A SPECIAL PURPOSE TIME SHARING SYSTEM

FOR

THE P.D.P.8 COMPUTER

A Preliminary Report By

Ad J. van de Goor

August 31, 1967

TABLE OF CONTENTS

page

INTRODUCTION........1

A. Reason for Timesharing........3

B. TS8 Configuration and Console Cost........4

C. Approach to the PDP-8 Timesharing System........4

CHAPTER I - STRUCTURE OF THE HARDWARE........7

A. Brief Description of the PDP-8 Processor........7

B. Primary Memory........8

C. Secondary Memory........9

D. Teletype Interface........10

CHAPTER II - STRUCTURE OF THE VIRTUAL MACHINE........12

A. Program Environment........12

B. Files........12

C. Console Control of the Virtual Machine........13

CHAPTER III - STRUCTURE OF SYSTEM PROVIDED FACILITIES SEEN FROM VIRTUAL MACHINE........14

A. Conversion of DEC Software........14

1. Editor........14

2. Desk Calculator........15

3. Assemblers: Macro and Pal........15

4. DDT........15

5. Loader........15

6. Fortran........16

B. Software to Operate in Conjunction with Computer Networks........16

1. Job Flow to Processors........16

2. Job Sharing........17

CHAPTER IV - STRUCTURE OF THE MONITOR........18

A. Map of Memory........18

1. Resident Programs and Files........18

2. Non-resident Programs and Files........20

B. File Structure........21

CHAPTER V - IMPLEMENTATION PLAN........23

LIST OF FIGURES

page
Figure I-1 PDP-8 TS8 Configuration........................24
Figure I-2 PDP-8 Major Register Block Diagram..............25
Figure I-3 Data Communications System Block Diagram........25
Figure II-1 State Diagram of Virtual Machine...............26
Figure IV-1 Disc File Structure............................27
Figure IV-2..28
Figure IV-3..28
Figure IV-4..28
Figure IV-5..29
Figure IV-6 User File Structure............................30

LIST OF APPENDICES

Appendix A Monitor Instructions............................31
Appendix B Current File Information........................33
Appendix C Input Output Buffers............................34
Appendix D Simulation Results..............................36

INTRODUCTION

SUMMARY

The system described in this report is being designed for 10 to 50 users, depending on the user demands. Though the system can stand alone, it is to work in conjunction with a larger system, the Remington-Rand 1108-EXEC 8, and the experimental IBM TSS-360/67.

In a general effort to make computing facilities more available to users and study the range of Time Shared Computer usage it is desired to design and study a special purpose time sharing system using a PDP-8 computer at Carnegie-Mellon University. The configuration is shown in Figure I-1. It is felt that particular jobs can be selected to best match particular computing facilities, and this proposal is a suggestion for such an experiment.

A time sharing system is one where several simultaneous users have access to a single computer or computer(s) through a network. Only a small degradation in performance (as measured by problem solving speed) occurs to any user and a user is supplied with essentially the same source (before time sharing) because the facility being requested (computing power, data files, etc.) requires only a small amount of the power for a short interval. The idea of Time Sharing, though very old*, has been attempted on the larger, most general computers, and as such, the behavior of such systems for users with small jobs entails a high overhead (or cost is large).

The system attempts to go beyond the obvious case of just concentrating messages (either on a character by character or a line by line basis) for

* C. Stratchey's paper (1959).
"Time-sharing in large fast computers"
UNESCO conference Paris 1959

the larger computers (although doing this function for only 20 users is worthwhile from an economic viewpoint).

The system is designed for three specific environments:

1. Companion mode: (used with a larger computer or computers).
   a. Editing a complete file: Files reside in the larger system and editing is done at periphery because interaction rates are high, and computing capacity required is low.

   b. Editing only pages: Text pages are transferred to the TS8 and edited.

   c. Trivial calculations: desk calculator, very small ALGOL and FORTRAN jobs.

   d. Pre-processing, special editing, and syntax checking for on-line languages like LCC, JOSS, BASIC, etc., prior to interpretation in a larger machine.
2. General purpose, stand alone time sharing with very limited capacity (suitable for secondary school or similar environment, where the computational demands are not extremely high).

3. Base for special purpose stand alone systems. E.g., shared data collection system; Time Shared Keypunch consoles for data entry and transformation; component of a network to concentrate messages, and do local processing.

The virtual machine provides the user with a console (a teletype KSR33 initially), a portion of the processor capacity for running 4K programs (larger programs, up to 64K, have to be managed by the user), 8 open files, shared file capacity of 200-400 text pages for all users, and a tape system which is used for long term file system.

A. REASONS FOR TIME SHARING

There are many commonly promoted reasons* for having a time-sharing system. Some are:

1. Better man-machine communication;
2. More efficient use of equipment and as such;
3. Effectively more computers available for use.

Better Man-Machine Communication

Literature on time sharing systems states that a time shared computer seeks to provide better service to users by providing better man-machine communications. The program debugging process can be made more efficient if the user can be present at the computer console and correct mistakes as they occur. Results are available very shortly after requests.

Efficient Use of Equipment

A problem common to all computers is that input-output equipment usually has a rate of transfer much lower than the cycle time of the computer. This means that the computer can be limited during input-output operations.

A time sharing system will make the whole process more efficient by allowing several I/O devices active at once. This, of course, is at the cost of overhead.

More Computer Use Available

In a time shared system assuming the slower in-out device, one effectively has a number of computers equal to the number of time-shared consoles. Con-

* See appendix in Time Shared Computers, G. Bell; ARPA Report, Carnegie Institute of Technology.

sidering that an additional console is much cheaper than an additional computer, a time shared system gives both more efficient operation and increased computing capacity.

F. TS8 CONFIGURATION AND CONSOLE COST

In the case of the TS8, we assume a hardware configuration of \$50,000 - \$75,000, and / user cost of between \$1000 and \$7500 for 10 to 50 users. Now, while these costs are substantially lower than for large scale systems (\$20,000 - \$50,000 / connected console) they are at the cost threshold of a stand-alone computer (<\$10,000).

Substantially more facilities are being provided, including user files, magnetic tapes, inter-console communication, and a larger basic facility. In fact, the two facilities it does not provide are: the running of large, unorganized programs and a large capacity filing system. The approach to filing does give the user the same ability that large systems provide.

C. APPROACH TO THE PDP-8 TIME SHARING SYSTEM

1. Methods of Sharing

Many choices are available for implementing the program and memory sharing for a Time Shared System. These range from pure multiprogramming to program swapping.

The proposed method for accomplishing the project is to use both multiprogramming and program swapping. The primary memory size (3 x 4096 words) is such that it contains 2 (4096 word) programs. A disk memory is used to store files and waiting programs. While a program is running, in core, the next program is swapped.

<u>Available Capacity</u>

We assume that jobs entering the system require very little computing (i.e., can be done while swapping occurs). We take the following example for 50 users to do editing.

The machine capacity required (worst case) for managing a typewriter line is approximately 0.4%/line. -20%

The machine overhead in the monitor (per user) is estimated to be less than 0.6%. (Assumes that 30 jobs are run /second.) - - - - - - - - -30%

The system degradation if the drum is transferring continuously is (1.5 MS /word stolen from core /4.5 Msec word transfer rate). - - - -34%

The remaining available capacity for processing is: - - - - - - - -16%

<u>Required Capacity</u>

Assuming 50 user jobs are entering the system each 8 sec (1 line of text) /job, and 10 MS of computing power is required for each request, then the average computing capacity required is:

$$\frac{.01 \text{ sec of computing}}{\text{job}} \times \frac{50}{8} \frac{\text{jobs}}{\text{sec}} = \frac{.065 \text{ sec of computing}}{\text{sec}} = 6\frac{1}{2}\%$$

The computing job load is only 50 jobs /8 sec or $6\frac{1}{2}$ jobs /sec.

The above simplification assumes perfect ordering of the requests. Also, in a given 34 MS period, the swap time, only about 5.1 MS (or 34 MS x 16%) of computing power (cycles to the processor) is available. This time, or capacity, when a new job is being loaded is called the "minimum job" or 5.1 MS.

This suggests that the system will not be disc bound, but compute bound. Since the job load is far less than the disk capacity (30 jobs /sec), the capacity loss due to disk transfers will be less or approximately only $\frac{6\frac{1}{2}}{30} \times 34\%$ or 6%, leaving 44% computing capacity available instead of 16%.

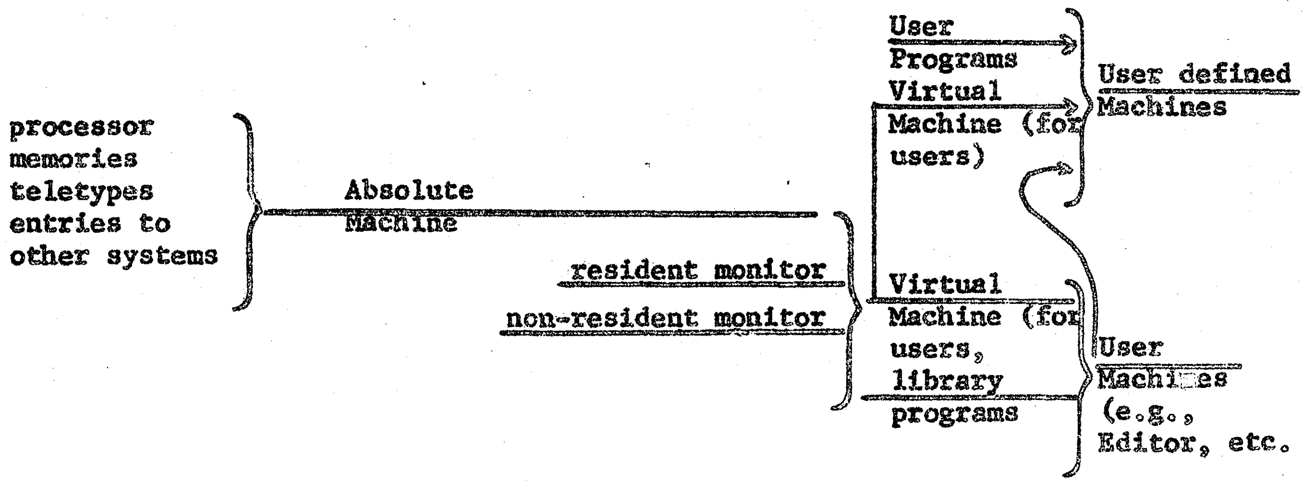
A simulation has been made for the system and is included in Appendix D.

For the simulation, assumptions about jobs were such that the average response time was less than a character time (or 100 MS) and the system response was limited by disk swap time. The simulation also considered cases when the capacity required was beyond the "minimum job".

2. Operation Under Time Sharing

Throughout the design of this system the hardware changes from a user's point of view will be a minimum such that programs which were running in the non-time-shared mode can be run in the time-shared mode (with only few modifications). On the other hand the system provides many more facilities which were not available in the absolute machine.

We define several machines according to the following figure: the Absolute Machine, the Virtual Machine, the User Machines and the User Defined Machines.



CHAPTER I

STRUCTURE OF THE HARDWARE

Figure I-1 shows the PDP-8 hardware structure. The minimum configuration has 3 x 4K of core memory, 20 teletypes, 1 disc file and 2 tape units. Except for some additional switches and special modifications to the PDP-8 processor, the hardware is standard.

The PDP-8 is a small, high speed, solid state circuit computer, and in its standard configuration has a 4K, 12 bit word 1.5 μs cycle time core memory.

A. Brief Description of the PDP-8 Processor

A block diagram of the basic PDP-8 processor taken from the PDP-8 handbook is given in Figure I-2. From this we see that it is built up out of the following blocks:

1. Accumulation (AC) *
2. Link (L) *
3. Program Counter (PC) *
4. Memory Address Register (MA)
5. Memory Buffer Register (MB)
6. Instruction Register (IR)

The additional blocks are:

1. Switch Register (SR) *
2. Major State Generator
3. Output bus drivers
4. Core Memory *

* Only accessible by program.

The core memory is divided into pages of 128 words.

Instructions are single address, one word long. The first 3 bits specify the operation and the remainder is used for addressing. In instructions which do not take an address the 9 bits are used for microprogramming. In the case of an input/output instruction they specify the I/O device (up to 64 devices) and microprogram a set of I/O command pulses.

Because the memory has a paged structure, the 9 bits for addressing are used as follows:

1. The first bit indicates indirect addressing.
2. The second bit indicates current page or page 0.
3. The remaining 7 bits are used to determine the location or line within a page.

In case of indirect addressing, the full 12 bits are used to specify a location in the 4K of core.

The core memory can be increased in units of 4K, until 32K. In this case two 3 bit registers are used to concatenate with the processor's address to form a 15 bit address. Communication among 4K memory modules is fairly difficult.

Hardware multiply and divide circuitry is not standard and is available as an option.

B. <u>Primary Memory and Communication between Primary and Secondary Memories</u>

From Figure 1-4 we see that the system has 3 x 4K of core memory. The resident part of the monitor is stored in 4K and the resulting 8K are used to contain 2 4K (or smaller) programs; one of which is being run while the other one is swapped.

A data break facility, the DM01, is a switch which allows several processors on controls to directly access the core memory. The word access interval can be as short as 1.5 μS /word.

C. Secondary Memory

Two kinds of secondary memory are used: (See Figure 1) They are:

1. Disc Memory

2. Magnetic Tape Memory

1. Disc Memory

The disc memory will be used for 2 purposes:

(1) Swapping device

(2) Storage of system files and temporary user files.

The fixed head (64 heads or tracks) disc memory is a Data Disc Inc. model F-6. The average access time is 16.6 Msec. and the transfer rate is 4.5 μsec /word (12-bit word), with 8192 words stored per track.

The capacity of the disc is 4096 blocks of 128 words or roughly, 0.5 million words.

The smallest amount of information which can be written on or read from the disc is 1 page = 128 words.

Since the disc and its controller require access to memory each 4.5 Msec, approximately 66% of the machine capacity is available while the disc is active.

2. Magnetic Tape Memory

Two tape units will be connected to the system. The 2 units are the DEC Tape Dual Transport type 555 together with the controller DEC Tape Control Type 552.

The word transfer rate of 1 12-bit word is 133 μsec; to transfer a page takes 18.2 msec.

The capacity of a standard tape is 1400 pages = 0.18 million words.

Since data on the tape is addressable and replaceable (unlike standard IBM format tape), a tape can be used for file storage, directories, etc.

D. Teletype Interface

The Teletype Interface used is the Data Communications Systems Type 680 from DEC. This system is built up out of the following units: (See Figure I-3).

1. Data Line Interface 681
2. Serial Line Multiplexer 685
3. Matricon patch panel 684
4. Teletype Connector panel 682
5. Telegraph level convertor 683.

The 680 allows up to 128 teletypes to be connected to the PDP-8. The 681 controls and executes transmission and reception of teletype information between the computer and the type 680 system. In order to do this it adds 2 instructions to the instruction set and takes 274 +7N core locations of program (N = number of teletypes connected) assuming 8 bit teletype lines are used.

The 685 is simply a switch which allows the 681 to be connected with any of the 64 teletypes.

For more information see "Small Computer Handbook."

E. Modifications to the Basic Processor

1. Addition of memory extension control (standard option)
2. Data Line Interface (standard option)
3. Trapping Logic (special modifications)

1. <u>Memory Extension Control</u>

This is the hardware necessary to be able to specify any of 8 possible blocks of 4K. A group of registers together with a set of instructions is added to the hardware.

2. <u>Data Line Interface</u>

See part D of current chapter.

3. <u>Trapping Logic</u>

Hardware will be added which provides for 2 states of running programs: monitor and user mode.

In user mode I/O instructions and the halt instruction will be trapped, and control will be transferred to the monitor.

A complete description of the required changes is not available yet.

CHAPTER II

STRUCTURE OF THE VIRTUAL MACHINE

This is the structure of the machine as the user sees it. Figure II-1 is a simplified model of the virtual machine. The different aspects of this are covered under the following headings:

A. Program Environment

B. Files

C. Console Control of the Virtual Machine.

A. PROGRAM ENVIRONMENT

Programs run on the time-shared system are almost identical with those running on the standard PDP-8. The time-shared system will have more possible instructions.

The size of a running program is $\leq$ 4K of core memory.

By means of program instructions the user is able to transfer to and from core files or parts of files. This enables a user to run programs $>$ 4K words.

B. FILES

The way a file looks like to a user is one consecutive block of the size of his file. The maximum size of a file is 64K words. The minimum addressable quantity of information in a file is 1 page = 128 words.

The maximum number of files a user can have at a time is 8. This is because of the limited disc space on which the files are stored.

C. CONSOLE CONTROL OF THE VIRTUAL MACHINE

All control of the virtual machine is done through the monitor. Monitor commands are initiated both from the console and the running program. A listing of both groups of commands is given in Appendix A.

Before entering a monitor command the state of the machine has to be switched from user to monitor mode; this is done by typing a control character. Upon completion of the monitor instruction the state of the machine will return to the user mode.

CHAPTER III

STRUCTURE OF SYSTEM PROVIDED FACILITIES SEEN FROM VIRTUAL MACHINE

A. CONVERSION OF DEC SOFTWARE

The greatest changes to existing software foreseen in this study will not be those necessitated by the time sharing facility but rather those influenced by the extended size of the virtual machine to 64K and the addition of high speed disk file accessing. All the standard software systems and subsystems will more naturally reside on the disks, be called by console monitor commands and initial loaded by the disk file handling routines for system files.

1. Editor

The current text buffer size in core which limits the size of text upon which the Symbolic Editor may function will be extended in the virtual machine. The Editor may be given 2 files containing up to 64K of text instead of the actual text being passed via the paper tape reader, and paper tape punch. It will be the responsibility of the Symbolic Editor to access the "text page" that it needs, via monitor subroutine calls. Furthermore, if the Editor requires increased work or data space, it will be required to do its own "paging" of information into core as described previously.

The I/O functions must be capable of handling the option of either paper tape or disk source files, although this function will be part of a separate file to paper tape copying facility.

2. Desk Calculator

The Desk Calculator function will be expected to operate in much the same manner as it does currently, perhaps with an extension to the repertoire of mathematical function.

3. Assemblers: Macro and PAL

The users of Macro and PAL will be categorized as system programmers. As such, they will be expected to know the dynamic state of their programs and will, therefore, be responsible for performing the functions of "page turning" when desired. The assemblers should provide macros for calling these and the other monitor subroutines.

4. DDT Debugging System

Only minor alterations are foreseen here. Expansion of table sizes will be made practical by the virtual 64K memory and the "page turning" concepts. DDT and the DDT symbol table should be part of a virtual file, and as such, invisible to a user program.

5. Loader

The loader must perform the functions presently accomplished manually of loading into core the primary executable module of each user's program. The loader will be controlled by the monitor which will pass to it the parameters indicating the file to be loaded and the option switches entered via the monitor console commands. Eventually, the loader will also have the function of link editing the non-monitor subroutines called by other languages. Each language processor requiring a link edit phase will use the same loader. Interpreters may require dynamic loading of large subroutines. Compiled programs usually function best with subroutines resident with the calling module when space permits. The loader should be capable of handling both functions.

6. FORTRAN Compiler and the FORTRAN Operating System

Most of the modifications to FORTRAN will be extensions rather than alterations. Initially only minor modifications would be made which allow input and output to be both named files. The compiler must be designed to handle larger programs than limitation now permit, up to a 64K program. The compiler does not produce machine language object code, but rather input to an interpreter under the FORTRAN Operating System. The interpreter will be burdened with the table of "paging" program segments based upon information passed to it in the form of tables by the FORTRAN compiler. During execution of programs the interpreter must check various compiler output tables in order to perform proper addressing ing and checking. Many of these checks may be turned off in a properly debugged program. This no-check option is particularly important in the functioning of this interpreter.

The FORTRAN Operating System will now be required to automatically perform the functions of compile, load and execute by creating and reading from temporary data files it establishes on disc instead of doing its intermediate I/O through paper tape. The load-link edit step should be accomplished by a call to the Loader from the FORTRAN Operating System.

B. SOFTWARE TO OPERATE IN CONJUNCTION WITH COMPUTER NETWORKS

1. Job Flow to Processors

All jobs passing through the PDP-8 will be assigned to a particular processor by being placed in the appropriate queue. The program to accomplish this function may have the same priority as any other user program sharing the PDP-8 control processor.

If a central data file is established for all the processors, the particular CPU assigned to accomplish the current task will be given only a pointer in its queue to the job. The particular CPU may then use either its own intermediate storage devices or those in the shared bank. All buffered output resulting from any CPU execution must be assigned back to the central data file where it will be added to the PDP-8 queue of jobs to be output onto peripheral I/O devices. The output functions will again be accomplished by a user-priority PDP-8 program which will refer to the monitor where its job queue will be stored. Storing this queue in the PDP-8 monitor will eliminate the need for further inter-program communication.

2. Job Sharing

Many jobs consist of several job steps, some of which require high speed CPU's, whereas others cannot justify the high overhead costs of the same processor. For example, the job may first be assigned to the PDP-8 for editing purposes, then to the UNIVAC 1108 for compilation and execution phases. Similarly, a PDP-8 assembly language program may first be assigned to the IBM/360 for a high speed assembly and then back to the PDP-8 for execution. When jobs are so shared, status words and condition codes must be appended to the jobs and passed between steps. Condition codes may designate the severity of errors encountered in a job step to be optionally checked before execution of the successive step.

(Implementation of a PDP-8 assembly language assembler on a larger machine, as used in the example above, is being planned for early implementation necessitated by the absence of a line printer on the current PDP-8 configuration.)

CHAPTER IV
STRUCTURE OF THE MONITOR

The total amount of core reserved for the resident monitor is 4096 words. This will not be enough to contain all the programs. Certain monitor programs have to be stored on the disc and be run like a user's program. The monitor programs which will reside in core permanently are those which will be used very often and/or are necessary directly to keep the system running.

A. MAP OF MEMORY

The monitor programs can be divided into 2 groups:

1. The resident programs and files
2. The non-resident programs and files.

1. Resident Programs and Files

(1) Interrupt identifier routines

(2) Scheduler routines

(3) Clock service routines

(4) Trap service routines

(5) Disc service routines

(6) I/O routines and I/O buffers

(7) Teletype handling and line editing

1.1 - 1.4 No software is written yet;that is why no numbers for required core are available.

1.5 Disc Service Routines. These routines control the swapping and handle the file access mechanism. In order to prevent too much swapping to be able to access information from the files, the monitor has resident a block of 8 words per open file. These 8 words will contain enough current history to make most of the file accesses direct without having to search through the whole file structure as described under part B of this chapter.

The information an 8 bit block contains is described in the Appendix B, "Current File Information".

Assuming that on the average there are 3 files open per user, then given there are 20 users, we need $20 \times 3 \times 8 = 60 \times 8 = 460$ words = 3.8 pages.

1.6 I/O Routines. This program takes care of the input, output and buffer service for the teletypes and the high speed paper tape reader punch.

It includes the routines of the 680 teletype interface; area for the I/O buffer and 2 status words for every I/O device.

Storage is required for:

680 teletype interface	$274 \times 7 \times 20 = 294$ words =	2.3 pages
I/O buffer		5.0 pages
Status words	$22 \times 2 = 44$ words =	0.35 pages
	total	7.65 pages

The above total does not include a program to pack and unpack the status words and the characters in the buffer and to assign and release buffers.

How the I/O buffers and the status words look like is described under "Input Output Buffer" in Appendix C.

1.7. Teletype Handling and Line Editing. The monitor has to be able to do very simple editing functions:

(1) Deleting a character

(2) Deleting a whole line.

(3) Recognize a carriage return

(4) Recognize an input termination character

(5) Recognize a character to move into monitor-console communication state.

Point 4 above is added because certain programs allow a user to type in some input string after which the system responds by typing the answer on the same line. Like DDT, for instance, allows a user to type in the name of a variable after which the system prints on the same line the current value of that variable.

The amount of core needed for buffers and the 680 program can be computed now.

I/O routines with buffers	7.65 pages
Current file buffers (sec. 1.5)	3.80 pages
Total	11.45 pages

A special page in the monitor has to be reserved for bringing in the page containing the user's directory, for making changes and copy certain parts in the file blocks. This area of 1 page can be used also to bring in the page with pointers in case of indirect. (See part B of this chapter.)

The area in the 4K free for the programs mentioned under A.1 is 32 - 12.45 = 19.55 pages.

Of the 19.55 pages, 1 page will be needed to contain a bit table for disc memory allocation.

2. <u>Non Resident Programs and Files</u>

These are:

(1) Syntax Analyzer

(2) Tape handling routines

(3) Error message file

(4) Main Directory

2.1 The Syntax Analyzer. This program analyzes the instructions given to the monitor. Together with the program, a list of all monitor instructions is stored. Appendix A, "Monitor Instructions", gives a listing of these instructions.

2.2 Tape Handling Routines. These include a set of routines to handle the type 555 DEC tape unit and a program to transfer from files on the disc to the tape files and vice versa.

2.3 Error Message File. This is a collection of error messages due to errors caused by a wrong use of monitor instructions.

2.4 Main Directory. A search program together with the main directory are stored in one file. The main directory contains all the legal user-numbers.

B. THE FILE STRUCTURE

With the File Structure is meant the way of access and storing of information on the disc file. Figure IV-1 shows the different levels of the file structure together with the 4 sub divisions. These are:

1. Non resident monitor file(s)
2. System directory (binary)
3. System directory (text)
4. Main directory

As discussed under part C of Chapter I, the smallest addressable quantity is 1 page.

1. Non Resident Monitor File(s)

In the resident monitor pointer(s) will be stored pointing directly to the non resident monitor file(s). Together with these pointers, information about the number of pages of the file(s) will be stored.

2. System Directory (Binary)

This directory will consist of a list of 4 word blocks where 1 block is used for every file in the directory. Figure IV-2 shows how the 4 words can be used.

3. System Directory (Text)

This directory will be identical to the System Directory (binary).

4. Main Directory

In this directory are stored all the legal user-numbers, together with a pointer to the corresponding User's Directory and some additional information. See Figure IV-3.

4.1 User Directory. All the files a user can have on the disc are contained in this directory.

Per file a block of 16 words is used. See Figure IV-4. The 4 pointers from Figure IV-4 can be used as direct or indirect pointers. This is determined by the control word. See Figure IV-5. The maximum size of a file to which can be pointed to directly is 4 x 4 = 16K. When the indirect pointer is used the maximum size of a file can be 64 x 8 pages = 64K.

Given the fact that we have 1 page for the User Directory and that it takes a block of 16 words per file, 8 files can be on the disc per user-number.

Figure IV-6 gives a block structure of the file system for user files.

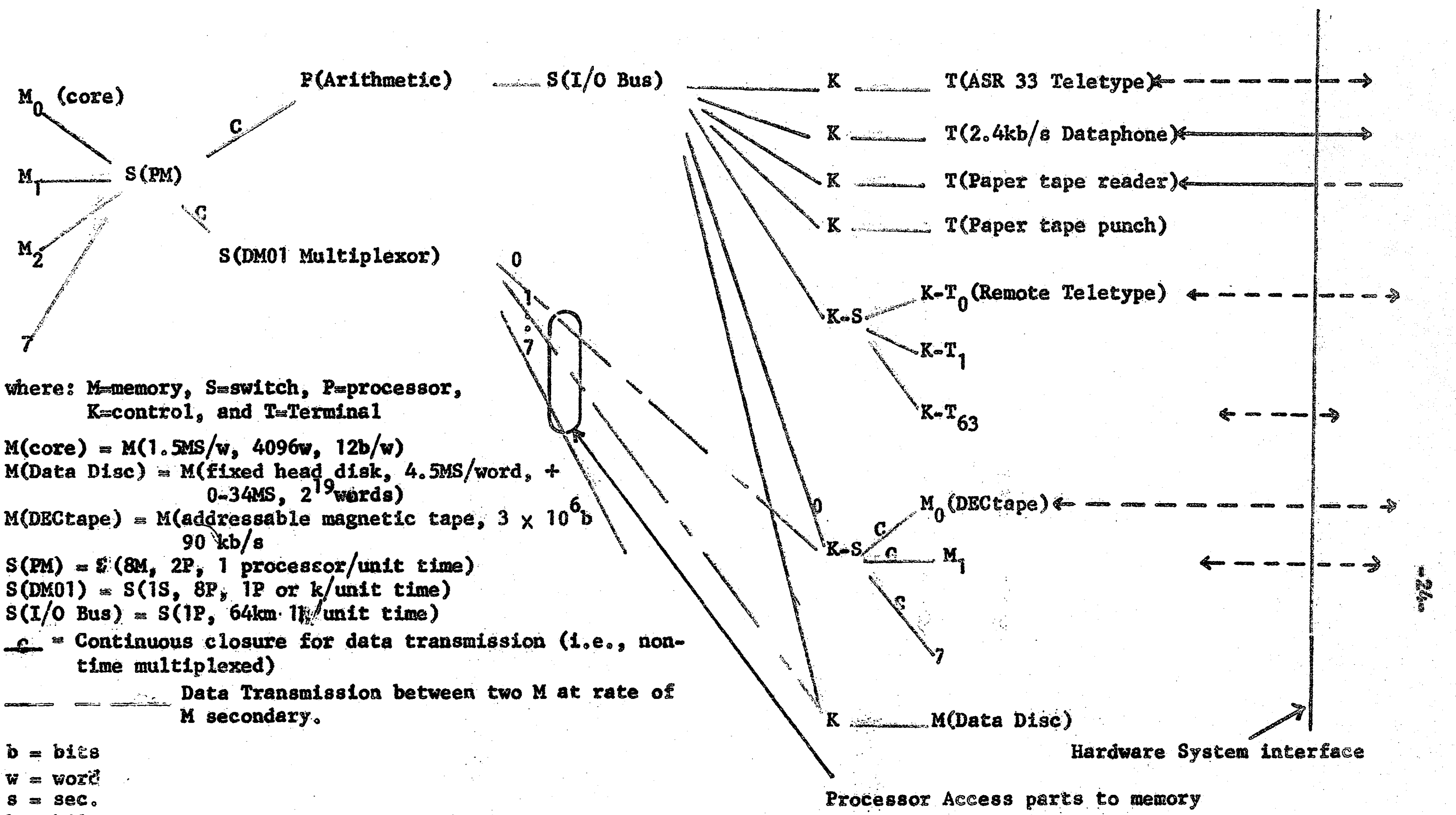
CHAPTER V

IMPLEMENTATION PLAN

The design of this machine is such that the system can be used as a stand alone system or in conjunction with a bigger machine.

The stand alone system will be the first stage in the design. By January, 1968 the monitor is planned to be written, together with changes in the existing software package. In parallel with this, the required hardware modifications will be done.

After the system as described in this paper will be realized, at some future time modifications will be made to use it in conjunction with a bigger system.



where: M=memory, S=switch, P=processor, K=control, and T=Terminal

M(core) = M(1.5MS/w, 4096w, 12b/w)
M(Data Disc) = M(fixed head disk, 4.5MS/word, + 0-34MS, 2^{19} words)
M(DECtape) = M(addressable magnetic tape, 3×10^6 b 90 kb/s
S(PM) = S(8M, 2P, 1 processor/unit time)
S(DM01) = S(1S, 8P, 1P or k/unit time)
S(I/O Bus) = S(1P, 64km 1k/unit time)
$\underline{c}$ = Continuous closure for data transmission (i.e., non-time multiplexed)
—— — —— Data Transmission between two M at rate of M secondary.

b = bits
w = word
s = sec.
k = kilo

FIGURE I-1. PDP-8 TS8 CONFIGURATION

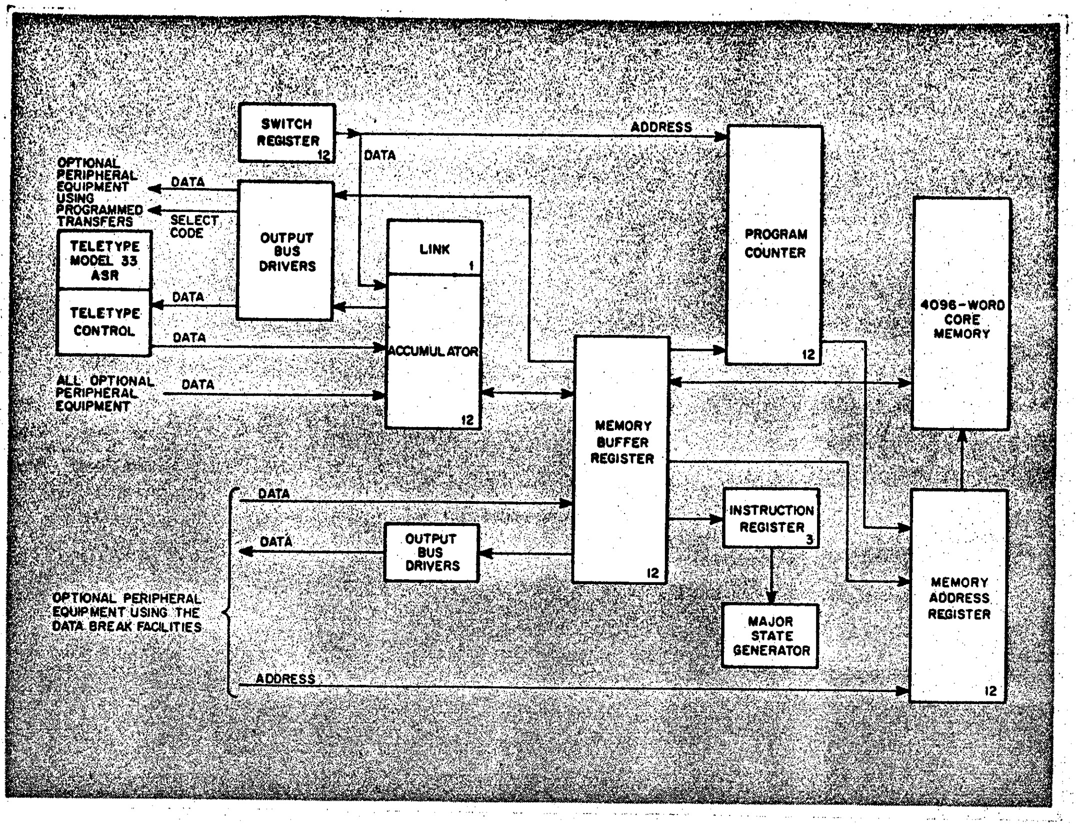


FIGURE I-2. PDP-8 MAJOR REGISTER BLOCK DIAGRAM

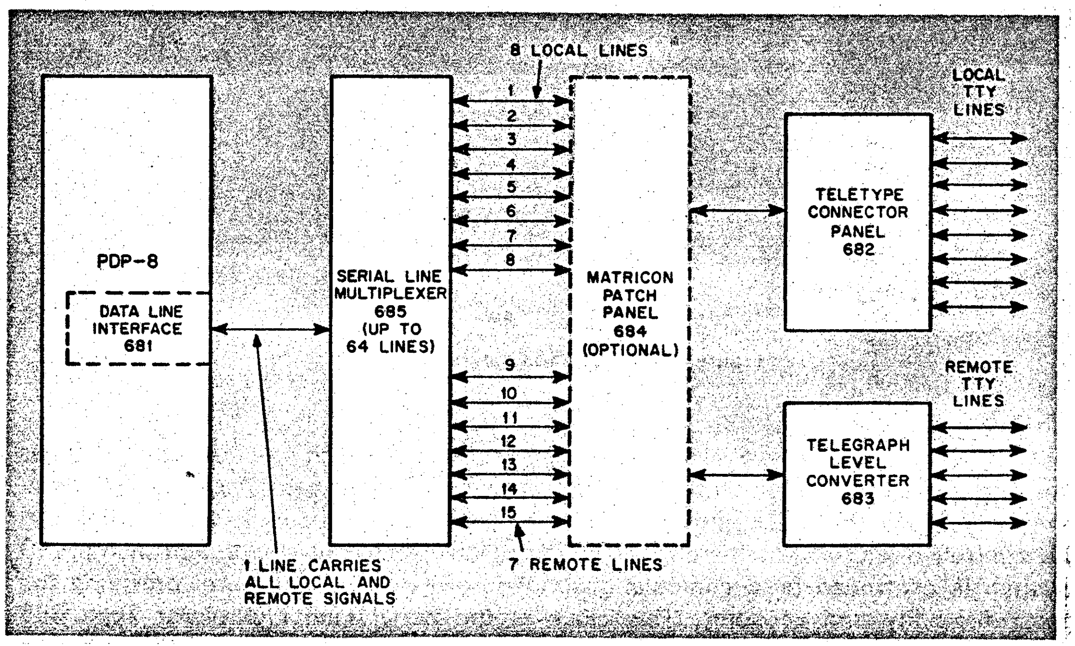

8 LOCAL LINES
LOCAL TTY LINES
PDP-8
DATA LINE INTERFACE 681
SERIAL LINE MULTIPLEXER 685 (UP TO 64 LINES)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
MATRICON PATCH PANEL 684 (OPTIONAL)
TELETYPE CONNECTOR PANEL 682
REMOTE TTY LINES
TELEGRAPH LEVEL CONVERTER 683
1 LINE CARRIES ALL LOCAL AND REMOTE SIGNALS
7 REMOTE LINES

FIGURE I-3. DATA COMMUNICATION SYSTEM BLOCK DIAGRAM

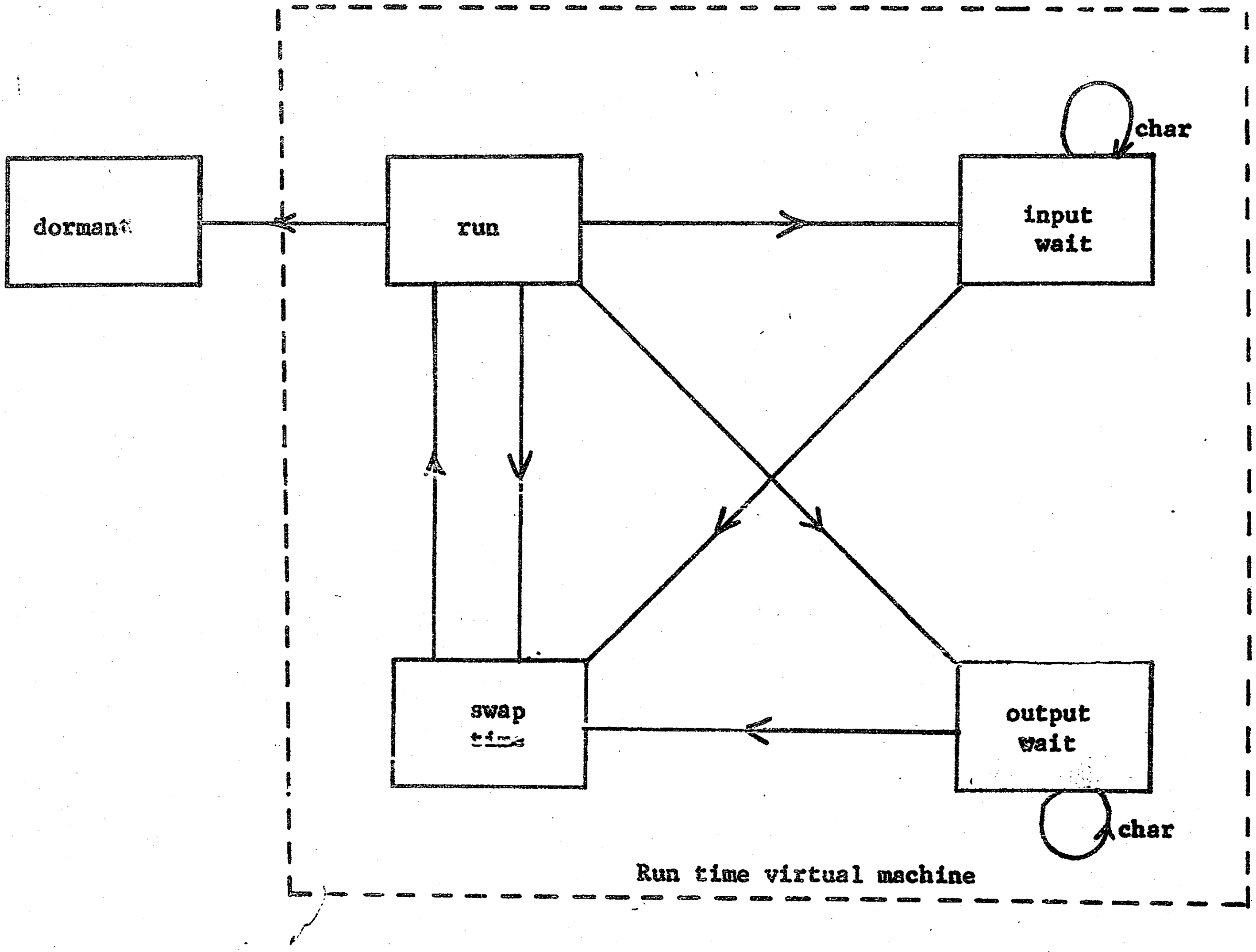


FIGURE II-1. STATE DIAGRAM OF VIRTUAL MACHINE

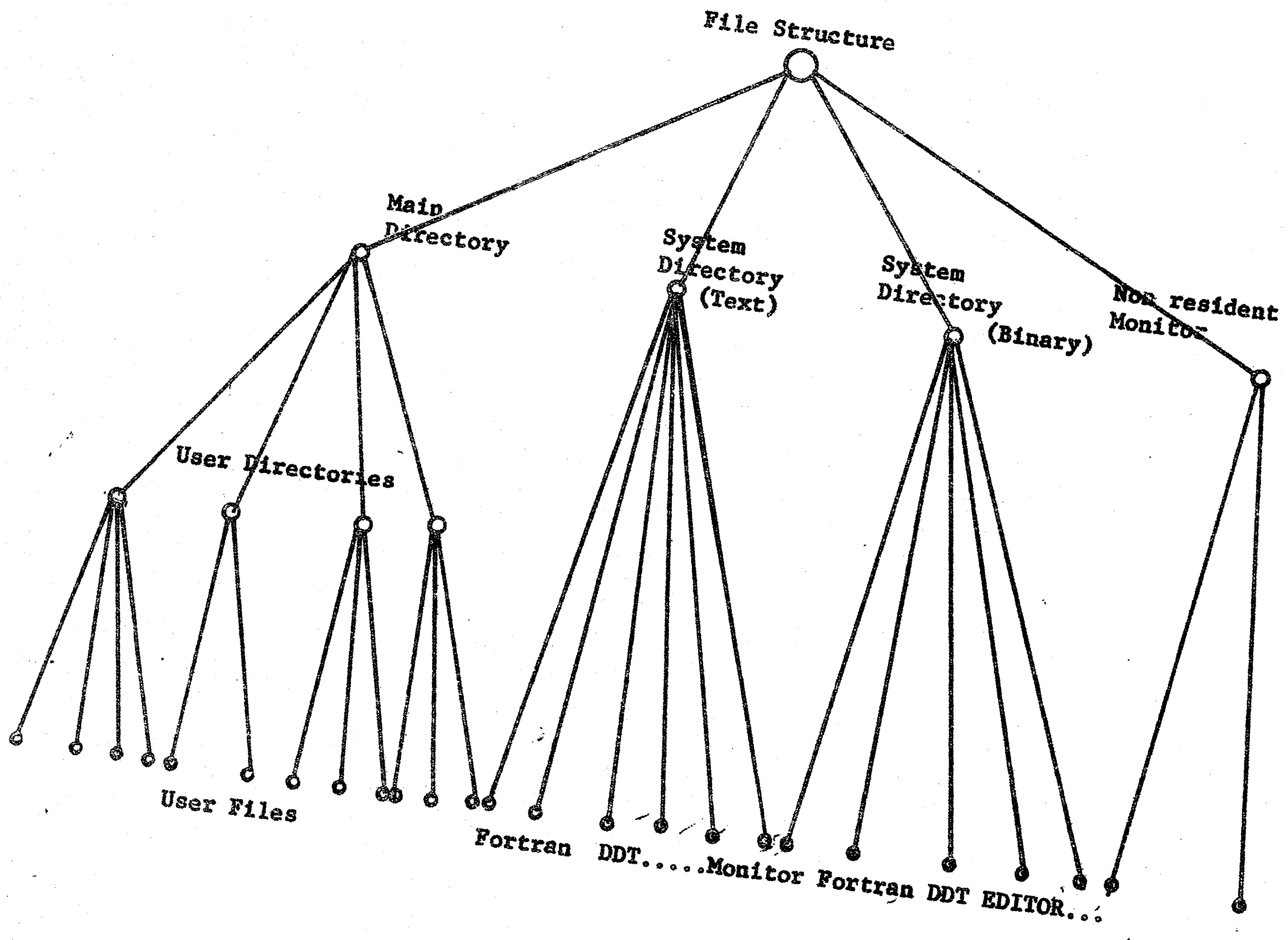


FIGURE IV-1. DISC FILE STRUCTURE

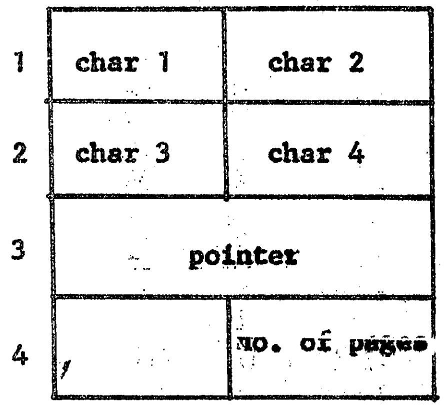


FIGURE IV-2

The first 2 words are used to store 4 characters, this limiting the name of a file to be 4 characters long at most. The 3rd word is a pointer which points to the first page of the file on the disc. The 4th word contains a counter to indicate how many pages the file is; the rest of the word is not used.

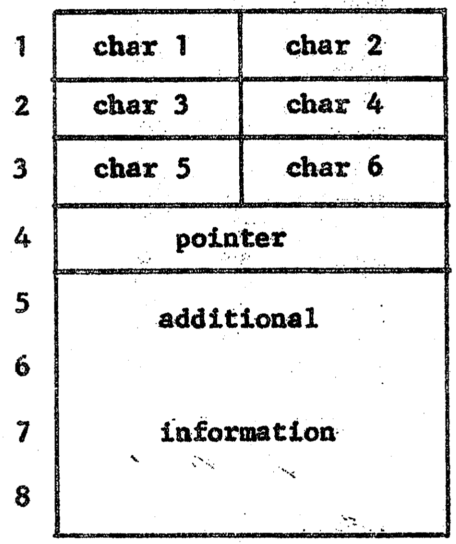


FIGURE IV-3

User-numbers are allowed to consist of up to 6 characters. The words 1-3 are used to store the user number. Word 4 contains a pointer pointing to one page containing the User's Directory. Words 5-8 are used for additional information storage, like: billing, etc.

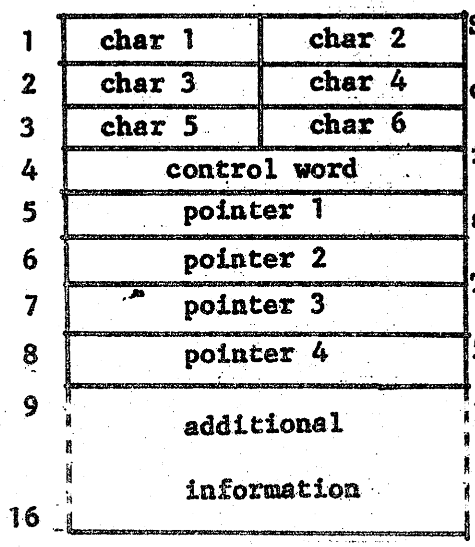


FIGURE IV-4

Similar to user-numbers, file names are allowed to consist of up to six characters. This is done also in order to be compatible with the DEC Tape file structure. The file name is stored in the words 1-3. The 4th word contains information about the 4 pointers 5-8. In the words 9-16 can be stored information like: read only, secret, etc.

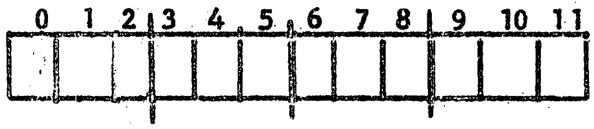


FIGURE IV-5

The control word 4 from the preceding figure is divided into 4 3-bit words, a 3-bit word for every pointer. The 3-bit word tells whether the pointer is used and if so, whether the corresponding pointer points direct or indirect, and in case of direct, to what size of a block it points.

0	0	0	pointer not used
0	0	1	points direct to 1 page
0	1	0	points direct to 8 pages
0	1	1	points direct to 32 pages
1	0	0	points indirect

When the pointer points indirect it always points to a 1 page block. Half of the words in that page can be used. When any of the words 1 through 64 are non-zero they are pointers which point directly to a block of 8 pages .

Only the 1st pointer of the 4 indicated in Figure IV-5 can point indirect. The 3 other pointers are not used then.

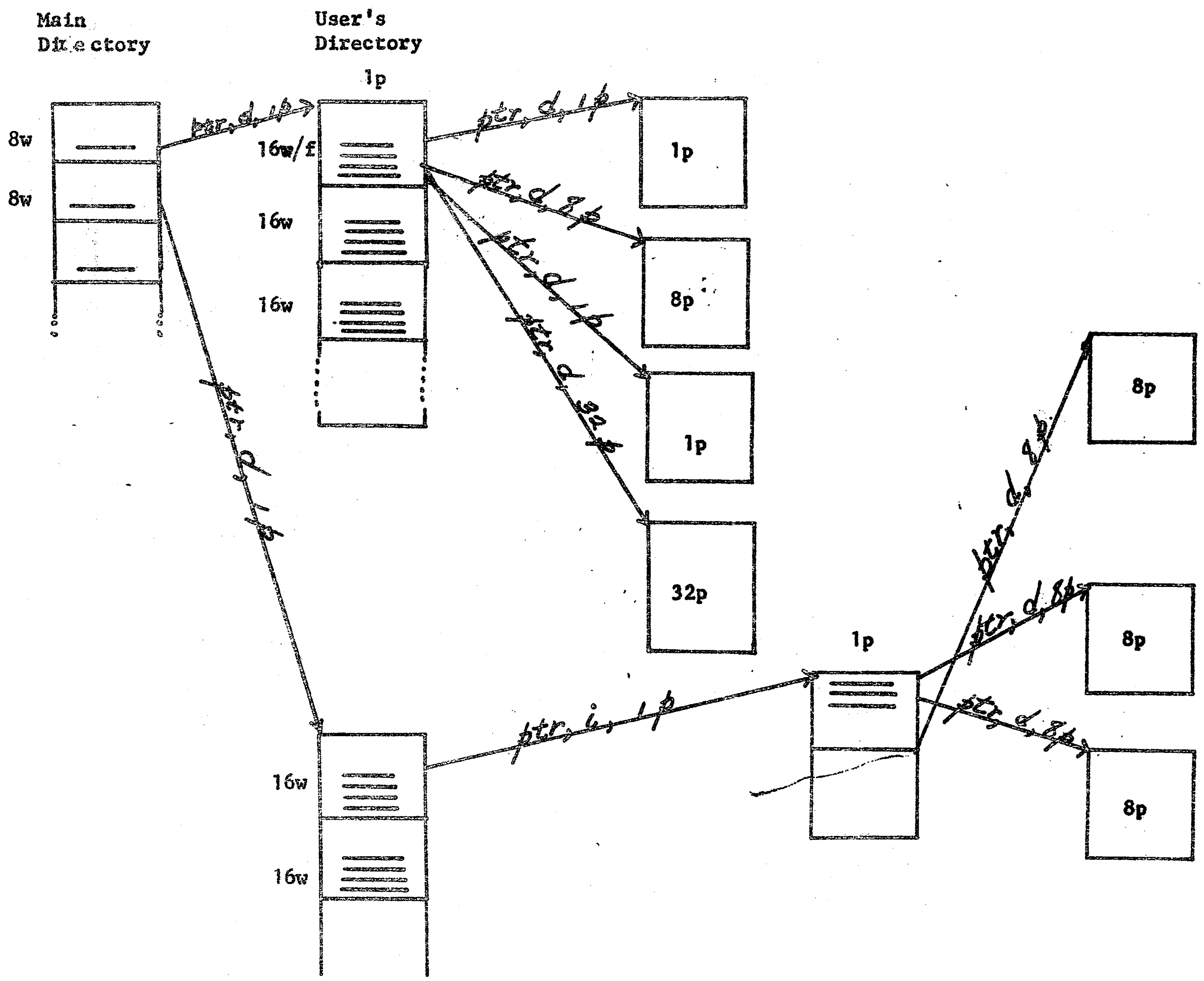


FIGURE IV-6 USER'S FILE STRUCTURE

APPENDIX A

MONITOR INSTRUCTIONS

When a C occurs in the right column the command can be issued from the console. When a P occurs the command can be issued by the program. In the list the commands are written with a full name, in the implementation mnemonics will be used however.

1. Logout		P,C
2. List	lists user's filenames	C
3. Open A	make file A accessable	P,C
4. Close A	make file A inaccessable	P,C
5. Delete A	deletes file A	P,C
6. Create A	creates new file A	P,C
7. Rename A,B	call file A, B	P,C
8. Load A, B, C, D	load pages A to A + B -1 of disc file C into core starting at page D	P,C
9. Dump A, B, C, D	load pages A to A + B -1 of core onto disc file C starting at page D	P,C
10. Load tape A,B,C,D	like Load only from tape	P,C
11. Dump tape A,B,C,D	like Dump only onto tape	P,C
12. Start A	start executing at core location A	P,C
13. Stop at A	stop executing at core location A	C
14. Stop	stop executing	C
15. Continue	continue processing	C
16. Macro A,B	apply macro assembler to file A put results into file B	P,C
17. Pal A,B	like macro	P,C
18. Fortran comp A,B	compile file A, result in B	P,C

19. Fortran exec. A,B		P,C
20. DDT A,B		P,C
21. Editor A,B		P,C
22. Change mode	change from user to monitor mode	C
23. Append A,B	concatenate file B to A	P,C
24. CPU	print Status of machine	C
25. TIME	print time	P,C
26. Users	print number of users	C
27. Assign A	assign device A	P,C
28. Release A	release device A	P,C
29. Set SWR	set switch register	P,C
30. Secret A	declare file A secret	C
31. Read only	declare file A read only	P,C
32. Read only others	declare file A read only for others	C
33. Free	make file free	P,C
34. Save	saves current machine state	C
35. Print A	print on teletype A	P
36. Read A	read from keyboard A	P
37. Read high speed	read from high speed reader	P
38. Punch	output on high speed punch	P

APPENDIX B

CURRENT FILE INFORMATION

As described in Chapter IV, part A under Disc Service Routines per open file a block of 8 words is stored in the monitor. The layout of this block is shown in Figure B-1.

1	Information
2	about file
3	Control word
4	Pointer 1
5	Pointer 2
6	Pointer 3
7	Pointer 4
8	Pointer to next block

FIGURE B-1

The first two words contain information about the file, like: file number, read only, secret, etc.

Word 8 contains (if not = 0) a pointer to the next block.

Words 3-7 are a copy of the corresponding words from the User's file directory. In case of direct all the information about the file is available in the monitor. In case of indirect the first pointer is used to point at the page the (at maximum 64) pointer are stored, and the 3 remaining pointers can be used to point at three 8 page file blocks. These 3 pointers are just a copy of 3 of the 64 possible pointers from the page the first pointer points at.

APPENDIX C

INPUT OUTPUT BUFFER

Because of the very limited amount of core memory available for the monitor programs only 5 pages were reserved for the I/O buffer.

The buffer area is divided into blocks of 8 words. Using packing techniques every block will be able to store 10 characters. See Figure C-1.

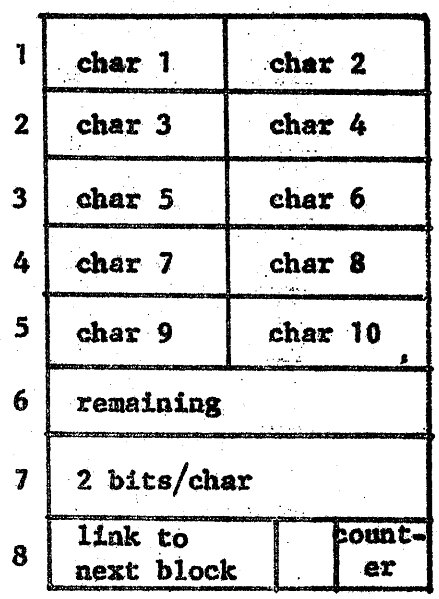


FIGURE C-1

The first 6 bits of every character are stored in the first 5 words. The remaining 2 bits/char. are stored in word 6 and 7.

The 8th word has a counter (4 bits) to keep track of the number of stored characters and a pointer to the next block.

The total number of 8 bit blocks is 5 x 128/8 = 820. So the pointer in Figure 1 has to be 7 bits long. A full input line consists out of 72 characters which means that the maximum number of blocks for input will be 8. For output we will restrict this to 4.

In case of the high speed reader (300 char/sec.) we will restrict the maximum number of blocks to 6 and 15, respectively.

From I/O buffer point of view a teletype is considered as 1 device having only 1 buffer which serves as either input or output buffer.

Somewhere in the monitor information concerning the teletypes and the high speed paper tape reader/punch and other possible I/O devices has to be stored.

The information is:

1. Usage of the buffer for input or output
2. Number of blocks the buffer consists of
3. Pointer to the first block of the buffer
4. Pointer to the last block of the buffer
5. The user the I/O device is assigned to.

All the information above can be contained in 2 words per device. See Figure C-2.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

I/O	block counter	pointer to 1st block
assignment		pointer to last block

FIGURE C-2

A block structure of the I/O buffering is shown in Figure C-3.

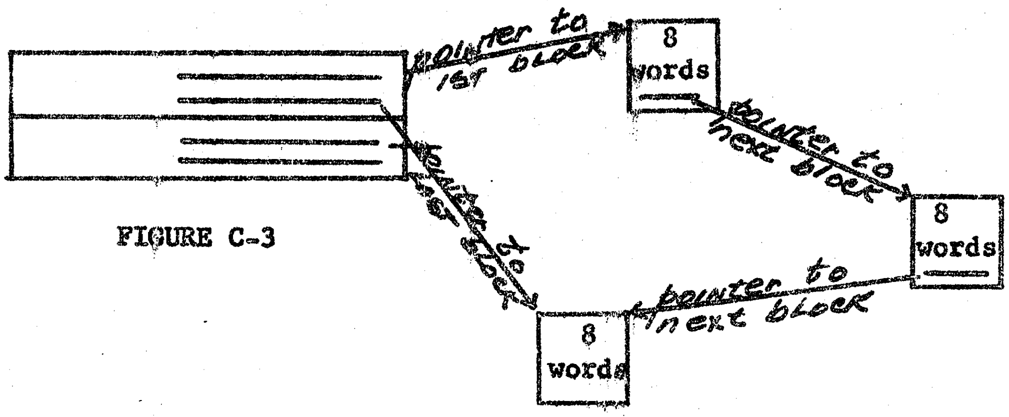


FIGURE C-3

APPENDIX D

The results of a simulation of groups of people editing is shown in the next pages.

The assumptions made were:

1. The time it takes to typewirte a line is distributed Normal with expected value = 8 sec and $\sigma = 2$. The tails < 1 sec and > 60 sec were cut off.
2. The computation time to process a line has a Normal distribution with expected value = 20 msec and $\sigma = 10$ msec. The tails <10 msec and > 60 msec were cut off.

The printout shows the array wait, which is the time a user has to wait after entering a full line to be able to enter the next line. The computation time, swapping time and overhead are included in this waiting time.

The distribution function and the cumulative distribution function of the waiting time are included in this appendix.

```
NUMBER OF USERS= 10      STEP SIZE= 5MSEC      AVERAGE WAITINGTIME=  46.46
WAIT[ 0]=   0.0      WAIT[ 1]=   0.0      WAIT[ 2]=   0.0      WAIT[ 3]=   0.0      WAIT[ 4]=   0.0      WAIT[ 5]=   0.0
WAIT[ 6]=   0.0      WAIT[ 7]=   0.0      WAIT[ 8]=  14.89     WAIT[ 9]=  43.86     WAIT[10]=  75.32     WAIT[11]=  96.3
WAIT[12]=  97.5      WAIT[13]=  97.7      WAIT[14]=  98.9      WAIT[15]=  99.0      WAIT[16]=  99.1      WAIT[17]=  99.3
WAIT[18]=  99.8      WAIT[19]=100.0       WAIT[20]=100.0       WAIT[21]=100.0       WAIT[22]=100.0       WAIT[23]=100.0
WAIT[24]=100.0       WAIT[25]=100.0       WAIT[26]=100.0       WAIT[27]=100.0       WAIT[28]=100.0       WAIT[29]=100.0
WAIT[30]=100.0       WAIT[31]=100.0       WAIT[32]=100.0       WAIT[33]=100.0       WAIT[34]=100.0       WAIT[35]=100.0
WAIT[36]=100.0       WAIT[37]=100.0       WAIT[38]=100.0       WAIT[39]=100.0       WAIT[40]=100.0       WAIT[41]=100.0


NUMBER OF USERS= 20      STEP SIZE= 5MSEC      AVERAGE WAITINGTIME=  48.42
WAIT[ 0]=   0.0      WAIT[ 1]=   0.0      WAIT[ 2]=   0.0      WAIT[ 3]=   0.0      WAIT[ 4]=   0.0      WAIT[ 5]=   0.0
WAIT[ 6]=   0.0      WAIT[ 7]=   0.0      WAIT[ 8]=  14.09     WAIT[ 9]=  37.16     WAIT[10]=  66.53     WAIT[11]=  89.71
WAIT[12]=  92.51     WAIT[13]=  94.21     WAIT[14]=  95.1      WAIT[15]=  96.7      WAIT[16]=  97.6      WAIT[17]=  98.7
WAIT[18]=  99.5      WAIT[19]=  99.7      WAIT[20]=  99.8      WAIT[21]=  99.9      WAIT[22]=  99.9      WAIT[23]=  99.9
WAIT[24]=100.0       WAIT[25]=100.0       WAIT[26]=100.0       WAIT[27]=100.0       WAIT[28]=100.0       WAIT[29]=100.0
WAIT[30]=100.0       WAIT[31]=100.0       WAIT[32]=100.0       WAIT[33]=100.0       WAIT[34]=100.0       WAIT[35]=100.0
WAIT[36]=100.0       WAIT[37]=100.0       WAIT[38]=100.0       WAIT[39]=100.0       WAIT[40]=100.0       WAIT[41]=100.0


NUMBER OF USERS= 30      STEP SIZE= 5MSEC      AVERAGE WAITINGTIME=  49.41
WAIT[ 0]=   0.0      WAIT[ 1]=   0.0      WAIT[ 2]=   0.0      WAIT[ 3]=   0.0      WAIT[ 4]=   0.0      WAIT[ 5]=   0.0
WAIT[ 6]=   0.0      WAIT[ 7]=   0.0      WAIT[ 8]=  13.49     WAIT[ 9]=  38.76     WAIT[10]=  63.84     WAIT[11]=  87.21
WAIT[12]=  89.31     WAIT[13]=  91.31     WAIT[14]=  93.71     WAIT[15]=  95.0      WAIT[16]=  96.3      WAIT[17]=  97.5
WAIT[18]=  98.4      WAIT[19]=  98.8      WAIT[20]=  99.5      WAIT[21]=  99.7      WAIT[22]=  99.7      WAIT[23]=  99.7
WAIT[24]=  99.8      WAIT[25]=  99.9      WAIT[26]=100.0       WAIT[27]=100.0       WAIT[28]=100.0       WAIT[29]=100.0
WAIT[30]=100.0       WAIT[31]=100.0       WAIT[32]=100.0       WAIT[33]=100.0       WAIT[34]=100.0       WAIT[35]=100.0
WAIT[36]=100.0       WAIT[37]=100.0       WAIT[38]=100.0       WAIT[39]=100.0       WAIT[40]=100.0       WAIT[41]=100.0


NUMBER OF USERS= 40      STEP SIZE= 5MSEC      AVERAGE WAITINGTIME=  51.13
WAIT[ 0]=   0.0      WAIT[ 1]=   0.0      WAIT[ 2]=   0.0      WAIT[ 3]=   0.0      WAIT[ 4]=   0.0      WAIT[ 5]=   0.0
WAIT[ 6]=   0.0      WAIT[ 7]=   0.0      WAIT[ 8]=  14.09     WAIT[ 9]=  37.16     WAIT[10]=  61.64     WAIT[11]=  81.72
WAIT[12]=  84.42     WAIT[13]=  86.91     WAIT[14]=  89.71     WAIT[15]=  92.11     WAIT[16]=  94.21     WAIT[17]=  96.0
WAIT[18]=  97.1      WAIT[19]=  97.9      WAIT[20]=  98.1      WAIT[21]=  98.7      WAIT[22]=  99.2      WAIT[23]=  99.4
WAIT[24]=  99.6      WAIT[25]=  99.9      WAIT[26]=100.0       WAIT[27]=100.0       WAIT[28]=100.0       WAIT[29]=100.0
WAIT[30]=100.0       WAIT[31]=100.0       WAIT[32]=100.0       WAIT[33]=100.0       WAIT[34]=100.0       WAIT[35]=100.0
WAIT[36]=100.0       WAIT[37]=100.0       WAIT[38]=100.0       WAIT[39]=100.0       WAIT[40]=100.0       WAIT[41]=100.0


NUMBER OF USERS= 50      STEP SIZE= 5MSEC      AVERAGE WAITINGTIME=  53.17
WAIT[ 0]=   0.0      WAIT[ 1]=   0.0      WAIT[ 2]=   0.0      WAIT[ 3]=   0.0      WAIT[ 4]=   0.0      WAIT[ 5]=   0.0
WAIT[ 6]=   0.0      WAIT[ 7]=   0.0      WAIT[ 8]=  12.39     WAIT[ 9]=  35.26     WAIT[10]=  57.84     WAIT[11]=  76.32
WAIT[12]=  79.02     WAIT[13]=  82.82     WAIT[14]=  85.91     WAIT[15]=  89.01     WAIT[16]=  92.01     WAIT[17]=  94.41
WAIT[18]=  96.6      WAIT[19]=  97.0      WAIT[20]=  97.8      WAIT[21]=  98.2      WAIT[22]=  98.3      WAIT[23]=  98.7
WAIT[24]=  98.9      WAIT[25]=  99.1      WAIT[26]=  99.4      WAIT[27]=  99.6      WAIT[28]=  99.7      WAIT[29]=  99.7
WAIT[30]=  99.7      WAIT[31]=  99.8      WAIT[32]=  99.8      WAIT[33]=  99.8      WAIT[34]=  99.9      WAIT[35]=  99.9
WAIT[36]=  99.9      WAIT[37]=  99.9      WAIT[38]=  99.9      WAIT[39]=  99.9      WAIT[40]=100.0       WAIT[41]=100.0
```

```
MODE 3,    LIMITS: MIN  0.00000 .+00, MAX  3.14685 .+01,    INT/COL 3.14685 .-01,          PAGE  1 OF GRAPH 75

   000       010       020       030       040       050       060       070       080       090       100
000 5---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
001 5         |         |         |         |         |         |         |         |         |         |
002 5         |         |         |         |         |         |         |         |         |         |
003 5         |         |         |         |         |         |         |         |         |         |
004 5         |         |         |         |         |         |         |         |         |         |
005 5         |         |         |         |         |         |         |         |         |         |
006 5         |         |         |         |         |         |         |         |         |         |
007 5         |         |         |         |         |         |         |         |         |         |
008 A         |         |         |        5|  3 4 1  |         |         |         |         |         |
009 A         |         |         |         |         |         |         |  5      3         | 1       |
010 A---------|---------|---------|---------|---------|---------|---------|-5-----4-3---------|--2------1
011 A         |         |         |         |         |        5|   4  1  |   3     |         |         |
012 A   1  3 5|         |         |         |         |         |         |         |         |         |
013 A1   23 4 | 5       |         |         |         |         |         |         |         |         |
014 A  21   345         |         |         |         |         |         |         |         |         |
015 1   32  4 5         |         |         |         |         |         |         |         |         |
016 1  23  4  5         |         |         |         |         |         |         |         |         |
017 A1 23 4 5 |         |         |         |         |         |         |         |         |         |
018 A 14   5  |         |         |         |         |         |         |         |         |         |
019 A5 4      |         |         |         |         |         |         |         |         |         |
020 2435------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
021 254       |         |         |         |         |         |         |         |         |         |
022 5 4       |         |         |         |         |         |         |         |         |         |
023 35        |         |         |         |         |         |         |         |         |         |
024 35        |         |         |         |         |         |         |         |         |         |
025 35        |         |         |         |         |         |         |         |         |         |
026 45        |         |         |         |         |         |         |         |         |         |
027 45        |         |         |         |         |         |         |         |         |         |
028 5         |         |         |         |         |         |         |         |         |         |
029 5         |         |         |         |         |         |         |         |         |         |
030 5---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
031 5         |         |         |         |         |         |         |         |         |         |
032 5         |         |         |         |         |         |         |         |         |         |
033 5         |         |         |         |         |         |         |         |         |         |
034 5         |         |         |         |         |         |         |         |         |         |
035 5         |         |         |         |         |         |         |         |         |         |
036 5         |         |         |         |         |         |         |         |         |         |
037 5         |         |         |         |         |         |         |         |         |         |
038 5         |         |         |         |         |         |         |         |         |         |
039 5         |         |         |         |         |         |         |         |         |         |
040 5---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
041 5---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
* * * * * * * * * * * * * * * * * * * * * * * *  END OF GRAPH NUMBER 75  * * * * * * * * * * * * * * * * * * * * * * * *

WAITINGTIME DISTRIBUTION                                      VERTICAL SCALE= 5MSEC/STEP

NUMBER OF PEOPLE EDITING= 1.*PLOTNUMBER                       HORIZONTAL SCALE SEE TOP OF PLOT
```

```
MODE 3,    LIMITS: MIN  0.00000 ,+00, MAX  1.00000 ,+02,    INT/COL 1.00000 ,+00,          PAGE  1 OF GRAPH 76

        000       010       020       030       040       050       060       070       080       090       100
    000 5---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
    001 5         |         |         |         |         |         |         |         |         |         |
    002 5         |         |         |         |         |         |         |         |         |         |
    003 5         |         |         |         |         |         |         |         |         |         |
    004 5         |         |         |         |         |         |         |         |         |         |
    005 5         |         |         |         |         |         |         |         |         |         |
    006 5         |         |         |         |         |         |         |         |         |         |
    007 5         |         |         |         |         |         |         |         |         |         |
    008 A         | 5341    |         |         |         |         |         |         |         |         |
    009 A         |         |         |    5 4 3|   1     |         |         |         |         |         |
    010 A---------|---------|---------|---------|---------|-------5-|-4-3--2--|----1----|---------|---------|
    011 A         |         |         |         |         |         |         |     5   | 4    3  2     1   |
    012 A         |         |         |         |         |         |         |        5|   4    3|  2    1 |
    013 A         |         |         |         |         |         |         |         |  5   4  |3  2   1 |
    014 A         |         |         |         |         |         |         |         |     5   4   32   1|
    015 A         |         |         |         |         |         |         |         |        5| 4  3 2 1|
    016 A         |         |         |         |         |         |         |         |         | 5 4 3 21|
    017 A         |         |         |         |         |         |         |         |         |   5 4 32|
    018 A         |         |         |         |         |         |         |         |         |      53 2
    019 A         |         |         |         |         |         |         |         |         |      5432
    020 A---------|---------|---------|---------|---------|---------|---------|---------|---------|-------5-3
    021 A         |         |         |         |         |         |         |         |         |       543
    022 A         |         |         |         |         |         |         |         |         |       543
    023 A         |         |         |         |         |         |         |         |         |        53
    024 A         |         |         |         |         |         |         |         |         |        54
    025 A         |         |         |         |         |         |         |         |         |        54
    026 A         |         |         |         |         |         |         |         |         |        54
    027 A         |         |         |         |         |         |         |         |         |         5
    028 A         |         |         |         |         |         |         |         |         |         5
    029 A         |         |         |         |         |         |         |         |         |         5
    030 A---------|---------|---------|---------|---------|---------|---------|---------|---------|---------5
    031 A         |         |         |         |         |         |         |         |         |         5
    032 A         |         |         |         |         |         |         |         |         |         5
    033 A         |         |         |         |         |         |         |         |         |         5
    034 A         |         |         |         |         |         |         |         |         |         5
    035 A         |         |         |         |         |         |         |         |         |         5
    036 A         |         |         |         |         |         |         |         |         |         5
    037 A         |         |         |         |         |         |         |         |         |         5
    038 A         |         |         |         |         |         |         |         |         |         5
    039 A         |         |         |         |         |         |         |         |         |         5
    040 A---------|---------|---------|---------|---------|---------|---------|---------|---------|---------5
    041 A---------|---------|---------|---------|---------|---------|---------|---------|---------|---------5
* * * * * * * * * * * * * * * * * * * * * * * *  END OF GRAPH NUMBER 76  * * * * * * * * * * * * * * * * * * * * * * * *

CUMULATIVE WAITINGTIME DISTRIBUTION                       VERTICAL SCALE= 5MSEC/STEP

NUMBER OF PEOPLE EDITING= 10*PLOTNUMBER                   HORIZONTAL SCALE SEE TOP OF PLOT
```