ETOS - A MULTI-LINGUAL TIME SHARING SYSTEM FOR THE PDP8

Author : Jim Dempsey

ABSTRACT

EDUCOMP's Time-shared Operating System (ETOS) is an integrated package of hardware and software designed for maximum utility of a computer. ETOS supports up to 16 users, with each user able to use the system as if he were operating on a stand-alone PDP8 with 64K bytes of memory.

This paper will contrast the various modes of computer operation and illustrate how the combination of batch, real-time, and interactive usage results in the optimum fulfillment of the all-important criterion of price/performance.

ETOS requires a 16K omnibus PDP8, clock, an RK8e controller with an RK05 drive, a backup peripheral for the RK05 drive, and a Time Share Control Module.

INTRODUCTION

The PDP8 was introduced in 1964. Since that time, over 20,000 PDP8s have been installed throughout the world. They have been used for every conceivable type of application from computer assisted instruction (CAI) to point of sales data entry (P.O.S) ~~CAI to POS~~. A complete array of peripherals have been interfaced to this versatile computer.

The inherent problem in this versatility has been non-standardization of hardware and software components. The large install-base has resulted in a proliferation of tailored software. One outcome of this situation is the fact that users usually can find other users who are performing similiar operations. Due to the flexibility of the machine, combining groups for greater efficiency and lower costs has often been impossible. A group of lab technicians running analog to digital conversion applications monopolize the machine so that data entry personnel spend their time and their employer's money waiting for the completion of the real-time tasks.

EDUCOMP decided that the variety of uses for the computer did not have to be mutually exclusive. ETOS was conceived with this objective.

GOALS

Before evaluating the effectiveness of a system, one must first define the goals of a computer utility[1].

1. Convenient remote terminal access as the normal mode of system usage.
2. Continuous operation with a minimum of interruptions.
3. A wide range of capacity to allow growth or contraction without either system or user reorganization.
4. An internal file system so reliable that users trust their only copy of programs and data to be stored in it.
5. Sufficient control of access to allow selective sharing

of information.

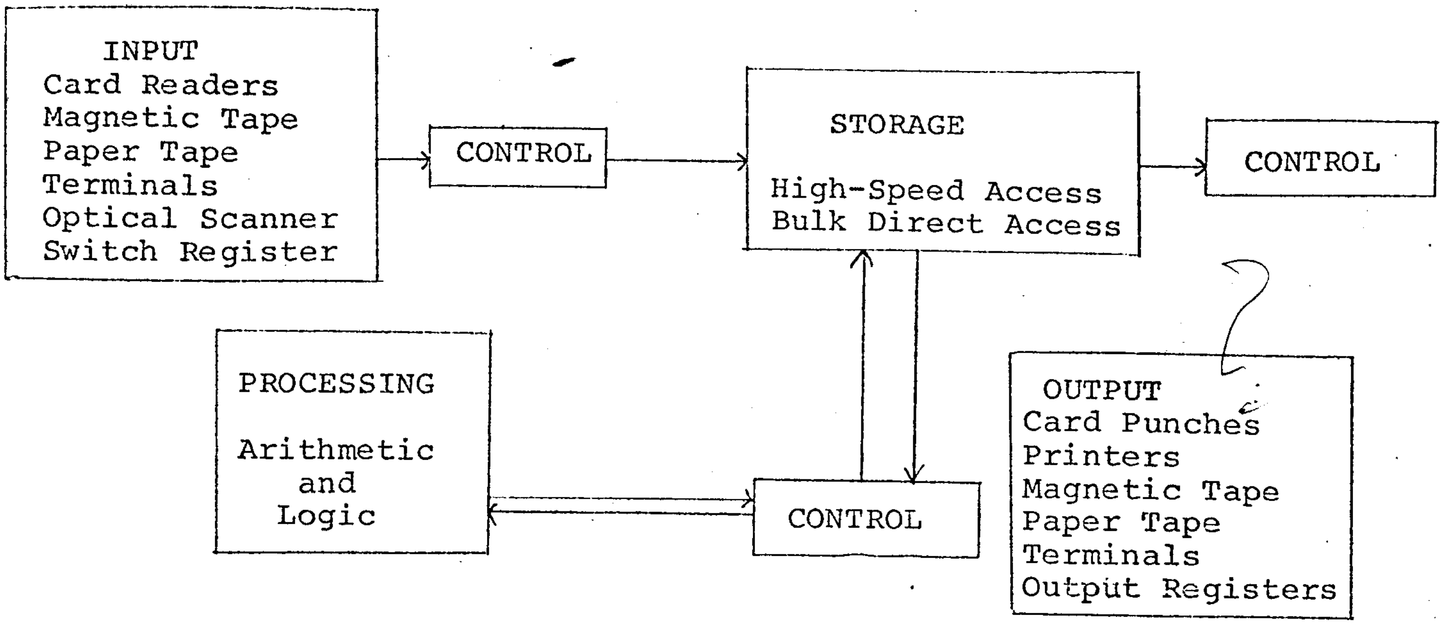
WHAT?

6. The ability to structure hierarchically both the logical storage of information and the administration of the system.
7. The capability of serving users with often conflicting objectives, such as educational, industrial, and administrative users.
8. The ability to support differing programming environments and human interfaces within a single system.
9. The flexibility and generality of system organization required for evolution through successive waves of technological improvements and the inevitable growth of user expectations.
10. An elevated degree of sophistication for the advanced user; yet, at the same time, ease of operation for the novice.

To determine if ETOS satisfies these requirements, it is necessary to examine the evolution of computers in general and, as an important subset, the PDP8.

HISTORY

The five basic functions ~~fo~~ of a digital computer are input, storage, control, processing, and output. These functions can be illustrated with the following flowchart[2]:



One of the earliest forms of computer I/O was paper tape. These systems were very slow for program development, and the file structures (boxes and racks of paper tape) were bulky and unreliable. On the PDP8, versions of BASIC, FOCAL, and Assembler used this method.

To achieve a more efficient programming system, the capability was developed to store programs and files on a mass storage medium. On the PDP8, the 4K disk/DECtape monitor fulfilled this responsibility. It provided users file and program storage and a keyboard monitor so that users could retrieve and run programs off the disk or tape with a few simple keyboard commands. Throughput of the system, especially editing, assembling, and debugging, was substantially improved. A batch mode was added to this monitor so that users could write programs on cards and increase throughput and the utility of the computer.

Up to this point, the use of the computer was restricted to single-user. In the interactive mode, program development for the one user is fast, but no one else can use the computer. File security is maintained, but the sacrifice in throughput is unacceptable. In the batch mode, the central processing unit is used efficiently, but peripherals will often remain inactive if a proportion of the users do not access them. Under these constraints, the peripherals are not available for a user who could be accessing them. With the variety of unsupervised jobs on the system, many batch environments are not suitable for file protection.

Another mode for utilizing the computer is real-time. This mode functions interactively or in a batch environment. Real-time may be defined as processing in which the input, computing, and output of the information must meet a deadline established by external conditions. An example is the application of the computer to perform analog to digital data conversion. An external device must be sampled every *x* seconds, or data would be lost permanently. The controlling job must be sure that no processing or other forms of I/O are occurring when the device necessitates data acquisition. The above characteristics restrict real-time to single-user mode in most systems.

The time sharing system was developed to remedy these problems. The first time-shared systems, however, supported only one language. Another important restriction was the requirement for enough memory on the system capable of maintaining all user jobs resident in memory. The ramifications of this was that an operating system requiring 4K words, at maximum, left 28K for users. If each task were only 4K, a maximum of 7 users could be supported. These initial systems also had to have a fixed size for each user, assigned at configuration time. If a user needed a little more memory for a special job, the entire system needed to be reconfigured. The beginning time sharing system on the PDP8 was FOCAL.

The next innovation for time sharing was the concept of virtual storage. Using virtual storage algorithms, the total amount of memory space taken up by user jobs, can exceed the physical amount of memory on the system. TSS/8 was developed for the PDP8. It supports BASIC, FOCAL, Assembler, and FORTRAN. However, under TSS/8, each user may only have a 4K job. This restricts the languages written for TSS/8 to a minimal subset of the majority of existing languages. ETOS removes this restriction by allowing each user a full 32K virtual address space. It uses the following technique to accomplish this.

To allow the processor to address more than 4K core (this being the limit imposed by a 12-bit word length), the extended memory control provides two three-bit registers, each allowing the processor to address 32K. The first register is the Instruction Field (IF) register, the contents of which are used as part of the memory address for instructions, pointers, and directly addressed operands.

At any point in time, a task requires, at most, 8K words. 8K is required if the Instruction Field and Data Field are different. Therefore, ETOS must only reserve 8K in the physical memory for the user to occupy. Since the resident portion of ETOS, which always remains in memory, occupies 8K, the minimum configuration requires 16K.

When ETOS is running, each task processes in user mode. This mode of operation causes the hardware to trap all Input/Output instructions, which include instructions which load and read the IF and DF registers. When the user executes a field instruction, the monitor maps the virtual address (the field specified in the change field instruction) into a physical address space. If the field which the user desires to access is not resident, the monitor must swap into memory an image of this field stored on the disk. The swapping, therefore, is performed in 4K word blocks. This mapping procedure is transparent to the user. He programs as if he had a real 32K word machine to himself.

ETOS also provides a special debug mode. In this mode, the user can emulate all IOTs. The monitor does not interfere with this emulation. The user may then write controlling programs which allow him to handle virtual interrupts, and other undefined instructions. The user could even write a TSS/8 emulator which would allow TSS/8 programs to run unmodified. Conversion from TSS/8 to ETOS would be made less painful with this capability.

To speed up the swapping process, EDUCOMP designed a time share control module. This board plugs into the onmibus like any other PDP8 module. The difficulty that this board overcomes, occurs in the case of the Change Instruction Field (CIF) instruction.

When this instruction is executed, it does not take effect until the next JMP or JMS instruction, thus making cross-field subroutine calls possible. As a result, the timesharing monitor, when it gets the CIF instruction, may only note the fact that a change is pending. No monitor trap is caused by the JMP or JMS. Consequently, the monitor must emulate all instructions from the CIF to the next JMP or JMS, or a hardware unit must trap JMP or JMS when it is enabled. We chose the latter course (RTS/8's background OS/8 support, the former) because of its greater efficiency. It is important to note that a computer with this additional module operates normally when the user processes stand-alone.

The most recent innovation on the PDP8 has been the development of OS/8 (formerly PS/8). OS/8 was developed so that the user could use the entire capacity of the machine. The 8K - 32K word job area assigned to the user allows the use of a set of sophisticated languages. OS/8 supports FORTRAN IV, Extended BASIC, COBOL, PAL8 and a wide variety of other languages and applications. It also contains improvements such as a system-wide command decoder utility which standardizes command syntax, a keyboard monitor for dispatching tasks, and system- buffered I/O. The main drawback of OS/8 is that it is a single-user system. Users may run sophisticated real-time applications, editing, assembling, debuggine, etc. in an interactive or batch mode. However, we regress to the situation in which personnel are fighting for machine-time.

ETOS makes available the full capabilities of OS/8 to each of many simultaneous on-line users. Several users may be running interactively, several users may be running in a batch mode from a batch file or from Hollerith or mark-sense cards, and several users may be running real-time. Jobs which are not terminal bound may be detached so that one terminal can initiate any number of tasks. Each user has available not only the sophisticated software capabilities of OS/8, but also the wide range of peripherals available.

Many time sharing systems have a serious drawback in their method of supporting peripherals. Often, devices not supported by the vendor are either impossible to implement by the user, or are costly to do so. Assuming they can be integrated into the system, they often complicate the system so that it cannot be supported. ETOS supports the extended arithmetic element, line printer, card reader, DECtape, high speed reader, high speed punch, and a variety of terminals. Line printer spooling is performed automatically. Assuming the IOTs are standard, ETOS will support terminals of any speed. Terminals may be hard-wired or remote.

For devices ETOS does not support, EDUCOMP has devised a unique plan for implementation. Device handlers are run as detached user jobs. Users who desire to utilize the device send this job message concerning the respective user task. This method has the advantage of separating the monitor and the device handlers. Users can debug their handlers external from ETOS. Therefore, ETOS can be more easily and satisfactorily supported, and modularity is maintained. Documentation is provided for the writing of device handlers and booting these jobs into the ETOS interrupt skip chain. Because the detached taks is booted into ETOS in this manner, the unusual peripheral can be supported as efficiently as a standard peripheral and time sharing need not be noticably affected.

The real-time support may also be used in conjunction with the device support. The user has the facility under ETOS to lock his job into core for a specified period of time. During this period, he may or may not decide to service interrupts. The user can actually program most real-time applications so that other time sharing users are unaware of the real-time processing. This capability will allow analog to digital conversion, digital to analog conversion, and simulation of events taking place in a critical period of time.

Currently, ETOS can talk to another computer system by treating the inter-computer link as another terminal on the system. The computers are linked together through terminal interfaces. Programs may be written which facilitate this synchronous transference of data. If the user desires faster and more communication, he may use the real-time facility to permit an asynchronous link.

PERHAPS A COMMENT ON WIDE USE AND VERSATILITY

EDUCOMP decided that OS/8 was very powerful, yet the user may desire a different operating system (e.g., COS300 with DIBOL). Therefore, no system file directory structure is imposed from the user's point of view; he would be presented with a logically contiguous addressing space, upon which he might impose any desired structure. ETOS is, thus, an environment in which a variety of substructures may be maintained.

ETOS has many advantages over OS/8 other than multi-user capabilities. The user has the facility to enter the System Command and Login/ Logout Executive (SCALE) which provides each user with absolute control facilities over his virtual machine. In essence, the user has a virtual front panel. All virtual registers can be displayed and modified. Virtual memory locations may also be examined or changed. These options are also available when the user is running. This provides a powerful debugging tool for Assembly language programs. Another advantage of SCALE is the safety provided for experimental programs. If the user's program halts or he destroys his copy of OS/8, no other users are affected, and the virtual machine with virtual OS/8 may be restarted from the terminal with a BOOT command.

File security and selective information sharing have also been the subject of a great amount of development effort. The user must first log in at an ETOS terminal. A password and an account number must be specified to allow access to system resources. If the password or account number does not match a table of valid accounts set up by the system manager, entry to the system is denied.

Each account number has a set of attributes assigned to it by the manager. Two types of file quotas are imposed. There can be a limit established on the amount of storage a user may use while he is logged in. The other limit is the amount of storage a user may have assigned when he logs out. Users can be inhibited from inadvertently or intentionally filling up the public disk structure. The account may also be designated as variable-length or fixed-length. A variable-length account will allow blocks not used to be available to all jobs. A fixed-length structure will reserve a prescribed amount of blocks so that the user can guarantee that he will have space for a particular operation. A user's files are, in addition, stored with a protection code which allows or disallows read/write privileges on those files of users under other accounts. Multiple users can, if the file protection code permits, read the same file simultaneously. An account may allow a user to mount his own private pack to extend his available storage area. Such private packs, dependent upon parameters provided when mounted, may be either totally private (single-user only) or semi-private (accommodating more than one account). Again, no directory is imposed, affording the user's operating system complete control over disk allocation. (OS/8 establishes all necessary directories automatically.)

There exist several attributes of an account separate from its file structure. The maximum amount of memory available to a job may be restricted. The system manager may impose limits to eliminate the situation wherein ETOS spends all of its time swapping and no time processing (thrashing).

The priority of a user is assigned at log-in time. ETOS utilizes a priority scheduler. Jobs which the system manager decides are more important than the average user, may be set to a higher priority. These tasks will then be processed before a job of a lower priority. There are eight levels in the priority run queue.

ETOS also has an internal priority depending upon the nature of the task. Input/Output has a higher priority than computing. Terminal input has the highest priority of I/O. In this manner, a user typing at the terminal will receive immediate response, while a compute-bound job will run a little slower than usual. This feature is important for user acceptance of the system. Data entry (e.g., POS) requires efficient terminal service.

One other feature is provided for efficient management of user tasks. When a user is swapped into memory, ETOS processes the job for a prescribed period of time. At the end of this time, or quantum, the user is swapped out and another user is swapped in for processing. The usual quantum is one hundred milliseconds. The system manager may specify that a user have a larger quantum for a critical application. There are eight possible quantums, from one hundred milliseconds to eight hundred milliseconds in increments of one hundred milliseconds. To perform the previously mentioned swapping, ETOS uses an optimized disk scheduler. Jobs are processed until the instant they are swapped out of memory. Efficiency is improved greatly with this optimization technique.

In addition to the sophisticated options for the processing of each user, a sophisticated package of job accounting is provided. The system manager can find out, at any time, which terminals are logged in, what they are running, how long they have been running, etc. A print-out of the utilization of peripherals may also be obtained. The manager can also obtain a summary of system usage, by account number. This information is for intelligently maintaining system accounts and their attributes. It can also be used for determining costs for each user on the system.

In reexamining the original goals, all of the areas have been addressed except one. This one is perhaps the most important one. This requirement is the flexibility and generality of system organization required for evolution through successive waves of technological improvements and the inevitable growth of user expectations. A great deal of time was allocated in the modularity of the monitor. Options which have been added are added as modules controlled by the monitor, rather than making the internal system so complex that it is unmodifible and unsupportable. Using this philosophy, concurrent time sharing, batch, and real-time have been implemented efficiently, yet simply. This concept of sophistication in conjunction with simplicity has allowed a wide range of applications. Educational institutions are running ETOS, including many students who have never seen a computer before. Administrative applications are running concurrently with these novice user jobs, requiring a secure file system. A large defense contractor performs complex engineering calculations while at the same time operators compile documentation under ETOS, using several word processing terminals. The purpose of ETOS and this paper has not been to glorify the technician, but instead to devoteitself to a neglected portion of the computer community, the user.

1-312-[illegible] -3100

REFERENCES

1. F. J. CORBATO and J.H. SALTZER
Multics - The First Seven Years
AFIPS Conf. Proc. 40 1972; AFIPS Press, New Jersey 1972
pp. 571 - 581

2. CHARLES J. SIPPL and CHARLES P. SIPPL
Computer Dictionary and Handbook
Howard W. Sams and Company, Inc., Indiana, 1974
pp. 491 - 526

3. ETOS Systems Manager's Guide V4
EDUCOMP, Connecticut, 1975

4. ETOS Systems User's Guide V4
EDUCOMP, Connecticut, 1975

5. WILLIAM P. BRADLEY
Managing a Virtual Memory Program
Infosystems, Vol. 21, No. 8, 1974, Hitchcock Publishing Co., 1974
pp. 36 - 38

6. W. F. BROWN, R. R. BIBAUD, G. L. HODGKINS
Planning for the Future Computer Complex
Computer Decisions, Vol. 5, No. 1, 1973, Hayden Publishing Company, 1973
pp. 30 - 35

7. ROBERT M. GLORIOSO and TIMOTHY D. CHASE
Design of Virtual Memory for Small Computers
Computer Design, Vol. 12, No. 12, 1973, Computer Design Publishing Corporation, 1973
pp. 67 - 72

8. JOHN R. HILLEGASS
As Timesharing Goes By
Computer Decisions, Vol. 7, No. 4, 1975, Hayden Publishing Company, 1975
pp. 56 - 60