AN INTRODUCTION TO UT-2D FOR SYSTEMS PROGRAMMERS

by

Waldo M. Wedel

October 1973 TIMS - 7

The writing of this document was supported in part by NSF grant GJ-1084

THE UNIVERSITY OF TEXAS AT AUSTIN

The purpose of this manual is to describe the principal structures and processes which make up the UT-2D operating system which has been developed by the Computation Center for the CDC 6600/6400 equipment configuration. This description is reasonably accurate at the time of publication.

From time to time changes are made to the UT-2D operating system to add new capabilities, to remove defects, or to improve efficiency. Whenever possible these changes are made in a way which is transparent to the user -- the external interface is not changed. These operating system improvements frequently do require changes to the internal interfaces of the operating system, i.e., changes to the data structures of the operating system, some of which are described in this manual.

The Computation Center reserves the right to modify the internal data structure of the operating system, without prior notice if the modification does not affect the external user interface as described in the Computation Center User's Manual. Significant changes to the internal interfaces will be reflected in future revisions of this manual.

TABLE OF CONTENTS

LIST OF FIGURES

I. Overview of UT - 2D.

UT - 2D is the operating system for the computer system at the Computation Center of the University of Texas at Austin. It was designed and developed by the Computation Center staff as an executive system for the Control Data 6600/6400 system. This manual is intended as an introduction to UT - 2D for systems programmers. This section of the manual gives a basic introduction to UT - 2D. Section II examines the system in considerable detail. This manual is not a substitute for studying the code of the operating system but it is hoped that it will make understanding the code easier.

I.A. Introductory considerations.

The following two sections offer a brief review of the architecture of the Control Data 6000 series computer systems and a discussion of the functions of UT - 2D as an operating system for these computers.

I.A.1. Introduction to the hardware.

A Control Data 6000 series computer may be considered to be composed of four logical hardware areas.

The central processor unit
Central memory (CM)
The peripheral processors (PP)
Peripheral hardware

The central processor unit is capable of performing high speed computations but has no input/output capability. It communicates only with central memory and extended core storage (ECS). The major difference between a Control Data 6400 and a Control Data 6600 is the central processor. The 6600 central processor contains 10 different functional units and is capable of executing some instructions concurrently. The 6400 has a unified processing unit which executes each instruction sequentially. The central processor can be switched between jobs in central memory by executing an exchange jump instruction. The exchange jump instruction exchanges the contents of 16 words in central memory, called the exchange package, with the current contents of the operating registers.

Under UT - 2D, the central processor compiles, assembles, loads, and executes user programs. It also is used to execute several system functions. Programming for the central processor is primarily done in higher level languages such as FORTRAN, ALGOL, etc. A powerful assembler for central processor code is available (COMPASS). Both the 6400 and the 6600 have the central exchange jump / monitor exchange jump (CEJ/MEJ) option and the integer multiply option installed.

Central memory is used for three purposes under UT - 2D.

1) To hold instructions for the central processor.

2) To hold data to be manipulated by the central processor.

3) To hold data to be moved to or from a peripheral processor.

The 6600 has 131,072 words (400,000 octal) of CM and the 6400 has 65,536 words (200,000 octal). Extended core storage (ECS) can be considered to be an extension of CM. On the University of Texas system, ECS can be accessed only by a central processor. It is used to hold operating system data which must be shared between the two computers and to hold jobs which have been swapped out of CM on the interactive machine.

The peripheral processor units (PPs) are small independent processing units each with 4096 12 - bit words of memory which are used to perform many system functions. The 6600 has 10 PPs; the 6400 has 7 PPs. The primary purpose of the peripheral processors is to perform system input/output (I/O). The peripheral processors can read and write central memory and transmit data to and receive data from peripheral devices using the data channels. The peripheral processors also perform most of the system control functions. Programs for the PPs are written in assembly language (COMPASS). No user programs execute in a PP.

The peripheral hardware attached to the University of Texas CDC 6600/6400 system is shown schematically in figure 1. The data channels connected to each unit are also shown for each machine. Many of the references contain additional information on the system hardware. The book by Thornton is a very good introduction. Control Data reference manuals containing specific details on peripheral equipment are also available.

I.A.2. Functions of UT - 2D.

The UT - 2D operating system controls the operation of the CDC 6600/6400 computer system. The system performs several major functions.

1) Resource Allocation. The system manages allocation of space in CM, ECS and on the disk mass storage units. It also allocates individual peripheral devices such as tape drives.

2) Central Processor Scheduling. The system assigns the central processor to those jobs requiring central processor service.

3) File Maintenance. The operating system maintains a unified file system consisting of output files, job files, local files, and permanent files.

4) Job Scheduling. The system schedules jobs to be run based on resource requirements.

5) Accounting. The system maintains a record of each job's utilization of resources for accounting purposes.

6) Performance Monitoring. An event probe is incorporated into the system software to record events significant to the performance of the system.

I.A.3. Maintenance of UT - 2D.

The source code for UT - 2D is written primarily in COMPASS although some utility programs are in other languages. The code is maintained in a compressed symbolic form on a file known as an old program library (OPL). The OPL can be edited by a system central processor program named MODIFY. MODIFY accepts an OPL and a set of corrections as input and produces a new program library (NPL) and an expanded source text file (COMPILE) incorporating the corrections. The COMPILE file can then be processed by an assembler or compiler.

The object programs produced by the assembly or compilation can be used to make a system load tape (Deadstart Tape). The system program EDITLIB will produce a new system load tape from an old load tape and the object programs.

Several conventions have been adopted in UT - 2D. Some of them will be presented here. For purposes of PP coding it is convenient to think of CM words as partitioned into five 12 bit bytes numbered from left to right beginning with 0. Thus, byte 0 is the leftmost 12 bits of a CM word. Bits within a byte or word are numbered from right to left, however. Thus a word is composed of bits 59...0 or of bytes 0...4, each of which has bits 11...0. By convention, octal values are represented as digits followed by the letter B. Thus a byte with bit 5 set has the value 0040B.

Throughout the text, many symbols are used to represent memory locations and commonly used terms. These are capitalized in the text whenever they are used. The symbols are generally defined on text files of symbol definitions which are accessible to the assembler. The listings of these symbol files contain the absolute values for these symbols. Most of the symbols are shown in a graphic format in the figures accompanying this manual.

I.A.4. Deadstarting UT - 2D.

The process of loading the system into the computer is referred to as deadstart (DS). A magnetic tape, the deadstart tape, is the source from which the system is loaded.

The computer is prepared for deadstart by mounting a deadstart tape on a tape drive and setting up a short bootstrap program on a panel of 144 toggle switches, the deadstart panel. The deadstart panel setting for UT - 2D is posted inside the cabinet door. It has been designed to be compatible with the customer engineer's panel setting, with the result that it is seldom changed. Although the panel settings do specify some options, they can be reset by keyboard entries during the deadstart procedure. When a button on the operator's console is pushed (the deadstart button), the program on the deadstart panel is loaded into PPO and executed. This program loads other programs which load the operating system into the computer. When the deadstart programs have terminated and deadstart is complete, the computer contains the following:

1) A file called the system library is stored on a mass storage device, normally one of the 6638 disks. This file is a partial copy of the deadstart tape. Programs can be loaded from the library into PPs or into CM as needed. The mass storage unit on which the library is stored is called the system device.

2) A program named central memory resident (CMR) is loaded into the low - order end of central memory. CMR contains a set of tables which are used to store status information about the system and the

central processor portion of the system monitor (CPM). Some of the tables are used by PP programs of the system to communicate with each other.

3) Some of the programs on the system library file are also stored in central memory, immediately above the area which contains CMR. These programs are stored in CM because from there they can be loaded into PPs much faster than they could be loaded from a mass storage device such as a disk. Only the most frequently used programs are stored in CM since CM is a valuable resource and any CM used to store programs can't be used for anything else.

4) The PP portion of the system monitor named MTR in PPO. MTR remains in PPO and is in overall control of the system.

5) The operator console driver program DSD resides in PP9. Information about the current status of the system is displayed by DSD on the two cathode ray tubes in the operator's console. The operator can control the operation of the system by typing commands on the console keyboard. DSD interprets and processes these commands.

6) The remaining PPs, the pool PPs, contain initialized values for the direct cells and the PP resident program, PPR, which serves three tasks. First, it has subroutines for communicating requests to MTR (MTR functions). Second, it contains a PP overlay loader for loading the PP with transient PP programs. Third, it contains a low-level disk driver used by the loader and also by all higher level disk driver routines. This third part is actually an overlay, 5DB, 5DG, or 5DE depending on which type of disk is being used at the moment.

I.B. Pool peripheral processors.

MTR, DSD, and PPR are loaded at deadstart and remain in the PPs into which they were loaded. PP overlays loaded by PPR remain in a PP until they have completed a specific task. PPR can then load another, often unrelated overlay to accomplish another task. The PP overlays are called transient programs. The PPs not containing MTR or DSD are called pool PPs.

I.B.1. Program execution.

A PP transient program may load additional overlays into its PP to help complete the primary task. The name of a PP overlay indicates its level and accessibility. All PP overlays have three-character names. If the first character is a letter, the overlay is a transient program which can be called by a central processor program or by the system (e.g. CIO). The overlay would be loaded at location 1000B or at the origin specified during assembly. If the first character of a PP overlay name is numeric, the PP overlay can only be called by the system (e.g. 1AJ). All primary overlays (programs) have names which begin with a letter or the number 1. Secondary overlays have names beginning with digits other than 1, usually 2. Their load origins are determined at assembly time to fit immediately above the program which loads them. Overlays which are assembled separately from a primary program usually have an origin of 2000B.

When a PP program is assembled a 60 bit header word is added to the beginning of the program. This word contains the name of the program, the actual load address, and the length of the PP program in CM words. When a PP overlay is loaded, the header word is loaded along with the overlay. Thus if a PP transient program is loaded at location 1000B, locations 773B through 777B of the PP will contain the header word of the PP program. This information is very useful when examining a PP memory dump.

Figure 2 shows a typical memory layout for a pool PP. Figure 3 shows PP words 0 to 77B, known as the direct cells since they may be addressed directly by a 12 bit PP instruction. Direct cells 0 - 67B are available for use by PP resident and the overlays in the PP. Direct cells 70B - 73B contain constants used by PPR and which may be used by any PP program but may not be altered. Direct cells 75B, 76B, 77B contain pointers to the communications area for the PP.

I.B.2. Communication between PP programs and MTR.

For each pool PP, there is an eight word PP communication area in CMR for communication between the PP and the system monitor. A diagram of the PP communication area is given in figure 7. The first word in the area is called the PP input register or IR. If the PP is idle, its input register contains zero. If MTR needs to run a transient program in a PP, it writes the name of the PP program left justified into the input register of the PP.

When the PP is idle, PPR scans the PP's input register every 128 microseconds. If it becomes non-zero, PPR issues a locate peripheral program (LPP) function to the system monitor. The monitor searches the system directory for the location of the program and returns the location of the program to the PP. PPR then loads the transient PP program at the address specified in the program header word and transfers control to 5 bytes past the beginning of the header word to begin execution of the transient program. The PP input register is not cleared by MTR until the transient program has completed execution. If a transient program needs to execute an overlay, it does this by jumping to a subroutine in PPR to load the overlay. The IR is not changed.

Parameters to the executing program may be included in the message written into the IR by MTR. These parameters may be read by the transient program or any overlay it loads. The parameters can appear anywhere in the low order 36 bits of the word.

If a PP needs to make a request to the monitor it puts a message in the second word of its PP communication area. This word is called the output register (OR). Byte 0 of the OR must contain a number which identifies the function requested of the monitor. The rest of the OR may contain parameters for the request. Additional information or parameters for the request may be placed in the area called the message buffer (MB). The message buffer consists of the six words of the communications area immediately above the IR and OR.

When monitor has completed the processing of an output register request, it will clear the leftmost byte of the output register to inform the PP that the request has been processed. If the monitor needs to communicate information about the processing of the request to the PP, it will store the necessary information in the remainder of the OR or in the MB.

Monitor requests are normally made through PPR. PPR takes a request formatted by an overlay, writes it into the OR and then waits for the leftmost byte to be cleared.

When a PP transient program has completed its function it sends a request to monitor to drop the PP. Monitor will clear the PP's input register to record the fact that the PP is available. The transient program then executes a jump to the PP resident idle routine which scans the IR waiting to load another transient program.

PP resident obtains the addresses of the words in the PP communication area for its PP from the contents of direct cells 75B, 76B, and 77B in the PP. These cells are set at deadstart when PPR is loaded and should not be altered by any overlay.

I.C. Central processor processes and memory allocation.

Central processor jobs can be introduced into the system from a batch terminal or from an interactive terminal. One typical way would be to punch a program on cards and read it in through a card reader. Typically a central processor job step involves reserving a block of memory for a process and then loading and executing it. A CDC 6000 series machine is typically run as a multiprogramming machine so that several central processor jobs can be in various stages of processing at the same time.

The central memory assigned to a process is always one uninterrupted block of words. The first word is called the reference address (RA). The length of the block is the field length (FL). All addresses used in a central processor program are relative to the RA and must be in the range 0 through FL-1. Any attempt to reference an address outside of these bounds is detected by the hardware. Therefore it appears to the user as if he were running in a computer with a memory the size of his field length.

I.C.1. Organization of central memory.

Central memory can be considered as divided into two categories, system CM and user CM. System CM is used for the storage of system tables. User CM is available for allocation to user jobs.

System CM.

System CM, diagrammed in figure 5, contains four basic types of information. In the lowest part are pointers to tables in higher CM. Next come a series of tables of system information. One of these tables is the PP communications area previously discussed. Other tables are used to keep track of the status of all jobs in the system. The central processor portion of the system monitor (CPM) resides above these tables. The area above CPM is used to store the CM resident peripheral library (RPL).

The CM above the RPL is allocated to user jobs. The lower portion of each block allocated contains the Control Point Area (CPA) associated with the user job (see figure 18). The CPA contains job status information. The remainder of each block is the CM which can be altered by a user process (i.e. the user's FL).

User CM.

The section of central memory above the user RA and below RA + FL is the portion of CM that a user process can alter. The first 100B words of this area are used for communication with the operating system. Figure 22 is a layout of this area. When a user process wishes to communicate with the operating system, a request is placed in RA + 1 of the user's field length. This word corresponds to the output register (OR) of a PP communication area and is generally referred to as "RA + 1".

The words from RA + 2 to RA + 63B are filled with control card parameters by the system. The name of the program and the number of control card parameters are placed in RA + 64B. Thus the process called by the control card PREP,CARDS,TAPE. would find the name PREP in the upper 42 bits of RA + 64B, the number 2 in the lower 18 bits, the name CARDS in RA + 2, the name TAPE in RA + 3, and zero in RA + 4 through RA + 63B.

RA + 70B through RA + 77B contain an image of the control card which called the process. Words RA + 65B through RA + 67B contain information which may be used by system routines such as the loader. RA + 65B contains the next available word for loading. RA + 66B contains the first word address of the program just loaded. RA + 67B contains the LDR completion bit.

I.C.2. Multiprogramming.

Several jobs can occupy different regions of memory at the same time. Each job has its own RA and FL. The number of jobs in central memory at any time is called the degree of multiprogramming. In general these jobs will be utilizing different system resources. One may be using the central processor while the others are receiving the services of a PP for I/O or some other function. Others may be waiting for central processor or PP services. Information on the system's response to various program requests and the system utilization of the resources is obtained by activating a software probe to record events which are significant to response time or utilization.

A process which does not require central processor service is placed in recall status. Two types of recall are used, auto - recall and periodic recall. If a central processor process issues a system request with the auto recall bit set, the system monitor will not reassign the central processor to the process until the system request is complete. If a central processor process issues a periodic recall request, the central processor is not returned to that process until the time specified in the request has elapsed.

The central processor is actually switched between processes using a round robin scheduling algorithm so that no single job can monopolize the central processor.

I.D. Files.

Files are the basic logical element manipulated by UT - 2D. Each job in the system is considered a file. All jobs operate on files of information and produce files of information. Consequently, the maintenance and manipulation of files is a major function of UT - 2D. Files are organized into logical records in UT - 2D. Logical records are defined in terms of physical record units (PRUs). A logical record consists of zero or more full length PRUs terminated by a short PRU. For a storage medium such as magnetic tape a full length PRU is defined as 512 CM words. Thus a logical record is zero or more consecutive 512 word blocks terminated by a block of fewer than 512 words.

Some storage media must be written in a constant length PRU. A disk or card are examples of such media. In these cases a word count is usually associated with each physical record and a short PRU is indicated by a low word count. A disk PRU is defined as 64 CM words; a card PRU is 15 CM words.

I.D.1. File types.

Files may be classified into five major types. They are system files, job or rollout files, output files, permanent files, and local files. System files are the dayfile, the system itself and several specialized system libraries. The dayfile is a record of system activity including accounting information. Job files represent user jobs. The information in a job file varies according to the state of the job. When a job file is in CM it consists of the control point area and a user FL. A job file on disk always contains a control point area image and may have an image of the CM field length. Such files are often referred to as rollout files. Output files are files which contain information produced by a job. OUTPUT, FILM, PLOT, and PUNCH files are examples of files of this type. Such files are retained in a queue until the device for which they are destined becomes available. Permanent files are data files which are retained by the system between jobs. Local files are scratch files local to a job. They are created as the job requires them and destroyed when the job is completed.

I.D.2. File name table, job status table.

All files except job files are maintained by means of a file name table (FNT). The FNT in UT - 2D is divided into two logical segments. The FNT for system, queue, and permanent files resides in the system CM area (see figure 9). The FNT for local files resides in the control point area (CPA) associated with each job (see figure 20).

Job file information is maintained in the job status table (JST) (see figure 10).

II. A closer look at UT - 2D.

The preceding sections have presented an introductory overview of UT - 2D in an attempt to present a basic background for the more detailed discussions of this section. The following subsections contain a more detailed discussion of UT - 2D system tables and processes.

II.A. Job processing.

One of the basic functions of almost any operating system is to process jobs for users. A detailed examination of the way in which this is accomplished is fundamental to an understanding of the operating system. It is convenient to distinguish between batch jobs and interactive or conversational jobs for the purposes of discussion.

II.A.1. Batch job processing.

A batch job can be introduced into the system from a card reader or a remote terminal. Typically it consists of one or more logical records, the first of which is a control card record. The control card record consists of a user identification card followed by a password card and one or more control cards. Control cards are commands to the system requesting system services. The logical records following the control card record are usually data required by the programs initiated by one or more of the control cards.

The batch job always produces results which are to be printed and optionally it may produce output to be punched, plotted, or filmed.

A batch job is first recognizable as a job when an entry is made for the job file in the job status table (JST) (see figure 10). This entry is made by one of the system I/O routines, GEMINI, TAURUS, or by CHG using the PP routine 1PS. The entry is flagged as ready for accounting processing by clearing the initialized job bit. The accounting supervisor, PISCES, uses the PP routine CPU to periodically scan the JST for jobs to be initialized.

When PISCES finds an uninitialized job it validates the user number and password against the validation file (QRT). If the user number or password are illegal, the job is terminated. Otherwise, a rollout file is created for the job. This minimal rollout file consists of an image of the control point area. The initialized job bit is then set in the JST entry and the job scheduler is called. The scheduler initiates job processing when sufficient CM becomes available by requesting the PP program 1RJ to roll in the job. When the job has been rolled in, MTR checks the control point status table (CPSL) for activity at the control point. If there is no activity at the control point the PP routine 1AJ is called to advance the job.

1AJ first examines the control point error flags. If no error has occurred, it then reads the next control card from the control card record. If a control card is present, 1AJ loads and executes overlay 2TS to process the control card. If the control card is one of the 1AJ special control cards, the processing is accomplished by a third level overlay, 3AJ.

If the control card names a local file at that control point, it is assumed that the file is to be loaded and executed. In this case the central processor loader, LDR= is loaded and control transferred to it. The mechanism for this control transfer is that the entry point found on the load file containing LDR= is placed in the P register image of the exchange package. The first 100B words of user low core are set up and a request for central processor (RCP) function is issued to MTR.

If the control card does not name a local file, the central library directory (CLD) (see figure 15) is searched for a routine with the same name. If such a name is found, the central routine is loaded and control transferred to it in much the same way as to the loader. If the name is not found in the CLD, the peripheral library directory (PLD) and resident peripheral library (RPL) are searched. If the name is in one of these directories, 1AJ enters a request for the PP program in the MTR delay stack (EDR function).

If the name is not found in the in one of these librariesat an error was made and the message ILLEGAL CONTROL CARD is issued and the job aborted.

If 1AJ finds the control card record empty, it calls the job completion processor 1CJ.

If 1AJ finds an error flag set, the error processing overlay 2EF is loaded and executed. Overlay 2EF issues a dayfile message indicating the nature of the error then searches the user low core area for recovery information. If this information is found and valid, an error recovery is attempted. Otherwise a dump of the exchange package and 100B words surrounding the P register image is produced on the OUTPUT file. The control card record is then examined for the presence of an EXIT card. If this card is found the card immediately following the EXIT card is executed. If no EXIT card is found the job is terminated by calling 1CJ.

After completion of error processing, if any, and before the next control card is read, 1AJ performs some cleanup operations to insure that all processes of the preceding job step are complete. These operations are 1) locating output buffers and emptying them, and 2) detaching any attached permanent files.

The job termination processor 1CJ releases all assigned equipment then clears the initialized job bit and sets the memory insignificant bit in the JST for PISCES. This combination of bits signifies a completed job which the scheduler rolls out. PISCES then computes the job cost and enters it in the job's dayfile. The dayfile is then written on the OUTPUT file and all special files are transferred to the system FNT as output files. All local mass storage files are deleted and job processing is complete.

II.A.2. Interactive job processing.

In this section a discussion of the differences between batch job processing and interactive job processing will be presented. Interactive jobs are entered into the system by TAURUS. Details of the entry process will be described in section II.G.

The differences in processing begin as early as the PISCES processing of the JST entry. If the user number is invalid and the conversational bit is set in the JST entry then PISCES signals TAURUS that the user number is invalid by not setting up a job file for the job and placing an error code in the JST for TAURUS.

If the job is logged in, then the control card record on the INPUT file which TAURUS set up for the job is read and control card processing is initiated as for batch jobs. When the control card record is exhausted 1AJ calls 1TM to request a new control card from the terminal. Interactive job termination differs from batch termination in that the file OUTPUT is printed only if the user requests it with a special form of the LOGOUT command.

II.A.3. Control point.

The control point has already been introduced in some of the preceding discussion. A control point is a logical concept used for locating a job file when it is in memory. UT - 2D supports a maximum of sixteen control points. The system table called CPSL (Control Point Status Location), Figure 6, has a one word entry for each control point. The word contains scheduling information, the RA and the FL. The RA and FL information is duplicated in the job's exchange package and in direct cells of each PP associated with a control point.

From the RA, the job's user CM and control point area may be located. A diagram of the control point area appears in Figure 18. The lowest portion of the control point area is occupied by the local file name table (FNT). This table is used to maintain the status of all local files used by the job. The control card buffer follows the local FNT. When PISCES initializes a job, it copies the job's control card record into the control card buffer. If the record does not fit, the excess card images are copied to a local file named CCFILE. 1AJ reads a control card from the control card buffer to the current control card area to initiate processing of a control card. If 1AJ reaches the end of the control card buffer, it refills it by reading CCFILE. The control card pointer always points to the next available control card in the buffer.

The local dayfile buffer is used to retain dayfile messages issued by processes running at this control point. The first word of the buffer is a word of pointers used for managing the buffer. When the buffer is filled, it is dumped to the local dayfile.

The PP rollout buffer is used to save pending PP requests when a rollout is to be performed. The requests are reissued when the job is rolled in. The control point status message is used for holding messages generated by the central processor process. This message is displayed on the operator's console by DSD. The exchange package is an image of the central processor exchange package (see figure 19) as described in the CDC 6000 series reference manual. The next five words are used to retain accounting information associated with the job. The next word contains flags used by the loader and 1AJ. The final two words contain various system limits for the job. The time, page, punch, and disk limits are kept here.

II.A.4. Special system jobs.

These jobs are initiated by an operator command to serve some special purpose for the system. The PP routine 1SS initiates each of these special jobs. They all run at a control point and all have special privileges.

The accounting processor, PISCES, has already been mentioned. It functions at the beginning and at the end of user job processing. At the beginning it verifies that the user is a valid user. At job termination it records the cost of the job in the system dayfile.

The PISCES main loop begins by checking to see if it is time to dump the system dayfile from disk to tape. If so, the system routine SKULKER is initiated to run the dump. PISCES then delays for two seconds by issuing two 1 second recall requests to the monitor. The PP routine CPU is called to search for jobs requiring PISCES processing. If no such jobs exist, control returns to the beginning of the loop. Otherwise, PISCES enters the job processing overlay, loading it if necessary.

System input/output and communication with the high speed remote batch terminals is handled by the job called GEMINI. The control point GEMINI contains no central processor code. When 1SS establishes a GEMINI control point, it places itself in the MTR delay stack. When it is recalled, it scans the EST for unassigned input/output equipment. If print, punch, film, or plot equipment is available and files are waiting for such equipment, 1SS initiates the proper driver by rewriting its own input register with the name of the driver. If an input device is unassigned and ready, the device driver is called in the same manner. 1SS also checks ECS utilization. If ECS is more than 80% full, it initiates a PP routine 1XX to dump the largest job in ECS to disk. The status of the event tape drive is also checked. If it is on, an event buffer is set up in the GEMINI field length and the low core buffer pointer is set to initiate event recording. The remaining tape drives are also checked and if they are on and ready, the tape label processor, 1TD, is initiated to assign the tape to the job requesting it.

Communication with medium speed batch terminals and low speed interactive terminals is handled by the control point TAURUS. TAURUS is currently being rewritten and its successor will be described in a revision of this manual.

II.B. Central processor system requests.

A central processor program must make requests to the operating system for input/output and other system services. All communication with the system is accomplished through word 1 of the user's field length. A request for system service is placed in word 1 (RA + 1), then the central processor must wait for that word to be changed to zero by MTR indicating that the request has been recognized. Two formats for system requests are used. These are shown in figure 23. The first format is used when the information in the request is completely self-contained. The second format is used when the request points to additional information required for processing the request. The procedure for issuing a system request is:

1) Check to make sure that the contents of RA + 1 is zero. If not, keep checking until it becomes zero.

2) Place the request in RA + 1.

The system monitor clears RA + 1 when the request has been honored. If the central processor can continue processing while the system is processing the request, it may do so. If the central processor cannot continue, it must wait, testing RA + 1 until MTR honors the request by clearing RA + 1.

The following requests can be made to the system by a central processor program.

END - Normal program termination.

ABT - Abnormal termination.

RCL - Enter recall status.

RCLP - Enter recall until operation is complete.

CIO - Standard input/output.

OPE - Open file.

CLO - Close file.

RSF - Release system files.

MSG - Issue message to dayfile or operator console.

TIM - Get time, clock, or date value.

GSN - Get job name.

SNP - Snapshot of central memory.

DMP - Dump memory.

INT - Interrupt central program.

RFL - Request field length.

PCC - Process control card.

RCC - Read control card.

II.C. Input/Output.

One of the most frequent requests for PP services made by a central process is a request for input or output of data. Communication of these requests is handled by means of a file environment table (FET). The layout of the FET is shown in figure 21. The actual request for I/O is made to MTR using a request for CIO in location RA + 1.

II.C.1. CIO - Combined input/output.

The major functions of CIO are to verify that the request for I/O is valid and to determine which of its overlays is to be called to accomplish the actual data transfer or positioning operation.

First the local FNT entry for the file is reserved by issuing a monitor RLF function. The buffer pointers FIRST, IN, OUT, and LIMIT are checked to be sure that they are in the user's field length and that LIMIT is greater than FIRST. The file name is checked to insure that it is legal and the operation code is verified. If all of this information is correct, the equipment driver overlay for the file is loaded and control is passed to the overlay to perform the operation.

The particular driver overlay to be called is determined from the EQ field of the local FNT entry for the file. This field contains an integer equipment number which points to an entry in the equipment status table (EST) (see figure 8) in system CM. The DEVICE field of the EST contains a two character mnemonic which determines which driver is to be called, for example MT (magnetic tape) results in a call to 2MT.

A major distinction between I/O devices is made in UT - 2D. Mass storage devices are devices for which the available storage is divided into logical entities called tracks. Typically the space on such a device is allocated to many different files belonging to many different users. In contrast a non-mass storage device can be assigned to only one file at a time. Magnetic tape drives and card readers fall into this category.

II.C.2. Mass storage device I/O.

A CIO request for a read operation on an existing mass storage device proceeds in the following manner. If the file is a random access file, the overlay 2PD is called to compute the sector address for the read. This is done by issuing a compute logical sector (CLS) function to the system monitor. CPM verifys that the track linkage is correct and returns the physical position of the sector on the device. 2PD clears the random indication and sets the position for the read operation. The PP overlay 2RD is then called to perform the read.

2RD first checks to see if there is sufficient room in the CM buffer area for at least one physical record unit (PRU) of information. If there is not, the operation is finished without reading any additional information. If sufficient space is available, 2RD reserves an I/O channel connected to the mass storage device and initiates the read operation from the device to its own memory by calls to the low level driver. The low level driver performs the actual read or write operations on the device. Use of the low level driver means that all mass storage devices look the same to 2RD, 2PD, 2WD, and other PP programs which access mass storage. 2RD then transfers the data into the CM area pointed to by the FET and advances the IN pointer. This process continues until the condition for termination of the operation is met (i.e. buffer full, end of record, or end of file). When the termination condition is met the data channel is dropped and control returns to CIO.

CIO then sets the file status complete and releases the local FNT entry for the file. If the user job has blocked waiting for the I/O to finish, it is unblocked.

A write operation is somewhat more complicated than a read. If the file is a random access file, 2PD is called as in a read operation. If the operation requested is a rewrite of an existing record or a write, the overlay 2WD is called and the record is written. If the file is not a random access file, a rewrite operation is illegal. A write operation on a sequential file which terminates by writing an end of record or end of file drops the remaining tracks on the file, if any are assigned. When the write operation is complete, CIO sets the file status complete and releases the local FNT entry for the file. The user job is unblocked if necessary.

Disk physical records

The actual physical record on disk is 502B bytes long. The first two bytes of each physical record contain a forward link to the next sector of the file and a byte count. Figure 24 represents an idealized mass storage device with 4 PRUs of 502B bytes per track. Four tracks and three files are shown. File A consists of one logical record of 750B bytes. The physical record with a zero link and zero length is the end of information (EOI). The fourth PRU on the track is unused. File B consists of a single logical record of 3000B bytes. The record crosses a track boundary and the fourth physical record of track 4001 points to track 4003. Sector 0 of that track is assumed to be the continuation of the logical record. File C is a file of two logical records. The first is 400B bytes long and the second is 750B bytes long.

II.C.3. Non mass storage I/O.

A read operation on a non mass storage device is handled in the same way as a mass storage device read operation except that a different device driver is called. Since random access is not defined for non mass storage devices read operations are initiated from the current position only.

The device driver follows a logic pattern very similar to that described for the mass storage driver 2RD. No low level overlays are required for the non mass storage drivers. All equipment manipulating functions are issued directly from the 2-level driver. The read operation proceeds as follows.

First the buffer is checked to be sure that there is room for at least one PRU of information. If there is room, a data channel connected to the device is reserved by issuing a request channel (RCH) function to the monitor. The device is then connected to the channel by issuing a proper connect code on the reserved channel. The status of the device is checked to insure that it is ready for data transfer. When the device is ready a read function is issued on the channel and data transfer to the PP memory begins. As data is read into PP memory the driver writes it to the CM buffer, advancing the IN pointer in the FET. This process continues until the read operation is complete. The equipment is then disconnected and the channel is dropped. The driver then returns control to CIO which completes the read operation as described above. I/O for conversational terminals is not handled in this manner. Section II.E describes the techniques utilized for these devices.

II.D. Peripheral processor resident routines.

Peripheral processor resident (PPR) consists of 15 subroutines which may be called by PP overlays. The first routine is the PP idle loop (entry point IDL). This routine is entered at the termination of execution of a PP overlay. First, it checks to see if any dayfile messages have been accumulated. If so, they are dumped to the dayfile by calling the overlay 2WM. If there are no dayfile messages, the loop scans the PP's input register every 128 microseconds and initiates execution of an overlay if the contents of the input register is non-zero.

The second and third entry points are related. The entry point EXR is used to load and execute an overlay. This entry point is used by the IDL routine to load and execute 2WM. The entry point PLL is used to load an overlay. It is used by the IDL routine to load a PP overlay and by EXR.

The entry point FTN is used for issuing MTR functions. The entry point WOC is used to wait for the contents of byte 0 of the output register to be cleared. This is done by MTR to signal the completion of a function.

The entry point DFM is used to record dayfile messages. Dayfile messages generated by a PP program are stored in the high end of the PP memory until the idle loop is entered. The idle loop initiates the actual writing of the messages to the dayfile.

The next three entry points format MTR functions. They are provided as distinct entry points because the functions which they format are so frequently used. Entry points RCH and DCH issue reserve and dereserve channel functions to MTR. Entry point PFR issues the MTR pause for relocation function if the storage move flag in the control point status word (CPSL) is set. It also updates direct cells RA and FL so that the PP can address the central memory associated with the job it is working on. The entry point MDC is used to alter the channel in a channel request according to a table of channels stored in memory. This is used when a device can be accessed by more than one channel and the requested channel is busy.

The remaining five entry points are all associated with the low-level mass storage drivers. The entry point SMS is used to set up the mass storage drivers. It loads and initializes the drivers. The entry points DMS, POS, WDS, and RDS are entry points to the mass storage drivers for disconnecting, positioning, writing, and reading mass storage respectively.

II.E. Multiprogramming.

As mentioned in a preceding section the support of multiprogramming is one of the key features of UT - 2D. This section will describe the role of the system monitor in supporting multiprogramming. The use of many of the low core tables in maintaining the state of the system will be described.

II.E.1. System monitor.

The system monitor performs many of the tasks which are fundamental to the operation of a multiprogramming system. The most important of these tasks are:

1) Scheduling.

The monitor determines which jobs present in the system and ready to be run are to be run.

2) Resource allocation.

The monitor must decide which jobs are to be assigned critical resources and when jobs are to be pre-empted from resources.

3) Performance measurement.

Code is incorporated into the monitor to record its performance of the above tasks and to record the performance of other components of the system. This code constitutes the "software probe".

4) Advance clock.

The system clock is connected to channel 14B. It has a cycle time of 4096 microseconds. The monitor frequently reads this clock and updates the time words in system CM.

The system monitor is composed of two separate modules of code. The module MTR is loaded into PPO at deadstart and remains in that PP until the next deadstart. The second module, CPM, is central processor code and resides in system CM. MTR and CPM often run in parallel, but must sometimes coordinate themselves in order to avoid timing problems in updating system tables. This is done by having MTR issue a standard function for processing by CPM while MTR waits.

The following sections will discuss points 1 to 3 enumerated above.

Scheduling

Scheduling is considered in the rather narrow sense of deciding which jobs will be run. Scheduling of resources is consisdered in the next section. The JST is the source of job information used by the scheduler. Jobs are scheduled for running on a least cost basis. Cost for batch jobs is determined by multiplying time remaining by the job field length. Time remaining is simulated for interactive jobs. The scheduling decisions are made by CPM at the request of MTR. If a decision is made to run a job which is not in memory, the PP routine 1RJ is called by MTR to resume the job. If a decision is made to remove a job from CM the PP routine 1SJ is called to swap the job out. The results of a job scheduler run are left in the control point action table (CPAT) (see figure 16). MTR takes action based on the CPAT.

Resource allocation

A variety of resources must be allocated by the system monitor. These include the central processor, central memory, mass storage, I/O channels, peripheral devices, and peripheral processors. central processor allocation decisions are made by CPM on a round robin basis. The central processor is assigned to each control point which needs central processor services for a 16 millisecond time quantum. The central processor state is examined after each quantum to determine if a hardware error exit was taken or if a program stop instruction is being executed. If so, the job is aborted.

CM allocation is accomplished by MTR. Any relocation of the job in central memory is done by CPM. A PP requests a change in CM allocation by issuing a request storage (RST) monitor function. A PP indicates that it is suspending execution for a storage move by issuing a pause for relocation (PFR) function.

CM is allocated in the following manner. New jobs to be run are loaded lowest in memory. Batch jobs therefore tend to migrate upward in memory. If a batch job requests a reduction in field length, that reduction is granted immediately and the hole which is left is added to the available CM. If a job requests an increase in field length, the holes above and below are examined to see if the request can be granted. If it can, the job is moved up or down as required by CPM. If the request cannot be granted, other available CM is examined to see if a move of other jobs can be made to satisfy the request. If so, the move is made; otherwise the job waits until enough high-cost jobs are swapped out to make the memory available. The waiting job itself may be swapped out during this period.

Mass storage allocation is accomplished using track reservation tables (TRTs) (see figure 17). Each allocatable track on a mass storage device is represented by 15 bits in the TRT. Tracks are allocated and released in response to reserve half track (RHT) and dereserve half track (DHT) functions issued by a PP. When a PP requires mass storage space for a file it issues a RHT function to the monitor. If the file is a new file, CPM locates an available half track on the least used mass storage device and assigns it to the file. If the file already has tracks allocated, the current track linkage is verified and an available track on the same device is assigned. The DHT function dereserves half tracks after verifying the linkage.

The TRTs reside in ECS so that they can be accessed by both machines. This permits complete sharing of files since all mass storage can be accessed by each machine.

Allocation of I/O channels is accomplished using a channel status table (CST). A PP which needs an I/O channel issues a reserve channel function (RCH) to the monitor. The CST is checked to see if the requested channel is reserved. If not the channel is assigned to the PP. If it is reserved the request is placed in a channel queue until the channel becomes available. The channel is dereserved by issuing a dereserve channel function (DCH).

The allocation of peripheral devices is handled similarly to channel allocation. The equipment status table (EST) is used for device reservation. The functions reserve equipment (REQ) and dereserve equipment (DEQ) are used for allocating devices.

PP allocation is accomplished using the PP input registers (IR) and two queues of pending PP requests, the delay queue and the ready queue. The PP input registers (IR) are used by MTR to keep track of which PPs are active. The queues are used by MTR to control PP allocation. The ready queue consists of requests which are ready to be processed. These are delayed requests whose delay has expired and requests which require immediate service. Delayed requests for PP services are made by issuing an enter delayed request (EDR) monitor function. A PP input register is cleared when the PP issues a drop PP (DPP) monitor function.

The monitor accomplishes performance measurement by placing selected events in an event buffer. An event is one to three CM words of information, usually representing a monitor request. Both modules of the monitor can place events in the event buffer so a "critical section" algorithm following Knuth (CACM May, 1966) is used to enter events in the buffer. MTR periodically scans the buffer pointers and if enough information is in the buffer for a tape record (1000B words) it places a request for the PP routine 1DB at the head of the ready queue.

The routine 1DB writes the event buffer to tape as long as at least a tape record of information is in the buffer. If the buffer does not contain 1000B words of information 1DB drops out. If the buffer is full when either of the two modules of the monitor tries to enter an event, a lost event count is incremented. The next time an event can be entered this count is entered into the buffer and the count is cleared.

The event buffer actually resides in the central memory buffer area of the system I/O driver GEMINI. The buffer is established when ISR finds that the special event dump tape drive in the EST has been turned on by the operator. IDB terminates an event tape dump when the end of tape is reached by turning off the tape drive.

The software event recorder is used heavily throughout the system monitor since the monitor's performance is critical to the performance of the entire system. In the following descriptions of MTR and CPM the event probe will be described in some detail to facilitate an understanding of the recorded events. These events are recorded only if the probe is turned on. The event recorder is activated by modifying certain instructions in the code of MTR and CPM when the event recording tape drive is turned on.

MTR, the PP portion of the system monitor, controls the overall operation of the system. The MTR main loop scans the PP output registers (OR) for requests for system monitor service. One OR is examined each cycle around the loop. If a request has not been issued, (i.e. contents of OR = 0) then MTR checks the current central processor exchange package to determine and set the central processor status. If a central processor request for monitor services has not been made (i.e. contents of RA + 1 = 0) or the central processor is idle, then the MTR outer loop is run. The PP output register address is advanced and the loop is started again.

If a PP has requested MTR service then the PP request subroutine is entered. CPM is activated, if necessary, to clear the MTR output register for event recording. The request is then checked for legality. If it is illegal, it is ignored and control is returned to the main loop. If it is legal then parameters are set up for the request processor subroutine and the request is recorded by the event recorder if it is active. The processing subroutine is then executed. The reply to the request is recorded upon return from the subroutine and the monitor reply is written to the PP's output register (OR). Control is then returned to the main loop where the system clock is updated and the central processor status is checked.

If a central processor request for monitor services has been issued then the central processor request subroutine is entered. The entry to the subroutine is recorded by the event probe. The request is then processed by the appropriate processor and the reply to the request is recorded. Control is then returned to the main loop where the system clock is updated, then the outer loop is run.

The outer loop consists of seven different subroutines one of which is executed each time through the outer loop. These routines perform the following functions.

1) Advance the system clock.
2) Check the central processor time quantum.
3) Call the intermachine communications processor.
4) Check the interactive time quantum.
5) Call the job scheduler.
6) Dump the event buffer.

7) Check delayed monitor requests.

The routine to advance the system clock simply reads the system channel clock and if this has advanced updates the system clocks.

The routine to check the central processor time quantum first checks to see if an error has been detected by CPM. If so, the error flag is set and the routine returns control to the outer loop. Otherwise, if the central processor quantum has expired then the central processor is exchanged by calling CPM to schedule a new job. The CPM reply is then stored and the routine exits to the outer loop.

The routine to call the intermachine communications processor is part of the software incorporated into the system to maintain communication between the 6600 and the 6400. It monitors the communications buffer (see figure 11) in system CM. When the communicator bit in the first word of this buffer is set, it schedules CPM to process the buffer.

The routine to check the interactive quantum checks to see how long an interactive job has been in CM. If the job has been in CM for more than a quantum of real time (currently one second) a scheduler run is requested to see if any other jobs are ready to run.

The routine to call the job scheduler first checks to see if job scheduler requests are pending. If they are pending, then CPM is called to schedule waiting jobs. The control point activity table (CPAT) is then scanned for jobs to be rolled in or out and the action is initiated.

The routine to dump the event buffer first checks to see if the event buffer is active. This is determined by reading the word (EBFP) in system CM which contains a pointer to the event buffer if the buffer is active. If the buffer is active then the subroutine which converts all the event recording instructions to an active state is called to make the instruction conversion if necessary. The buffer is then checked to see if at least one tape PRU (1000B words) of events have been recorded. If so, a call for the PP routine 1DB is entered at the head of the ready queue to begin dumping the buffer to tape. Control is then returned to the outer loop.

The check delayed request subroutine first checks the delay queue for entries with expired delays, which are placed in the ready queue. The ready queue is then checked and all ready requests which can be initiated are started by assigning a PP to process them. If a PP cannot be assigned to the request it is left in the ready queue. Otherwise the assignment is made and the request is deleted from the queue.

If the delay request was issued by a central processor program asking for recall (RCL) the request is cleared from the ready queue and the recall status of the control point is cleared. The clock is then advanced if necessary and the request queue is advanced. When the ready queue is exhausted or no free PPs remain control returns to the outer loop.

CPM, the central processor portion of the system monitor, is activated by a monitor exchange jump (MXN) from MTR. The CPM main loop begins by switching the central processor time charges from a user control point to the system monitor. Next system display areas in system CM are updated for display by DSD. Requests for central processor monitor services are then honored beginning with requests from MTR. Each pool PP output register (OR) is then scanned for central processor monitor functions or MTR replys to functions. Time bookeeping is then done and CPM exits by exchange jumping to a control point or the IDLE package if no control points want the central processor.

Function processing begins by checking to see if the function is an MTR function, which is ignored. If a CP monitor function is found it is processed by jumping to the proper subroutine.

II.E.2. Monitor functions.

The basic processing of monitor functions was described in the preceding section. Processing of individual functions will now be described. This section lists all PP function requests by function number with the name. Functions 1 through 22 and 27 are processed by the central processor portion of monitor independently of MTR. Functions 23 through 43, except 27, are processed by MTR, or by CPM at MTR's direct request while MTR waits.

(01) RMS - Request mass storage returns equipment number of least used mass storage device with the number of sectors requested available. This function is used when opening new files to distribute files evenly between mass storage devices.

(02) REQ - Request equipment requests a specific item of peripheral hardware for the control point. The EST is searched for the

equipment. If the equipment is found and is available it is assigned to the job requesting it. The assigned equipment bit in the JST is also set. If the equipment is not found in the EST the PP requesting the equipment is notified. If the equipment is assigned to another job the requesting PP is also notified.

(03) DEQ - Drop equipment releases the equipment assigned by REQ.

(04) CES - Change equipment status is requested to change the status bits in the EST.

The next four monitor functions handle mass storage allocations. Mass storage devices are partitioned into logical units called sectors and tracks. A sector is the smallest unit of storage available on a device and is sometimes called a physical record unit (PRU). The size of a PRU varies depending on the device. For all mass storage it is 64 CM words (100B words). Space on mass storage devices is allocated in half-tracks. The number of sectors per half-track also varies with the device. A track table (TRT) is maintained in ECS by CPM to record all reservations. The TRT elements are linked into a chain for each file. The FNT entry for each file in the system which resides on mass storage contains a pointer to the first track for the file.

(05) RHT - Request half track performs one of three possible functions depending on the format of the request. It may be used to reserve a chain of tracks, to extend or follow a chain of tracks, or to follow a chain of tracks without extending it.

(06) DHT - Drop half track may be used to drop a track chain, to truncate a track chain, or to remove a track from the middle of a

chain.

(07) CLS - Compute logical sector is used to find a new position on a mass storage device by tracing down the track chain.

(10) DTL - define track linking is used by the deadstart to initialize or recover track tables.

The next two functions are used to read and write ECS. Only CPM can read and write ECS under UT - 2D, since it is used only for swapping and data which must be shared by the two machines. User programs are denied access to ECS by setting FLX in their exchange packages to zero.

(11) REC - Read data from ECS.

(12) WEC - Write data to ECS.

The next three functions are monitor service functions for PPs.

(13) LPP - Locate peripheral program searches the peripheral program libraries (PLD and RPL) for a named routine and returns its location.

(14) PDE - Process device error produces an error message for a PP which has detected a data channel error or device error.

(15) FJN - Format job name simply formats a job name from the JST for printing or messages.

The following three functions are used to control allocation of data channels to PPs. The reserve channel functions check to see if the channel is already assigned to a processor. If it is, the request is placed on a channel queue until the channel becomes available. Otherwise the channel is assigned to the PP by setting the PP number in the channel table in system CM. The drop channel function dereserves the channel for the PP and then checks the channel queue to see if any requests for the channel are pending. If so, the request is honored, otherwise the channel assignment is cleared.

(16) RSY - Reserve system channel reserves a channel to a system device.

(17) RCH - Reserve channel reserves an I/O channel for the PP.

(20) DCH - Drop channel drops a channel reservation and optionally charges a job for the data moved across the channel.

The next function is part of the software event recorder.

(21) REV - Record event causes a record of the event to be placed in the event buffer if the probe is active.

The next function is used to alter bytes in the system limit word (SLMW) of the control point area.

(22) ACB - Alter control point byte alters bits in a given byte of a given word of the control point area. WARNING: The ACB function is a historical dreg and should be used only to alter the system limit word, SLMW. Other words should be altered directly by the PP, but only when the job is reserved.

The next three functions are used by PPs to reserve files or words for actions on files or tables associated with particular words. The first two functions are performed by CPM, at the request of MTR.

(23) RGF - Reserves a file in the global FNT for action by the processor. Reservation is indicated by clearing bit 0 of the FNT entry. The PP sets bit 0 when the activity on the file is complete.

(24) RLF - Reserves a file in a local or control point FNT.

(25) RWD - Reserve word reserves any word in CM in the same manner as RGF reserves a file.

The following function is used for diagnosing system problems. When the system monitor is in step mode it processes only one function and then waits for a signal from the operator to process another. The function is issued only by the display driver at the operator's request. It is processed by MTR.

(26) STM - Step monitor sets and clears monitor step mode.

The following function is used by MTR to request services of CPM.

(27) MTR - MTR function request for CPM. An MTR function is issued to CPM as a part of the processing of several monitor functions. The functions which CPM performs at the request of MTR are:

1) Scheduler run.
2) Storage move.
3) Verify system tables.
4) Process intermachine communication buffer.
5) Record event.

The remaining functions are processed by MTR. Some of them require CPM services.

(30) RPP - Request peripheral processor is used by a peripheral processor to request the services of an additional peripheral processor.

(31) EDR - Enter delayed request is used by a peripheral processor to enter a request for a peripheral job after a specified delay.

(32) DPP - Drop peripheral processor is used to signal MTR that the

PP is no longer needed. The input register (IR) is cleared and the available PP count is updated.

(33) ABT - Abort control point is used when an error is detected by a PP program which requires termination of the job.

(34) RCP - Request central processor is used to indicate that the control point is loaded with a job which can be executed.

(35) RCL - Recall central program is used to indicate that a PP activity requested by the program at a control point is complete and that the central program may be taken out of recall status.

(36) DCP - Drop central processor is used to indicate that the control point no longer contains a program fit to run. It is also used (followed by a RCP later) to enable a PP to examine a control point's exchange package or make other timing dependent accesses to the control point memory.

(37) CEF - Change error flag is used to set the error flag in the JST to indicate a PP error.

(40) RST - Request storage is used to request a change in the amount of CM assigned to a job.

(41) PFR - Pause for relocation is used to indicate to MTR that a PP is in a state such that MTR can move a job up or down in CM.

(42) SCH - Schedule jobs is issued to request a scheduler run.

(43) SCP - Swap control point is issued to signal a change in the swap status of a control point as reflected in CPAT. The usual case is to force a rollout.

The last four functions are handled cooperatively by MTR and CPM. RST and PFR will be considered first. If the PP is requesting a decrease in field length, the new field length is granted and the CM which was released by the request is recorded. Control then returns to the PP. If an increase was requested, a check is first made to see if sufficient storage is available to grant the request. If no storage is available, the reply field length is unchanged. If a gap of sufficient size is found in CM, a storage move is executed to give the job the required FL. The actual storage move is executed by CPM on request from MTR. The function PFR is a vital part of this process since a PP executing at a control point must suspend addressing CM during a storage move. The PP indicates this by issuing a PFR function. PP resident resets the RA and FL values in PP memory after each PFR function to insure that the PP has the correct RA and FL for addressing CM.

The function SCH is issued to initiate a run of the job scheduler. The scheduler works from a table in CM known as the central processor action table (CPAT) and the JST. When a scheduler run is requested the scheduler scans the JST to find jobs whose status has changed since the last scheduler run. The category and cost for the job are recalculated and a new category is assigned. The JST is then sorted to establish the job priorities. The scheduler then scans the CPAT to find jobs in a running state and sets the sign bit of each such job. The sorted JST is scanned and for each job to be run, the sign bit in the CPAT is cleared or a swap-in request is stored in the CPAT. The CPAT is again scanned and for each sign bit still set, a swap out request is entered in the CPAT. MTR scans the CPAT and calls either 1SJ to swap jobs out or 1RJ to roll jobs in. 1SJ issues a SCH function at the end of a swap out to initiate a scheduler run for the newly available central memory.

II.F. Intermachine communication.

The University of Texas computer system consists of a CDC 6600 and a CDC 6400 computer system. The two systems are physically interconnected as shown in Figure 1. This section describes the intermachine communications used to coordinate the operation of the two computers. The computers are run as two separate communicating computers.

The UT - 2D permanent file system is described in this section since its operation is controlled by information passed back and forth by the two machines.

II.F.1. Capabilities, files, and jobs.

The basic communication between the two machines is handled by DSD and MTR. The MTR processing was described in section II.E.1. DSD examines the high order byte of the coupler communications buffer header word for the DSD transmission bit to be set. If it is, DSD writes the buffer to be transmitted. Similarly, if DSD detects a coupler status indicating transmission from the other machine then it reads from the coupler to the buffer. After the coupler is read, the monitor communicator bit is set in the buffer header word to indicate that a buffer has been read and needs to be processed by the monitor. If no buffer is waiting to be written the coupler is scanned periodically for messages from the other machine. If both machines want to write, the 6600 does the write. This occurs only after a deadstart and is sufficient to establish the alternating communication protocol.

Three classes of messages are sent by this mechanism. Inquiries about the status of files are passed between machines. If a file is to be transferred from one machine to another the FNT entry for the file is moved into the buffer as a reply. Every ten minutes a time of day request is passed to the machine with the channel clock for a check on time of day and date. A capability table, figure 13, is passed between the machines defining the I/O capabilities of each machine. This defines which machine will process files such as output files waiting to be printed.

Four types of files are transferred by this mechanism. The system validation file (QRT) is passed between machines upon demand by PISCES. Queue files are passed to the machine which contains the equipment for processing them. Job files are generally passed from the 6400 to the 6600 since all non-interactive jobs are usually run on the 6600. Permanent file set entries are passed on demand. More detailed discussion of permanent file mechanisms are presented in the next section.

Notice that only FNT entries are passed across the channel coupler. In order to access these files the TRT for the mass storage devices must be available to each machine. Consequently, the TRTs are stored in ECS where they are rapidly accessible to each machine, and interlocked by use of bits in the ECS flag register.

II.F.2. Permanent files.

Permanent files under UT - 2D are resident on magnetic tapes. Each tape contains a directory for the permanent file set which is stored on that reel. When a job makes a reference to a file in the file set the reel is copied to disk and an entry is made in the system FNT which points to the directory for that file set. The directory is then scanned for the particular file referenced and a working copy of the file is created on a 6638 disk.

A job reference to a permanent file actually triggers a search of both machines FNT for the directory for the file set referenced. This search is implemented through the coupler communicators. All permanent file set loads are run on the 6600 if possible.

File sets which have been altered are copied back to tape periodically to preserve changes. If a file set remains unreferenced for more than a few hours it is purged from the disk.

REFERENCES

Control Data 6000 series Computer Systems Reference Manual. Pub. No. 60100000, Control Data Corp. Arden Hills, Minn, 1973.

Howard, J. H. The Coordination of Multiple Processes in Computer Operating Systems. TSN-16, Computation Center, The University of Texas at Austin, Austin, Texas, 1970.

Thornton, J. E. Design of a Computer: The Control Data 6600. Scott Foresman, 1970.

Figure 1

CDC 6600-6400 CONFIGURATION

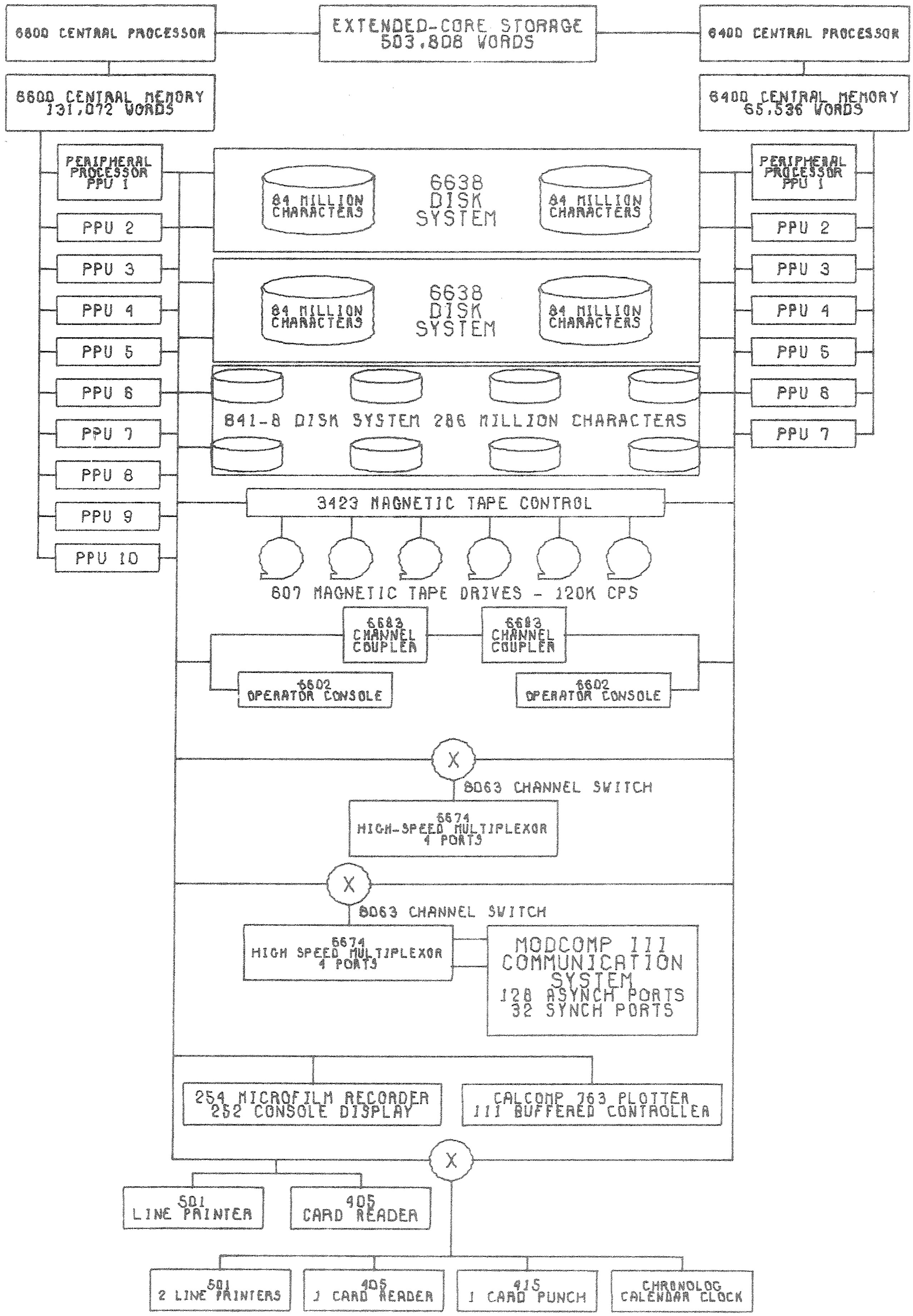
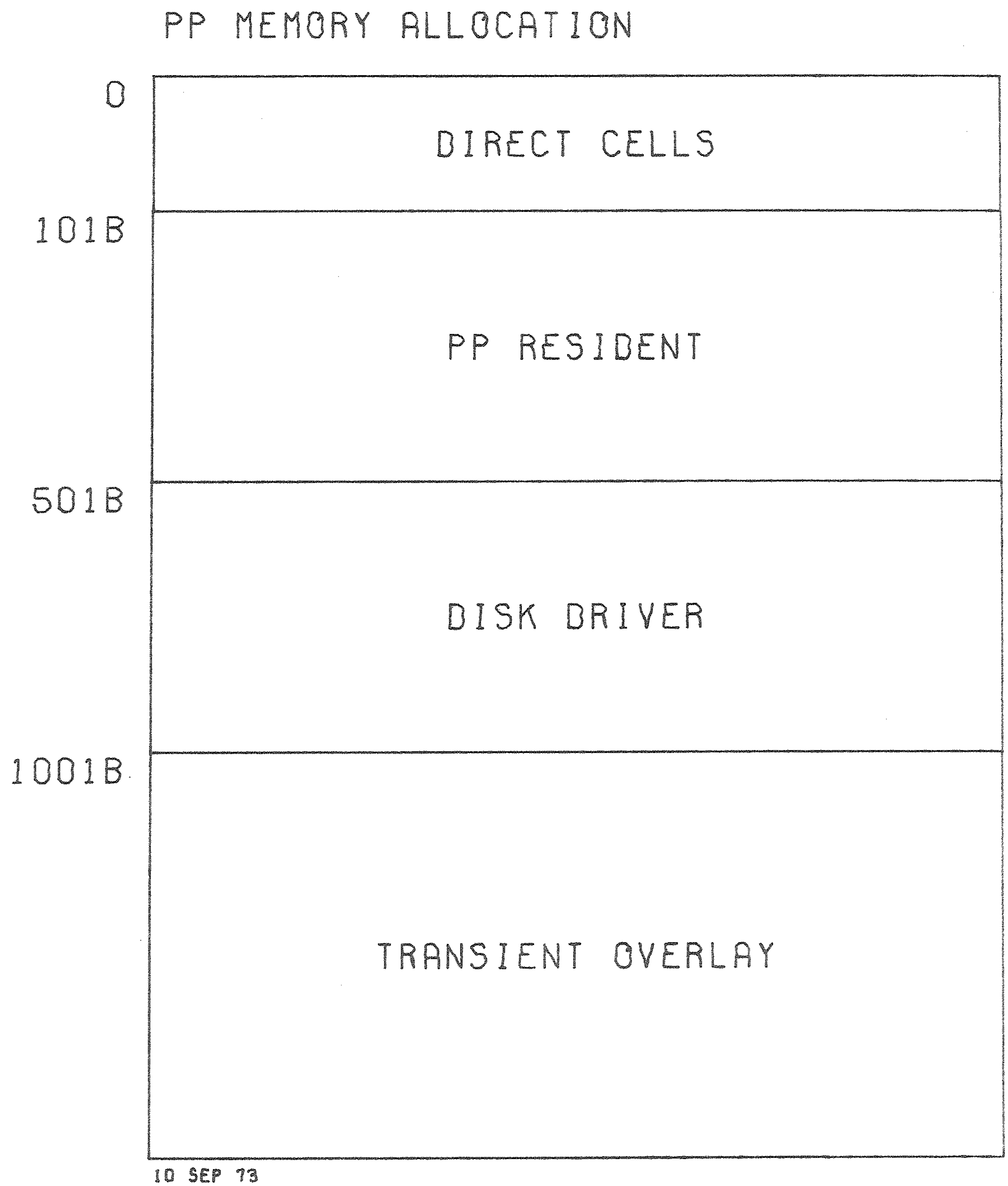


Figure 2


PP MEMORY ALLOCATION
0
DIRECT CELLS
101B
PP RESIDENT
501B
DISK DRIVER
1001B
TRANSIENT OVERLAY
10 SEP 73

UT-2D PP DIRECT CELLS

0X	T0	T1	T2	T3	CH	EQ	TN	SN
1X	CM					LA	NM	
2X	FS							
3X	ES							
4X	BA					BS		
5X	IR					RA	FL	FA
6X	FT		IN		OT		LM	
7X	ON	HN	TH	TR	CP	IA	OA	MA

T0 IS CHANGED AS A SIDE EFFECT OF CERTAIN PP INSTRUCTIONS

CM,FS,ES,BA,IR ARE CM READ-WRITE BUFFERS

IR USUALLY CONTAINS A COPY OF THE INPUT REGISTER

CM IS USED TO READ-WRITE THE OUTPUT REGISTER

RA,FL,ON,HN,TH,TR,CP,IA,OA,MA ARE MAINTAINED BY PPR AND SHOULD NOT BE CHANGED BY THE PP PROGRAM

FT,IN,OT,LM ARE USED BY I-O PROGRAMS TO CONTAIN CIRCULAR BUFFER POINTERS

CH,EQ,TN,SN ARE USED BY THE DISK DRIVERS

Figure 4

UT 2D CENTRAL MEMORY ALLOCATION

ZERO	TABLE POINTERS
CPSL	CONTROL POINT STATUS TABLE
PPCA	PP COMMUNICATIONS AREA
EST	EQUIPMENT STATUS TABLE
FNT	FILE NAME TABLE
JST	JOB STATUS TABLE
CCB	COUPLER COMMUNICATIONS BUFFER
DFB	DAYFILE BUFFER
ECB	ECS CIO BUFFER
EDB	ECS DISPLAY BUFFER
CLD	CENTRAL LIBRARY DIRECTORY
PLD	PERIPHERAL LIBRARY DIRECTORY
CMXA	CPU MONITOR EXCHANGE AREA
MEMA	MEMORY MAP FOR DISPLAY
PCT	PROCESSING CAPABILITY TABLE
OCB	OPERATOR COMMUNICATIONS BUFFER
CPM	CPU MONITOR
RPL	RESIDENT PERIPHERAL LIBRARY
	CONTROL POINTS
TRT	TRACK TABLES IF NO ECS

11 JUL 73

UT 2D TABLE POINTERS

	ZERO	0				
*1	CCBP	CCB	LIMIT	EDB	ECS ADDR	
*2	PLDP	PLD	LIMIT			RPL
	DFBP	DFB	IN	OUT	LIMIT	
*3	FNTP	FNT	LIMIT	FREE		ECB
	ESTP	EST1	EST2	EST3	LIMIT	COUNT
	JSTP	JST	LIMIT	FREE		
*4	CLDP	CLD	LIMIT			EBFP
	OMDL	OPERATOR MESSAGE				
	CTIL	CH 0	CH 1	CH 2	CH 3	CH 4
		CH 5	CH 6	CH 7	CH 10	CH 11
		CH 12	CH 13	CH 14		
	MFLL		MAX FL	STD TKS	USED FL	MACH FL
	CPEF					CP EFF
	PPEF					PP EFF
	CPIT			CP IDLE TIME		
	PPIT			PP IDLE TIME		
	CPMT			CP MONITOR TIME		
	TIML	HH.MM.SS.				
	DTEL	DD MMM YY				
	SYSL	SYSTEM LABEL				
	RTCL			TIME SINCE DEADSTART		
	SIFL	MODE	MESS	CP STEP	IDLE	

MODE
0001 - NO UNLOAD
0002 - PASSWORD GATING
0004 - INDEPENDENT
0010 - 6400
0020 - NO QRT
0040 - VACATE
0100 - UNCOUPLE
0200 - NO BATCH
0400 - DFLT LOADER = CP

2000 - DISCOUNT TIME

MESS
0001 - SWAP
0002 - PUNT
0004 - RFL
0010 - 1XX
0020 - PFM
0040 - RFL

IDLE
0001 - BATCH
0002 - JOBS
0004 - EQUIPMENT
4000 - SCHEDULER

*1: EDAL = EDBP = CCBP
*2: RPLP = PLDP
*3: ECBP = FNTP
*4: EBFP = CLDP

11 JUL 73

Figure 6

UT 2 CONTROL POINT STATUS TABLE

(CPSL+I) = CPSL[I]

STATUS	ERROR	PC	AC	RA/100B	FL/100B

STATUS: AWPX------RM
- A = ACTIVE CP
- W = WANT CPU
- P = AUTO RECALL
- X = RECALL
- \- = UNUSED
- R = ROLLOUT
- M = MOVE

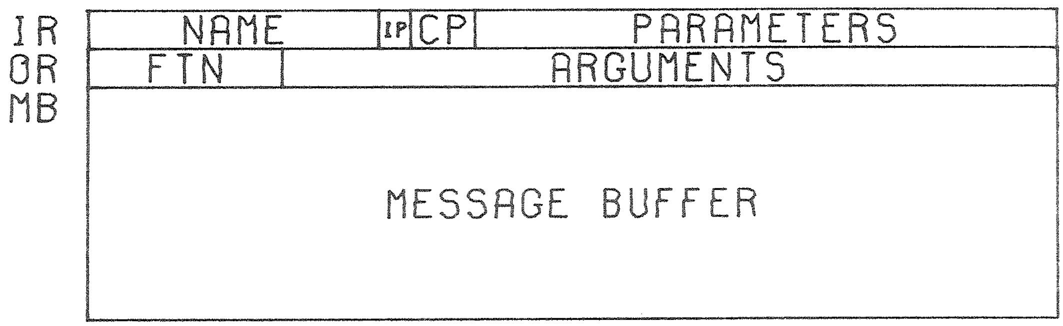
ERROR:
- 0 = NONE
- 1 = TIME LIMIT
- 2 = ARITHMETIC ERROR
- 3 = CP OR PP ABORT
- 4 = PPRB OVERFLOW
- 5 = PP CALL ERROR
- 6 = OPERATOR DROP
- 7 = TRACK LIMIT

PC: (PERMANENT COUNT)
= NUMBER OF PERMANENT PPS

AC: (ACTIVITY COUNT)
= NUMBER OF PPS + NUMBER
OF DELAY STACK ENTRIES

11 APR 72

UT 2 PP COMMUNICATIONS AREA



I = INTERNAL USAGE
P = AUTO RECALL

11 APR 72

Figure 8

UT 2 EQUIPMENT STATUS TABLE

(EST1+N) = EST1[N]

JOB	CHNLS	FLAGS	TYPE	CODES

(EST2+N) = EST2[N]

		ID 2	ID 1	FILE

(EST3+4*N) = EST3[N]

MS

ERR CT		N FREE	SL	FLAG REG
E TRT ADDRESS		F FREE		F SYST

DS

ERR CT		OCB POINTER

NON-MS

ERR CT	SLOT	REEL
	STATUS MESSAGE	

FLAGS:

1 = OFF
2 = UNAVAILABLE
4 = DOWN
10 = 6000 SERIES
20 = MASS STORAGE
40 = SYSTEM DEVICE
100 = END
200 = REPEAT
400 = SUPPRESS
1000 = GO
2000 = UNLOAD

CODES:

EQUIP	6681	UNIT

E = ECS TRT

11 JUL 73

Figure 9

UT 2 FILE NAME TABLE

FILE NAME				PR	TYPE	F
ID	EQ	FT	CT	CS	LIMIT	

F:
1 = NOT RESERVED

TYPE:

	00X	01X	02X	03X
XX1		OUTPUT		
XX2	USERPF			
XX3	SYSTEM	DUAL		
XX4	PUNCH	PUNCHB	PAB	
XX5	FILMPL	FILMPR		
XX7	PLOT	PLOTI	PLOTW	PLOTWI

11 JUL 73

Figure 10

UT 2 JOB STATUS TABLE

JOB NAME				CP	FLAGS
ID	EQ	FT	TL	FL	PL/OAE
CAT		COST		SLICE	SORTNDX
WAIT STATUS					

FLAGS:

1	=	NOT RESERVED
2	=	EQUIPMENT ASSIGNED
4	=	CONVERSATIONAL JOB
10	=	SUBMIT JOB
20	=	INITIALIZED JOB
40	=	SPECIAL JOB
100	=	MEMORY INSIGNIFICANT
200	=	ZAPPED JOB
400	=	RESTART JOB
1000	=	PFD ATTACHED
2000	=	DISCOUNT JOB

WAIT STATUS:

READY		ROLLOUT TIME			0
COMMENT	0	JOBADDR	MSGADDR		1
KILL	1				1
EXECUTE	2				1
DROP	3				1
ABORT	4	DUMP			1
GO	5	SET SW			1
LOGOUT	6	LN TIME	TMLIMIT		1
TIMEREQ	7		ADDRESS		1
REQUEST	SLOT⌐REEL				*2*
ASSIGN	EQ TYPE				3
PAUSE	0				4
RTK	1	TRACKS			4
RTL	2	TIME			4
REQMT	3	MT	NN	.	3
FILWAIT	FILENAME				5
PFDWAIT	PFD⌐NNNN				6
TTYWAIT		ROLLOUT TIME			7
OCBWAIT	DISPLAY CODE FOR OCB WAIT				10

2+2000 = MULTI-REEL
2+4000 = WRITE-RING

11 JUL 73

Figure 11

COUPLER COMMUNICATIONS BUFFER

CCB	FLAGS			DELAY
CCE	Q[1]	R[1]		N[1]
	MESSAGE[1]			
	. . .			
	Q[K]	R[K]		N[K]
	MESSAGE[K]			
	0			
	IGNORED			

FLAGS - 4000 - NEEDS CPM PROC
2000 - NEEDS DSD XMSN
DELAY - DSD CYCLES BEFORE XMSN
Q[I] - QUERY NUMBER
R[I] - 0-QUERY, 1-REPLY
N[I] - MESSAGE+HEADER LENGTH

SYSTEM FILE MESSAGES

0	1	R			3
1	NAME				31B
2	FST ENTRY				

WORD 2 = 0 IF R=0

PFD MESSAGES

0	2	R			3
1	PFDNNNN				21B
2	FST ENTRY				

WORD 2 = 0 IF R=0

TIME+DATE MESSAGES

0	3	R		3
1	HH.MM.SS.			
2	DD MMM YY			

WORDS 1,2 = 0 IF R=0

11 JUL 73

Figure 12

COMMUNICATIONS (CONT)

CAPABILITY MESSAGES

Word			
0	4		11B
1...10B	SENDER-S CAPABILITY TABLE		

FILE MESSAGES

Word			
0	5		3
1	FNT ENTRY		
2	FST ENTRY		

JOB MESSAGES

Word			
0	6		3
1	JST WORD 0		
2	JST WORD 1		

11 APR 72

CAPABILITY TABLE (PCT)

EACH WORD IS DIVIDED INTO

OTHER MACHINE	THIS MACHINE

0	SE	JST	JOBS	JOB PROCESSING
1		FNT	OUTPUT	PRINT EQUIPMENT
2			SYS	SYSTEM FILES (QRT)
3			PFD	PERMANENT FILES
4			PUNCH	PUNCH EQUIPMENT
5			FILM	FILM EQUIPMENT
6			CLOCK	CLOCK EQUIPMENT
7			PLOT	PLOT EQUIPMENT

JOB PROCESSING FLAGS
- 0001 - BATCH CAPABILITY
- 0002 - SPACE FOR LARGE BATCH
- 0004 - INTERACTIVE JOBS

EQUIPMENT FLAGS
- 0001 - IDS 00...07
- 0002 - IDS 10...17
- 0004 - IDS 20...27
- ...
- 0200 - IDS 70...77

IDS
- 0X,1X,2X CENTRAL SITE
- 4X,5X TAURUS RJE
- 6X,7X EXPORT RJE

PFD AND QRT FLAGS
- 0000 - VACATE MODE
- 0377 - NORMAL MODE

11 JUL 73

Figure 14

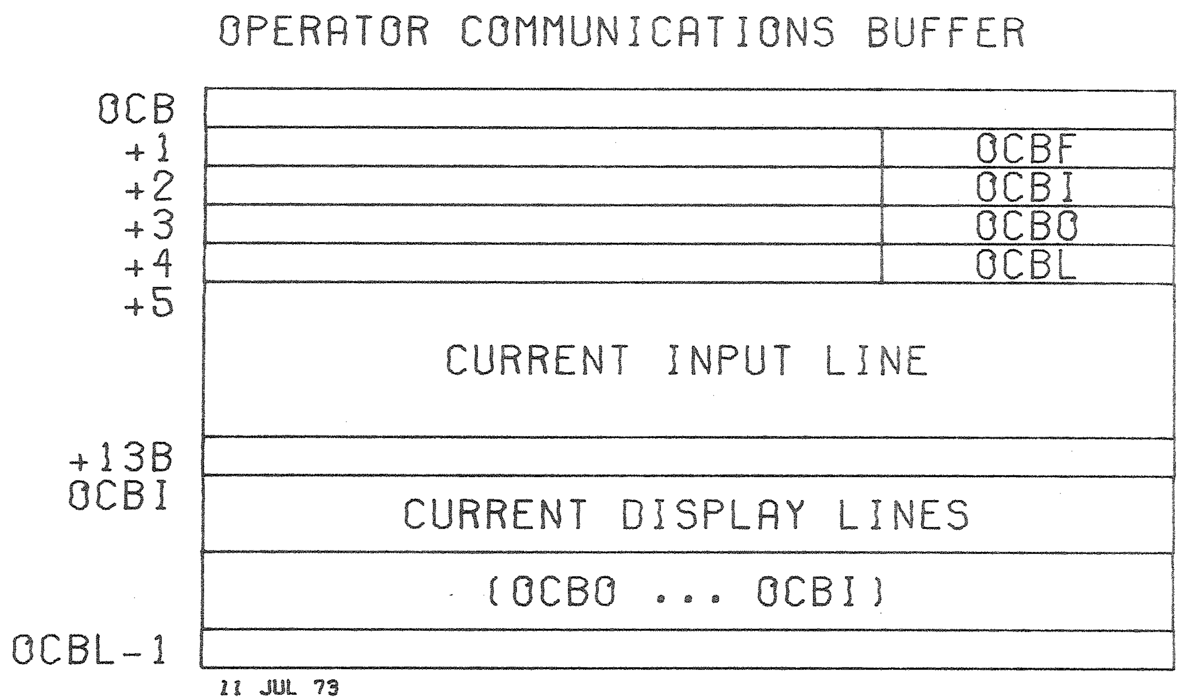

OPERATOR COMMUNICATIONS BUFFER
OCB
+1
+2
+3
+4
+5
+13B
OCBI
OCBL-1
OCBF
OCBI
OCBO
OCBL
CURRENT INPUT LINE
CURRENT DISPLAY LINES
(OCBO ... OCBI)
11 JUL 73

Figure 15

CENTRAL LIBRARY DIRECTORY

0	NAME					FLAGS
1	SN	EQ	FT	CSUM	FL	LSC

FLAGS: 1 - RESERVATION
2 - OVERLAY
4 - CC LOADABLE (CC)
10 - PRIVILEGED PROGRAM
20 - LIBRARY
40 - FLEXIBLE FL
100 - CLD CREATED

FL - MIN. FL/100
LSC - LOADER STAT. COUNT

PERIPHERAL LIBRARY DIRECTORY

NAME		ORIGIN	TRACK		SE

ORIGIN INCLUDES HEADER
TRACK,SE = DISK ADDRESS

RESIDENT PERIPHERAL LIBRARY

0	NAME		ORIGIN		N
1	PP CODE				
N	NEXT ENTRY				

N AND ORIGIN INCLUDE HEADER

11 JUL 73

Figure 16

CPU MONITOR EXCHANGE AREA

CMXA	CP MONITOR PACKAGE
IDLA	IDLE PACKAGE
TXJA	XJ TRAP PACKAGE
CPAT	CP ACTION TABLE

CP ACTION TABLE

	INDX	ACTION

SET BY SCHEDULER
PROCESSED BY MTR
INDX - JST INDEX
ACTION:

0 - VACANT CP	1 - CALL 1RJ
2 - ROLL IN	10 - ACTIVE
11 - CALL 1SJ	12 - ROLLOUT

15 OCT 73

Figure 17

TRACK RESERVATION TABLES

E 4000	E 4001	E 4002	E 4003
E 4004	E 4005	E 4006	E 4007
	. . .		
E 7774	E 7775	E 7776	E 7777

E XXXX - ENTRY FOR TRACK XXXX

ENTRY FORMAT

R	F	P	LINK

R - 1 IF TRACK RESERVED
F - 1 IF FIRST TRACK OF CHAIN
P - EVEN PARITY BIT
LINK - NEXT TRACK OF CHAIN
(0 IF LAST TRACK)

SHORT SECTOR DEFINITION TABLE

STORED IN SECOND HALF OF ECS TRT
SEE TRT FOR PACKING

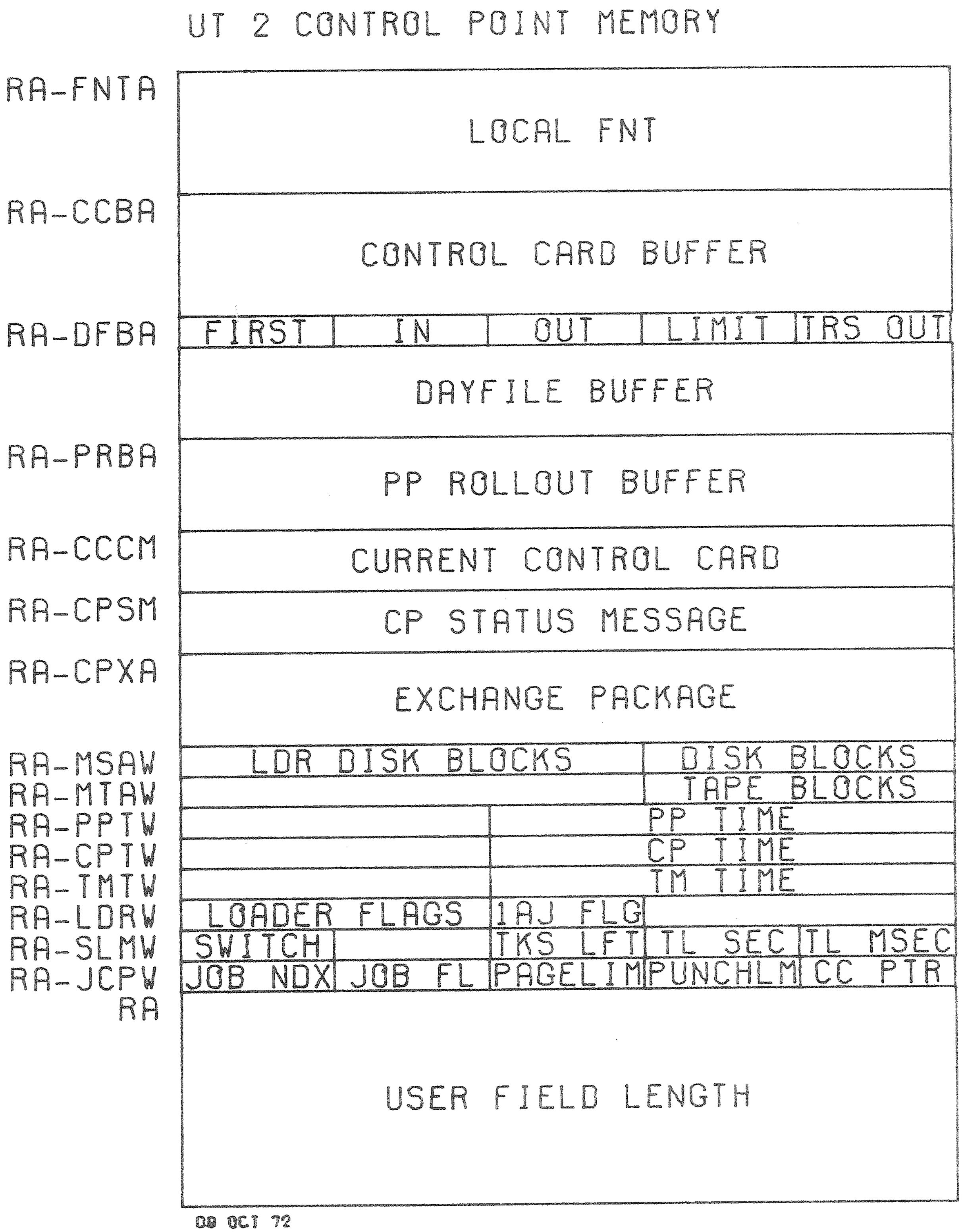
ENTRY FORMAT

	P		0	1	2	3	4	5	6	7

P - EVEN PARITY BIT
I - 1 IF SECTOR I SHORT
(LAST WORD OF SECTOR
CONTAINS WORD COUNT.)

11 APR 72

Figure 18


UT 2 CONTROL POINT MEMORY
RA-FNTA
LOCAL FNT
RA-CCBA
CONTROL CARD BUFFER
RA-DFBA
FIRST
IN
OUT
LIMIT
TRS OUT
DAYFILE BUFFER
RA-PRBA
PP ROLLOUT BUFFER
RA-CCCM
CURRENT CONTROL CARD
RA-CPSM
CP STATUS MESSAGE
RA-CPXA
EXCHANGE PACKAGE
RA-MSAW
LDR DISK BLOCKS
DISK BLOCKS
RA-MTAW
TAPE BLOCKS
RA-PPTW
PP TIME
RA-CPTW
CP TIME
RA-TMTW
TM TIME
RA-LDRW
LOADER FLAGS
1AJ FLG
RA-SLMW
SWITCH
TKS LFT
TL SEC
TL MSEC
RA-JCPW
JOB NDX
JOB FL
PAGELIM
PUNCHLM
CC PTR
RA
USER FIELD LENGTH
08 OCT 72

6X00 EXCHANGE PACKAGE

	P	A0	
	RA	A1	B1
	FL	A2	B2
	EM	A3	B3
RA(ECS)		A4	B4
FL(ECS)		A5	B5
	MA	A6	B6
		A7	B7
	X0		
	X1		
	X2		
	X3		
	X4		
	X5		
	X6		
	X7		

11 APR 72

Figure 20

UT 2 LOCAL FNT

DISK

FILE NAME					FLAGS
	EQ	FT	CT	CS	STATUS

TAPE

FILE NAME					FLAGS
	EQ		BYTES	MODE	STATUS

FLAGS:
- 1 = NOT RESERVED
- 2 = READ ONLY
- 4 = NON-RETURNABLE
- 10 = EOR ENCOUNTERED (READ NS)
- 20 = LOCAL PFD

MODE:
- 3 = DENSITY
  - 1 = HI
  - 2 = LO
  - 3 = HY
- 4 = X
- 10 = RO
- 20 = WR
- 40 = PE
- 100 = MR
- 200 = LB
- 400 = REWIND AT EOT

11 JUL 73

UT 2D FILE ENVIRONMENT TABLE

FET	FILENAME				OP
FET+1	DEV	FLAGS		K	FIRST
FET+2					IN
FET+3					OUT
FET+4	FNT	PRU			LIMIT
FET+5					
FET+6	CURR POS			NEW POS	

OP = CIO OPERATION CODE (EVEN)

FIRST = BUFFER ORIGIN

IN = FIRST VACANT WORD IN BUFFER

OUT = FIRST FULL WORD IN BUFFER

LIMIT = BUFFER LIMIT

K = FET LENGTH IN EXCESS OF 5

DEVICE = 2 LETTER DEVICE MNEMONIC IF MASS STORAGE
4000+MNEMONIC IF NOT MASS STORAGE

FLAGS = 4XXX RANDOM ACCESS FILE
1XXX USER PROCESSING BIT
X4XX ERROR PROCESSING BIT

FNT = ADDRESS OF LOCAL FNT ENTRY

PRU = SIZE OF ONE PHYSICAL RECORD ON DEVICE

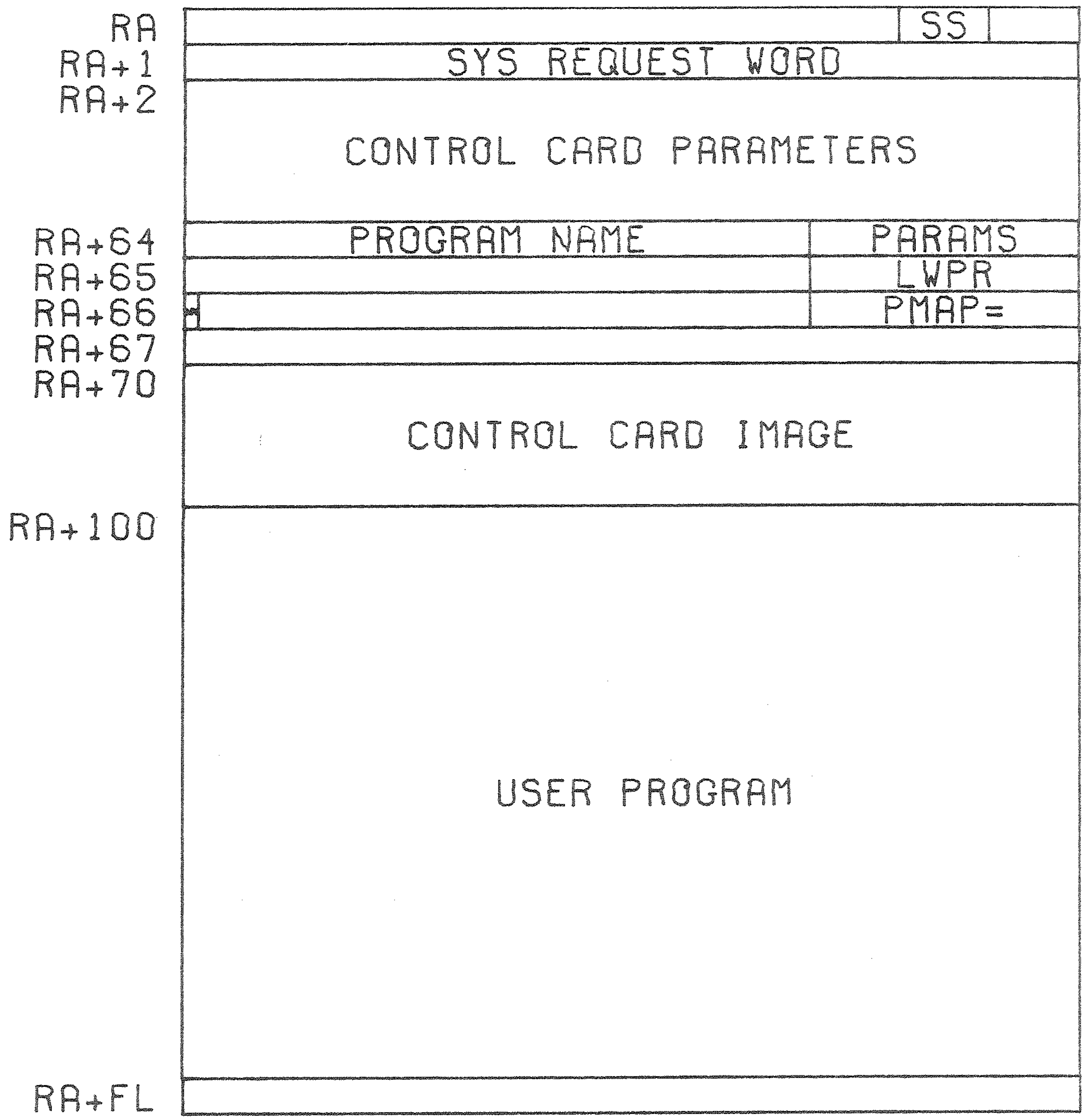
CURR POS = CURRENT LOGICAL SECTOR +400000

NEW POS = DESIRED LOGICAL SECTOR +400000

17 JUL 73

Figure 22

USER CENTRAL MEMORY



LWPR - LAST WORD ADDRESS OF PROG
XJPR - BIT 59 OF RA+66
= 0 DO NOT USE XJ
= 1 XJ INSTRUCTION OK

17 JUL 73

Figure 23

SYSTEM REQUEST FORMAT

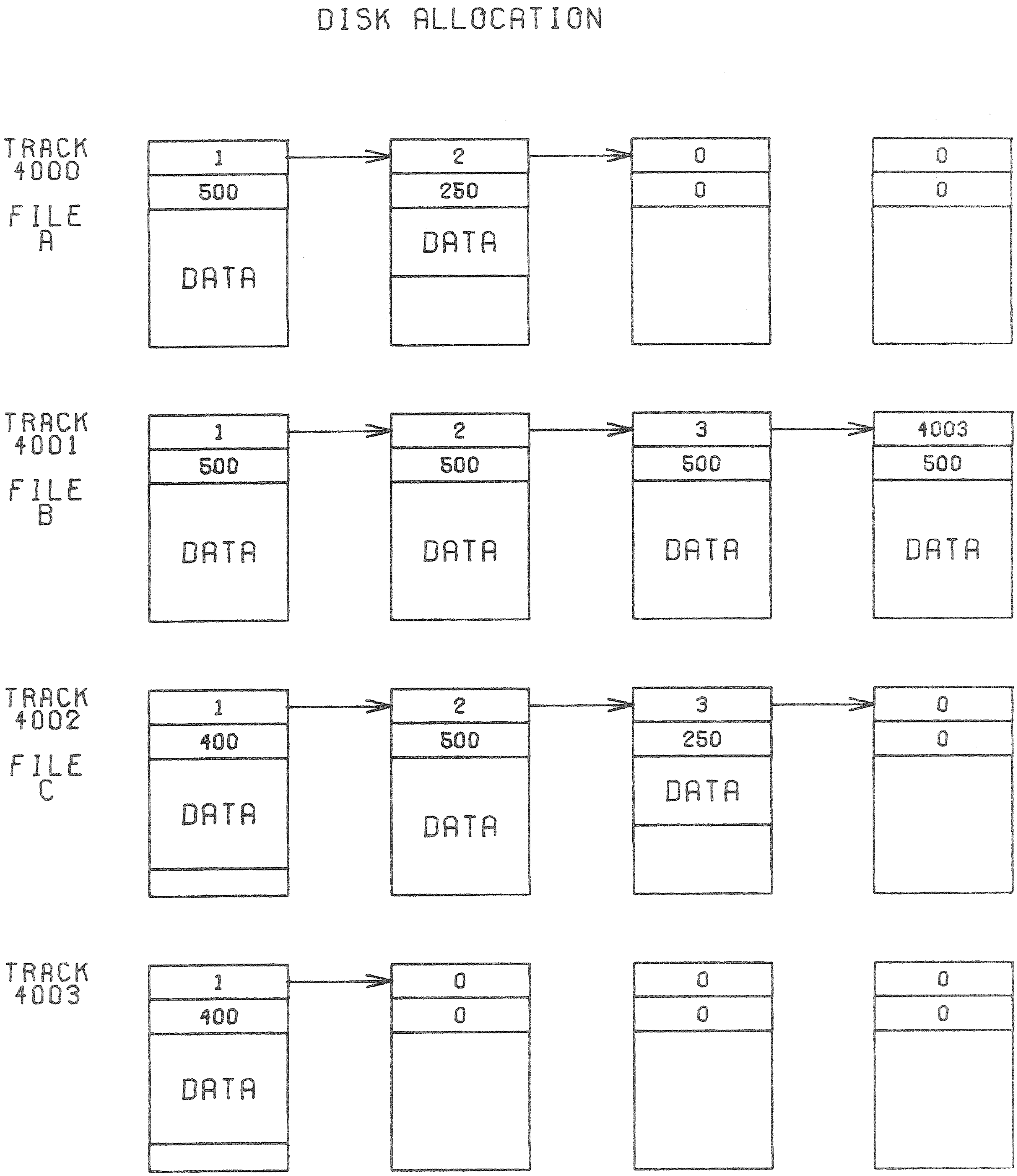
FFF	RCL	P2	P1

FFF	RCL	P3	P2	P1

FFF = SYSTEM REQUEST NAME
RCL = 20 IF AUTO RECALL
P3 = TERTIARY PARAMETER
P2 = SECONDARY PARAMETER
P1 = PRIMARY PARAMETER

17 JUL 73

Figure 24


DISK ALLOCATION
TRACK 4000
FILE A
1
500
DATA
2
250
DATA
0
0
0
0
TRACK 4001
FILE B
1
500
DATA
2
500
DATA
3
500
DATA
4003
500
DATA
TRACK 4002
FILE C
1
400
DATA
2
500
DATA
3
250
DATA
0
0
TRACK 4003
1
400
DATA
0
0
0
0
0
0