and UNOS operating system provide such attractive capabilities, why don't the established giants of the minicomputer industry come out with similar products? The issues here are ones of market position and compatibility. Traditional suppliers have built systems around proprietary buses out of necessity, and more recently as a way to provide compatibility — not with new standards, but with their existing base of old systems. In many cases, OEMs and quantity purchasers have designed products into those buses and often into the assembly language of those machines. For these OEMs, the choice is either to continue following the traditional vendors' product plans or to break loose from the proprietary systems and move to higher level standards.

(Continued)



The Universe 68 leapfrogs 16-bit machines, combining the low cost and small size of micros with the large addresss space, high throughput, 32-bit architecture, and high-speed bus structures inherited from mainframes (via superminis). The 32-bit Universe 68/05 (under $10,000) and ever-lower-cost VAXs make 16-bit machines lose viability.



The Universe 68 family includes single- and dual-height chassis configurations; interactive terminals; the UNOS real-time, UNIX-compatible operating system; and board-level products.

10 The Business Side: A Win-Win Proposition

(Continued)

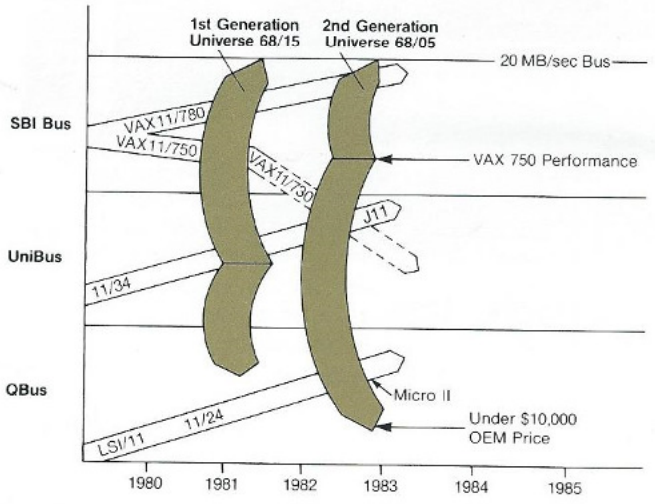
With Digital's recent VAX announcements, it is clear that users of 16-bit minis will want to move up to 32-bit systems. This means new operating systems, new languages, and, most likely, a new bus architecture.

The product that the OEM wants is a system with 32 bits, high bandwidth bus, standard operating system and languages, and a price in the micro-system range. These are the key concepts of the Universe/UNOS system.

For existing products, market pressure will dictate the move to new architectures, higher levels of performance, lower costs, and standard components. Users can bite the bullet and start the transition into the next generation, or wait for a competitor to provide the incentive. Some existing OEM products would benefit greatly from a ground-up transition to a new environment, gaining from the experience of the existing products and taking advantage of the reliability, maintainability, and portability of high level languages.

For a company starting up a new product or project, the choice is clear cut: they can work with a large supplier who uses older proprietary architecture and build on the security of that environment; or they can take full advantage of a new generation of capabilities, built on industry standards, compatibility, and multiple suppliers. The latter option can provide the "biggest bang for the buck" needed to enter new markets and remain competitive in existing markets.

Charles River Data Systems can help system integrators make the move to the new generation of computer capability; we have over ten years experience manufacturing and servicing computer systems and peripherals. At the same time, we offer advanced technology 32-bit systems that take advantage of new concepts in hardware and software architecture.



The Universe 68 breaks traditional minicomputer price/performance restrictions. Historically, the established minicomputer leaders have offered improved performance at lower cost, but without allowing new machines to break into the price domains of older ones. Lower-cost VAXs have broken this pattern, while the Universe 68 leapfrogs it.

Our business is built on the success of our customers. The more successful they are at putting together the right systems, either for re-sale or in-house use, the more systems they will buy from us. We have put together the business elements that can help foster that success from the start-up project through future generations. We want to work with our clients to provide the products described here, and to identify future products. We can't do everything, which is why we have focused on the OEM channel for selling our computer systems. But we can build winning systems, and plan on doing so for some time to come. Our customers' businesses and feedback will help us to produce next year's winners. And of course their success will make us all winners.

Does Vendor Offer:	CRDS	Competitor #1	Competitor #2	Competitor #3
1. 32-bit capability	yes			
2. Real-time transaction orientation	yes			
3. Complete systems hardware and software	yes			
4. Unbundled systems/components	yes			
5. Software only, for use on your hardware	yes			
6. Hardware manufacturing licenses	yes			
7. Standard operating system compatibility	yes			
8. Standard languages	C, PASCAL/FORTRAN/COBOL			
9. Standard/non-proprietary buses	VERSAbus SASI bus VME bus Multibus			
10. Multi year software discounts	yes			
11. System source code availability	yes			
12. Contracts with quantity discounts or cumulative discounts	yes			
13. Retroactive discounts for quantities in excess of estimate	yes			
14. Update service for software	yes			
15. Years of systems experience	Since 1973			
16. A solid market & business Direction that fits with your future	Ask us.			

Vendor selection checklist. UNOS/Universe systems offer crucial capabilities not available from many suppliers. For example, as higher-performance 32-bit microprocessors become prevalent, a full 32-bit data and address bus will be required.

We'd like your input.

We're working constantly to improve our products and our understanding of our customers' needs. Would you help us by taking a few minutes to complete this questionnaire? We value your comments and thank you for your assistance.

If you need more information on Universe 68 systems, please fill out and return one of the postage-paid reply cards attached to the back cover of this book. Or, if you need immediate help, call our U.S. headquarters (617) 655-1800; Mid-Atlantic regional office (201) 666-3900; Western regional office (602) 863-7739; or our European headquarters (U.K.) 44-4955-56545.

Name _____ Title _____

Company _____ Department _____

Address _____ Phone _____

_____ Company Size: _____ Employees

City _____ $ _____ /year gross sales

State _____ Zip _____

What computer systems do you use now?

Vendor	Type	#/yr.	Vendor	Type	#/yr.

Describe your application: _____

In what industry areas does your company participate?

___Education ___Communication ___Medical Care

___Manufacturing ___Transportation ___Energy/Resource Development

___Business Applications ___Banking/Insurance ___Other

Will these systems be used internally_____, or provided to 3rd parties _____?

What kind of systems are you looking for currently?

___KB Memory ___MB Disk ___Graphics (Res:___ x ___) Other:_____

___Terminal ___Printer Speed ___Tape

___8-Bit ___16-Bit ___32-Bit

What languages? (Indicate "1" for Required, "2" for Desired)

___COBOL ___RPG ___BASIC ___ADA ___LISP

___FORTRAN ___PASCAL ___C ___APL ___FORTH

___OTHER _____

What application or additional software facilities? (Indicate "1" or "2" as before)

___DBMS	___ISAM	___Source Code Management	
___Report/Gen Query	___3270	___3780/2780 (RJE)	___HASP
___Ethernet	___IEEE 488	___X.25	___LAN

___Others: _____

_____

How many systems will you purchase for this application in the next 18 months?

Development Systems _____ 1st System Required: Month _____ Year _____

Production Systems _____

Is your project funded?_____ If not, when do you expect funding? _____

Please describe anything else important to your application. _____

_____

_____

What should CRDS' next steps be? _____

_____

_____

What computer publications do you read?

___Byte
___Communications of ACM
___Computer Decisions
___Computer Design
___Computer (IEEE)
___Computer Systems News
___Computerworld
___Datamation
___Digital Design
___EDN
___Electronic Design
___Electronic Engineering Times
___Electronic News
___Electronics
___Electronic Products
___IEEE Spectrum
___Infoworld
___Info Systems
___ISO World
___Mini-Micro Systems
___Software News
___Systems & Software
___Other _____

PLEASE RETURN TO: Marketing Support Department
CHARLES RIVER DATA SYSTEMS, INC.
Four Tech Circle
Natick, MA 01760

For More Information

- ☐ Please send me more information on Universe 68 computer systems.
- ☐ Have a sales representative call me.
- ☐ First system required by _____ (month), ____ (year).
- ☐ Estimated requirements _____ systems per year.

Name _____

Title _____

Company _____

Street _____

City _____ State _____ Zip _____

Telephone _____ Ext. _____

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 29 NATICK, MA

Postage will be paid by addressee

Charles River Data Systems, Inc.
4 Tech Circle
Natick, MA 01760

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

CHARLES RIVER DATA SYSTEMS

U.S. Headquarters: 4 Tech Circle, Natick, MA 01760/Tel (617) 655-1800/Telex II (710) 386-0523
Mid-Atlantic Regional Office: (201) 666-3900; Western Regional Office: (602) 863-7739
European Headquarters: 24 Palm Close, New Inn/Pontypool, Gwent/South Wales, U.K. NP40DE
Tel 44-4955-56545/Telex 851-498627

Printed in U.S.A. IGTU-BR02