KAWASAKI STEEL TECHNICAL REPORT
No.27 (November 1992)
Hot-Rolled, Cold-Rolled and
Surface Coated Steel Sheets
and Electronics and Instrumentation

Realtime UNIX "UNOS" and Its Application

Kazuyuki Sakurada, Satoshi Kuroiwa, Kensuke Iwaki, Osamu Sekigawa

Synopsis :
Kawasaki Steel launched the Universe super-microcomputer family of systems to the Japanese market in October 1988. The Universe systems run under the operating system, UNOS, which is compatible with AT&T UNIX System V. The UNOS is an entirely independent operating system developed to provide capabilities beyond the limited real-time and file system capabilities of standard UNIX. Such extensions are essential in commercially based applications which require prompt response and high reliability. Since its introduction to Japan, the UNOS operating system has been meeting a large number of real-time application needs. Typical application of UNOS is online transaction processing in which database resides and exchanges its data frequently through multiple I/O ports.

The body can be viewed from the next page.

Realtime UNIX "UNOS" and Its Application*


Kazuyuki Sakurada
Manager,
Engineering & Technology Dept., Integrated Systems & Electronics Div.


Satoshi Kuroiwa
Project General Manager, Factory Automation Systems Div., Toyota Motor Corp.


Kensuke Iwaki
General Manager,
Data Systems Sales Dept., Systems Business Operations No. 2, C. ITOH Techno-Science Co., Ltd.


Osamu Sekigawa
Manager,
Engineering & Technology Dept., NTT Chuo-Teleconnet Co., Ltd.

Synopsis:

Kawasaki Steel launched the Universe super-microcomputer family of systems to the Japanese market in October 1988. The Universe systems run under the operating system, UNOS, which is compatible with AT&T UNIX System V. The UNOS is an entirely independent operating system developed to provide capabilities beyond the limited realtime and file system capabilities of standard UNIX. Such extensions are essential in commercially based applications which require prompt response and high reliability. Since its introduction to Japan, the UNOS operating system has been meeting a large number of realtime application needs. Typical application of UNOS is online transaction processing in which database resides and exchanges its data frequently through multiple I/O ports.

1 Introduction

Commercially based application of UNIX**, which is a standard computer operating system (OS), has begun only in recent years. Although UNIX has a history of more than 20 years, commercial application has been delayed for several reasons, among which reliability problems involving the UNIX system itself require particular mention.

In order to bridge the gap between the service level offered by UNIX and market needs, Kawasaki Steel Corporation is marketing a super-microcomputer trade named Universe,*** in which a realtime UNIX called UNOS**** is used as the basic OS. Major considerations in UNOS are total throughput and reliability.

This report argues the advantages of UNOS from a new perspective by comparing UNOS with the general purpose UNIX. It also discusses commercially based applications of UNOS and the results obtained, based on several representative examples.

2 Realtime UNIX "UNOS"

2.1 Purpose of Development of UNOS

The origin of UNIX can be traced to a system first realized by Ken Thompson of Bell Labs in the United States in 1969 in the process of transferring a computer game called Space Travel to a DEC PDP-7. The history of UNIX includes UNIX Version 6 (announced in 1974) and UNIX System 5 (1983). At present, UNIX is used as a standard OS in a wide range of hardware from workstations to mainframe computers.

UNIX provides a multi-user/multi-task environment, and offers functions which make possible a dialogue exchange between the operator and computer.[1] The basic design concept is the equal allocation of system resources (hardware and software) to all users. To this end, the timesharing method of providing service was adopted.[2]

This concept is adequate for applications in the field

* Originally published in *Kawasaki Steel Giho*, 24(1991)1, 1-7

** The UNIX operating system was developed and is licensed by UNIX System Laboratories, Inc.

*** Universe is a registered trademark of Charles River Data Systems, Inc. Kawasaki Steel Corporation is the sole sales agent for the Universe computer line in Japan.

**** The UNOS operating system was developed and is licensed by Charles River Data Systems, Inc.

of research and development, for example, in computer-supported design and software development. However, the conventional UNIX handles all processes on an equal basis, is therefore unable to cope with commercially based applications in which multiple processes of varying urgency and importance are incorporated into a single system.

UNOS is a realtime UNIX-type operating system developed by a group headed by Jeff Goldberg (who also participated in the development of UNIX at Bell Labs) to remedy the weaknesses of the conventional UNIX. The first version was announced in 1981 by Charles River Data Systems, Inc. in the United States.[3] Thereafter, a Japanese language processing function, a window function, and a variety of LAN-compatible functions were added by Kawasaki Steel.

Although UNOS maintains compatibility with UNIX System 5 (in conformity with SVID, Second Ed.[4]), it is technically distinct from UNIX in the following respects:

(1) UNOS is equipped with an autonomous process priority management configuration which makes it possible to cope fully with I/O bound applications; this is not possible with the conventional UNIX system.
(2) The kernel (which is the core of the OS) is the object of realtime processing in UNOS, making it possible for the system to accept external interrupt processing demands even during system calls.
(3) The synchronous processing function for multiple processes was strengthened to facilitate the configuration of multi-task processing systems
(4) A highly reliable filing system capable of high-speed access was developed.
(5) Multi-processor systems are possible, permitting the use of distributed processing in order to increase total throughput.

2.2 Control of Priority of Processes

In practical use, processes in which I/O access does not frequently occur during processing, such as technical calculations and CAD applications, and processes in which I/O access is common, such as transaction processing, often coexist in the same system. In this paper, the former are termed "CPU bound processes," while the latter are called "I/O bound processes."

Processing throughput in CPU bound processes depends on the command execution speed of the computer. Once a process is begun, it is desirable to minimize process changes and continue with the execution of the process until completion. Because CPU bound processes are not expected to respond immediately to interruptions, it is acceptable if they are assigned a low priority level where the right of execution is concerned.

In contrast, it is essential that I/O bound processes respond to interruptions quickly. They must be given priority in taking over the execution right when an initiation request occurs, even if only brief use of CPU resources is required.

In reality, individual processes have both CPU bound and I/O bound aspects, which typically differ in strength. Because individual processes are managed by the kernel, the kernel should ideally be able to observe the distinctive characteristics of each process and control the operation of the processes on the basis of the results.

In the UNOS system, process management was developed in response to this requirement. In the conventional UNIX, on the other hand, all processes are considered to be CPU bound and are controlled as such, sacrificing the response characteristics needed by I/O processes.

Figures 1 and 2 show examples of automatic adjustment of the priority level of CPU bound and I/O bound processes by the UNOS process control function. For each time interval Δt, which is a function of the priority level, the process priority level is reduced only by a prescribed, system-specific value. Naturally, it is possible to fix the process priority level at a constant value by setting identical upper and lower limits. The expression Δt is referred to as "quantum time" and is the means of activating the scheduler in response to the passage of time in each case.

When an I/O wait occurs during the time Δt, the process switches into the rest mode, and the process priority level is corrected to an intermediate value between the existing value and the upper limit value. The

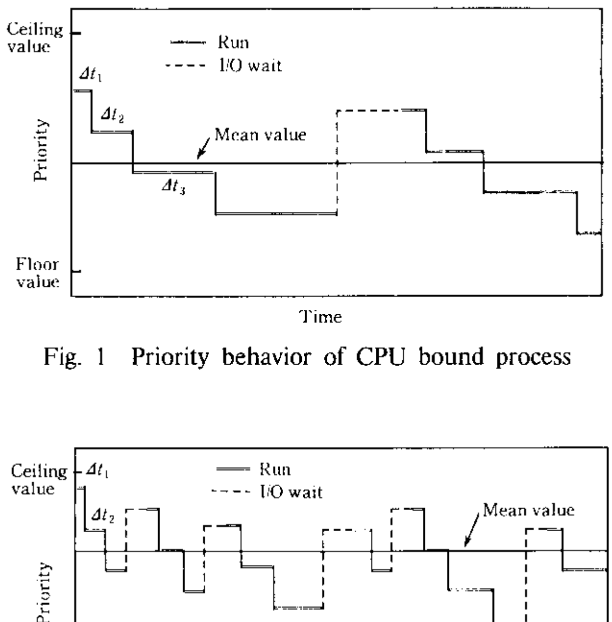

Fig. 1 Priority behavior of CPU bound process

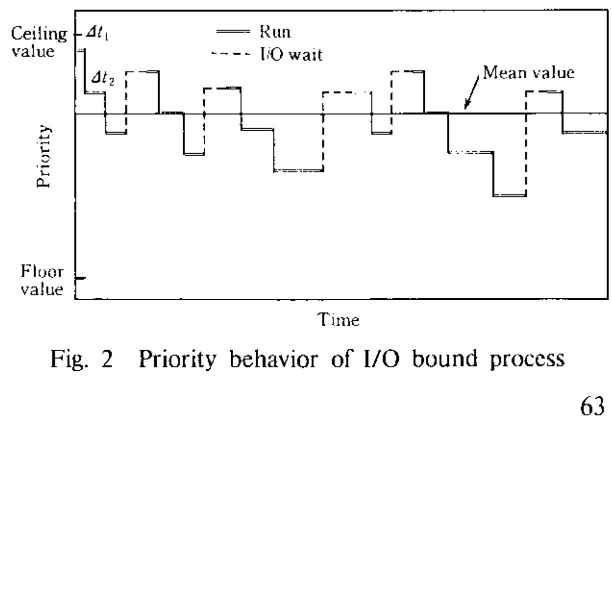

Fig. 2 Priority behavior of I/O bound process

scheduler is activated, and the execution right is passed to another process. However, even when an I/O wait does not occur during Δt, the scheduler activates when Δt expires, and the process execution right is transferred if another process with a high priority level is waiting for initiation.

In the UNOS system, the time interval Δt is a function of the process priority level. The value of Δt is small when the priority level is high (20 ms minimum) and large when the priority level is low (4 s maximum).

From the figures it can be understood that the priority level of I/O bound processes is automatically adjusted to a higher average level than that of CPU-bound processes; on the other hand, a CPU-bound process can continue to occupy the CPU for a considerable period of time if process switching is not initiated by an outside I/O interruption.

2.3 Realtime Kernel

The reason UNOS is called a realtime UNIX is because scheduling processing for process switching is executed immediately in response to interruptions. Interruptions of the UNOS system can be broadly divided into the following three categories:

(1) Interruptions from outside hardware, such as tty interruptions
(2) Exceptional interruptions
(3) Trap interruptions originating in a system call

UNOS is provided with an interruption handling function for each type of interruption, and is designed in such a way that the scheduler is activated when interruption handling is complete. The scheduler compares the priority levels of processes on standby as previously mentioned, and allocates CPU resources to the process with the highest priority. It should be noted that the fact that the scheduler activates after every interruption handling operation is a major point of difference between UNOS and the conventional UNIX. This feature makes process switching possible even when a kernel is engaged in the execution of a system call, and is the basis for realtime processing.

2.4 Mechanism of Process Synchronization

Although CPU bound processes are generally run freely and without interference from other CPU bound processes, the mutual timing of processes is frequently a consideration with I/O bound processes. A process synchronization mechanism acting through the kernel is therefore needed to coordinate I/O bound process timing.

UNIX systems possess a synchronization mechanism in the form of "signals," but generally speaking, virtually all 32 signals are previously defined by the system, and the range available to the user is limited. For this reason, a synchronization mechanism is commonly realized in UNIX systems by using an interprocess communication function such as message queuing. However, the kernel must make a detailed check of a number of internal tables in this method, and synchronization processing therefore requires considerable time. Because realtime performance is not achieved, this is a problem in the conventional UNIX.

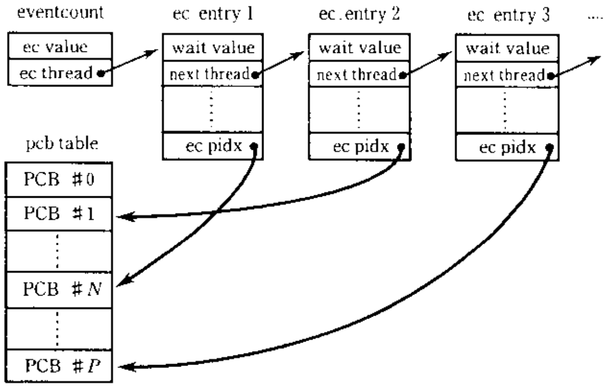

Fig. 3 Internal mechanism of eventcount (ec)

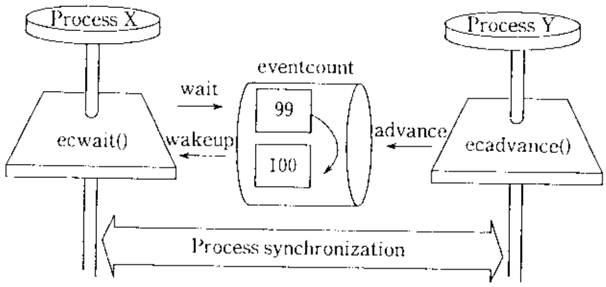

Fig. 4 Process synchronization mechanism

To solve this problem, a new function called "eventcount" was introduced in UNOS. Eventcount (EC) is a special full-time memory file comprising a 32-bit integer type counter and pointer. Three types of EC are available, one which can be used as desired by the user, one used internally by the UNOS kernel, and one used by the device driver. The mechanism, shown in **Fig. 3**, is basically the same in the three types.

The counter value in the EC is a cumulative sum of individual ecadvance () system calls. If the wait value in the ec_entry table indicated by the pointer value and the ec value are in agreement, a status flag for the process control block (PCB) table indicated by the ec_entry table mentioned above can be rewritten, and the object process is converted to the initiation waiting status. Thereafter, the execution right is passed to the scheduler. This is the mechanism for initiating a process. In the ec_entry table, the chaining order begins with the lowest waiting value in order to minimize the time spent by UNOS in table searches.

An example of the synchronization of two processes using ec is shown in **Fig. 4**. In conceptual form, this

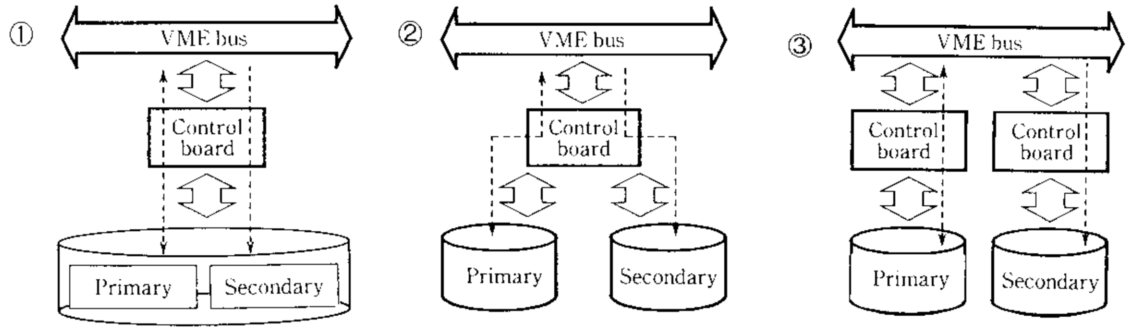

Fig. 6 Logical pairing between two disk partitions

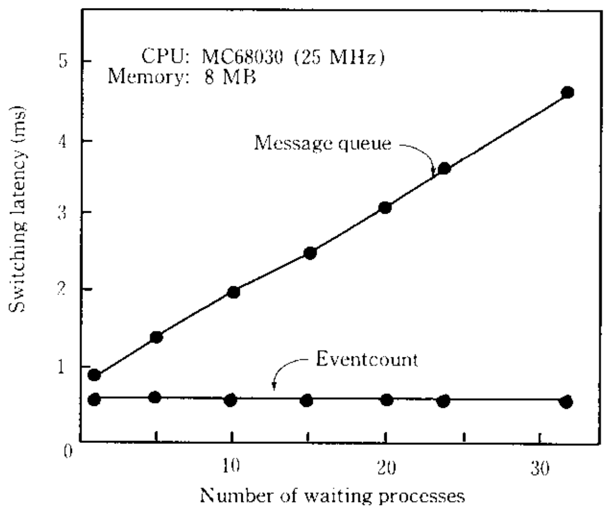

Fig. 5 Process switching latency

example shows how process X, which is programmed to remain in the waiting state until the ec value reaches 100, "wakes up" after the ec value is increased sufficiently by process Y.

The results of a trial comparison of process-switching processing performance with the UNOS ec function and UNIX message queue function are shown in **Fig. 5**. The time required for process switching was measured as processes in the waiting condition activated and then successively returned to the waiting state. Groups with various numbers of processes were tested to determine the relationship between the number of processes and switching latency (time required for switching). As can be seen in the figure, the time required for process switching increased with the message queue method as the number of processes in the waiting condition increased, but remained unchanged when ec process switcing was used. Although the message queue function is used in the conventional UNIX system, practical high-speed process synchronization can be realized by using the ec function. This is a distinctive feature of UNOS.

2.5 File System

The UNOS file system improves performance and reliability by means of several unique features. Measures for improving performance in accessing magnetic disks include contiguous files and two-level scatter files. The disk shadowing function enhances reliability.

Disk shadowing prevents the loss of data from magnetic disks due to hardware problems by establishing a logical pairing between two disk partitions and then updating the data contained in both partitions in parallel operations. If a problem occurs with the main partition, the backup partition automatically becomes the main partition, and processing can continue. As shown in **Fig. 6**, three types of logical pairing are available, corresponding to the response level required by the problem.

3 Universe Computer

Universe is a computer platform with a Motorola 32 bit CISC type microprocessor (MC 680XX) and a VME bus as the standard I/O bus. An MC 68040 (25 MHz) chip can be used for maximum calculation performance, producing a measured drystone value of 29 050 s^{-1}. When an MC 68030 (25 MHz) is used, the measured drystone value is 4 950 s^{-1}.

Universe offers a suitable environment for the UNOS operating system. Excellent access to magnetic disks and communication networks is achieved with a functionally-distributed I/O processor. In addition to various serial communication functions, the communication functions to which the Universe computer is particularly suited include world standard LANs such as TCP/IP, Full MAP, and Mini-MAP. Universe also meets the requirements of ME-NET[5], a LAN for field-bus applications which is being actively promoted particularly by the auto industry in Japan. Many of these network functions were developed by Kawasaki Steel for application in the field of factory automation, which requires a high degree of system reliability.

It may also be mentioned in passing that Kawasaki

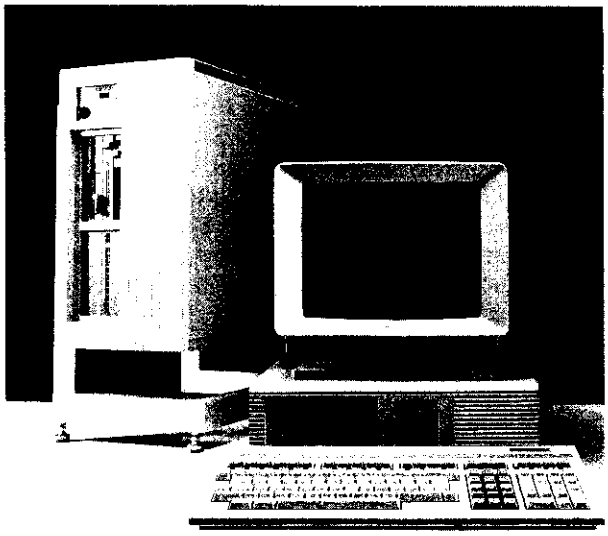

Photo 1 Universe 400 (left) and Universe 200 (right)

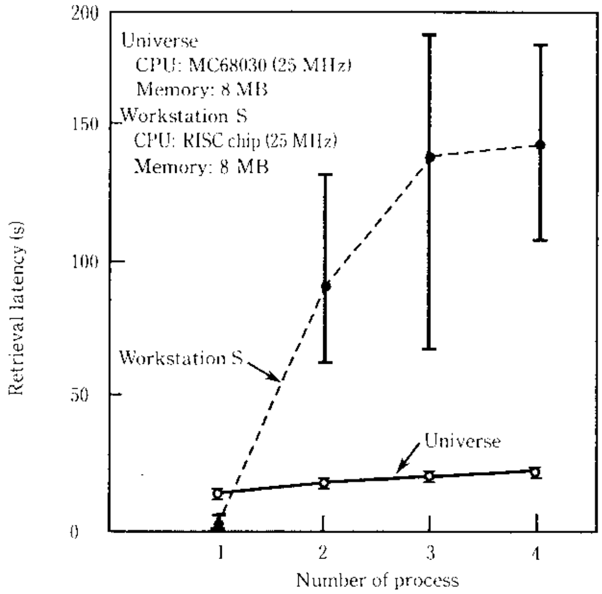

Fig. 7 Comparison of informix benchmark test's results between the Universe computer and a workstation widely used in the market

Steel has been marketing Universe computers (**Photo 1**) in Japan since October 1988.

4 Evaluation of Data Base Processing Performance

A data base management system (DBMS) function is indispensable to meet the needs of commercially based applications. The Universe computer supports the Japanese language edition of informix[6] software (a product of the American firm Informix), which is the mostly widely used relational DBMS worldwide.

The informix DBMS has a server/client configuration. Communication between the client software and server, which is a data base engine, is by means of structured query language (SQL) commands, which are a superset of the standard query language as specified by the American National Standards Institute (ANSI). Networking is possible, as is distributed processing.

Because the server accepts accessing by multiple clients, its process characteristics are strongly I/O bound, and I/O-bound processing performance has a major impact on total system throughput.

A tree-structure file called an "index" is incorporated in the informix DBMS. The speed of access to data is increased by following the index from point to point according to pointer values which are also included in the system. However, because the index is stored in a magnetic disk, server processing performance is greatly affected by the performace achieved in disk I/O access characteristics.

Figure 7 shows the results of an investigation of processing performance with a Universe computer (3.2 MIPS MC 68030) and a commercial UNIX type workstation produced by another company (the comparison machine, called workstation S in the figure, used a 17 MIPS RISC chip). Identical informix-SQL software (Edition 2.10.03 C) was used. In this benchmark test, the time required for multiple clients to search identical data bases for data corresponding to object conditions was measured in five separate trials. The average, maximum, and minimum times are shown in Fig. 7.

A remarkable difference in the performance of UNIX and UNOS was seen as the number of clients increased, as is obvious in fig. 7. In spite of the much higher MIPS value of the commercial UNIX workstation, the UNOS-mounted Universe computer was clearly superior in processing performance when serving more than one client. Moreover, there was little variation in processing time with the Universe, indicating that it is possible to accurately estimate processing time requirements with this machine.

In sum, however, the remarkable difference between these two machines reflects the difference in the realtime processing performance of the UNOS and conventional UNIX systems.

5 Examples of Application

As can be understood from the foregoing, the UNOS OS provides the optimum to processing conditions where high-frequency interruptions from numerous I/Os are involved. This is made possible by a realtime kernel and the unique priority-scheduling function of the UNOS system. In addition to these features, the high speed and high reliability realized in file processing make UNOS an appropriate OS for data base processing.

It is of course desirable to select applications in which these features can be used to best advantages. Such applications generally involve what is termed "on-line transaction processing (OLTP)."[7] OLTP is used in processing systems in which messages and transaction data generated at input terminals are updated or referenced in realtime using a data base.

The following presents several representative examples of the use of UNOS in commercial OLTP applications (The reader is also referred to the literature cited[8-11] for other examples which cannot be discussed here).

5.1 Automotive Part Production Management System

A single automobile typically includes approximately 50 000 parts, some of which are assembled into critical mechanical components such as constant velocity joints and differential gears. Many of these critical mechanical components are manufactured by the automaker itself. However, with frequent model changes and the increasing number of models in makers' product lines, there is a constant need to register items with new specifications. Thus, the production line for these mechanical components is a typical example of a multi-kind production line.

The system described in this section was developed to optimize order generation and control the progress of production and level of inventories in in-house auto component production processes at Toyota Motor Corporation, where the component assembly process is a super-small lot/multi-kind operation characterized by sharp changes in required quantities.

To make it possible to cope flexibly with changes in product delivery and lot size requirements, as well as with future system expansion, a distributed processing system was configured with multiple Universe computers and personal computers networked in a LAN. The system is shown schematically in **Fig. 8**.

Based on daily order information transmitted by the host computer (IBM 3090), the system indicates the optimum component processing and order amounts. As results, it transmits the production records to the host computer.

In the administrative offices, tables of parts assembled into mechanical components and standard processing tables are recorded in the data base, and production order sheets are automatically output.

In the plant, optimization processing which matches orders for each part, inventory levels, and production plans (ordering plans) is applied to all lines. Processing instructions are given and production data is collected using bar code control sheets.

The previously mentioned informix is used in the data base, which facilitated the development of a complex software for registering, collating, and searching a large number of data base tables. Repeated processing calculations comparing order generation, inventories, and production (ordering) plans can be conducted while the data base is being accessed. In this regard, the high-speed characteristics of UNOS data base access (discussed in Sec. 4) were particularly useful. Even when multiple client applications are accessed simultaneously in the same data base table, well-coordinated processing is pos-

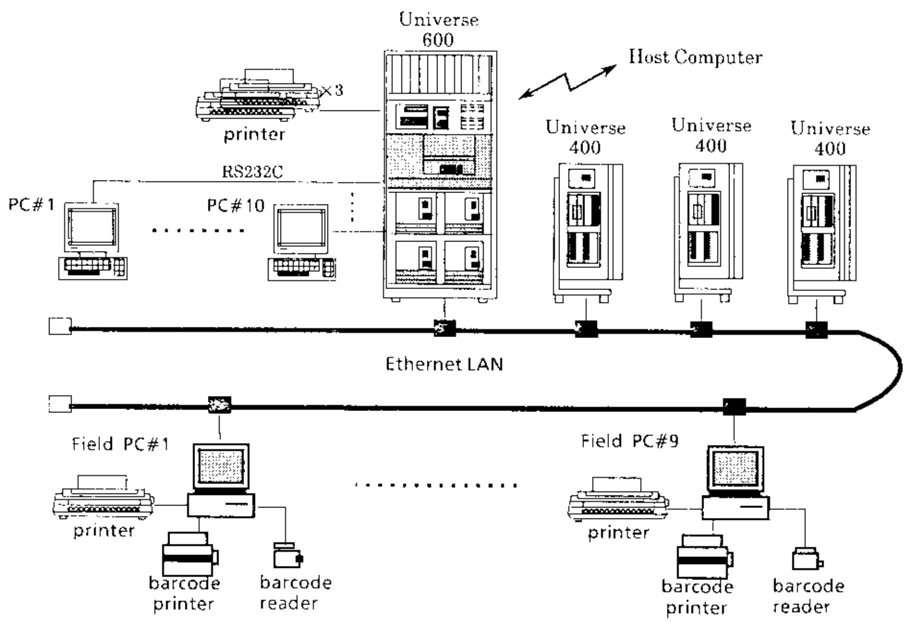

Fig. 8 Computer system of automobile parts manufacturing shop

sible with the informix exclusion processing function.

5.2 Data Entry System

The purpose of data entry systems is to transform the various types of data which play a central role in an information-intensive society into a form which can be handled by a computer. Generally, the data entry industry comprises service contractors, who perform the actual data input task on behalf of a requesting client, and in-house users, who perform the work themselves. Recently, however, the volume of data which clients handle has risen rapidly, resulting in an increased need for expanded, large-scale data entry systems with many input terminals.

Although C. Itoh Techno-Science Co., Ltd. has marketed data entry systems for many years, they have developed and are now marketing a new system trade named G-5EX, which is capable of meeting the market needs described above. The G-5EX is configured as a conventional system with a Universe computer as the host. A large number of operator keyboards with customized *kanji* (Chinese character) keyboards are linked to the Universe by serial communication. Input data is collected and output on magnetic tape or optical disks.

The key codes which are input at terminals by operators pass through a serial circuit and reach the main CPU by way of the I/O processor card and VME bus in the Universe computer. The main CPU performs *kanji* conversion for multiple keyboards and returns a specific *kanji* code to the dam terminal where the signal originated, and the result appears on the operator's screen. Although the average operator keystroke speed is 5 stroke/s, with a maximum of 10 stroke/s the G-5EX has demonstrated that it is capable of supporting as many as fifty operator terminals. **Photo 2** shows a typical workplace scene at the Data Entry Center of Densan Corporation.

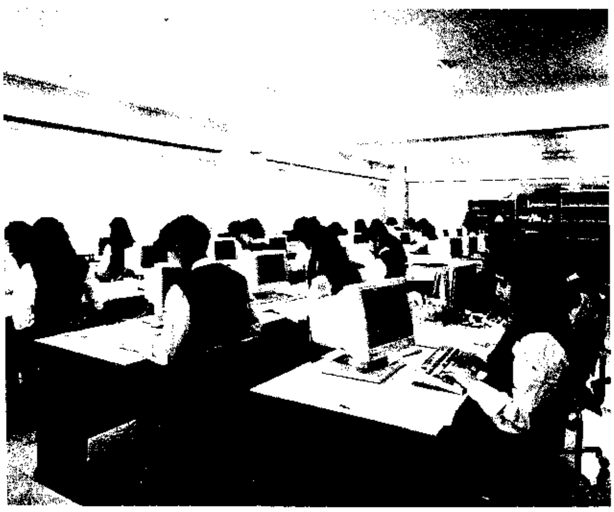

Photo 2 View of Data Entry Center office (by courtesy of Densan Corp.)

5.3 Telecontrol System

Telecontrol systems were introduced in 1969 as telemetering systems which read consumption meters for public utilities such as gas, water, and electricity by way of telephone circuits. At present, these systems are used in an operating-agency function in controlling data searches in remote areas, and in data collection and data signal generation, and are expected to be expanded into telemeter VAN systems in the future.

NTT Chuou-Teleconnet Co., Ltd. considers the telecontroller systems to be part of the infrastructure of an information-intensive society, and has already begun marketing and providing service for a telecontroller system trade named ATEX-L for the LP gas industry, where the need for this type of system is greatest. Using NTT's "no-ring" service, the system reads customers' meters by silently selecting the specified terminal and receiving the signal (terminal pickup method).

The configuration of the system is shown in outline in **Fig. 9**. A Universe computer loaded with UNOS, is located at the Telecontroller Center, along with the Center's network control units (NCUs) and modems. NCUs attached to the gas and other meters being controlled and control units for communications with the Center are located in each customer's home. The NCUs in the customer's home communicate with those at the Center via the telephone network. Gas sales offices are equipped with ordinary personal computers and printers, making it possible for individual offices to exchange information with the Center electronically using telephone circuits.

Not only can the gas sales office's task of meter-reading be automated when this system is used, but gas leaks and other abnormalities can be detected by remote control using sensors located in the customer's home. It is also possible to shut off the gas valve in the customer's home by remote control in an emergency.

This system is required to manage the records for a wide range of customer homes, with one center being responsible for the records of approximately 100 000 customers. UNOS was adopted because its features satisfy the essential requirements of realtime processing in data base search and updating operations and high reliability in file management.

6 Conclusions

A large number of application software programs have been written for UNIX, which continues to occupy a well established place as the industry standard OS. However, because both CPU bound and I/O bound processes are present in commercially based applications, as exemplified by on-line transaction processing (OLTP). A realtime processing function and a proces scheduling mechanism capable of allocating computer resources in

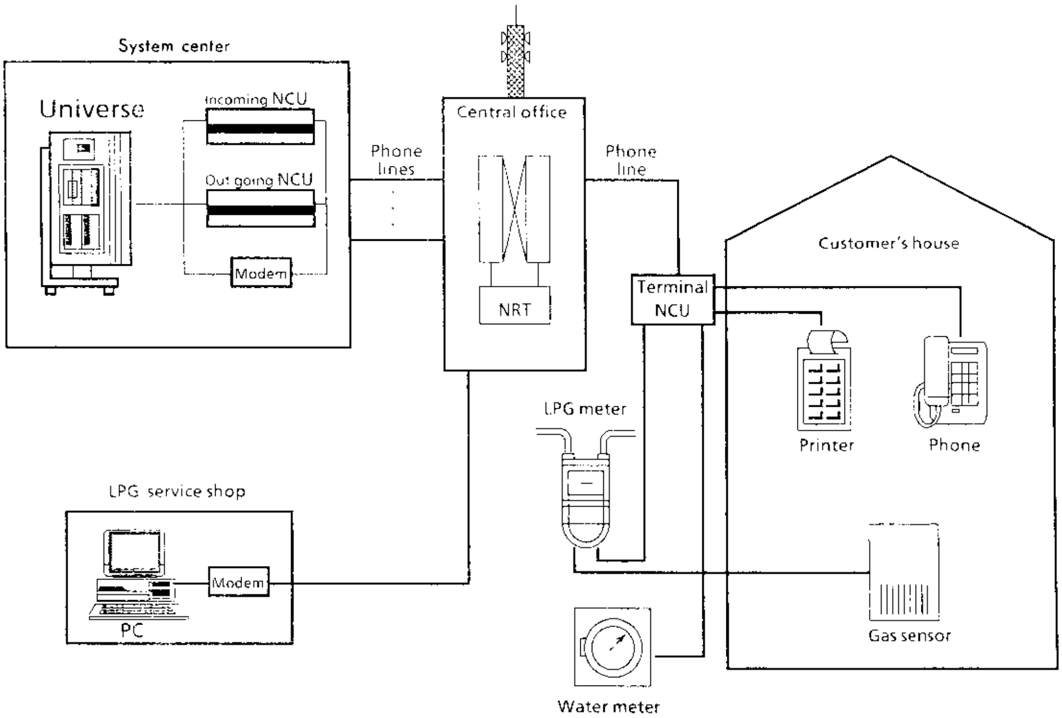

Fig. 9 Configuration of telecontrol system

line with the special features of the respective processes are therefore necessary in realizing a high-throughput system.

The authors have constructed a practical system using a realtime UNIX system called "UNOS" for commercially based applications with these functional requirements. The conclusions of this work are as follows:

(1) In data base search and update operations, UNOS offers higher speed and fewer variations in processing time than the conventional UNIX.
(2) For frequent I/O interruptions, UNOS is superior to UNIX in realtime response characteristics.
(3) The UNOS eventcount function is useful in realizing synchronization among processes easily and in realtime.
(4) UNOS is provided with reliability-enhancement functions such as disk shadowing which make it possible to construct a system with high reliability.

UNOS has been developed by Kawasaki Steel to respond to the requirements of Japanese language use and the needs of a variety of types of networks, and is demonstrating its effectiveness in a number of fields of practical application, including several presented in this paper.

The future will perhaps see striking growth in the use of UNIX systems in OLTP applications in manufacturing, finance, distribution, communications, and other industries. Although tendency to enhance the realtime function of the conventional UNIX system is already in evidence,[12] it is hoped that research and development will continue to progress in this field.

References

1) AT & T Corp.: UNIX System V User Reference Manual, (1984)
2) D. M. Ritchie and K. Thompson: "The UNIX Time-Sharing System," *The Bel Technical Journal*, 57(1978)6, Part 2
3) Charles River Data Systems, Inc.: "Universe VME Family Technical Overview", (1987)
4) AT & T Corp.: System V Interface Definition, (1985)
5) ME-NET Secretariat: ME-NET Overview, Verl. O Rev. 1, (1991)
6) Ascii Corp.: informix-SQL USER'S MANUAL, (1989)
7) J. Gray: Online Transaction processing Systems, (1991), [McGraw-Hill]
8) H. Watanabe and K. Sakurada: "Application of Real-Time UNIX to FA," Jidoka-gijutsu (Mechanical Automation), 22(1990)6, 38–42
9) H. Torikoshi, T. Ikeda, T. Hosokawa, S. Okamoto, and I. Morioka: "Automatic Testing System for Sheet Metal," *CAMP-ISIJ*, 4(1991)2, 326
10) All Nippon Airways Co., Ltd.: ANA Flight Academy, (1990)
11) O. Nakao: "Telemarketing Strategy by Voicelink," *Telemarketing journal*, 4(1991)5, 35–42
12) UNIX International, Inc.: System V Road Map, (1991)