Stanford Department of Computer Science
Report No. STAN-CS-8 l-839

February 198 1

SHORT WAITS

by

Arthur L. Samuel

DEPARTMENT OF COMPUTER SCIENCE
Stanford University

Short WAITS

by
Arthur Samuel

Computer Science Department
Stanford University
February 10, 1981

Short WAITS

Abstract

This is an introductory manual describing the SU-AI timesharing system that is available primarily for sponsored research in the Computer Science Department. The present manual is written for the beginner and the user interested primarily in the message handling capability as well as for the experienced computer user and programmer who either is unfamiliar with the SU-AI computer or who uses it infrequently. References are made to the available hard-copy manuals and to the extensive on-line document files where more complete information can be obtained.

The principal advantages of this system are:

1) The availability of a large repertoire of useful system features; 2) The large memory; 3) The large file storage system; 4) The ease with which one can access other computers via the ARPA net; 5) The file transfer facilities via the EFTP program and the ETHERNET; 6) The XGP and the DOVER printers and the large collections of fonts available for them; and 7) The fast and convenient E editor with its macro facilities.

Contents

Short WAITS

Introduction

This is an introductory manual to the WAITS operating system and to the SU-AI time-sharing computer, intended for the beginner and as a ready reference to the more complete documentation available elsewhere. Especial attention is given to the message handling facilities and to the commands and programs that are generally useful in the writing and handling of letters, texts, documents and files of all sorts and in the preparation, compiling and execution of computer programs. A companion manual, Essential E, CSD Report STAN-CS-60-796, describes the most commonly used text editor and is recommended as collateral reading.

Much of the material referenced in this manual is covered in greater detail in the Monitor Command Manual, Stanford Artificial Intelligence Laboratory, Operating Note 54.6. This may be read on-line by typing **READ** MONCOM<CR>. Update information is available by typing **READ** MONCOM.UPD<CR>. Additional documentation is available on-line in many individual files, both with respect to the matters covered in this manual and on the many other facilities available on the system, such as compilers for all of the more commonly used computer languages (as well as for the newer ones currently under development), cross compilers, assembliers, packages of mathematical routines, sorting and merging routines, etc.. A partial list of this on-line documentation is given in Appendix II. Ready reference to this material is available through the mechanism of the HELP command as described below. Lists of the hard-copy Artificial Intelligence Laboratory Memoranda may be read on-line by typing READ AIMLST <CR> and of the Operating Notes by typing READ SAILON<CR> although too much reliance should not be given to these older reports as they tend to become obsolete in a very short period of time.

The SU-AI computer with its **WAITS** operating system and the associated large, and well integrated, file system is used primarily for sponsored research.

Before you use the SU-AI computer you should make arrangements for authorization. You will be assigned some disk space to store your personal files and you will be assigned a Programmer Name which can usually be your initials or a 3-letter nickname. If you intend to work from a remote terminal, you will also need a password. If at all possible, you should begin your experience with this system using a DataDisc terminal and this manual will assume that you do.

Details as to the method of obtaining authorization may be read from any DataDisc terminal (even without being logged in) by typing HELP LOGIN<CR>.

1. Your Computer Terminal

There are several types of terminals currently in use on the SU-AI computer and they differ principally in regard to the number, names, and uses of the special keys that distinguish a computer terminal from a typewriter. The use of a DataDisc or DataDisc-compatible terminal will be assumed in this manual.

There is one key that is common to all currently used terminals, the CALL key, and it is usually located on the far upper right. This key is used to call the system monitor. You will be interacting with the monitor when you first sit down at a terminal and whenever it prompts you by displaying a period at the beginning of a line. If the system does not print a period or if it simply fails to respond, then you rnay have to strike the CALL key (once or twice) to call the monitor. There are more graceful ways of getting back to the system monitor that you will learn in time, but the CALL key is always the Key of Last Resort.

You should note that there is a full set of keys for the numerals and, specifically, that the letter "l" is not used for the numeral "1" and the letter "O" is not used for the numeral "0". **Confusing, isn't it?** There is, of course, the usual space bar and there is a carriage return key (usually marked RETURN) that has some special functions. We will call this key the **CR** key. Then there are a number of other keys that are not the same on different types of terminals, as will be explained below.

You may have noted the use of capitals in referring to the named keys on the keyboard. This convention will be used throughout this manual. When there might be danger of confusing the name of the key with text that is to be typed, the key's name will be shown in angle brackets, thus, **<CR>**.

1.1 The DataDisc Keyboard

This explanation will be easier to understand if you are at a DataDisc terminal as you read it.

In the first place, most of the keys carry two symbols and there are two different shift keys (and actually two of each kind), one kind marked SHIFT to get upper-case letters and the other kind marked **TOP** to type the additional characters that are shown on the tops of the keys, i.e., above the main typewriter characters.

Then there are a number of unusual keys such as the **ESC**, FORM, **VT**, **BS**, LINE and **ALT** keys. These are all very useful but you can ignore them for the moment. You may need to use some of the remaining special keys even to log-in.

The *CONTROL* and the *META* keys (and there are two of each) are shift keys in that they are held down while other keys are struck. They transform the meaning of the struck keys into commands, a Monitor command if you are typing to the System Monitor, an E command if typing to the E editor, etc.. In the text that follows, when a key is to be struck with one of these two keys held down, this will be indicated by having the word *CONTROL* or *META*, in angle brackets and italicize, thus, *<CONTROL>* or *<META>*, just before the designated key symbol (or name) and it will apply to this one key stroke only.

If the system (or any other program, for that matter) is typing information (and scrolling

it) faster than you want to read it, type <CONTROL> <BREAK>, that is, depress and hold the CONTROL key while you strike the BREAK key. The typeout will be interrupted and the following legend will appear at the top of your screen:

```
********** →HOLDING← **********
```

Typing <*CONTROL*><CLEAR>, that is, holding the **CONTROL** key and striking the **CLEAR** key, permits the typeout to resume.

The BREAK and **CLEAR** keys when used without the CONTROL key have quite different meanings. The **CLEAR** key, in particular, is useful for deleting an entire line of text that you have typed in error.

I.2 Other Keyboards

A number of other types of keyboards are currently being used; some, such as the earlier DataMedia keyboard, do not have a *META* key and have an EDIT key that is used to replace the *CONTROL* key. If you are planning to use one of these, it might be well for you to seek some personal advice as to how to proceed.

2. Log-In and Log-Out

If you plan to use a DataDisc terminal, your first task will be to find a DataDisc terminal that is free, as evidenced by the following legend on the screen:

```
TAKE ME I'M YOURS!
```

There will be some other information that need not concern you for the moment. Occasionally you will find that someone's work is appearing on the screen even though no one is actually sitting at the terminal. There are ways to displace the remote user but, until you are quite familiar with what you are doing, it might be wise to find another terminal.

During busy times, and this means most afternoons, you may see something like:

```
Q: 115          No DD channels left; type ESC CALL to wait for one.
```

This means that the maximum permitted number of DataDisc terminals are in use. Should this be the case you can reserve a place in the queue by striking the **ESC** key followed by the CALL key. The number of your terminal will be added to the list (after the letter Q). When your turn comes, your screen will go blank and you will hear a beep. You should then log in without delay.

2.1 Log-In

You log in by typing L PRG then a carriage return, using the Programmer Name (in place of PRG) that has been assigned to you. On the DataDisc terminals the carriage return key is labelled RETURN. We will refer to this key frequently and we will use the abbreviation CR or <CR> instead of the word RETURN.

The full name for the logging-in command is LOGIN but the WAITS monitor is very tolerant and allows one to truncate commands, typing only enough letters to disambiguate the command. Also, no distinction is normally made between upper and lower case letters, but be careful, a few programs are not so tolerant. It happens that two Monitor commands, namely LOGIN and LOAD, begin with LO, so LO is not an acceptable truncation while LOG and LOA are. In some cases, and the Log-In command is one of these, an initial letter is assigned, arbitrarily, to one of two or more commands that begin with the same letter. So L PRG<CR> works.

An alternate way to log in is to type L 1,PRG where the comma separating the 1 from the PRG specifies that you want to receive all system messages, even those you have already seen in prior sessions and you want the monitor system to perform those specific tasks that you have specified in a previously established OPTION.TXT file (to be explained below). Using a slash separator instead of a comma (or omitting the 1, entirely, as above) will limit the system messages to those that are new since your last log-in. Using a period will produce a fast log-in, with no messages and with the OPTION.TXT file ignored. To summarize these and other punctuation effects:

Symbol	Meaning	Symbol	Meaning
,	All messages, uses OPTION.TXT	/	New messages, uses OPTION.TXT
.	No messages, ignores OPTION.TXT		Note: L PRG is like L 1/PRG
!	Password required, then like /	%	Sets new password, then like /

The 1 in your log-in command (or the absence of anything before your PRG) says that you will be working on your project 1 and that most of the files that you will want or will create will probably be in your 1,PRG directory area. The 1 is called a PRoJect identifier (PRJ) and you may find it convenient to have several different PRJS as you accumulate files of different sorts. Any desired string of not over 3 alphanumeric characters can be used as a PRJ, however, certain system programs expect you to have the numeral 1 as an available PRJ. The two parts, namely PRJ and PRG when taken together are frequently refered to as your PPN (Project-Programmer Name).

If you are required to use a password, the system will now request it. Your password will not appear on the screen as you type it, so be careful.

If everything is in order, the system will acknowledge that you are now logged in by typing something like the following:

```
Job 41     SU-AI WAITS 9.12/D Assembled 01/13/81
Wednesday     14-Jan-81     1526     74F 23C
```

Several messages will probably appear that will not concern you at this time. You may want to read them anyway and practice using the <*CONTROL*><BREAK> command to stop the

typeout. Finally, the system will type a period at the beginning of a line, which is its way of telling you that it is now ready for your next command.

2.2 Automatic Log-In Services

When you log in normally (e.g. .LPRG<CR>, various optional automatic services are available. These options are specified in an OPTION.TXT file. If you are a beginner, you may, if you wish, ignore the rest of this section and work without an OPTION.TXT file for awhile.

An OPTION.TXT file consists of a single page with a series of commands specifying actions that are to be taken on log-in and log-out and in response to several other commands as will be noted later. A simple file might containing the following:

```
LOGIN: NOMAIL,WHO,BEEP;
```

This instructs the monitor to do three things on log-in: 1) (NOMAIL) To inform you if there is any new mail but not to ask you if you want it displayed; 2) **(WHO)** To display two lines of information at the top of the screen (the **WHO LINES**); and 3) **(BEEP)** To emit a beep when the system completes any operation taking more than 15 seconds.

Many other log-in services are available. You might get an expert to help you with this. Later, you will want to refer to Section 5.1 in the Monitor Command Manual (available on-line by typing READ MONCOM<CR> to the system monitor) to learn how to establish an OPTION.TXT file that meets your own particular needs. There is also a brief summary file obtained by typing READ OPTION<CR>.

2.3 Kjob (to Log Out)

To Log Out, you type K<CR>, that is, the letter K followed by a carriage return. This assumes that you are talking to the system monitor which, as mentioned earlier, usually prints a period at the start of a line to indicate that it is waiting for you. The system will reply by displaying a summary of your current usage that will look something like the following.

```
Job 41,[1,ALS]    Logged off TTY72      15:25  14-Jan-81
1.10 hours, console time.
0.96 minutes, cpu time.
143.14 pages average core.
Kjob
```

If you are using a telephone line, be sure that you clear the phone connection properly. If you want the logout process speeded up, type K/F<CR>.

Always plan to log out as soon as possible to make space for other users. Inactive jobs are logged out automatically after a certain period of time so it is a good idea to save your editing

before leaving your terminal even if you plan to return shortly.

One convenience might be mentioned--to remind yourself of something at the time of log-out, you can create a LOGOUT.MSG file containing a reminder message. The message will be typed back to you on log-out and the file will be deleted.

3. Editors

Editors come in two flavors. The basic editor is the System Line Editor. You have been using this editor even as you log-in to the system. You will be using it much of the time even when you are using the E editor, since E actually relinquishes control to the System Line Editor for most of the within-a-line tasks.

The next level of editors are what might be called File Editors. There are several varieties of these, (E, SOS and TECO) are the best known. E is, by far, the most popular and the use of E will be assumed in this manual. You can use h to prepare and edit all of your files, text files, computer programs and the source files for document compilers. Actually, E contains some facilities for formatting text so one can, and some people do, prepare documents with E alone.

Three document compilers are currently available on the SU-AI computer, as will be described in a later section. These require a well-edited input file containing text and formatting instructions and they produce an output file that can be sent directly to a printer. Some of these programs permit a limited amount of operator intervention to repair errors, but the most common way to operate is to go back to a file editor (usually E) to edit the source file and then to reinitiate the document compiler.

3.1 The System Line Editor

When you type commands to the system monitor and when you type text to the E editor, you will be using the System Line Editor. This Line Editor is quite special and it has many features that greatly simplify typing to the computer as compared to typing on a typewriter. You will need to know some of these features before proceeding.

You should first observe the CURSOR, an underscore mark that moves along as you type. Actually there are two underscores (unless you are on a DM) when you are typing to the system monitor, one appearing under the first character of the string of characters and the second that will be one character position beyond the last character that you have typed. This second underscore will be called the **CURSOR** and it always shows where the next typed character will appear.

A few essential Line Editor commands are:

<BS> If you mistype a character, the BACKSPACE key (labelled BS) can be used to erase it. Striking the **BS** key repeatedly will continue to erase characters from the end of the line. When not at the end of a line <BS> behaves like **<CONTROL>** <BS>, as described below.

<CONTROL> **<BS>** If you discover a mistyped character in an earlier portion of the line, you can move the CURSOR back to the wrongly typed character (without having to disturb

the characters that follow) by first depressing the CONTROL key and holding it down while you strike the **BS** key to get the CURSOR back from the end of the character string, and so to immobilize the automatic erasure feature. Thereafter, you can strike the **BS** key as many times as necessary (either with or without holding the CONTROL key) to move the CURSOR back as far as desired. Once positioned, you need only strike the desired key to overwrite, that is, to remove the unwanted character and to replace it by the desired one.

<CONTROL><FORM> If you want to get back to the first character in the line, hold the CONTROL key and strike the FORM key. Striking the FORM key, without holding the CONTROL key, will overwrite with an unwanted character.

<CONTROL><SPACE> If you should go too far back, you can move the CURSOR forward by depressing and holding the CONTROL key while typing with the space bar. If you strike the space bar without holding the CONTROL key, you will be overwriting the existing character at the CURSOR position with a space.

<CONTROL><TAB> Holding the CONTROL key and striking the TAB key will get you back to the end of the line. Striking the **TAB** key, without holding the CONTROL key, will overwrite with a **TAB.**

*<CONTROL>*D You can delete an unwanted character (and move the rest of the line of text to close the gap) by positioning the CURSOR under the unwanted character and depressing and holding the CONTROL key while you strike the letter D (for delete).

<META> <char> You can insert characters (and automatically move the rest of the text along to make room), by first positioning the CURSOR at the position in the line where you want the character to appear. You then depress and hold the *META* key while you type the character or characters that are to be inserted.

<CR> When you have completed a line of text or a command to your satisfaction, you can terminate the line of text or command by simply striking the RETURN key without having to move the cursor to the end of the line and the entire line will be accepted for processing. If you are using the text editor E, a line marker in the form of an arrow will be advanced to the next text line.

CONTROL><CR> If you have just completed a monitor command by typing a <CR> and if the system fails to understand your command string (or even if it executed your command and you want to repeat the command, as is, or slightly modified), you can retrieve the string for re-editing by typing <CONTROL><CR>. Remember, this means holding the CONTROL key down while you strike the RETURN key. This only works for the command string that you have just terminated so don't strike any other keys before giving this command.

The System Line Editor has many more features that you will want to know, but these few will get you started. Fortunately, most Data Disc terminals have a list of the line-editor commands displayed just above the keyboard. If you would like an up-to-date copy of this list, type HELP DDKEY<CR> to the system monitor. The monitor will ask for confirmation. Terminate your Y answer with a <CR> (the Helper doesn't tell you this). You will be told when the **XGP** starts to print your request and you should go to room MJH 433 to get it without delay.

3.2 The E Editor

Your next task will be to learn how to use the E editor. You can easily do this by a session or two with the program called **ETEACII**. Type HELP ETEACH<CR> and follow directions. This will create a special file (appropriate to the terminal that you are using) on your directory area. This file will instruct you and allow you to try out what you are learning on the file itself The file will rernain in your file area, so that you may return to it at a later time, but do delete it when have learned enough to proceed. You do this by typing DEL TEACH<CR> if you have been using a DataDisc terminal or DEL TEACH.DM<CR>, if using a DataMedia terminal.

You will also want to get a copy of the CSD Report STAN-CS-80-796, called ESSENTIAL E. This is more easy to read than the comprehensive on-line manual E.ALS[UP,DOC], the Manual of Last Appeal. You can refer to this on-line E manual by typing ER?<CR> to the monitor, or while using E by holding the CONTROL key and typing the question mark symbol ? (this time without the usual **<CR>**, since it is a one-character command) and you can get back to the file you are editing by typing <*CONTROL*>H, i.e., the letter H with the CONTROL key and without a <CR>.

The ER command should be used to read normal (user-generated) files so that they will be protected against accidental damage and you should use the ET command only if you plan to alter the file. The READ command is used to read the documentation files that are available for general use.

4. Files and File Directories

A file is a stored collection of information, perhaps a program (either in source form or compiled), data, mail, a report or texts of any sort. Most files on the SU-AI system are stored on permanently-available disk units. Later, you may want to store some of your data on magnetic tape, and if so, please refer to Appendix A 9.2 in the Monitor Command Manual and perhaps get someone to help you the first time.

The disk units are amazingly reliable and their contents are copied to back-up magnetic tapes at frequent intervals by a system program called DART. Monitor commands to run DART can be used to retrieve files that have been removed from the disk or damaged, although, of course, the files are restored as of the last time they were saved. As an added feature, there is a command PUMPKIN that allows you to leave an order to have desired files restored for you.

4.1 File Identifiers

Files stored on the disk carry a file identifier of the form FILNAM.EXT[PRJ,PRG], where FILNAM is a name of 6 or fewer characters (usually alphanumeric). The 3- or fewer-character extension EXT is normally used to identify the file as to its type, and may be omitted for files that are not used as source files for other programs.

Should you be so unfortunate as to use more than your allotted file space, some of your files may be purged and the system purger uses the extension to decide which of your files it will

delete. Files with an extension of TMP are the first to go! You can specify the order yourself with a PURGE: line in your OPTION.TXT file (READ PURGE<CR> for details).

Some commonly used extensions and their meanings are:

DMP Binary core image	LST Listing output file	SAI SAIL source file
DO ADO file	MSG MAIL message file	TEX TEX source file
DOC Document file	PAS PASCAL source file	TXT Text file
FAI FAIL source file	PUB PUB source file	UPD Document update
F4 Fortran source	REL Relocatable core image	XGP File for XGP

The PPN, that is, the [PRJ,PRG] portion of the file identifier, further identifies the file and prevents it from being confused with other files that may bear the same name. This portion also specifies the file directory to which the file is assigned. For your first files, the PRJ will be the numeral 1. The PRG will be your assigned programmer name. You can read or store files bearing the PRJ,PRG that corresponds to the PRJ,PRG under which you are logged in (unless you are aliased to another directory, as explained in the next section) without typing the brackets and the bracketed information. If you are referring to files on other directories, this information usually must be included. Because of the [PRJ,PRG], you can be as free and fanciful as you like in creating file names, but do try to make them meaningful.

Document files on [UP,DOC],[S,DOC] and on a few other public document directories follow a different convention and use the extension, EXT, to contain the Programmer Name of the person who created the file. As a convenience, documents on these directories can be read with E by typing READ FILNAM<CR> without an EXT and without the [PRJ,PRG]. Please note the distinction, READ expects to find the file on one of the public document directories, whereas ER and ET expect to find the file on your log-in(or alias) directory or on the specified PPN directory.

4.2 ALIAS and ACCESS (to change default file PPNs)

You will soon feel the need for a convenient way to change PPNs in order to simplify the use of files in a different area. You do this by typing AL PRJ,PRG<CR>, where PRJ,PRG is the desired default PPN to be used for all subsequent file references. If your log-in PRG is to be retained you need only type AL PRJ<CR>. Once aliased to another PPN, files in the aliased PPN are referenced without the need to specify the PPN and files on your original log-in area do require the PPN to be specified. To return to your log-in PPN your type AL<CR>.

Type HELP UFD<CR> or READ MONCOM<CR> to learn how to establish other file directories under your own PRG.

If you alias to another user's PPN, you may refer to his unprotected files without having to specify the PPN but you do not acquire any of the owner's special privileges and your logged-in PPN is still checked for protection purposes.

ACCESS is similar to ALIAS in format but it requires you to supply an appropiate password and then it gives you owner access to the desired file area (running a brief program to do this). READ ACCESS <CR> for full details.

In the rest of this manual we will refer to your aliased PPN frequently and it should always be

understood that this will mean your log-in **PPN** if you have not aliased elswhere or to the area that you are accessing if you have given an **ACCESS** command.

4.3 File Protection

In addition to the protection afforded by the use of passwords (which itself does not protect files), it is possible to specify a protection code for each file directory and for each file. The code specifies the access allowed to the file for its owner (i.e. a user logged in under the file's **PPN),** for other authorized users, and for guest users. The system is not totally secure, and indeed it is not intended to be since it is customary for users to share programs and data with each other. Most users do not protect their files at all except to write-protect certain critical files to prevent accidental damage. If you feel that file protection is necessary, the three-octal-digit protection code (nnn) is obtained by adding together the desired values as shown in the following table. This code can be appended as a switch, /PR=nnn, to **RENAME** command. The user may impose an additional form of protection, while reading a file with E, by entering it with an **ER** command, by using a /R switch, or by giving the E command <*CONTROL*>XREADONLY<CR>.

The protection code values are:

User may not	Change PR	Read	Overwrite
Owner			100
Local user	040	020	010
Guest	004	002	001

(for files, 400 means Don't Dump, 200 means DELete protected)
(for directories, 400 means Password required for remote use only)

4.4 E Prepared Files

Files prepared by the E editor are usually divided into pages of arbitrary lengths, as dictated by the subject matter and they normally contain a directory page. This directory page is automatically kept up-to-date by E as the file is written, with information as to the relative location of the pages on the disk and with page identifying information extracted from the first line on each page. This directory can bc of considerable use to the writer of the file and to others if care is taken to see that the first line on each page properly characterizes the material on the page. The **HOW** file is an example where this is done. Special E commands simplify the generation of appropiate directory lines for programming-language files (listing labels, procedure names, etc., as desired, to facilitate directory searches).

Most of the compilers and document-preparation programs on the system are written so as to recognize and ignore the directory page and to ignore the pagination.

4.5 Filehacks

Filehacks are simple shorthand names for certain commonly-referenced files. They usually consist of a back slash \ followed by a name that may be truncated to only enough letters to disarnbiguate it from other filehacks. For example, typing ET\G<CR> will enter the GRIPES.TXT[2,2] file.

Filehack names may be used with the COPY command and while using the E editor. Some useful filehacks are listed below. Some of the files are not maintained in E format and some of the filehacks imply /R (readonly) mode, as shown in the list. To override a default /R mode, you can explicitly specify /–R after the filehack.

FILEHACK	Abbr	/R?	File
\BBOARD	\B	No	The system bulletin board file
\CSD	\C	Yes	CSD bulletin board file
\DIGEST	\DI	Yes	The current AP news digest
\DOW-N	\DO	Yes	The system downtime forecast file
\FORWARD	\F	Yes	The forwarding file for the MAIL system
\GRIPES	\G	No	The system gripe file
\NEWS	\NE	Yes	The current NYT news summary
\NOTICE	\NO	Yes	The system message file
\NS	\NS	Yes	Your current NS notifications file
\MAIL	\M	No	Your current mail file
\PLAN	\P	Yes	Your current plan file

The E editor supports two other useful filehacks, the use of the question mark ? to refer to the on-line E manual and the use of the partial symbol ∂ to refer to your own mail file. Typing ET∂<CR> is a good way to start the day (after logging in!).

4.6 To DELete, RENname or COPy Files

Files may be deleted by typing DEL FILNAM .EXT [PRJ,PRG] <CR>. If several files arc to be deleted, their names may separated by commas and typed in one command. The [PRJ, PRG] s may be omitted if the files are on your aliased PPN(or on your original log-in PPN in you are not aliased away). You may prefer to use the DIRED program (see below) if you have several files to delete.

Files are renamed by typing REN newname←oldname<CR>, where the names must include extensions (if used) and PPNS if different from your alias PPN. RENAME may be used to change the protection by typing REN oldname/PR=nnn<CR> where nnn is the desired protection (see table under Protection above). Renaming has the advantage over copying and deleting the original file in that only the directory references are altered and no actual copying of the file is needed.

Files are copied (without destroying the original file) by the COPY command. As always, the [PRJ, PRG] portion of the name may be omitted for files on your alias area. An example will illustrate this and several other points:

```
COP XXXYYY.TMP←XXX.ALS[UP,DOC](2:*),YYY.TMP(3,6:9)<CR>
```

There are no such files, but if there were, a new file would be written on your alias area containing all but the first page of file XXX followed by pages 3, 6, 7, 8 and 9 from file YYY (also from [UP ,DOC]) appended. Note that the **PPN** is sticky, that is, once one has been specified for one source file, it need not be repeated and so both source files would be from [UP, DOC] . The resulting file would, of course, not contain an E directory page and you would be asked if you want one created when you try to edit the new file with E.

If you want to copy one or more files into your own area, as separate files, each with the same name and extension as in the source, you may omit the destination entirely and simply type:

```
COP NAM1.EX1[PRJ,PRG],NAM2.EX2,NAM3.EX3,NAM4.EX4<CR>
```

If you list several files, watch your periods and commas, they must be right. If you try to write over a file that already exists, you will be given the option of replacing the original file or of aborting the request. If the source file or files do not exist you will be told and the *<CONTROL>* <CR> command may be used to recall your typed string so that you may correct or modify it (but not if you have typed Y or N to answer any of copy's questions).

A word of caution-While PPNs are sticky, as shown, for the **COPY** command, different sets of rules are used in other situations.

4.7 Listing Files—SPool, XSpool, UNSpool, Qspool and DOVer

While it is possible to list files directly on the line printer and on the **XGP** printer, these devices are in nearly constant use and there is a serious scheduling problem. The preferred procedure is to spool the request and allow the system to schedule the actual listing on the basis of the time of the request and the projected size of the listing. Spooling has the added advantage to the user of not tying up the user's job while the listing is actually being done. Large jobs may also be scheduled for delayed spooling to a time, say at 4 A.M. (/HOLD=400) when the printer will be in less demand. The listings are normally preceded by a title page for easy identification. Delayed listings of **XGP** listings are not recommended at present because of the poor stacking capability of the **XGP** and inadequate storage facilities, but do try to schedule large listings during off hours if at all possible.

Files are spooled for the line printer by typing SP FILNAM. EXT [PRJ,PRG] <CR>, and for the **XGP** by typing XS FILNAM.EXT[PRJ,PRG]<CR>. If the PPN is not specified, your alias PPN is assumed. If there is no EXT and the given file name does not exist without an extension then **XGP** is assumed. You may follow the file name with a list of page numbers or page ranges (enclosed in parentheses) that are to be printed, for example (2,4,6:8) will cause pages 2, 4, 6, 7, and 8 to be printed.

Several files may be spooled by one command, with the file names separated by commas. The file name may carry switches. If the switch follows the name, it applies to that file only. A switch listed ahead of a file name applies to that file name and all following names.

An asterisk may be used in place of a name, an extension, a PRJ or a PRG. This will match any term except extensions of RPG, DMP or REL. This may, of course, result in the spooling of more than one file. If you use an asterisk, it might be wise to add an /ASK switch which will cause SPOOL to list the files it has selected, one at a time, and ask for a Y confirmation before

spooling each file. Alternatively, the file names and associated specifications may be read from an auxiliary file if you use the @ sign followed by the name of this auxiliary file in the spool command.

Most of the files that you may want to spool will probably be normal text files, that is files written in **ASCII** code, but it is possible to spool binary files with the output printed in octal by appending the switch /OCTAL or /DUMP.

A few of the more useful spooling switches are:

Switch	Action
/DElete	**Delete file after printing**
/NOHeading	**No headings at top of each page**
/NOTItle	**No title page**
/REPeat=n	**Make n (decimal) copies (not sticky)**
/OCTal	**Octal listing**
/ASK	**Ask for confirmation**
/Hold=hhmm	Hold **in queue until time hh hours and mm minutes**
/Qspool	**Report status of queue after entering request**

Typing UNSP<CR> will cause the spooler to list the files in the queue that carry your PRG, one at a time and ask if you indeed do want them removed from the queue. Typing Q<CR>, or Q/X<CR> if you are only interested in the **XGP**, will cause the present contents of the queue to be displayed.

It is now possible to print files on the Dover printer. This facility is still under development as a part of a general system program called **PRESS**. At the present time, typing **R** PRESS<CR> or simply DOV<CR> will run this program and information is then available by typing HELP<CR> in response to the program's ready signal of Press>.

4.8 Listing and Searching File Directories

As mentioned earlier, the **PPN** portion of the file identifier is used to assign the file to a directory. To see a listing of a file directory, of, say, the [UP ,DOC] directory, type DI [UP ,DOC] <CR>. For your own aliased directory, the **PPN** may be omitted or, if you want full information, type DI/FU<CR>. If you have more than a screen full, remember the **<CONTROL> <BREAK>** command to delay scrolling.

To save a copy of your directory in a file, type DI FILNAM←<CR>. If you have several PRJ's and want them all listed, type DI FILNAM← [*,PRG] <CR>. Don't, I repeat, DON'T use [*,*] or you will fill up all of your allowed file area, and maybe all of the available disk space as well, with all of the directories on the entire system, and it will take a long, long time!!!

There is a new program (**R WILD**< CR>) available with greater (wild card) search capabilities than with the **DI** command. For details, type READ WILD<CR>.

4.9 Directory Editing, DRD or R DIRED

The DIRED program allows the user to edit a directory in a manner quite analogous to the way one edits a file using E. You call DIRED by typing DRD<CR> to the monitor. You will be presented with a listing of the files on your aliased PPN directory. By default, this listing will be sorted alphebetically by FILNAM and EXT. Other sortings may be specified by your OPTION.TXT file or they may be specified at the time DIRED is called.

You can modify DIRED's display by using commands that are similar to the E text-editor commands, and in so doing, you will be preparing a list of actions that will be executed later, either by an explicit command or by a normal exit from the program. For example, if you delete a file listing from the display by using the <*CONTROL*> <*META*>D command, the file itself will be listed for later deletion. You can also rename, spool and change the protection on files, all by using this program. READ DIRED before trying anything fancy. You can abort the actual execution of your action list, if you get into serious trouble, by using the CALL key.

4.10 Monitoring File Usage, DSKSIZ, R CKSUM and CK

The command DSKSIZ<CR> reports your disk usage and your disk allotment, or that for a specified PRG typed as an argument. Two switches may be used, /F for a fast summary (recommended) or /V (for verbose) if you want to be given the opportunity of having the information recorded in a file for later reference and, perhaps, the opportunity to write a second file PURDAT to contain a list of the files for the purger to delete.

The R CKSUM< CR> command allows you to specify a list of files that you want monitored with a record kept of all pages to which any changes have been made since the last reporting. Common uses of this command are to monitor changes to \BBOARD, additions to errata and UPD files relating to programs of interest, and changes to files that are being worked on by several people simultaneously. For full details type READ CKSUM<CR>.

The monitor command CK<CR>, reports the existence of new mail and of CKSUM changes. CKSUM changes, so reported, may then be viewed (either automatically or by giving the monitor command CONT<CR>, with the choice previously specified by an R CKSUM command). One enters E with the pertinent files listed as referenced files so that one may switch between them by the <*CONTROL*>H command and with the pertinent pages marked so that they may be viewed, one after the other, by using the <*CONTROL*>M command.

5. HELP and READ HOW to learn all about it

The HELP command and the HOW file greatly simplify the task of locating WAITS system information and supplement the use of the READ command which is limited to those system features for which there are separate documentation files.

5.1 HELP

The Monitor command HELP NAME<CR>, where NAME is the name of any one of many system commands or of many user-available programs, is all that you must type to get information regarding the named command or program.

In many cases, a brief resume will appear on your screen with enough information for you to use the command or program and with references to where more information is to be found. Alternatively, you may be told of a program that can provide the service requested or that can produce a document containing the desired information and you will be asked for a Y confirmation that you want this program to be run. Typing HELP HELPER<CR> will supply one-line descriptions of many of these messages and services, and typing HELP<CR> (without a NAME), will list all the possible names.

Finally, you may be told that no HELP file exists but that a file does exist that may contain the information that you seek. The necessary R E;FILNAM command will appear on your screen only awaiting the typing of a <CR> for confirmation that you do indeed want to see this file. Most of these files are on the UP,DOC and S,DOC file directories.

Appendix H contains a complete list of all HELP resumes, services and R E;FILNAM references as of 11 Dec. 80. Typing READ HELP<CR> will provide information as to how you can write HELP files for the [3,2] directory.

5.2 READ HOW

The HOW.ALS[UP,DOC] file is a (currently incomplete) reference file of information regarding monitor commands and useful system programs. It is organized with the material on each topic restricted to a single page and with the first lines of each page so arranged that the file's directory page forms an easily-used table of contents. The file is read by typing READ HOW<CR>(how else?). The directory page can be used as a browsing aid to find out what is available. In many cases the material in the HOW file is sufficient either to allow one to do simple things without the need for further study, or to prod one's memory as to the necessary details.

It is hoped that many users will add to the HOW file by writing HOW-formatted pages for subjects on which they are well versed. Mail such information to ALS for proofing and final inclusion in HOW. Corrections to existing pages are also welcome.

6. System Programs

Several system programs have already been mentioned on earlier pages of this manual. There are a large number of such programs that are of great value in simplifying the user's tasks. A few of these will be described in this section.

Commands like LOGIN tell the monitor to run particular system programs in your core image, that is in memory space specifically assigned to you by the system. Many of these programs are evoked by simply typing their names, usually followed by some parameters. These commands

are all documented in the Monitor Command Manual (**READ** MONCOM<CR>) with recent additions and corrections in MONCOM.UPD. Cryptic one-line descriptions of these commands are given in Appendix C.

There are a few system commands that act without occupying the user's core space and hence may be issued without destroying a core image that the user may wish to retain. A special escape command <ESC>. (where the period is a part of the command) may be used to exit from an operating program and so to allow the necessary commands to be typed. After giving the desired command one can reenter the operating program by the <BREAK>. command (where the period is again a part of the command). Some of these system commands may also be given while using the E editor by special E commands, for example, to SPOOL a page of text or to MALL some text from a file being edited to another person.

Other equally useful programs are called by the **R** command; **R** CKSUM<CR> , is an example. These are usually documented in separate manuals. Note that **R** is not an abbreviation for **RUN** but is a different command and it requires the program to be found on a special system directory [1,3]; this **PPN** will be used even if you should specify another one. The RUN command is used to run user's programs as will be explained later. Many of these (**R** FILNAM) programs are documented by **HELP** and **HOW** but a substantial number are not. A complete list of DMP files on [1,3] is given in Appendix R.

Just to complicate matters, there is yet another category of commands that are called by name but that actually run the SNAIL system program. SNAIL, in turn, may select and call the correct system programs to process a user's source file and finally may load and initiate the execution of the user's compiled program, all this without further user intervention. The distinctions between these various catagories of commands is somewhat artificial as far as the user is concerned and the only thing to note is that some commands are called directly by name while certain others require the **R** FILNAM method of calling.

6.1 Compiling and Running User Programs, with SNAIL

Computer programs can exist and be stored as files in at least three forms, 1) as one or more Source files, written in either an Assembly language or some Higher-Level language that can be processed by an Assembler or Compiler to produce Relocatable program modules, 2) as one or more Relocatable (REL) files that can be linked together by a Link Loader to produce an executable Core Image, and 3) as a Dump (DMP) file that can be loaded directly into core for execution. User-created programs that are no longer under development often are kept in DMP form and can then be run by the the RUN command. The SNAIL program is designed for use with user-created programs while they are still undergoing development and are subject to frequent recompilations.

Some compilers consist of two separate programs, an initial Compiler that produces a file written in an intermediate language for some hypothetical machine and a final Translator that translates this intermediate language into the final machine-dependent form. At the present time, the SNAIL, commands deal only with single-stage compilers which may, of course, involve multipasses as long as the external handling of intermediate-language files is not required. It is customary to write **DO** files that take the place of SNAIL commands for dealing with these two-separate-stage compilers.

SNAIL commands are actually of three classes, those having to do with the compiling of computer programs, which will be discussed in this section, those relating to document compilers and a third and less used class that incorporate some of the features of the **DO** commands (see the Monitor Command Manual for these).

SNAIL commands allow the user to delegate to the computer many tasks relating to: 1) the checking as to the need for a recompilation, 2) the choice of the proper assembler or compiler to produce a new Relocatable (REL) file, 3) the production of an assembly listing (if wanted), 4) the choice between loading a Dump (DMP) file or link-loading the specified (REL) files to produce a Core Image, 5) the choice of a suitable debugging program, 6) the production of a new DMP file, and 7) the starting of execution either of the program proper or of the debugger. **SNAIL** commands are remembered and may be reinitiated by giving the command only (without file names and switches) or they may be reinitiated directly from the E editor by the <*CONTROL*>XGO<CR> command.

The date written is kept with each Source file, REL file and DMP file and when a SNAIL command is given, the 3 dates are compared and (unless ordered otherwise by switches) the amount of work is minimized. This is done, 1) by not recompiling a source file if there is a corresponding REL file and if this REL file carries a later date written than does the source file, and then, 2) if loading is required, by omitting the linking operation and by loading the DMP file if one exists and if this, in turn, carries a later date written than do the REL files.

The six compiler commands and their actions (when not modified by switches), are:

SNAIL Command	Abbrev	Produce REL file	Link Load	Produce Core Image	Include Debugger	Will Start Execution with
COMPILE	COM	may				
LOAD	LOA	may	may	will		
PREPARE	PRE	may	may	will	will	
EXECUTE	EX	may	may	will		user program
TRY	TRY	may	may	will	will	user program
DEBUG	DEB	may	may	will	will	debugger

These commands take as arguments one or more file identifiers, separated by commas. Individual programs may also be broken into two (or more) files having only one END statement at the end of the last file. This is convenient if a common program portion is to be shared by more than one program. Such segment names are separated by plus signs rather than by commas, and they compile into a single REL file carrying the name of the last segment. Alternatively, the arguments (that is, the names of the files to be used by the SNAIL command and their associated switches) may be contained in a command file and refered to by an @ sign followed by the name of the command file, e.g. LOAD @SYS.

The choice of the proper compiler is made by SNAIL on the basis of the file's EXT. If standard extensions are used, **SNAIL** will refuse to use an incorrect compiler. The file extensions need not be explicitly stated in the command if the source files carry standard extensions. In those cases where recognized extensions are not used, the desired compilers may be specified by switches. The following extensions and compiler switches are recognized:

EXT	Switch	Processor	EXT	Switch	Processor
FAI	/FAIL	FAIL assembler	SAI	/SAIL	SAIL compiler
MAC	/MACRO	MACRO assembler	F4	/F4	FORTRAN compiler
PAL	/PALX	PALX assembler	BLI	/BLISS	BLISS compiler
MID	/MIDAS	MIDAS assembler	AL	/AL	AL compiler
S1	/S1	FASM assembler	PAS	/PASCAL	PASCAL compiler

Switches may also be used to interfere with the automatic date-written dependencies noted above, to specify the debugger that is to be used, to produce a DMP file, and to produce various types of listings. Some of the more important switches are:

Switch	Abbrev	Meaning
/COMPILE	/COM	**Re-compile always without checking the REL dates**
/CREF	/C	**Request a cross-reference** listing
/DDT	/D	**Use DDT if a debugger is required**
/LIBRARY	**/LIB**	**Search this term as a library**
/LIST	/L	**Make a listing file**
/MAP	/MAP	**Produce a loader map of global symbols**
/RAID	/RA	**Use RAID if a debugger is required**
/SAVE	/SAV	**Make a DMP file of the core image**

For a complete list see MONCOM.BH[S,DOC] Section A 1.7. Switches may also be passed to the translators and to the link loader.

6.2 The DO Command

The **DO** command instructs the monitor to load the line-editor buffer with text from the file named in the command, which should contain a series of commands. The system then proceeds to execute this series of commands just as if the user had typed them directly. The command string may be quite elaborate, the only proviso being that the total length of the string must not exceed the size of the buffer (about 140 characters). You can even circumvent this restriction by splitting your intended string of commands at a point where the command string leaves the system at monitor command level, inserting a **DO** command at this point calling a second DO command string (on the same page) and having the rest of the commands in this second (and third, fourth, etc.) **DO** command string.

Certain necessary character conversions are required in **DO** files to allow special keys, such as *<CONTROL>*, *<META>* and **<CR>**, to be indicated. The <CR> and <LINE> key strokes that may be used in writing the command string are ignored, as are the E-editor directory and page marks. The following special symbols are used:

Symbol	Meaning	Symbol	Meaning
↔	**Translates to CR**	α	**Adds *CONTROL* bit to the next character**
↓	**Translates to LINE**	β	**Adds *META* bit to the next character**
≠	**Translates to** ALT	$\alpha\beta$	**Adds both CONTROL and *META* bits**
λ	**Translates to CALL**		**Terminates and separates DO strings**
⊗	**Translates to ESC**	≡	**Quotes the next character**
⊗—	**Translates to BREAK**		**(so that an α, say, can actually appear)**

Note that these symbols and those used for E macros are similar but not identical, so refresh your memory before writing either type of string.

The question mark symbol ? performs the special function of designating the character that immediately follows it to be a variable name to be defined by the user at the time the **DO** command string is executed. Suppose the character is A (i.e., ?A). At the first occurence of this ?A in the command string, **DO** types out A = and waits for the user to type in a text string ending with **<CR>**. This string (but not the <CR>) is substituted for every occurrence of ?A in the command string and the execution proceeds.

If several **DO** command strings are stored in the same file, they must be separated by the | symbol. The command **DO** T(3)<CR> calls for the execution of the third command string in the file named T.DO.

The **DO** command is remembered in a **TMPCOR** file as if it were a **COMPILE-type SNAIL** command, so the exit-mode commands in E (<*CONTROL*>XGO<CR> for example) will also repeat the **DO** command. Thus, users of languages that are not recognized by **SNAIL** can write **DO** files to perform similar functions.

Many of the features mentioned above can be illustrated by a **DO** file, named T.DO, that was used in preparing this manual and that contains the following text:

```
ET TEXT.TEX↔αXparen{↔αXdefine ∩↔α(α)αβ↓↔|
R maxtex↔\input text.tex↔|
ET ?A.TEX↔αXparen{↔αXdefine∩↔α(α)αβ↓↔|
R maxtex↔\input ?A.tex↔|
DOVER ?A.PRE↔|
```

The first string in this file (called by the monitor command **DO** T<**CR**>) starts E-editing the file TEXT.TEX. It then defines the parenthesis-finding commands to refer to the braces { and } and it defines a macro, callable by the E command <*CONTROL*>∩, that will search for a matching pair of braces, an operation that is very useful in checking TEX input files.

The second command string (called by the monitor command **DO** T(2)<CR>) starts the TEX document compiler (MAXTEX, in this case) and gives it the required input line specifying the file TEXT.TEX. When an error is reported by TEX, one can go back to E by the TEX e command without disturbing the retained **TMPCOR** established by the **DO T(2)** command so that TEX can be re-initiated from E by typing <*CONTROL*>XGO<**CR**>.

The third and fourth strings are similar to the first two except that the file name is given by the ?A convention so that the name of the source file may be typed-in at the time that the **DO** command is given. The fifth string is simply a duplicate of the **DOVER** command that TEX generates (with the file name in this same ?A form).

6.3 <ESC># <CR> Terminal-Input Macros

Terminal-input macros may be used to shorten the typing of frequently-used hard-to-type system commands. They are evoked by typing <ESC># <**CR**>, where # is an identifying number in the range from 5 to 20. Each macro definition is limited in length to 18 characters, although it may contain a **DO** command which extends the limit to 140 characters or more. Terminal-input macros remain associated with the login terminal even when aliased or mapped away. They are redefined automatically at login if they are saved in a CHRMAC.CHR file and if there is a LOGIN: **CHRMAC;** statement in one's **OPTION.TXT** file.

To create a terminal-input macro, type **R** CHRMAC<**CR**>. Then type the number to be used to identify the macro followed by a <**CR**>, then the macro itself. Type <ALT>X to exit from the defini'ng mode, type the > symbol to write the macro into a file, and finally, an E to exit from the program. The program is self documented, responding to a question mark ? and to an <ALT>? with lists of the options available while in the initial and the macro-defining modes, respectively. For more information, **READ MACROS** <**CR**>.

6.4 Who, When, Where and Finger

These four commands provide the user with information as to who is currently using the system and their current activity, WHEN they last logged off (even if they are currently logged in), WHERe they are located, and puts the FINGer on them (identifying them by name and physical location). These commands may be given simply by typing the capitalized portions of the command names as just used (with a <CR>) or they may be modified by programmer names given as arguments. Without an argument, WHEN refers to the interrogating user only while the other three commands provide information on all users.

WHO presents a dynamic display that is kept updated to show who is actually getting service with some information about the total usage. WHO normally exits automatically after two minutes and can be terminated earlier by typing either a <SPACE> or a <CR>. Since the material to be presented will usually more than fill the screen, it may be scrolled by typing <FORM> or ↑ for up and <VT> or ↓ for down, or it may be restricted in scope (while it is running) by typing certain single letter commands (without <CR>s) of which the following are the more useful:

R Only recently running jobs	M My jobs only
N Restore normal display	E Exit, leaving info. on screen

Still other commands are available that permit the detailed specification of the PPNs or job numbers that are to be displayed, either directly or indirectly through a referenced file. As an example, the command *PRG<CR> wil restrict the display to only jobs belonging to PRG. For full details type READ WHO <CR>.

WHO should not be confused with the two very useful Who Lines that can be made to appear at the top of the screen by the <ESC> w command (also automatically established at log-in by an OPTION.TXT request). Try it! and READ WHOLIN<CR> for an explanation of what you see.

The other three programs present static (possibly scrolled) displays. **WHEN** requires no further explanation. WHERe, without an argument, provides a list of all jobs in numerical order and a limited amount of statistical information about each job. FINGer provides a somewhat different set of details about the jobs and identifies each user by both PPN and name and gives the physical location of the terminal being used. If the FINGER request is for specific PPNs, and the individuals are not logged in, then the times of their last log-outs are reported.

6.5 FIND, DFIND and OFIND

The FIND command is used to locate information in a file by searching for a key (or several keys) within the specified file. The key may be any arbitrary string of characters without regard to their surroundings. The case of letters is disregarded unless specifically restricted, as noted below, and a space between words in the key will match zero or more spaces, tabs, CRLFS and formfeeds. If the file name is omitted, the system phone directory is assumed. Actually, the FIND command and its relatives DFIND (for searching the unabridged dictionary word list), and OFIND (for referencing one's OPTION.TXT file for detailed specifications), are very useful for other types of searches as well. One can, for example, search one's mail file by specifying IN ∂ or the system NOTICE.TXT file, by specifying IN ∂*.

The key may be specified in quite general terms, using the following symbols:

Symbol	Meaning	Symbol	Meaning
,	Comma used to separate strings	≡,	Comma is used in string
≡x	A lower case x	≡X	An upper case X
∀ A	n y character	3	Any char. except a delimiter
1	Any char. except the following	|	Any delimiter
00	Any num. of char. that follows	≡	Quotes the next character
{xyz}	Any one of the embraced chars.	∞{xyz}	Any num. of any embraced chars.

Hits are reported by typing out a portion of the text. By default, a paragraph is used for the FIND command and a line is used for the DFIND command. It is also possible to set a maximum number of lines that are to be printed on each side of the hit. The hit list may be sent to a named file by adding WRITING FILNAM to the command string, and in this case only the number of hits will be reported on your terminal.

The complete syntax of the FIND, DFIND and **OFIND** commands is:

find [**within** <delim>][**surround** <**num**>] <**key**> [omit[ting][**only**] <omits>][**in** <file>][writing <**file**> 1

where [] indicates optional elements, the delimiter used with WITHIN must be MSG, LINE, PAGE or PARAGRAPH, and < num> specifies the maximum number of lines to surround the key. For more information READ FIND.

The E editor has an extended command <*CONTROL*>XDFIND that interfaces with the DFIND program (Note that the <*CONTROL*>XFIND command in E is something else entirely!). This command starts up a phantom job to do the search, and a summary of the results are sent to your terminal. The summary includes the number of hits, as well as the first, last, and (if different) shortest "hit" lines. Incidentally, if you are a poor speller you might be interested in the **SPELL** program.

6.6 MAIL (to Users and to the System), SEND and TALK

The MAIL program is used to send messages to users and sometimes to special disk files for certain purposes. All users of the system make use of the MAIL program in some form or other. Even non-authorized users are able to use MAIL to communicate with authorized users via the system. The more important commands in this group are:

Command	Meaning
MAIL	**Send a message to one or more message files and notify adressees**
SENd	**Send a message to the terminals of one or more designated users**
GRipe	**Send a message complaining about a system problem**
REMind	**Schedule a message to be sent at some later time**
PLAn	**Create a file describing how to find you when not logged in**
EVent	**Send a message to all users about an event on a given date**
BATch	**Schedule the execution of a command string at some later time**
LAter	**Schedule the execution of a given program at some later time**
CK	**Report existence of new mail (and of CKSUM data, see section 4.10)**
ET∂	**Edit E-formatted message file**
RCV	**Edit non-E-formatted message file (now seldom used)**
CANCEL	**Delete REMIND, BATCH, or LATER requests or queued MAIL'**

The TALK command is a separate system feature that enables two logged-in users to link their terminals (even if of different types) so that everything either of them types will appear on both screens. This feature can be surprising to the person not initiating it, so it is wise to reach an agreement, usually through the exchange of SEND messages, prior to establishing the link.

The MAIL command, itself, will now be described. Each user of the system is automatically assigned space in a special message file, as required, to which messages can be sent. A logged-in user, on receiving a message in this way, is notified of the message arrival and of its subject on the next line of the screen if in monitor mode or at the bottom of the screen if using the E editor, unless a GAG command is in effect. If not logged-in, notification is given on the next log- in.

While it is possible to mail single-line messages without a subject listing, this format is more suitable for ei SEND messages, as noted below.

To mail a message with a subject, one first types MAIL/SUBJECT PRG *subject*<CR>, where PRG is the programmer name of the intended recipient and subject is a one or two word note as to the content of message. The MAIL program will then ask you to type the message, which must be terminated by *CONTROL*> <*META*><LINE>. In fact, if you simply type MAIL<CR>, the system will prompt you with requests for the name of the recipient, the names of other persons to receive copies, the subject and finally the message itself with a reminder to end your message with the usual termination of <*CONTROL*> <*META*> <LINE>. This is the so-called hand-holding mode, which is more verbose and time consuming but may be less confusing to use.

The normal Line-Editor editing commands may be used to edit the text as you are writing it. In fact, you can actually delete a <CR> and back up into an earlier line. An alternate procedure is, however, usually superior. One simply types <*CONTROL*> <*META*>E when the need to edit an earlier line arises. This causes a special MAIL$E.TXT file to be created and the text that you have written to be put into this file, with the command and destination information on one page and your text on another page. E is entered with the text page displayed and you can now edit the file (including the command page if necessary) with all the full power of E. This only works if you are already typing the message itself. If you are still typing the command line, the correct command to switch to E is <ALT>. When the file has been completed and edited to your satisfaction, type <*CONTROL*>XRUN<CR>. This exits from E, sends the message and then deletes the file. Should you exit from E without typing <*CONTROL*>XRUN<CR>, you can reenter the file with E in the normal fashion and then exit by typing <*CONTROL*>XRSYS MAIL< CR >. The message will be mailed and the file deleted as before.

The MAIL schemes so far described do not preserve a copy of your outgoing mail. You can, of course, copy yourself, but an even simplier way is available. If one has an OUTGO.MSG file on one's normal login directory, a copy of every outgoing mail message is automatically sent to this file. The /-OUTGO switch will override this provision for any specific message.

Mail may be sent to several recipients by appending the necessary number of PRGS separated by commas, or the recipients may be specified in a distribution file that is referred to by the at sign @. If the message is to be sent to all users, the asterisk (*) may be used.

The MAIL program may be evoked while in E and either the contents of the Attach Buffer

or the entire current page (or selected lines using an argument) may be mailed, by using the E command <*CONTROL*>XMAIL PRG<CR>. If you want to mention the subject, you must include a switch **(/SUBJECT)** and the form is <*CONTROL*>XMAIL/SUBJECT **PRG text of subject** <cr>.

The **SEND** command is similar to MAIL except that it sends the message directly to the logged-in recipient. Such messages are usually restricted to single-lines since the recipient may not be in a position to receive a multi-line message directly. If the designated recipient is not logged in, the sender is given the option of having the message mailed. The single-line format of the **SEND** command is:

SEND PRG message< **CR** >

where **PRG** is a programmer name and *message* is the one-line text you want sent.

The other MAIL related commands are all self-documenting and can be entered in the *hand*-holding mode by simply typing their names.

6.7 The DART program

DART is a program that saves disk files on magnetic tape and restores files from tape to disk. It is used to make periodic backups of the disk, and it is available to users who may find it necessary to restore a file that has been purged or otherwise lost from the disk. For full details **READDART.**

6.8 Keyboard and Display Mapping

The <EXC>#M command (with no <CR> and issued from a DD terminal), where # is a line number assigned to another DD terminal, maps one's terminal to the designated line, i.e. detaching one's terminal from its home line and attaching it in parallel with the terminal that is assigned to the designated line. Alternately, one can use the command <ESC>[PRG] to map to the line assigned to user **PRG.** When mapped, anything typed on the mapper's keyboard (with e few exceptions, such as the <ESC>M command) is accepted by the system as having been typed on the mappee's keyboard. The <ESC>M command returms the mappee's terminal to its home line.

Mapping to another's terminal without the user's knowledge, unless for good reason, is unethical (like reading papers on someone else's desk) and it is potentially dangerous since a false key stroke on the mapper's part can cause a lot of trouble. You should not try mapping until you have developed a communality of interest with other users that will justify such action. It is comforting, however, to know that an expert will be able to observe your display at another terminal when you need help.

It is possible to hide a DD terminal, when necessary, by the <ESC>H command. The normal custom is to resort to hiding only on rare occasions. The <BREAK>H command cancels the hide command.

6.9 Miscellaneous System Commands

A few of the many other system commands that you will soon want to be able to use might include: the UNDELETE command, with a format UNDELETE newfile←oldfile, that works only if the disk space has not yet been reassigned to another file, so try it right away; the **R SRCCOM** command that allows one to compare two source programs, and the corresponding, but less useful, **R BINCOM** for binary files. See the listings in the Appendices to get an idea of the many commands and services that are available.

7. Document Compilers

As mentioned earlier, Document Compilers accept source files and produce output files that can be sent to an output device to produce finished documents. Three such programs, **POX, PUB** and TEX, are generally used. Two dialects of TEX called MAXTEX and ARKTEX are favored by many.

Document compilers, by their very nature, are rather hard to use, at least if you want to do anything at all fancy. One way to get started is to slavishly copy some document source file as to overall format and simply replace the text by your own text. For example, if you like the format of the present manual, simply copy SUAI.TEX[TEX,ALS] into your own area under the desired name for your new file, save page 2, as is, and replace the text gradually with your own words, and then use MAXTEX as described below.

Files prepared for the **XGP** may be viewed on **DD** terminals by using the **R** XGPSYN<CR> program which is reasonably well self-documented.

7.1 POX

POX is the oldest of the available document generators. It generates an **XGP** file for the Xerox Graphics Printer and will produce justified text using different fonts. Some people still use **POX** for many of their document needs.

The principal shortcomings of **POX** are the lack of facilities for the automatic handling of such matters as Section, Footnote and Equation numberings and the automatic preparation of a Table of Contents. Confirmed users of the TEX program fault **POX** for its lack of finesse in taking care of the many subtle features that are involved in producing book-quality text.

For more information on **POX** type READ HOW<CR>. There is an **XGP** file that can be spooled on the **XGP** by typing XSP POX.XGP[UP,DOC]<CR>.

7.2 PUB

PUB is an advanced text justifier and page formatter intended primarily for use by programmers. It can automatically number pages, sections, figures, footnotes, etc. and can print their numbers in Roman numerals as well as in digit or letter form. It can generate cross references,

tables of contents, and indexes. Page layout is flexible, and allows multiple column output. Line formatting includes tabs, underlining, superscripts, subscripts, centering, and justification. Macros programmed in a SAIL-like string-processing language can generate text to be printed in the document. The output of the compiler is a file which can be printed on the terminal, on the line printer, or on the **XGP.**

A complete manual exists for this program and can be consulted on-line by typing **READ** PUB<CR>. Updates and corrections are available by typing **READ** PUB.UPD<CR>.

7.3 TEX

The TEX program is the most recent of the available document generating programs. We *can* do no more than to quote from the introduction in the Computer Science Dept. Report No. STAN-CS-78-675 entitled **TAU EPSILON CHI.** by Donald E. Knuth.

"This is a handbook about TEX, a *new* typesetting system intended for the creation of *beautiful* books-and especially for books that contain a *lot* of mathematics. By preparing a manuscript *in* TEX format, you will *be* telling a computer *exactly* how *the* manuscript is to be transformed into *pages* whose typographic quality is comparable to that of *the world's* finest printers; yet you won't *need to do much more work* than would *be* involved *if you were* simply typing the manuscript *on* an ordinary typewriter. In fact, your total work will probably be significantly less, *if* you consider the time *it* ordinarily takes *to revise* a typewritten *manuscript, since* computer text files *are* so easy *to* change and to reprocess."

If you cannot get a copy of this report you might buy a copy of **TEX** and **METAFONT** by D.E.Knuth, Digital Press (1979), available at the bookstore. **As** a last resort you might type **READ** MANUAL.TEX[TEX,DEK] to the monitor and try to read the source file from which the **TAU EPSILON CHI** report was prepared.

Two versions of TEX are currently available. The older version, called by typing R XGPTEX<CR>, prepares an output file for printing on the **XGP,** and the newer version, called by typing R TEX<CR>, prepares an output file for printing on the **DOVER.**

Both programs respond with an *, and they then expect a reply of \input FILNAM<CR>, where FILNAM is a file with the extension of **TEX** that contains text and all of the necessary instructions expected by TEX. The \ before the word input tells TEX that this is the symbol used in the manuscript as an escape symbol before commands. Actually any infrequently used keyboard character could be used, but the \ is the most common. It is used in some of the auxilary input files that you may find convenient, such as the file BASIC.TEX[1,3], which will load a basic set of commands. If you want to use this BASIC.TEX file, the first line in your file should read: \input **basic.**

Upon completion of the compilation, the necessary printing command is loaded into the monitor input buffer, only awaiting your typing of a terminating <CR> before execution.

MAXTEX and **ARKTEX**

Several dialects of TEX are being developed for the less-than-expert hacker. One preloaded version, called MAXTEX, has information on 42 fonts (named abc...z ABC...L<>@;) and uses macros to take care of such matters as: handling of chapters, sections, etc.; automatic table

of contents (as well as plates and figures); several styles of pages, with different headings; comments, annotations, footnotes; paragraphing details (enumerate, itemize, display, indent, etc.). Otherwise, it is used just as the ordinary TEX in the system. This manual was prepared using **MAXTEX.**

MAXTEX is called by typing R MAXTEX<CR> to the monitor. There are three files, KERMAC.TEX, PAPMAC.TEX and LETMAC.TEX, that take the place of BASIC.TEX and provide a more elaborate set of macros for special features. Documentation is somewhat frugal at the moment; **READ** MAXTEX for more details. Also see the MAXTEX page in HOW.ALS[UP,DOC].

ARKTEX is a macro package for use with the TEX processor. This package supports output of a variety of styles and formats. These formats include straight text with page numbers in one of several locations or without them, letter with Stanford letterheads, multi-column output, and book format output. The book format is the most developed format of this series and it supports chapters and sections that automatically count. Support for tables of contents and indexes is being improved at the time of this writing. For more information, **READ** ARKTEX.

A version of TEX that preloads the fonts used by ARKTEX is available as ARKTEX. To use it, type R ARKTEX instead of R TEX. Your document should begin with \input **ARKTEX** to obtain the latest version of the macro package.

8. Interfacing with other Systems

The **SUAI** computer is linked to the **ARPA** network, a facility organized by the Advanced Research Projects Agency of the Department of Defense to connect computers at various research centers funded by **ARPA,** allowing people at one host to use the resources of another host. The device which provides the interface between our computer and the network, called an IMP (Interface Message Processor), can bc used by user programs like any other I/O device. Two main system programs are provided for connecting to other computers by console commands: the user TELNET program and the File Transfer Protocol program **(FTP).** The former allows you to use your terminal as if it were a terminal of the remote host computer; the latter provides high-speed transmission of data between hosts. For more information **READ** TELNET and READ FTP.

A link is also in the process of being developed with the **ETHERNET.** For the current file transfer protocol, **READ EFTP.** Other reports are sure to be available in the near future, so ask someone about this matter.

9. UUOs and the UUO Manual

UUOs are monitor calls that make use of certain machine instruction codes that would otherwise be unused or illegal. Such calls are primarily of interest to those persons who program in Assembly languages, although it behooves all programmers, particularly those who use **SAIL** (or any other language that permits machine-code insertions) to at least know of the existence of such calls and of the UUO manual. This (276 page) manual is available on-line by typing **READ** UUO<CR>, and in hard copy as Stanford Artificial Intelligence Laboratory Operating

Note 55.5. Please do not- try to get a copy unless you really need it and do not under any circumstances try spooling it.

Some of the many matters handled by UUOs have to do with establishing input and output channels, setting up buffer rings, opening files, initializing input and output devices, establishing shared upper-memory segments so that more than one user can share a program or program segment, sending and receiving inter-job messages, the getting and setting of job information, operating the Dialer and operating video and audio switches.

Appendices

Appendix C

The following Monitor Commands are available on **SU-AI** WAITS9.12/d as assembled on 1/13/81. They may be truncated as shown by capitals. This list does not include programs that are run by an **R** FILNAM command (see Appendix R).

For a similar list with longer explanations **READ** CMDNAM<CR>. For more information try HELP <command name> <CR> and if no information is available type **READ MONCOM<CR>**. The Index is on page 144 of that file.

@	cross-reference list with ATSTGN	DFind	find word in dictionary
ACcess	access user file directory	DIAl	talk to comp. by phone
ADvance	advance mag tape	DIrectory	type file statistics
ALias	set disk PPN	DO	exec. commands from file
Assign	assign device to job	DOVer	print file on Dover
ATtach	attach job to terminal	DQavg	type disk queue average
BACkspace	back mag tape	DRd	run directory editor
BATch	schedule delayed execution	DSksiz	report disk use and quota
BOok	book mode edit with E	DTn	DM simulator TELNET
CAncel	delete reminders	DUmp	dump files to mag tape
CContinue	cont. job, term. in mon. mode	E	examine core image
CDetach	cont. job, detach terminal	EDit	edit file with SOS
CEtv	create file with E	EFInd	find string (E-format)
CFork	cont. job, detach term., new job	EFTp	transfer file over EtherNet
CKmail	see if user has new mail	ELfqfix	fix jammed ELF queue lock
CLruwp	clear upper seg, write protect	EOt	mag tape positioning
COMpile	compile program	ERead	E edit file readonly mode
CONtinue	cont. job, term. in user mode	ETv	edit file with E
COPy	copy files	EVent	list event in system messages
Core	set or type core size	EXecute	compile, load, and run prog.
CREAte	create file with SOS	FIles	type status of files in use
CREF	cross-reference list with CREF	FIND	find char. string in a file
CStart	start job, term. in mon. mode	FINGer	type name and loc. of users
DAytime	type time of day or job times	Finish	close and release device
DDt	enter DDT or RAID	FIXimlac	init. IMLAC terminal
DE	deposit into core image	FLush	clear term. buffers
Deassign	deassign device from job	FOrk	detach, make new job
DEBug	compile, load prog. start debug	FTP	ARPA net file transfer
DECide	ask computer for advice	Get	load core from file
DELete	delete files	GRipe	report system bug
DETach	detach iob from term.	HAlt	stop job (↑C)

HELLo	type name of curr. mon. version
HELP	explain system program
HOSt	find who given net host is
HOT	read AP or NYT news
KAttach	kill job, att. another
KIll	kill another job
Kjob	kill this job
KLog	kill this job, Login
LAter	run prog. later
LISP	run LISP
LIST	list file on line printer
LOAd	compile and load program
LOCate	loc. dump no. for file
Login	make a new job
LOGOut	kill this job
MAIl	send mail to a user
MAKe	create file with TECO
NOEdit	no edit key on terminal
NS	read AP or NYT stories
OFind	find string per OPTION.TXT
OTN	old TELNET protocol
PDetach	detach job, make it phantom
PJob	type job using device or term.
PLAn	create plan file
PLIst	list PUMPKIN requests
POX	compile doc. with POX
PPpn	print PPN of job
PREpare	compile, load with debug.
PROCess	execute command string
PROVe	run SYS:PROVE
PTy	type term. attached to job
PUB	compile doc. with PUB
PUMpkin	ask for off-line restore
Qspool	type spooler queues
R	run [1, 3] program
RCv	receive mail for user
READ	read system doc. with E
REAP	mark files for dump and del.
REASsign	reassign device
REEnter	start prog. at reenter addr.
REMind	create reminder message
REName	rename file or change prot.
RER	run prog. from SYS, give it text
RERUn	run prog. from DSK, give it text
RESEt	reset a job
RESOurces	type available resources
RESTore	restore file from tape
RETry	try again to send queued mail
REWind	rewind mag tape
RSl	reserve service level
RUn	run program from DMP file
SAve	save core image in file
SD	SUPDUP to some net host
SENd	send message to user's term.
SETUwp	set upper seg. write prot.
SLevel	type asigned service level
SPool	request line printer listing
SSave	save two-seg. core image
Start	start job, term. in user mode
SUpdup/SD	run SUPDUP program
SYstat	type system status
TAlk	talk to another terminal
TECo	edit file with TECO
TELnet/TN	talk to ARPAnet computer
TESt	run SYS:TEST.DMP
TIme	type runtime for job or system
TList	list files on dump tape
TN	TELNET to some net host
TRAnsfer	copy files, delete source
TRY	compile, load with debug., run
TTy	set terminal parameters
TUrkey	who last used DART tape?
TYpe	type a file
UDpufd	modify UDP file directory
UFd	modify DSK file directory
UNDelete	try to recover deleted file
UNPRotect	reduce protection of a file
UNPUmp	undo PUMPKIN requests
UNReap	undo REAP requests
UNSpool	delete spooler request
VERify	run SYS:VERIFY
WIIEN	see when PRG logged out
WHEre	type job info for PPN
Who	display system status
XGplist	list file on XGP
XPart	prepare file part for XGP spool
XSpool	request XGP listing
Zero	clear DTA, UDP, or DSK dir.

HELP resumés, services and RE;FILNAM references are available for:

A2E	COMBIN	EFUN	HOWBIG	MACLSP	PC	SIMPLE	TOSCIP
ACCESS	COMPIL	EMACS	HTSWTS	MACROS	PCHECK	SIX12	TTY
ACRONY	COPY	ESCAPE	HUNK	MACSYM	PCP	SIXSYS	TTYCMD
ADDINF	CRDIR	ETEACH	HYRS	MAGIC	PCREF	SL	TTYESC
ADS	CRE	ETV	IIKEY	MAIL	PDLOV	SLAC	TTY JAM
ADVENT	CREF	EVENT	IIPOX	MAP	PFORM	SLDEC	TTYSET
AIM248	CROSS	EXAMPL	ILISP	MARGIE	PIX	SLIBRY	TVFONT
AIMDOC	CRYPT	EXERCI	IMLAC	MARS	PK	SLR1	TYPE
AIQUAL	CRYPT0	EXT	IMLKEY	MAXTEX	PLAN	SNAIL	TYPREL
AIWORD	CSD	F	IMPSTA	MBOX	PLNR	SNOBOL	UDP
AL	CSET	FAIL	INDENT	MENTOR	POLL	SORT	UDPUFD
ALIAS	D	FASBOL	INDEX	MERGE	PONY	SOS	UFC
ARKTEX	DART	FCOPY	INFO	METAFO	POX	SOUP	U-FD
ARM	DDFONT	FCROX	IO SAIL	MF	POXOLD	SPASM	UNDELE
ARPA	DDHACK	FELT	JAMLIB	MICROS	PPK	SPEED	UNXGP
ARPANE	DDKEY	FILDMP	JARGON	MIDAS	PPSAV	SPELL	USEM11
ASSIGN	DDT	FILE	KALAH	MIX	PRLISP	SPIDER	USEMUS
ATSIGN	DFTP	FILES	KBDMAP	MLISP	PROGS	SPINDL	USERGD
ATTACH	DFTPCH	FILEX	KIMKEY	MLISP2	PROVE	SPOOL	wo
AUTISM	DILACKS	FIND	KJOB	MONCOM	PRUNE	SPSUB	VARIAN
AUTOLO	DIABLO	FINDAF	KL10	MSORT	PTY	SRCCHK	VERIFY
BAIL	DIAL	FINGER	KLDIF	MTRX	PTYJOB	SRCCOM	VISREV
BATCH	DIALNE	FIXPNT	KRL	MUSIC	PUB	SSORT	WHEN
BBOARD	DIET	FLOW	KWIC	NCOMPL	PUBMAC	STICKY	WHERE
BCOMPL	DIR	FMUNGE	L	NET	PURGE	SUDS	WHO
BCPL	DIRED	FNTCHR	LAKOFF	NETDOC	PURGER	SUPDUP	WHOLIN
BIBOP	DISPLA	FOL	LATER	NETWRK	Q	SUTIP	WHOPHN
BINCOM	DISPLY	FOLMRG	LAWS	NEWESC	RADIO	SWR	WILD
BKG	DLNFTP	FOLPRM	LCFMAN	NEW10	RAID	SWRFRM	WISE
BLAISE	DLNSFT	FONT	LEHRER	NEWLSP	RAIDUP	SYMBOL	WL
BLISS	DM	FREEFO	LEIBNZ	NEWPRV	RCV	SYSTEM	XGP
BLOOD	DMACS	FSCALE	LETTER	NEWS	RECAUX	TALK	XGPDVI
BLOOP	DMKEY	FTP	LIFE	NOEDIT	REDUCE	TAPE	XGPSYG
BOOK	DO	FUDGE2	LIFXGP	NOTICE	REMIND	TEACH	XGPSYN
BUNDLE	DOC	GENRAL	LINGO	NS	RESOLV	TECH2	XGPTYP
BUREAU	DOVER	GEOMED	LINK	OPTION	RETRY	TECO	XIP
CANCEL	DRAW	GO	LINK10	OTLSER	REVED	TECORD	XLATE
CANON	DRD	GPRINT	LISP	P	RIVEST	TELNET	XPART
CARDS	DSKSIZ	GRFX	LISP16	P2P	RLISP	TEMPER	XSPOOL
CCLOCK	DTLNET	GRIPE	LIST	PACK	RSL	TENDMP	YUMMY
CGOL	DTYPE	GRUMP	LISTS	PAM	RUNES	TENGWR	YUMYUM
CHESS	DVI	GUEST	LOA	PAS	SAIL	TERMS	Z80
CHRMAC	DVIXGP	FLADAM	LOADAV	PASCAL	SAVE	TEST2	ZERO
CKMAIL	E	HAR 1	LOGIN	PASDEF	SCHEME	TEX	370
CKSUM	ECL	HELIB	LOGOUT	PASDOC	SCIP	TFX	6500
CMDNAM	EDFONT	HELP	LRNMU1	PASHLP	SCRIBE	TIMES	6800
CNVR	EDITOR	HELPER	LRNMUS	PASINS	SD	TIP	68KASM
CODE	EFIND	HOST	LSPARC	PASMAN	SEND	TIPUG	6TO10
COLIST	EFTP	HOW	MACLIS	PASNOT	SFA	TMPCOR	8080

Appendix R

The following DMP files, on the [1,3] directory, may be run by an R FILNAM<CR> command. If HELP documentation is unavailable, many of these programs are still easy to used as they may contain self documentation or they may request the necessary input data interactively.

A	CKSUM	DTLNET	FREEFO	LOGRUN	PAS	REPENT	TEC	WAVES
ACRDEM	COLIST	DTN	FSCALE	LSPMON	PAS2	RESOLV	TECH2	WHEN
ACRONY	COMB1	DVIFTP	FSIM	MACLSP	PASCAL	RHY	TECO	WHERE
ACRPAP	COMBIN	DVIXGP	FSIM2	MACRO	PASOLD	RMS	TELNET	WHO
ACRROD	COMPLR	E	FTP	MAGGOT	PC	ROUTE	TEMPER	WILD
ACRSIM	CONVRT	EC	FUDGE2	MAIL	PCP	RPG	TENDMP	WISE
ADAM	COOL	ECL	FUNC	MAINSA	PCPLOT	RSEXEC	TENGWR	WL
ADH	COPY	EDFONT	GAL	MAP	PCREF	RSL	TEX	W-LA
ADS	CPRINT	EDIT	GEOMED	MAXTEX	PDRILL	RUNOFF	TEXDOC	WLABS
ADUDP	CRAM	EFIND	GLOB	META	PEDIT	S	TFDRD	WLNET
ADVENT	CRDIR	EFTP	GO	METAFO	PERUSE	S1LPT	TFTOPL	WLUMAP
AFPM	CRE	EMACLS	GOGAME	METER	PETAL	SAIL	TFXGEN	WM
AID	CREF	ENCODE	GREEN	MF	PFORM	SAMCMP	TFXPR	WOMBAT
AKRNIM	CROSS	ERAID	GRHOCK	MFRPVC	PICTUR	SAMPI,A	TIMES	X
AL	CRU1	ESPERA	GRNJOB	MFRSYN	PM	SCIP	TINGLE	XAP
ALIGN	CRU2	EX	GRUMP	MICROS	PIXUR	SCORE	TJ6PUB	XBIGMF
AMPSCL	CRYPT	EXB	HARRY	MIDAS	PK	SCRIBE	TMPCOR	XCOMPL
ARKTEX	CRYPT0	EXBFTP	HAUNT	MIXSCR	PLISP	SEGMEN	TSNAIL	XGPDVI
ARKXGP	CSTART	EXHALE	HELP	MIXSND	PLOT	SHIT	TSTJOB	XGPJOB
ARMDPY	D	EXL	HG	MLISP	PLTOTF	SHUFFL,	TTEX	XGPLOT
ATSIGN	DA	EXMRG	HOST	MLISP2	PLTPRE	SIMPLE	TTYSET	XGPMF
B	DAEMON	EXR	HQSTAB	MLISPC	PMFW	SLAC	TVFONT	XGPQUE
BASIC	DART	EXS	HOSTAT	MMFOL	POINTY	SLR1	TYPJOB	XGPSYG
BCOMPL	DAZZLE	F	HYPNO	MP	POLL	SMPLS	U	XGPSYN
BILLBD	DBCK	F40	IC	MPV	POX	SNAIL	UDPUSE	XGPTEX
BINCOM	DCHESS	FAIL	IIIGO	MS	PPK	SOB	UEDDT	XGPTYP
BLAISE	DCSTAT	FAILSA	IIIPOX	MSORT	PPL	SOLO	UNDOC	XIP
BLIS 10	DDFONT	FASBOL	IL	MTEX	PPSAV	SOS	UNPACK	XLATE
BLISS	DDHACK	FASM	IMPSTA	MTRX	PR	SPASM	UNPOX	XLD
BLOOP	DDJOB	FASM2	INDEX	MUS10	PREPOX	SPELL	UNXGP	XLISP
BLTST	DDT	FCOPY	INFO	MUS11	PRESS	SPIDER	V	XM
BRAVO	DDTLSP	FCROX	INHALE	MUSIC	PRIV	SPINDL	VAR2P	XPART
BRKSND	DDUSE	FED	INR	MYMF	PRN	SPOOL	VARIAN	xs
BUDGET	DEDIT	FILDMP	JMCWC1	NAME	PROFIL	SQUEEZ	VARL	xsv
BUNDLE	DFTP	FILEX	JMGSYN	NCOMPL	PROVE	SRCCHK	VCGEN	YH
BUREAU	DIABLO	FIND	JOTTO	NETWHO	PRUNE	SRCCOM	VCLIP	ZERO
BUZZ	DIAL	FINDAF	JUST	NEWLCF	PSEG	SSORT	VCWP	11FTP
CAL	DIET	FINE	KAFFEE	NEWMUS	PTEX	STEREO	VERCH	11TTY
CALLIT	DIGTIZ	FINGER	KALAH	NEWS	PTY	SUPDUP	VERIFY	12T
CAM	DIRED	FIRWRK	KRL	NPOX	PTYJOB	SURVEY	VIDXGQ	1LISP
CANCEL	DIZZY	FIXIM1	KWIC	OPUMA	PUB	SW	VLISP	3DFLY
CANTOR	DLNFTP	FIXIML	L	ORACLE	PUB2	SWR	VM	6TO10
CCLOCK	DLNSTA	FLTAPP	LESCAL	P	PUMA	SYNDMP	VM15	
CHAT	DM	FMUNGE	LIFE	PACK	Q	SYNJOB	VM22	
CHATER	DMCHK	FNTCHR	LIFXGP	PACKMS	RAID	SYNTER	VMR	
CHECKE	DO	FOL	LINK	PAGE	RCV	SYNTH	VNEW	
CHRFNT	DOCTOR	FOLISP	LINS	PAL	RDPY	SYSDWN	VPROVE	
CHRMAC	DPYHAX	FONT	LISP	PALX	READCN	TALK	VTEST	
CHRTST	DRAW	FONTCA	LOADAV	PAM	REDUCE	TAP	VWOLF	
CINEMA	DRW	FRAID	LOADER	PARRY2	RELFLY	TAPE	WAIT	

Notes

Index

Notes