TOPS-10 AND UTS OPERATING SYSTEM COMPARISON

June 13, 1972

R. A. Anderson
D. M. Brown
D. Custis
S. Kravitz
S. Pond

R. A. Anderson
D. M. Brown
D. Custis
S. Kravitz
S. Pond

The purpose of this paper is to provide a hardware/software comparison of the Universal Time Sharing (UTS, Xerox) and TOPS-10 (DEC) operating systems. The paper will begin by discussing, for each operating system respectively, the following topics:

- General Capabilities
- Hardware Configuration Requirements
- File Management
- System Measurement
- Scheduling Techniques
- Reliability and Recoverability
- Time-Sharing Response Time
- Re-entrant Processors
- Program-System Protection
- File Security

The paper will conclude by comparing the strengths and weaknesses of each system relative to the aforementioned topics.

The TOPS-10 Operating System offers by means of a user command language the following capabilities:

- Time-sharing
- Multiprogrammed Batch Processing
- Remote Batch
- Real-Time

This operating system executes batch operations in a background mode (maximum of 127 concurrent jobs) while supporting up to 127 local and remote interactive terminals. The time-sharing system supports such composition, editing, and debugging programs as FORTRAN, COBOL, MACRO-10, BASIC, AID, and EDITOR. The TOPS-10 OS offers a relatively extensive instruction set (366 instructions) including floating point and byte manipulation instructions. The word size of the system is 36 bits and utilizes 7-bit USACII characters. This OS facilitates a parallel mode of operation by allowing data channels and arithmetic processors simultaneous access to separate memory modules, i.e., asynchronous, interleaved memory modules, each containing four ports, are provided. The TOPS-10 OS supports real-time tasks at a microsecond response level by allowing such tasks to interface directly with the priority interrupt system. Real-time response is limited only to the ability of the hardware to respond to interrupts.

The TOPS-10 OS was designed on the concept of modularity, thereby allowing a wide variety of hardware and peripheral equipment to be used by the system. All peripheral equipment is software-supported through the expandability of the DEC-10 monitor; i.e., peripheral handling routines can be added to the monitor with no changes to existing software required. The standard TOPS-10 system requires the following hardware:

- Card Reader
- Card Punch
- Line Printer
- Dectape Transport
- Magnetic Tape Transport
- Disc Drive
- High-Speed Swapping Drum
- Alphanumeric Terminals

The minimum amount of core memory suggested for the TOPS-10 OS is 64K. However, if the system is to support a heavy load of Batch and Time-Sharing users the system will not be effectively utilized at 64K, 128K is suggested. Memory is expandable in 32K modules to 256K on a single processor system and 4096K on a dual processor system.

The file management facilities of the TOPS-10 OS allow for the handling of sequential and random access files. Files allocated to And/Or created by a user are usually referred to by an alphanumeric name which the operating system equates to a particular physical device and location. The system maintains for each user (normally a time-sharing user) a user master catalog which identifies the name, access method, protection level and physical location of each permanent file assigned to a particular user. The number of files a user may create is limited only by a quota imposed on the user by the installation. File storage may be explicitly created by a user before referencing a file or dynamically allocated by the system at execute time. The system provides for simultaneous use of a file among several users through the use of protection codes. The file management capabilities provided by TOPS-10 relieves the user of any considerations of device dependence,; i.e., a user may use a file without any a priori knowledge of the device (disc, pack, disc, drum) on which the information is stored. The system also organizes files independent of access method so that it is not necessary to completely reorganize a file to change from sequential-access to random access methods.

The TOPS-10 OS supplies a standard set of system measurement and tuning facilities. To measure system performance, dumps of selected system tables, scheduling and dispatching queues and accounting information may be requested. A set of system diagnostic routines are also available which monitor system activity and create tables of performance information. This performance information is scanned periodically by report generators which display in a tabular format such statistics as CPU idle time, channel busy time, etc. Tuning facilities exist in the form of SYSGEN options which installations may vary to accommodate their specific needs. At system generation time, the option exists to vary parameters determing time-sharing slice time, queue sizes, core resident portions of OS, etc.

The scheduler is the "nucleus" of the TOPS-10 OS. The scheduler controls the scheduled use of the system; i.e., determines the sequence of time allotments to the users. The scheduler maintains a queue of jobs requesting allocation. The jobs in this queue are ordered according to external priority and system demands. As a job reaches the top of the queue, it passes to a queue maintained by the peripheral allocator. Upon allocation of requested peripherals, the job is passed to the core allocator. After receiving the necessary core, the job is passed to the dispatcher. The dispatcher maintains an ordered queue of jobs ready for execution. As jobs are dispatched from this queue, a "quantum" of execution time is assigned to the job. If the job completes during its quantum, it is passed to a wrapup routine. If the job exhausts its quantum without completing, it is returned to the end of the dispatch queue. If a job is interrupted before completing its quantum due to initiation of an I/O operation, the job is returned to the dispatch queue and will usually regain control when the I/O operation has completed. The number of jobs which are allowed to pass to the scheduler can be controlled by the operator. The operator may type "SCHEDULE N" which has the following effect based on the value of N:

N=0 Normal time-sharing-batch operation

N=1 No further LOGINS allowed

N=2 No further LOGINS from remote terminals

During the course of a job's execution, the job will reside in only one queue at any point in time. Primary among these queues are the following:

- Run Queue
- I/O Wait Queue
- I/O Wait Satisfied Queue
- Sharable Device Wait Queue
- Teletype Wait Queue
- Teletype Wait Satisfied Queue
- Stop Queue
- Null Queue

Recoverability and reliability are very important parts of any system. The TOPS-10 OS functions in conjunction with the following safety features:

. Power FailSafe

. Automatic Restart

. Temperature Protection.

If system power fails, an interrupt is generated to initiate a program which saves all registers and status information needed to facilitiate a restart. If a power surge occurs, a sequence of power-down operations will be initiated and automatic restart will occur when power returns. Similarly, in the case of sudden temperature variations, an interrupt is generated to save register and status information. Besides these hardware features, TOPS-10 provides at discrete intervals snap dumps, core image and system table dumps, file system backup copies, etc. The purpose of these dumps is to facilitate restart and aid in the debugging of system errors.

Time-sharing response time is an important gauge of an operating system's efficiency. Current DEC literature indicates that a uniprocessor system can handle up to 63 terminals and a dual processor system up to 127. However, the literature fails to indicate how response times vary with the number of users. My experience as a TOPS-10 time-sharing user (Irvine) indicates that time-sharing response time is extremely poor when the system is trying to maintain a moderately heavy batch load while supporting more than 10 terminals. The system responds favorably when there are no more than 5 time-sharing users. However, when the number of users exceeds 5, the degradation in response time is quite apparent.

In the TOPS-10 OS, all software which may be requested simultaneously by more than one user is pure code, i.e., re-entrant. The time-sharing Monitor, FORTRAN Compiler, COBOL Compiler, etc., are all re-entrant routines. This re-entrant capability has the effect of conserving core in that a single copy of a re-entrant program may be shared by a number of users at the same time. Re-entrancy is accomplished by two relocation registers which allow a user area to be divided into two logical segments which may occupy noncontiguous areas in physical core. Re-entrant programs are always composed of two segments--a low, nonsharable segment which usually contains just data, and a high, sharable segment which contains instructions and constants. The sharable segment is "write" protected to prevent modification.

User-system protection is accomplished in two ways by the TOPS-10 OS. Firstly, a master-slave operating mode exists which prevents a user from executing any master mode instructions which might impair the integrity of the system. Secondly, user programs are prevented from adversely affecting the system or another user program by assignment of a protection register to a user program. The protection register contains the maximum relative address the user can reference. Thus, if a relative address generated by a user program is greater than the contents of its associated protection register, a monitor trap occurs and the user program is aborted. Thus, no user program can access any storage location except those specifically assigned to the job. As an additional protective measure, all re-entrant code is "write" protected to prevent contamination by unauthorized users.

File security is maintained among users of the TOPS-10 system through the use of protection codes. These codes, which are assigned when the file is created, describe access privileges of the person who created the file and other authorized users. Access codes of 0 to 7 may be assigned to a file where 0 allows all access privileges and 7 allows no access privileges.

Access priviliges specify whether a file may be read, written, executed, updated, renamed, or appended to and to whom such privileges apply. Files may be created with any combination of access privileges, thereby allowing a user to restrict the use of a file in any manner desired. To insure file system integrity and protect the user against inadvertent destruction of his own files, user files are periodically saved on backup storage. This provides the capability of selectively restoring a damaged file or a set of files if a user so requests. Thus, the operating system protects a user's files from unauthorized access by other users and from inadvertent mistakes made by the file originator.

Universal Time Sharing System (UTS), developed by Xerox Data Systems, is an operating system utilized primarily for its time-sharing capabilities. UTS is the result of several years of experience on other XDS operating systems including BCM, BTM (Batch Time Sharing Monitor), RBM (Real-Time Batch Monitor), and BPM (Batch Processing Monitor). In fact, BPM is a subset of UTS giving this operating system all the batch capability of BPM in addition to time-sharing.

UTS provides the following concurrent general capabilities:

- Time-Sharing
- Multiprogrammed Batch Processing ← Remote Batch
- Real-Time Processing

Time-sharing cpabilities allow for up to 128 users to initiate and access the on-line services from a variety of character-oriented terminals. User programs are stored on the high-speed XDS Model 7212 RAD for rapid access to core memory for execution. Batch processing may be initiated via 1) local batch, 2) remote batch using high-speed remote batch terminals, or 3) terminal batch. Batch and time-sharing programs are completely compatible so that large batch programs may be checked out in the time-sharing environment for monitoring purposes and readily entered into the batch stream when the user desires to run large amounts of data. This removes the necessity for a user to sit idle at a terminal during a long run. Real-time processing consists of core-resident real-time programs utilizing the Sigma high-speed hardware interrupt system. Nonresident modules may be established at SYSGEN time. Interrupts may be armed, enabled, and triggered via program control. When an interrupt occurs, foreground action takes place on a high priority basis. In addition, peripherals may be dedicated to real-time processing at system generation time to insure the fastest response time.

UTS offers META-SYMBOL (assembly language), COBOL, BASIC, and FORTRAN IV in addition to various application languages (GPDS, 1401 SIM, SL-1). Features for manipulating programs include capability for:

- Creating or modifying source files
- Creating relocatable object modules (ROM's)
- Creating load modules
- Modifying object module symbols
- Load and execute object programs with library search
- Executing object programs under control of one of the debugging systems
- Resume execution of programs interrupted or stopped by the user for debugging purposes
- Save core image of program being executed and retrieve it for continued execution
- Copy and delete permanent files.

An EDIT system enables the user to modify files with ease. In addition, various other processors enable the user to manipulate files and other items related to his terminal environment. The file system is device independent, meaning the user need not modify his program to change from one I/O device to another.

UTS operates on the Sigma 6, 7, or 9. We will discuss the Sigma 7 hardware. The Sigma 7 system consists of basically:

- Memory of up to 8 magnetic core storage units
- A central processing unit (CPU) which addresses core memory, fetches and stores information, performs arithmetic and logical operations, sequences and controls instruction execution, and controls the exchange of information between core and other elements of the system.
- Input/output system controlled by one or more input/output processors, each providing data transfer between core memory and the peripheral input/output devices and operating simultaneously with the CPU.

The Sigma 7 is a word-oriented virtual memory machine, a word consisting of 32 bits plus a parity bit.

- Data quantities in core addressable as 1-, 2-, 4-, or 8-byte quantities.
- Memory expandable from 8,192 to 131,072 words in increments of 8,192 or 16,384 words. Recommended minimum system is 80K.
- Sixteen general-purpose registers, expandable to 512 in increments of sixteen.
- Hardware memory mapping (optional)
- Real-time priority interrupt system with 240 levels. Interrupts may be armed, enabled, and/or triggered via program control.
- Interruptability of "long" instructions to insure faster response time.

The Sigma 7 supports a wide variety of peripheral equipment, including magnetic tape units (7- or 9-track), rapid access disc, card readers, card punches, display terminals, keyboard printers, line printers, paper tape readers and punches, graph plotters, and data communication equipment. A typical system might consist of CPU with memory map connected via multiple memory ports to core memory modules, a selector I/O processor for handling the swapping RAD, and a multiplexor I/O processor for handling a card reader, line printer, card punch, tape unit, disc unit, and communications controller with several remote terminals.

Memory management is done via a hardware memory map. Physical and virtual memory is segmented into 512-word sections called pages. Whereas virtual memory is contiguous, physical memory need not be. Physical pages are scattered throughout core based on their availability. The hardware memory map which is loaded into the user's JIT specifies where a user's virtual memory page is physically located. Sigma 7 utilizes the "all-in" or "all-out" concept for loading and execution of programs in core.

The UTS file structure is designed for utilization of random access storage media. Three types of files may be used, namely 1) consecutive or sequential, 2) keyed or indexed sequential, and 3) random. Consecutive are those which are created and accessed in order. Keyed files consist of a collection of related records, with each record identified by a key. Random files provide the user with a contiguous random access storage area to be organized at his own discretion. The file organization itself consists of a system of pointer tables. A master catalog of accounts points to the file directory location for each user. The file directory points to the file information table (FIT) which in turn points to the files.

UTS maintains a system of performance monitoring to facilitate system measurement and tuning. Results are displayed via a dedicated terminal and items are displayed at regular intervals. Individual installations may select which types of information they want to monitor, which may include CPU use, processors (CPU time), processors (number of users) I/O rates, console time, number of users, number of interactions. Reports may be generated with specified system functions being measured. Based on this information, a system may be generated (SYSGEN) or modified to tailor the system to a particular installation or job stream. In addition, XDS uses a unique hardware measuring piece of equipment called ADAM. This equipment enables the user to measure instruction utilization, in counts or time, particular routine usage in the monitor (for possible modification of monitor overlay structure), disable time for interrupts, etc.

The swapping and scheduling algorithm is event (carriage return, interrupt, time slice exhaustion, etc) driven. Each job is in one of 29 processing states. Jobs are queued within each state. The 29 states are given an order of execution and an order for swapping. The first state group in the execute list is searched. First in core gets the CPU and first out on RAD gets a swap in priority. The swap list is then searched from bottom to top for first in core, which is then swapped out. Three quanta are observed to maximize efficiency. They are as follows:

- QMIN — Minimum time slice (may not be interrupted until this time is used); usually 50-150 ms.
- QMAX — Maximum time slice (may be interrupted before max time)
- SQUAN — Swap quanta keeps job in core until it has used specified amount of CPU which avoids overhead in swapping or thrashing.

UTS has various reliability and recoverability features which include:

- Watch dog timer for detection of software or hardware errors causing system hangup
- Power fail-safe interrupt to save and restart the system in the event of power failure
- Memory parity error checking
- Error log
- Automatic system restart or recovery
- Snapshot of failed monitor
- Diagnostic peripheral exercisors.

Diagnostic data may be processed by diagnostic programs which will analyze core dumps.

Response is a most important aspect of a time-sharing system. Response time of a relatively short period of time may seem like an eternity to a user sitting at the terminal. It is also difficult to gauge. Obviously, the more loaded down a system becomes with users will result in slower response time. UTS advertises an average response time of 2 seconds in its pamphlet, which, if holds true, is quite excellent. My limited experience with UTS at UCI indicates that the response time is very good even with a good number of users (20 or more) on-line.

With a 25-32K core resident monitor, shared processors (re-entrant routines) are a necessary feature of UTS. They help to minimize the amount of core needed by the monitor and swap time as well. UTS has approximately 40 share processors which include its compilers. It has approximately 30 nonshare processors. In addition, the some 86 monitor modules are structured into 7 overlay segments to minimize core requirements for the resident monitor.

Program-system protection is maintained by a master/slave mode concept and by keys and locks. In master mode, anything goes (monitor mode). In slave mode (user programs), "privileged" instructions are denied. These are instructions relating to I/O or those changing the basic control state of the computer. Also, each memory map page has a 2-bit access control code for each page indicating write-read-access, read-access, read, or no permissions for that particular page. Independent of the memory map is the key-lock concept. Each page of actual core memory has a 2-bit write lock. Write keys are carried with a user addressing core. Writing is permitted only if 1) wirte key is "00" (skeleton key), 2) write lock is "00", or 3) write key equals write lock.

File security is maintained by some of the usual password and file permission conventions. Files also have an alphanumeric name associated with them. Jobs can create files only under their own account number. Periodic file dumps are taken to maintain recent copies of files in case of system failure.

Two large systems such as UTS and TOPS-10, while alike in many ways, have a multitude of design and philosophy differences. Both systems provide the same general capabilities. The TOPS-10 system seems to support a mixed environment of batch, time-sharing, and real-time tasks to better advantage than does UTS. However, UTS provides more sophisticated time-sharing services and can sustain a heavier time-sharing load than can TOPS-10. The 36-bit word length of the TOPS-10 system offers some advantages over the 32-bit word of the UTS system. A 36-bit word allows for greater numerical precision in arithmetic operations. The 36-bit word also allows for the storage of five 7-bit USACII characters/word while the 32-bit word allows for the storage of only four EBCDIC characters/word.

Both systems support hardware of approximately the same access and/or transfer rates. The UTS system, however, is unique in its use of a rapid access disc to decrease swapping overhead. In regard to memory organization and space, the TOPS-10 system utilized two-way or four-way interleaved memory modules, whereas the UTS system does not. Secondly, the UTS system partitions core between batch and time-sharing operations, whereas partitioning is not performed under UTS. The use of interleaved memory modules is of significance, for this allows channels and the arithmetic unit to concurrently access memory. Memory protection is accomplished under TOPS-10 by protection registers which impose a limit on the domain of relative addresses generated by a user while UTS utilizes a lock and key method; which is the better protection scheme is debatable.

In the area of reliability and recoverability, use of re-entrancy program system protection and file protection, neither system presents any distinct advantages over the other. Both systems provide hardware and software recoverability mechanisms and insure the integrity of the file systems by maintaining duplicate file copies. Both systems maintain a master-slave mode to prohibit a user from operations detrimental to system integrity.

The general conclusions drawn from the comparison of the TOPS-10 and UTS systems are the following:

- The TOPS-10 system seems to support a mix of batch, time-sharing, and real-time tasks more effectively than the UTS system.

- The TOPS-10 OS provides better real-time response than UTS.

- The UTS OS supports the time-sharing environment more effectively than TOPS-10.

Inter-Office Memorandum

To Distribution

Date May 20, 1974

From Jerry Packer

Location Atlanta

Subject DEC COMPETITIVE REPORT

Organization XCM - JP039

XEROX

Attached is a very interesting report that was written by Jon Stewart who is now at the University of Southern Mississippi. As some of you old-timers may remember, Stewart was the individual at L.S.U. who wrote the damaging Sigma 5 BTM/DEC-10 comparison report (Circa 1968). He subsequently worked for DEC as an analyst and supported the University of Mississippi account which has turned out to be something less than a success for DEC. The report is somewhat personal and is written to his old cronies at DEC. However, he draws some interesting comparisons between the DEC and Xerox Systems, both hardware and software. Happy reading!

Distribution:

Atlanta District Personnel
RMM's (J. Bedford, R. Giordano, K. Felderstein)
Eli Anfenger C2-12
Phil Becker I4-05
Ed Bryan A1-85
Bob Donaldson I5-12
Dave Escoffery
Mike Harvey I5-06
Dick Litschgi A1-03
Bob Mariani I4-05
Larry Tashbook I4-05
Bill Walker I8-03

OOOO: JON,STEWART 03/17/74 13:12

OOOO: JON,STEWART 03/17/74 13:12

OOOO: JON,STEWART 03/17/74 13:12

OOOO: JON,STEWART 03/17/74 13:12

TO: ALL GOOD REBELS (IN THE GENERIC SENSE)

FROM: JON A. STEWART, UNIVERSITY OF SOUTHERN MISS.
BOX 267, SOUTHERN STATION, HATTIESBURG,MS.39401
HOME ADDRESS: 705 HILLENDALE DR.,H'BURG,MS.39401
HOME PHONE: (601) 544-5755

I AM TAKING THIS OPPORTUNITY TO BRING SOME OF MY GOOD FRIENDS UP-TO-DATE ON BOTH THE XEROX SIGMA 9 AND MY OWN RETURN TO "ACADEME"--(WHAT I DID AT OLE MISS COULD HARDLY BE CALLED "ACADEMIC"). ALSO, I HAVE AN ULTERIOR MOTIVE IN SOUNDING OUT THE POTENTIAL FOR EMPLOYMENT OF ANOTHER GOOD REBEL--AND VERY CAPABLE SYSTEMS PROGRAMMER-- WHO IS CURRENTLY EMPLOYED HERE AT USM, BUT WHO WAS FORMERLY EMPLOYED BY XDS IN THE SIGMA (6,7,9) BENCHMARK GROUP AT EL SEGUNDO-- SHADES OF BOB GOOD AND HOGAN!!

FIRST AND FOREMOST, ASHLEY, THE UTS SYSTEM (NOW CP-V BUT USM IS WAITING FOR 800 RELEASE-- NOW RUNNING UTS DOO) IS QUITE A GOOD ONE-- THOUGH IT HAS A FEW SERIOUS DESIGN FLAWS-- PRIMARILY IN THE MEMORY AND FILE MANAGEMENT AREAS-- WHICH DO PROVE A HANDICAP (THOUGH NOT SEVERE WITH CURRENT LOADING HERE) AT TIMES. !!!BUT!!! THE SIGMA 9 HARDWARE IS REFRESHINGLY STABLE COMPARED TO MOST OF THE DEC-10 INSTALLATIONS I HAD THE OPPORTUNITY TO BE AROUND FOR AN EXTENDED PERIOD OF TIME. SINCE COMING HERE IN OCTOBER WE HAVE PROBABLY HAD LESS THAN 8 TO 12 HOURS OF UNSCHEDULED DOWN-TIME (EXCLUDING SOFTWARE-RELATED PROBLEMS AND OPERATIONS SNAFUS). MEMORY (AND CHANNEL) PARITY ERRORS ARE JUST NON-EXISTENT (WELL, PERHAPS ONE IN THAT FIVE MONTH PERIOD). MEMORIES ACTUALLY CHECK PARITY-- SO THERE IS SUCH A THING AS A BUSS CHECK WHICH RESULTS WHEN PARITY IS GOOD AT ONE END OF THE MEMORY BUSS BUT BAD AT THE OTHER END); ALSO, THE MAP REGISTERS AS WELL AS OTHER ADDRESS-RELATED PATHS DO CARRY AND CHECK PARITY. THE DISKS (EXCUSE ME, DISCS -- A LA XEROX) WE HAVE ARE 7262 UNITS (ABOUT LIKE RPO3'S IN CAPACITY, BUILT BY XEROX, BUT NOT ANY LONGER)-- THEY ARE THE WEAKEST HARDWARE COMPONENT AT PRESENT-- REQUIRING FREQUENT ALIGNMENT BECAUSE OF ELECTRONIC DRIFTING CAUSED BY A COOLING PROBLEM (ACCORDING TO F/S). THEY WILL BE REPLACED EVENTUALLY WITH 7275 (XEROX DESIGNATION) DISKS, WHICH ARE 3330-LIKE UNITS BUILD BY CDC. MAG TAPES GAVE TROUBLE HERE AFTER INITIAL INSTALLATION (1600 BPI, 120 KB) BUT AFTER INITIAL SHAKEDOWN THEY'VE HARDLY SQUEAKED ONCE. CARD READERS AND PRINTERS (WHICH ARE HEAVILY USED-- ESPECIALLY IN THE ACADEMIC BATCH LAB WHICH EXISTS REMOTELY-- ACTUALLY ABOVE THE MACHINE ROOM) HAVE BEEN MUCH MORE RELIABLE THAN WERE THOSE AT OLE MISS-- AND THE HEAVY DUTY EDF PRINTER (XEROX DOESN'T CLASSIFY IT THAT WAY, USM DOES) WHICH WAS , I BELIEVE, BUILT BY NCR IS MUCH BETTER QUALITY THAN THE DATA PRODUCTS PRINTER-- ABOUT THE BEST PRINTING QUALITY OF ANY DRUM PRINTER I'VE SEEN. RIBBON CONSUMPTION WHICH BECAME A COSTLY PROBLEM AT OLE MISS IS NOT A PROBLEM ON THESE PRINTERS (PERHAPS HEAVIER THAN ON IBM BUT NOT NEARLY AS BAD AS THOSE DATA PRODUCTS PRINTERS). CARD READERS AND THE PUNCH HAVE READ-CHECK STATIONS (I.E., A SECOND READ STATION FOR THE READERS); THE CARD PUNCH (FROM UNIVAC) DID GIVE MULTIPLE PUNCH-CHECKS-- WHICH CAUSED OPERATIONAL GRIEF-- BUT IT HAS BEEN SLOWED DOWN AND DOESN'T SEEM TO CAUSE MUCH COMPLAINT NOW.

THE HEAVIEST CPU LOADING WE'VE HAD ON THE SYSTEM WAS A SIMULATED (STIMULATED ??) ONE-- 16 BATCH JOBS, 10 TIME-SHARING, AND 7 GHOSTS-- SOMEWHAT LIKE DETACHED JOBS). IN THIS SITUATION THE SLOW-SPEED RAD (ONLY 384 KB TRANSFER) WHICH XEROX UNFORTONATELY BID (IN ORDER TO UNDER PRICE DEC??) JUST COULDN'T HACK THE SWAPPING LOAD AND LOST TIME EXCEEDED 10%. MONITOR OVERHEAD, AS THEY REPORT IT, NEVER EXCEEDED 7% DURING THAT RUN. IN THAT LOAD STUDY THE 30 USER JOBS (2 GHOSTS ARE NECESSARY TO RUN THE SYSTEM) WERE ALL 20K IN SIZE AND COMPUTE-BOUND AND RAN INDEFINITELY-- I.E., UNTIL CANCELLED; THE UTSPM (UTS PERFORMANCE MONITOR) WAS RUN CONCURRENTLY TO EXAMINE THE STATISTICAL DATA BASE BUILT-IN AND COLLECTED BY THE MONITOR. THE FIGURES WERE REPEATABLE AND CONSISTENT WITH FINAL JOB TIMES REPORTED AFTER THE JOBS WERE TERMINATED. TO COMPARE WITH DEC IT WOULD BE NECESSARY TO ADD THE 5 (OR SO) JOBS NECESSARY TO RUN THE BATCH SYSTEM. THE REASON FOR THE LIMIT OF 10 T/S JOBS WAS THAT'S ALL THE PORTS WE HAD THEN (NOW UP TO 15, GOING TO 24 BY JULY, WE HOPE).

THE BATCH THROUGHPUT OF THIS SYSTEM IS QUITE EXCEPTIONAL-- PROBABLY TWO-THIRDS OF THE STUDENT JOBS ARE UNDER FLAG-- BUT MANY ARE BIG JOBS (COMPILER AND ASSEMBLER PROJECTS) RUN UNDER META-SYMBOL OR EXTENDED FORTRAN IV. ALSO, A COBOL COURSE OF SOME 60 OR SO STUDENTS ADDED AN ADDITIONAL HEAVY LOAD TOWARD THE END OF THIS QUARTER (JUST FINISHED). MY BROTHER RIC IS NOW AT THE UNIVERSITY OF HOUSTON (1108) AND OF COURSE, MISS. STATE NOW HAS AN 1106, SO I EXPECT TO GET SOME CONFIRMATION OF MY FEELING THAT THE BATCH ON THE SIGMA 9 IS AS GOOD, OR MAY EVEN EXCEED, THAT OF THE EXEC-8 1108 SYSTEM-- WHICH IN MY FEW BRIEF CONTACTS WITH IT WAS NOT REALLY THAT IMPRESSIVE (THOUGH IT HAS A GOOD MULTI-PROGRAMMED BATCH REPUTATION). ONE THING'S FOR SURE THE XEROX ARCHITECTURE IS MUCH BETTER FOR THE MIXED TIME-SHARING AND BATCH ENVIRONMENT THAN THE 1106/1108 OR EVEN THE 1110 (WHICH SOME OF YOU KNOW I HAD A CHANCE TO STUDY BEFORE LEAVING OLE MISS). OUR USUAL "REAL" PEAK LOADING IS 24 JOBS (6 OR SO BATCH STREAMS, 15 T/S JOBS, 3 SYSTEM GHOSTS -- WHICH ARE ACTUALLY MONITOR EXTENSIONS LIKE DAEMON). THE ENCLOSED NEWSLETTER WILL TELL YOU SOMETHING ABOUT THE CHARACTERISTICS OF THE LOADING-- AND OTHER STUFF. ACTUALLY THE END OF THIS QUARTER (MID-FEBRUARY) SAW THE HEAVIEST REAL LOADING YET-- THE COBOL, ASSEMBLER AND COMPILER CLASSES AS WELL AS SOME SNOBOL, APL, SL-1-- AN ANALOG SIMULATOR-- AND CIRC-- CIRCUIT ANALYSIS PACKAGE, AND THE USUAL END-OF-PERIOD EDP,ETC.

NOW ABOUT T/S--AND I HAVE NO COMPLAINTS EXCEPT THE EDITOR (USING IT NOW) AND THE LACK OF SOME OF THE USEFUL MONITOR-RELATED-UTILITY FUNCTIONS--E.G., WILD CARD DIRECTORY SEARCHES, FILE-EXTENSIONS, RE-NAMING AND PROTECTING FEATURES; THE LATTER TWO FUNCTIONS ARE DONE BY COPYING; OF COURSE THERE IS NO FILE-EXTENSION, BUT FILE PASSWORDS AND EXTENDED NAMING (UP TO 31 CHARACTERS, USUALLY A LIMIT OF 10 OR 11 IMPOSED BY SUBSYSTEMS) ARE REALLY MUCH MORE VALUABLE IN MANY WAYS. ALSO THE PREVALENCE OF THE KEYED FILE (LIKE ISAM ONLY A MONITOR ACCESS METHOD--WIDELY USED BY ALMOST ALL SYSTEM PROCESSORS) IS A NOTABLE UNIFYING INFLUENCE TO CROSS-LANGUAGE COMMUNICATIONS-- SUCH AS I HAVE BEEN DOING RATHER EXTENSIVELY WITH BASIC, APL, FORTRAN AND SNOBOL. TO GO ON...

1) GENERALLY, MEMORY MANAGEMENT (DUE TO DEDICATED VIRTUAL ADDRESS SPACE, THE MAPPING REGISTERS,ETC.) IS A MAJOR PROBLEM WITH UTS-- THOUGH NOT PARTICULARLY CONCERNING ANYONE HERE AT THIS TIME; OF COURSE, THERE IS A "GOOD"-- SOME WOULD SAY OTHERWISE-- OVERLAY LOADER WHICH MAKES THE 64K USER-VIRTUAL-ADDRESS RESTRICTION NOT TOO MUCH OF A PROBLEM; ALSO, SOME PROCESSORS (E.G., A SIGMA 9 SIMULATOR, THE LISP INTERPRETER WRITTEN BY DAVE HENDERSON) HAVE BUILT-IN DEMAND PAGING.

now 92K

2) SYSTEMS MAINTENANCE IS SOMETHING I'VE NOT HAD TO DEAL WITH MUCH HERE-- BUT I DO WORK SOME WITH THE SYSTEMS GROUP AND MY IMPRESSION IS THAT, IN GENERAL, IT'S ABOUT 10 TIMES BETTER THAN SAY WITH IBM-- BUT PERHAPS TWICE AS DIFFICULT AS WITH THE DEC-SYSTEM 10; THE DPSP PROCESSOR WHICH ASHLEY IS AWARE MUST BE USED TO ADD SHARED PROCESSORS IS NO BIG DEAL-- THERE ARE A FIXED NUMBER OF SLOTS SET UP FOR SHARED PROCESSORS, BUT THAT CAN BE MADE HIGH ENOUGH TO ALLOW EASY ADDITION OF OTHERS AT ANY TIME-- FOR EXAMPLE, SNOBOL4 GOES IN AND OUT OF THIS CATEGORY FROM TIME TO TIME, AS DOES THE SIGMA 9 SIMULATOR,ETC. IT IS SAFE TO SAY THAT SYSGEN CONSUMES A LOT MORE CPU TIME THAN WITH DEC-- ESTIMATES RANGE FROM 45 MINUTES TO 1 HOUR OF CPU-TIME-- AND THERE ARE A LOT MORE KLUDGEY STEPS INVOLVED; I'LL BE TEACHING AN OPERATING SYSTEMS COURSE THIS NEXT QUARTER AND WILL KNOW A LOT MORE WHEN I FINISH THAT ONE. CAN YOU GET ME SOME FREE DEC-10 TIME SOMEWHERE TO USE IT FOR COMPARISONS AGAINST UTS?? DON'T SAY OLE MISS. IN GENERAL DELTA (LIKE DDT) IS MUCH BETTER THAN DDT-- AND I FORGOT TO MENTION, THERE IS A GOOD FORTRAN DEBUG PACKAGE CALLED FDP, WHICH IS QUITE POWERFUL AND EASY TO USE; ANOTHER ASIDE, COBOL DEBUGGING ON-LINE IS NOT TOO GREAT,YET, BUT THEY SAY THERE WILL BE AN INTERACTIVE DEBUGGING PACKAGE SOME DAY. NOW BACK TO MONITOR MAINTENANCE; PATCH ***CARDS*** NO LESS MUST BE READ-IN AGAIN AFTER RE-LOADING THE MONITOR FROM THE "PO" TAPE (ESSENTIALLY MAGRIM, A MONITOR AND ALL OF SYS). THE PATCHED COPY IS THEN WRITTEN ON THE RAD, AND THEREAFTER UNLESS PROBLEMS DEVELOP (NEW PATCHES TO BE ADDED) BOOTING OCCURS VERY EFFICIENTLY FROM THE RAD. ANOTHER INTERESTING POINT: AUTOMATIC RECOVERIES SEEM TO WORK VERY WELL, IN ABOUT 30 SECONDS OR LESS; THUS ALLEVIATING ANY NEED FOR A "SMART/FAST" OPERATOR--YES, ASHLEY, THERE IS AN OPERATOR! SYSTEMS PEOPLE GET AN AUTOMATIC DUMP,ETC. TO LOOK AT LATER AT THEIR LEISURE. ALSO, THE POWER-FAIL-AUTO-RESTART SEQUENCE

USES WORK AND COUPLING THAT WITH THE REQUIRED MOTOR-GENERATOR SET MEANS THAT POWER FAILURES (EVEN THE UP AND DOWN KIND) DON'T REALLY CAUSE MUCH CONCERN-- WHICH WAS NOT THE CASE AT OLE MISS DURING THE STORMY SEASON.

3) THE FILE MANAGEMENT (I.E., DISK SERVICE,ET AL) SEEMS QUITE INFLEXIBLE AS FAR AS CONFIGURATIONS AND PARAMETERS GO-- LIKE ONE BIG PUBLIC STRUCTURE (WHICH SINKS OR SWIMS TOGETHER !); BUT, OTHER THAN THE OCCASIONAL HARDWARE REALIGNMENT-- AND SOMETIMES ACCOMPANYING REFRESH AND RESTORE, THERE HAVE BEEN VERY FEW PROBLEMS WHICH WERE APPARENT TO THE USERS. I HAVE HAD ONE FILE LOST SINCE I CAME HERE FIVE MONTHS AGO, AND HAVE NOT HAD TO SPECIFICALLY RECOVER ANY FILE FROM THE BACKUP TAPES (SINCE THAT ONE FILE DIDN'T BOTHER ME ANYWAY). CRASHES ARE VERY INFREQUENT AND THERE IS AN AUTOMATIC SPACE-RECOVERY TECHNIQUE (CALLED HGP RECONSTRUCT-- FOR "HEADER GRANULE PACK") WHICH GETS BACK THE LOST-BLOCKS WHICH MIGHT RESULT FROM CRASHES, AND AT VARIOUS OTHER TIMES (DURING BOOTING OR OPERATOR RECOVERY) THERE ARE BUILT-IN CONSISTENCY CHECKS WHICH VERIFY THE INTEGRITY OF THE FILE SYSTEM. SINCE THE EDITOR USES KEYED FILES, AND IS UPDATING LINE BY LINE, YOU SELDOM LOSE ANYTHING WHEN A CRASH OR AUTO-RECOVERY DOES OCCUR. ALL OF THIS CONTRIBUTES TO A MUCH MORE STABLE SYSTEM-- ESPECIALLY TO THE UNSOPHISTICATED USER.

4) FILE SYSTEM THROUGHPUT HAS NOT REALLY BEEN PUSHED HERE AT ALL-- AND I COULDN'T GUESS AT ITS ULTIMATE POTENTIAL HAVING OBSERVED ONLY VERY SLOW DISK-ACCESS RATES; UTS DOC DOES LITTLE (OR NO) OPTIMIZATION AND FILLS UP THE ONE PUBLIC STRUCTURE FROM ONE END GOING THAT AWAY...SO PROBLEMS DO OCCUR EVEN NOW WITH FILE CONTENTION. ALSO, THERE IS NO DKSPRI SO EVERYONE SEEMS TO GET EQUALLY BAD FILE-SYSTEM RESPONSE WHEN SOMETHING WEIRD AND WONDERFUL IS GOING ON-- I HEARD FROM MIDDLE TENN STATE UNIVERSITIES THAT THEY EVEN OBSERVED SOME LINE PRINTER SLOWDOWN DURING A BENCHMARK RUN BY XEROX (WHICH I UNDERSTAND DEC WON HANDILY, BUT HONEYWELL GOT THE BUSINESS ANYWAY)-- WHICH COULD HAVE BEEN DUE TO THIS INABILITY TO GET ANY DATA TO PRINT, ASSUMING SYMBIONT SPACE ON PACK.

5) NOW, WE TALK ABOUT QUEUE MANAGEMENT AND SOME MORE ABOUT BATCH. AND I HOPE CHUCK O'TOOLE HAS REALLY GOT THE OLD SYSTEM SWINGING BY NOW, AS HE DID AT PITTSBURGH! IT IS VERY NICE TO SEE,HEAR,FEEL (WHEN YOU HAVE TO DEAL WITH BATCH, OR JUST GET A LISTING) IMMEDIATE RESPONSE. THERE JUST AREN'T ANY DELAYS FOR UTS QUEUE-MANAGEMENT FUNCTIONS. SUBMITTING 16 BATCH JOBS FROM ONE TERMINAL SUBMITTED JOB (TO GET A LOADING) RESULTS WITHIN A FEW SECONDS (SAY 5) IN 16 STREAMS BEING FIRED UP-- WHEN THE CONTROLS SO ALLOW. BATCH, BY THE WAY, REQUIRES NO LOGON (A SMALL SECURITY BREACH-- SINCE ANYONE KNOWING ACCOUNT AND NAME CAN LOGON UNDER THE SUPER-PRIVILEGED :SYS ACCOUNT-- NATURALLY STUDENTS DISCOVERED THIS BY READING THE FORTRAN DOCUMENTATION, WELL ??) AND THERE IS "DEDICATED" SYMBIONT (SPOOLER) SPACE WHICH HOWEVER OVERFLOWS FROM RAD TO PACK IF REQUIRED. GENERALLY PRINTING AND CARD READING OCCUR INSTANTANEOUSLY WHEN THE APPROPRIATE DEVICE(S) ARE NOT OTHERWISE BUSY. ALSO FROM AN OPERATIONAL POINT-OF-VIEW BATCH HERE IS MUCH, MUCH SIMPLER THAN THE DEC VERSION (I'D SAY ABOUT THE SAME FACTOR AS BETWEEN SYSTEMS MAINTENANCE ON THE DEC 10 AND IBM 360/370 OS/VS1/VS2/VM...). IN OTHER WORDS, IT IS POSSIBLE FOR AN IDIOT TO OPERATE THE SIGMA 9 COMPUTER-- IT IS A QUIET JOKE HERE THAT, ON OCCASION, THIS HAS ACTUALLY BEEN OBSERVED (BY THE GREMLINS AT NIGHT) AND I KNOW FOR A FACT THAT THIS EASE-OF-OPERATION WOULD HAVE HELPED IMMENSELY AT OLE MISS, THOUGH UNFORTUNATELY NOT SOLVING ALL THEIR PROBLEMS. TO MAKE A LONG STORY SHORT-- IF YOU HAVE TO HAVE BATCH (AND THERE ARE THOSE WHO SAY "YES") THEN HAVE IT GOOD , FAST AND EFFICIENT SO THAT OTHERS CAN GET SOME WORK DONE. MONITOR OVERHEAD DURING OUR PEAK LOADING TIMES (WHICH HAVE BEEN UP AROUND 70-80% OF THE AVAILABLE CPU, WITH CONSIDERABLE LOST TIME DUE TO THE SLOW RAD) HAS NEVER EXCEEDED 10%-- THAT IS, SWAPPING, SCHEDULING, I/O, THE SPOOLERS ALL INCLUDED. OF COURSE THIS IS PARTLY DUE TO LOW CPU-REQUIREMENTS FOR I/O SERVICING BECAUSE OF A DECENT MULTIPLEXER CHANNEL.

6) THE UTS SCHEDULER LOOKS AT A PARAMETER CALLED THE BATCH BIAS-- WHICH , HERE, IS ALWAYS SET TO FAVOR TIMESHARING-- TO GET THE BEST RESPONSE POSSIBLE FOR TIMESHARING. HOWEVER, WHEN BATCH BIAS IS SET THIS WAY AND THERE IS A TIMESHARING JOB WANTING THE CPU IT SEEMS THAT IT WILL ALWAYS GET IT FIRST. SO I STARTED UP A 32K BASIC JOB WHICH SAID "10 GOTO10" AND IT SEEMS THAT THE BATCH SYSTEM SCREECHED TO A "HALT." WELL, NEEDLESS TO SAY SOMETHING WILL HAVE TO BE DONE ABOUT THAT EVENTUALLY--

MAYBE CP-V HAS ALREADY ADDRESSED THAT PROBLEM. IN GENERAL, CONTRARY TO MY OWN EXPERIENCE WITH DEC-10 BATCH, WHICH I FUTILELY TRIED TO MINIMIZE, THE CONTROLS FOR BATCH UNDER UTS ARE QUITE GOOD. LOGICAL PARTITIONING WORKS VERY WELL FOR RESOURCE ALLOCATION CONTROL -- NO CONFLICTS CAN ARISE FOR MULTIPLE TAPES, MULTIPLE LARGE OR CPU-CONSUMING JOBS,ETC. SINCE THE LOGICAL PARTITIONING AND ALLOWED NUMBER OF ACTIVE STREAMS CONTROL THIS QUITE WELL.

7) COBOL PRODUCTION EDP IS QUITE EFFICIENT (RE: HOGAN STUDY ON THE COMMERCIAL INSTRUCTION SET). WHEREAS AT OLE MISS (WITH OPERATOR AND SYSTEMS PROGRAMMER SNAFUS, BATCH BOMB-OUTS, COBOL INEFFICIENCY, LARGE COMPILING RUNS-- 60K PLUS FOR THE ACCOUNTING SYSTEM-- A LA DAVE K., JOHN C. & DON F.-- DID I MISS ANYONE??) IT WAS FAIRLY ROUTINE FOR END-OF-SEMESTER ACCOUNTING, STUDENT RECORDS,ETC. TO CONSUME 30-40% OF THE CPU OVER SEVERAL DAYS RUNNING, I HAVE YET TO OBSERVE MORE THAN ABOUT 15% CPU-LOADING DUE TO THE EDP GROUP HERE EVER A SIMILAR PEAK-LOAD PERIOD, AND USM IS LARGER IN ENROLLMENT BY A FEW-HUNDRED THAN OLE MISS; BUT NOT DOING QUITE AS MUCH EDP DEVELOPMENT WORK; ALTHOUGH THEY DO HAVE A COUPLE OF ON-LINE ADMINISTRATIVE APPLICATIONS UNDERWAY (ADMISSIONS AND PERSONNEL) ALREADY.

8) LANGUAGE PROCESSORS, UTILITIES AND THE APPLICATIONS SOFTWARE USED AT USM SO FAR (SL-1, BIOMED, CIRC, GPDS , OTHERS HAS BEEN GOOD TO EXCELLENT IN QUALITY. VERY STABLE. WORKS AS DOCUMENTED. VERY GOOD DOCUMENTATION (PERHAPS THE BEST IN THE INDUSTRY FOR THE SCIENTIFIC USER-- MUST BE THE SDS GROUP). I HAVE PERSONALLY USED FLAG, EXTENDED FORTRAN-IV, DELTA,FDP, PCL (LIKE PIP), THEIR ONE EDITOR (EDIT), BASIC, APL, AND HAV YET TO FIND A SERIOUS BUG IN ANY OF THEM (OTHERS HERE HAVE COMPLAINED BITTERLY ABOUT FLAG... I HAVEN'T USED IT MUCH). FLOYD-- XEROX APL IS EXCELLENT! DON'T LET ASHLEY TELL YOU OTHERWISE. DAVE HENDERSON SAYS XEROX CORPORATE REALLY DOES USE IT INTERNALLY FOR THINGS LIKE FORECASTING (DID YOU EVER HEAR WHAT HAPPENED TO THE MORMON CHURCH--DID THEY GO WITH XEROX?). APL ROOT SEGMENT IS 27 PAGES (SAY 14K) AND IT HAS FOUR OVERLAYS ALL 5 PAGES OR LESS. IT HAS A VERY GOOD FILE I/O PACKAGE (BETTER THAN THAT OF APLSS) AND IN THE RECENT RELEASE HAS A BUILT-IN GRAPHICS FEATURE-- FOR USE WITH THE TEKTRONIX 4013 OR 4010. I AM BUILDING A GOOD APL LIBRARY (WE DON'T HAVE A 2741 YET BUT XEROX DOES SUPPORT THIS TERMINAL AS A STANDARD ITEM-- FREE-OF-CHARGE). SNOBOL4 WAS OBTAINED FROM U.T./ARLINGTON AND IT IS O.K. BUT A REAL CPU- AND CORE-BURNER. I'M NOW CONVERTED TO SNOBOL FROM APL AND READ MANUALS AT HOME IN BED-- OF COURSE, BOB GOOD REACHED THAT POINT EVEN EARLIER WITH COBOL-- BUT THEN I NEVER BELIEVED HIM EITHER. AND HOW'S CAROL & JENNIFER-- THEY SHOULD BE BE OK WHEN THEY GET WARMED UP AGAIN-- IT DOESN'T TAKE LONG! ALSO, THERE'S NO GOOD REPLACEMENT FOR RUNOFF ("TEXT" IS VERY 2741-ORIENTED AND, AT A QUICK GLANCE, NEEDS SOME HUMAN-ENGINEERING, AS DOES THIS EDITOR, WHICH IS CERTAINLY NO SOS, OR EDITS). FLOYD, CAN YOU GET ME THE APLSS "SOS-LIKE" EDITOR WRITTEN IN APL? IS IT PROPRIETARY? ASHLEY--ALSO-- HELP WE NEED A LISP COPY OF DOCTOR. GOT DOCTOR WORKING IN SNOBOL VERSION (FROM U. OF ARIZONA) BUT AS PREVIOUSLY NOTED IT'S A REAL BURNER. ASHLEY & BOB G. AM SENDING A SNOBOL4 RUN (WITH CPU TIMES) WHICH YOU MIGHT WANT TO CHECK OUT ON THE 1070. HENDERSON'S VERSION OF LISP SEEMS QUITE GOOD-- BUT I'M NO LISP HACKER!

****NOTE-- IF YOU GET THIS FAR, KEEP READING.......

ENOUGH ABOUT SYSTEMS COMPARISONS. I'LL BE DOING A LOT OF THAT THIS QUARTER IN THE UPCOMING OPERATING SYSTEMS COURSE-- WILL KEEP YOU POSTED AND HOPE YOU LET ME KNOW HOW THE DEC-10 IS PROGRESSING-- YOU NEVER KNOW, I MAY WANT TO BUY ONE FOR MY COMPUTER COMMUNE SOME DAY. I DON'T EXPECT TO HAVE MUCH DEALING WITH OLE MISS (FOR OBVIOUS REASONS-- IN THE NEAR FUTURE) BUT I DO EXPECT TO RE-ESTABLISH UNO (NOW OFFICIAL) ACQUAINTANCES AND CONTACTS. WE HEAR SOME TALK FROM XEROX THAT THERE MAY BE A SIGMA 9 LIBRARY SYSTEM (SPECIAL FUNDING) AT TULANE UNIVERSITY. WE'RE HAVING A SCIENCE FAIR DAY (APRIL 19 & 20TH) AND, SINCE THERE WILL BE INDUSTRIAL EXHIBITS , I AM HOPING ONE OF YOU CAN GRAB A GT40 AND LOTS OF HAND-OUTS AND COME DOWN FOR A FEW DAYS-- I'M SERIOUS! I HAVE CONTACTED THE KNOXVILLE OFFICE (JERRY ROBERTSON) ABOUT THIS-- LEFT A MESSAGE-- HOPE SOMEONE WILL COME-- WE DO HAVE A COUPLE OF SMALL COMPUTER SYSTEMS UNDER CONSIDERATION WHICHMIGHT BE OF INTEREST TO DEC-- ASSUMING THEY STILL WANT TO DO BUSINESS IN THE STATE OF MISSISSIPPI. ASHLEY, COULD YOU BRING A TURTLE? YOU WERE RIGHT ABOUT OLE MISS-- THAT'S JUST WHAT THEY NEEDED TO TRAIN THEIR SYSTEMS PROGRAMMERS.

**** FURTHER NOTE-- CONCERNING DAVE HENDERSON.....

NOW, CONCERNING DAVE HENDERSON. RESUME IS ENCLOSED GIVING

THE BEST OF THE SYSTEMS PROGRAMMERS I HAVE KNOWN-- AND STUDIED (WHEN THEY WEREN'T WATCHING!). CLEARLY A LITTLE ECCENTRIC AND REBELLIOUS-- SEEMS AS THOUGH I HEARD THE TERM UNMANAGEABLE ONCE OR TWICE, COULDN'T HAVE BEEN AT DEC, THAT RADICAL COMPANY. ALSO, DAVE HAS A VERY GOOD APPLIED AND THEORETICAL MATHEMATICS BACKGROUND-- IT APPEARS THAT VANDERBILT HAS A GOOD COMPUTER SCIENCE PROGRAM -- AND AN AMATEUR'S INTEREST IN NUCLEAR PHYSICS. IT IS MY OPINION THAT DAVE WOULD BE A BIG ASSET TO ANY OF THE SYSTEMS DEVELOPMENT GROUPS IN MAYNARD (DEC-10 OR SMALLER SYSTEMS). I'VE CONTACTED RICHARD DOWELL (NOW IN MIAMI AS MOST OF YOU KNOW) AND HE IS CHECKING OUT THE PITTSBURGH AREA THROUGH RAFF ELLIS. DAVE IS INTERESTED IN LEAVING HERE BY JUNE 30TH (HIS CONTRACT EXPIRES THEN) AND, THOUGH HE PREFERS THE SOUTH (IT IS WARM, YOU MUST ADMIT THAT FLOYD!), HE DOES WANT TO GET CLOSE TO AND TAKE SOME COURSES IN AGOOD COMPUTER SCIENCE GRADUATE PROGRAM (CMU, MIT,CASE, THAT TYPE OF PLACE). DAVE HAS INTEREST IN BOTH OPERATING SYSTEMS AND LANGUAGE PROCESSORS. HE CERTAINLY KNOWS UTS AND THE XEROX SYSTEMS SOFTWARE INSIDE AND OUT-- MUST BE SOMETHING TO SAY FOR WORKING IN THE BENCHMARK GROUP.

*****KEEP IN TOUCH. JON *****

A COMPARISON OF

TOPS-10

AND

UTS

by Maria Plaza
(DEC)

The material presented herein was gathered from various publications of Digital Equipment Corporation and Xerox, and from report or rumor by various individuals associated with both companies.

TOPS-10 a mnemonic for the Total Operating System for the PDP-10, is the operating system which currently runs on Digital Equipment Corporation's DECsystem-10. UTS, or the Universal Timesharing System, is the operating system which runs on Xerox's Sigma 6, 7, and 9. The current status of both of these complex, yet modular, pieces of software is the result of about seven years of effort involving design specifications, coding, improvements, maintenance and redesign. UTS, which was released around October, 1971, seems to have evolved from other Sigma operating systems, namely BCM, BPM, RBM, and BTM. TOPS-10, which should now be properly called the DECsystem-10 Monitor, or the DEC-10 Monitor, has undergone rather modular changes since its inception in 1967. The new name for the operating system was given when DEC included the ability to handle the new KI10 processor and dual processor systems. However, the name TOPS-10 is still widely used, and I will use it here.

As far as general capabilities, both operating systems offer batch, timesharing, and real time. Currently, I've heard that UTS supports single stream batch, since its multi-stream batch isn't out yet. The DEC-10 Monitor can support 14 job streams under MPB, short for Multi-Programmed Batch, which has been out for about a year and a half. Real time jobs which can be run only in executive mode under UTS can be run either in executive mode or user mode under TOPS-10.

Both systems offer timesharing capabilities for up to 127 users.

In addition to the above, TOPS-10 also offers remote batch facilities, whereas UTS does not. With the inclusion of remote software in the operating system, up to eight remote stations can be handled by communications through a PDP-11. Each of these stations is a PDP-8/i which can have a card reader, line printer, and a concentrator for up to sixteen interactive terminals. Currently, there are five DECsystem-10's with remote batch stations. As of January 1, 1972, it was rumored that about ten installations were running UTS and over 200 installations were running TOPS-10.

The hardware configurations on which these two operating systems function can vary widely. The DEC-10 Monitor can be on a dual processor system, and it can run either under a KA10 processor with a maximum memory size of 256K words or a KI10 processor with a maximum memory size of 4096K words, where a word is 36 bits. As far as I could gather, UTS can be run on a Sigma 6 or 7 with a maximum memory size of 128K words, or a Sigma 9 with a maximum memory size of 256K words, where a word is 32 bits. A workable minimal system seems to require 80K for UTS and 64K for TOPS-10. Each system can expand its memory size in 16K increments.

The secondary storage areas for both of these systems can be a combination of both fixed head and moveable head disks. UTS, however, requires a fixed head disk, which Xerox calls a RAD, on which it stores jobs that have been swapped out of core. The DEC-10 Monitor can swap jobs onto a fixed head disk (drum) or it can swap onto a moveable head disk (disk pack). It can even swap onto both types of devices in the same system. Of course, it is much slower to swap on disk packs; however, for those customers who find it satisfactory, it means a lower cost for a system without a drum and its controller. The disk packs on the DECsystem-10 have an average access time of 47.5 ms or 41.5 ms depending on the type of drive used, while those on the Sigma have an average access time of 75 ms. The software routines concerned with these moveable head disks take advantage of positioning optimization in both operating systems. However, the DEC-10 Monitor also takes advantage of latency optimization.

The peripherals which these two operating systems support are the usual, i. e., card readers, line printers, magnetic tapes, and interactive terminals, all of various types. In addition, the DECsystem-10 also has DECtapes, random access magnetic tapes developed by DEC, which are bidirectional and redundantly recorded.

To support these hardware configurations both Xerox and DEC supply a software system which is quite large. Both are modular and can be tailored for particular hardware configurations and for certain types of system usage. However, they still occupy a good percentage of main storage. Some numbers I found concerning UTS were 31K for the monitor, 16K for overlays and 8K for tables which give an operating system of 55K. Therefore, on an 80K system, only about 25K is left for users. TOPS-10 for a 32-user system, for example, consists of a 23K monitor and 8K of tables, which give a total of 31K. This would increase by 1K for five additional users. It would also increase 4K for remote batch stations and 1K for another processor. Nevertheless, on a typical system with 32 users, if there is 64K of memory, a 31K operating system leaves 33K of user core.

Well, how do these operating systems go about allocating some of this available core to user jobs? UTS allocates on the basis of pages, where the pages are 512 words and are kept track of with a hardware map. TOPS-10, on the other hand, allocates in increments of 1K or 1024 words. Under UTS when a user job is brought into core and it needs several pages, it doesn't need to get contiguous pages. However, currently under TOPS-10 with a KA10 processor, a user program can be separated into at most two non-contiguous segments positioned by dual hardware relocation and protection registers. Each of these segments, one referred to as the low segment and the other as the high segment, has to be a number of 1K chunks that are contiguous in core. This constraint often requires shuffling the other jobs residing in core to consolidate unused core and make one contiguous chunk. This process, of course, does take some time and adds to monitor overhead. The KI10 processor, however, will use a paging scheme with a 512 word page, i. e., when the software to take advantage of paging is released to the field. Just as a matter of interest here, I have an interesting time comparison concerning paging with the Sigma and DECsystem-10. A particular report claimed that on a context switch, the Sigma memory map hardware had to be completely reloaded and might take 200 us for a large program while DEC's KI10 processor simply switches page tables in less than 3 us. Nevertheless, whether it's pages or segments, both of these operating systems still require that the entire user program be in core before execution can begin.

How large might some of these jobs be that have to be brought into memory? Well, due to the constraint on the number of bits allowed for addressing, the maximum number of words that a user can address under UTS is 128K and under TOPS-10, 256K. In some configurations there may be more physical memory than what the user can address. In others, the maximum size for a user program would be physical memory minus the size of the resident operating system..

It is possible, and often true in both systems, to have more than one user in core. Yet, unless a dual processor DECsystem-10 is considered, only one program can be in the execution process at a particular time. Obviously, with several timesharing users and batch jobs there is quite a bit of contention for core in both systems. This problem is alleviated somewhat by having shareable, or reentrant, high segments under TOPS-10 and shareable processors under UTS. An employee of Xerox stated that on a typical UTS system there were 40 shared processors and 30 non-shared processors, included in the system software. FORTRAN and BASIC are shared processors, for example, but COBOL is not under UTS. Under the DEC-10 Monitor BASIC, FORTRAN, ALGOL, MACRO (the assembly language), and COBOL are shareable, along with several of the other utilities on the system. To illustrate this savings in core space, suppose there are 3 BASIC users. Instead of each of them having a 12K high segment of the BASIC compiler in core, only one copy need be in core saving 24K. Under TOPS-10 any user can create a high segment and with care in programming can make it a usable shareable high segment.

Since several jobs could be in core under UTS or TOPS-10, how does the operating system decide which job is going to be executed next? First of all, a scheduling routine only considers as candidates those jobs which are in a compute queue, or as we might say, those jobs which are ready to run. UTS schedules jobs in a round-robin fashion through the compute queue and assigns each job one quantum of run time, or one time slice, which is usually set at 2 seconds but can be changed. If I understand correctly, there is no bias here under UTS on the size of the job. The DEC-10 Monitor has at least three normal run queues, called PQ1, PQ2, and PQ3 (PQ meaning processor queue), and some systems, depending on the desires of the systems programmer or systems manager, have from one to 16 high priority queues (HPQ's). When the scheduler goes to pick the next job to run, it checks the HPQ's first, if there are any, starting with the highest. All jobs which are ready to be executed and in an HPQ are executed before jobs in the normal run queues. Of course, the privilege for running in an HPQ can be designated only to certain users. This is especially helpful to real time jobs. When there are no jobs to be run in an HPQ, the scheduler, which is invoked every 1/60th of a second, checks PQ1 for a job. If there are none in PQ1, it goes to PQ2; if none there, it goes to PQ3. The quantum run times for jobs coming from these three queues vary. Jobs from PQ1 get one half of a second, those from PQ2 and PQ3 get two seconds. When a job is first ready to be executed it will go into an HPQ, if one was set and the job had the privilege to set it. Otherwise, it will go into one of the three normal run queues depending on its size. A job up to 4K goes into PQ1, between 4K and 16K into PQ2, and otherwise into PQ3. After a job has computed for a time equal to its initial quantum, it stays in the same HPQ if it's in one, or it goes to the bottom of the next processor queue (PQ). Those from PQ3, however, go to the end of PQ2. Also, and very importantly, all non-HPQ jobs go into the front of PQ1 after they've gotten an I/O complete. Consequently, a compute bound job will circulate in PQ2 and PQ3 and get to PQ1 only after doing I/O. This scheme is usually good for the timeshared user who is looking for quick response time or terminal I/O .

Not only does size affect the initial processor queue assignment in the DEC-10 Monitor, but it also affects the frequency of swaps for a job. Since it would take a longer time to swap out a larger job, those jobs have a greater in core protect time. This in core protect time is computed dynamically depending on the type of swapping device and the size of the job. UTS also has a parameter called SQUAN, or swap quantum, which controls the frequency of swaps. However, I believe it is set by the systems programmer or systems manager, but I don't think it varies for the size of job.

Now what sort of file system do these jobs work with? First of all, UTS allocates file space on disk in granules,

i. e., 512 words at a time, whereas TOPS-10 allocates in blocks of 128 words. Allocating smaller chunks could lessen the amount of wasted disk space, but perhaps increase overhead. Consequently, under TOPS-10 the systems programmer can change the allocation to be a certain multiple of 128-word blocks. The cataloguing scheme of UTS seems to be only one level, and it extends over all the disk packs in the system. Under TOPS-10, however, it is multi-level and can be over selected disk devices in the configuration. The hierarchy of these levels is a master file directory which points to a user file directory which can point to subfile directories which can point to other subfile directories down to six levels. (This subfile scheme is helpful for batch jobs that are running under the same user numbers and creating or deleting files of the same name, since it eliminates the confusion.) There can be up to eight of these hierarchies or file structures in a DECsystem-10, and they can be public or private. It is also possible to remove a file structure from the system when it is running. As an example, a configuration with six disk packs could have one pack as a private structure, one as another structure, and three of them together as another structure.

Whether the files exist in the single level catalogue of UTS or the flexible scheme of TOPS-10, they each have a protection associated with them. This is to prevent misuse by unauthorized users. From what I could gather, UTS can allow certain users to read or write a file that already exists and can have a password associated with a file. TOPS-10 has the ability to distinguish the creator of a file from those users in the same project as the creator and from all other users. The creator of the file can then assign access privileges to three groups; i. e., himself, his project, and all others. These privileges range from 0 to 7 and are as follows:

7 No privileges
6 Execute only
5 Read
4 Append
3 Update
2 Write
1 Rename
0 Change protection

where each number also gives the privilege of those higher numbers. For example, an access privilege of 4 allows one to append, read, and execute. A default protection is set for each file, and it can be changed by the systems programmer. This gives a pretty comprehensive protection scheme.

With the great quantity of information contained in all these disk files, both systems must provide a method of file backup. In the event of disk failures causing file losses, both systems have a method of saving files on magnetic tape and retrieving them when needed. In the event of a software crash, the DEC-10 Monitor doesn't seem to have too many problems preserving files, since it stores much of the file information on disk. However, I've heard that UTS must determine the status of disk files from an examination of core which might have been messed up by the crash. Thus, the contents of the disk might get lost.

Crashes may not be as fatal as losing the contents of a disk, yet they are still frowned upon in any computer center. In environments where UTS and TOPS-10 run, several timesharing users can be interrupted from their work when the system halts and they receive no response at their terminals. This could cause several irate users. Of course, both operating systems are continually being improved to eliminate as many software crashes as possible. Also, measures are taken to provide quick recovery or reload procedures, error reports, meaningful crash analysis methods, debugging facilities and easy patching methods. The end result never seems to be good enough, though, for all those users who expect perfection out of a computer!

Nevertheless, improvements continue, often making it quite difficult to keep pace with the most recent capabilities and techniques of an operating system. Therefore, very close study and a great deal of usage is necessary to become well acquainted with either UTS or TOPS-10. It also makes a paper of this nature no easy task, if one wants accurate facts.