AAEC/E580

AAEC/E580

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

UNED - AN INTERACTIVE TEXT EDITOR FOR IBM370 COMPUTERS

by

R.J. CAWLEY

September 1983

ISBN 0 642 59786 3

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

UNED - AN INTERACTIVE TEXT EDITOR FOR IBM370 COMPUTERS

by

R.J.Cawley

ABSTRACT

A powerful line-oriented text editor is described. The editor command set resembles that of the standard UNIX editor. The editor, which runs on any IBM370 machine under the MVS operating system, is simple to use and the user needs little or no knowledge of the operating system or job control language.

CONTENTS

1. INTRODUCTION

UNED is an interactive terminal-oriented facility that allows users to manipulate and alter text files (typically "card-images" on disk). It provides a simple way of doing all the things that can normally be done when handling a deck of punched cards. The advantage of UNED lies in the power, scope, and speed with which these familiar things can be done. For instance, changing every occurrence of a string, say "WRITE(7,40)" to "WRITE(8,40)", is specified concisely with a simple command:

```
*S/WRITE(7,40)/WRITE(8,40)/
```

and is completed in a matter of seconds, even for a file consisting of several thousand card images.

In addition, provided that the user builds a suitable jobcard and job control statements into the file, the file can be submitted to be run as a job, exactly analogous to feeding a card deck onto a card reader. The job submission facility also provides a powerful method of merging separate files into a single file during the job submission process.

The syntax of the editor commands is based upon that of the standard UNIX editor [Thompson and Ritchie 1975], and although somewhat terse is easily learnt and provides powerful editing capabilities. UNED is, however, much more friendly to the user than the UNIX editor, always querying the user when copious quantities (one line or more) of data may be destroyed, or not saved.

2. GETTING INTO UNED

On the AAEC's Lucas Heights Research Laboratories (LHRL) network, UNED may be started in a number of ways. The simplest method is described in the text; see Appendices B and C for other methods.

The user must first become known to the system. To do this the space bar on the terminal is pressed, and the computer responds with the prompt:

ID:

The user must then enter the account number, which the computer verifies. If the account number is incorrect, the above prompt will be reissued, otherwise the user is prompted by

$

The user should then type UNED followed by carriage return. This should start the editor. At this stage, UNED may print a news dataset at the user's terminal. Eventually UNED will prompt the user for instructions by displaying on a new line the single character "*". At this stage, the text buffer is empty.

3. COMMAND SYNTAX

Throughout this description, optional information is indicated by enclosing it within the characters "<" and ">":

<optional information>

All UNED commands have the same basic format. A command starts with an optional line range followed by a command mnemonic, which is optionally followed by additional arguments. The general format of a command is

<line range><command><command modifiers><arguments><option>

All fields (the "P" command is defaulted if the command is omitted) are optional. The various fields are as follows:

<line range>	specifies the range of lines to be operated upon;
<command>	specifies the operation to be performed;
<command modifiers>	special restrictions upon scope of command, e.g. restrict command action to a specific column;
<arguments>	data required to perform the command operation, e.g. original and replacement strings for string substitution;
<option>	further options, e.g. print result of command at the terminal.

The line range may consist of a single line number, two line numbers separated by a comma (or semicolon), or a default value if the line range is omitted. In line-oriented commands, the default value is usually the last line referenced, whereas in file-oriented commands, the default is usually all lines of the file.

3.1 Line Addressing

UNED is oriented around the basic unit called the line. Lines of text within the text buffer are addressed (identified) by their position relative to the beginning of the file. Thus the

first line is addressed as line number one, the second as two, and so on.

There are two ways to specify the address of a line. The first is simply to type the line number of the line of interest, the second is to specify the distance of that line from the current line. The user is always positioned at a particular line, referred to as the "current line". The current line has the name "." to allow it to be easily referenced. Whenever a command is given which is not preceded by a line address, the address of the current line is assumed.

A common way to address a line is by relative addressing. Any line address which begins with a "+" sign is assumed to begin ".+1". Similarly, lines beginning with a "-" sign are assumed to begin ".-1". Thus ".-7" (equivalently "-6") addresses seven lines before the current line, and "+4" (equivalently ".+5") addresses five lines after the current line. Another commonly addressed line is the last line of the text buffer which is named "$" for easy reference.

Addition does not need to be specified within an expression if the context is unambiguous. Thus ".3" is equivalent to ".+3", the "+" sign being implied. This of course is only possible when connecting the symbols "." and "$" with each other or with numeric values.

Line addresses are separated by commas or semicolons. If a semicolon is used, the current line is immediately set to the previous address (before the next address is interpreted). This feature is useful for simplifying complex address specifications.

UNED supports a number of abbreviations for commonly accessed lines and line ranges. The "page length" inherent in some of these abbreviations is defined as a line count, and may be defined

by the user (default is twenty lines). The following abbreviations are used:

.	the current line,
$	the last line of the text buffer,
-	the previous line,
+	the next line,
*	the entire text buffer (1,$),
/	the next line through to the last line (.+1,$),
\	the first line through to the current line (1,.),
!	the next "page" (.+1,.+n),
^	the previous "page" (.-n,.-1),
%	the "page" surrounding the current line (.-m,.+m),
&	the last non-trivial (not single line) line range referenced on an "L" or "P" command,

where n is the current page length, and m is half the current page length.

The abbreviated operators "+" and "-" may be concatenated to address several lines away from the current line, e.g.

+++ specifies .+3, and

---- specifies .-4.

4. COMMANDS

4.1 Adding Text

There are three commands ("A", "I", and "C") which allow lines of text to be added to the text buffer. Each of these commands place UNED into "input mode". When in input mode, text may be entered without its being interpreted as a command. After

entering all the text required, simply type carriage return to enter "command mode".

4.1.1 Append

The first command for adding text is "A" (append). This command allows one or more lines of text to be added to the buffer AFTER the line address specified. The format is

```
<line address>A<option>
<text>
<cr>
```

where

<line address>	specifies the line after which text will be added (default is "." and 0 means add at the beginning of the text file);
A	is the command mnemonic;
<option>	if present is the character "P" which requests UNED to prompt for each line of text input by printing a line number at the terminal; the character "L" may also be specified, and specifies no prompting for input lines (this is the default option);
<text>	represents one or more lines of text to be added to the text file, each line terminated by a carriage return;
<cr>	is an immediate carriage return (empty line) which forces a return to "command mode".

A special immediate form of the "A" command is available. The format is

<line address>A/string/<option>

This form adds the string as a new line after the specified line, and optionally prints the added line at the terminal according to option:

<option> = P preceded by line number,

<option> = L without line number.

Note that the text string is enclosed within the delimiter characters "/". The delimiter character can be any character which is not present in the text string to be added. This version of the command returns immediately to command mode.

4.1.2 Insert

The second command for adding text is "I" (insert). "I" allows one or more lines of text to be added to the text buffer BEFORE the line address specified. ("A" adds after the line address.) The format is

<line address>I<option>
<text>
<cr>

where the interpretation is similar to that of the "A" command, except that a line address of value zero is not permitted.

An immediate form of the "I" command is also available. The format is

<line address>I/string/<option>

The interpretation is similar to that of the immediate form of the

"A" command.

4.1.3 Change

The third command for text addition is "C" (change). This command allows one or more lines of text to be replaced with as many lines of text as the user desires. The format is

```
<line range>C<option>
<text>
<cr>
```

where

<line range>	specifies the range of lines to be replaced;
C	is the command mnemonic;
<option>	if present is the character "P" which requests UNED to prompt for each replacement line of text by printing a line number at the terminal.

If only a single line address is given, that line alone is replaced. If no line address is given, the current line is replaced.

An immediate form of the "C" command is available. The format is

<line range>C/string/<option>

where

/string/	is the text (enclosed within delimiters, here "/") of the single line which is to replace the given line range;
<option>	if present specifies the printing of the replaced line at the terminal as: <option> = P preceded by line number, <option> = L without line number.

The user is prompted for verification of the command prior to any text being deleted.

4.2 Deleting Lines of Text

The "D" (delete) command allows one or more lines of text to be deleted. The format is

<line range>D

where

<line range>	is the range of lines to be deleted;
D	is the command mnemonic.

If only one line address is specified, that line alone is deleted. If no line address is specified then the current line is deleted. The user is prompted for verification of the command before any text is deleted.

4.3 Moving Text

The "M" (move) command allows lines of text to be moved from one place to another. The format is

<line range>M<line address><option>

where

<line range>	is the range of lines to be moved;
M	is the command mnemonic;
<line address>	is the line after which the moved text is to be placed;
<option>	is one of the characters "P" or "L" which if present causes the moved text to be printed at the terminal, preceded by line numbers if "P".

4.4 Duplicating Lines of Text

The "T" (duplicate) command allows one or more lines of text to be duplicated. The format is

<line range>T<line address><option>

where

<line range>	is the range of lines to be duplicated - the default line range is simply the current line;
T	is the command mnemonic;

<line address> specifies the line after which the duplicated text is to be placed - the default is to place the duplicated text immediately after the current line;

<option> if present is one of the characters "P" or "L", and specifies that the duplicated text be printed at the terminal (preceded by line numbers if "P").

4.5 Changing Text

There are three commands ("S", "Z" and "V") which allow text within a line to be modified.

4.5.1 Substitute

The commands "S" (substitute) and "Z" (global substitute) allow a given text string to be replaced by another string. The format for the "S" command is

<line range>S/string1/string2/<option>

or

<line range>Sn/string1/string2/<option>

or

<line range>S/string1/string2/string3/<option>

or

<line range>Sn/string1/string2/string3/<option>

where

<line range>	is the range of lines to be modified;
S	is the command mnemonic;
/	is a string delimiter character, and may be any non-numeric character not contained within string1, string2, or the optional string3;
n	if present restricts the string match of string1 to column "n" of each line in the given range;
string1	is the character string to be replaced;
string2	is the replacement character string;
string3	if present must also be matched within a line before replacement of string1 by string2 is permitted;
<option>	if present is one of the characters "P" or "L" which specifies that each line modified is to be printed at the terminal (preceded by line number if "P").

The format for the "Z" command is identical to the "S" command, the difference in action being that for the "S" command; only the first occurrence of string1 on each line is replaced by string2, and, for the "Z" command, all occurrences of string1 on each line are replaced.

String1 and string2 are remembered by UNED until another "S" or "Z" command changes them. This feature is useful since UNED permits an abbreviated form of the "S" and "Z" commands:

<line range>S<option>

or

<line range>Sn<option>

where the last used values of string1 and string2 are implied.

4.5.2 Overwrite

The "V" (overwrite) command allows text beginning at a nominated column to be overwritten (replaced) by another string. The format of the "V" command is

<line range>Vn/string/<option>

or

<line range>V<option>

where

<line range>	is the range of lines to be modified;
V	is the command mnemonic;
n	is the column number at which the new string will be placed;
/	is a string delimiter character, and may be any non-numeric character not contained within string;
string	is the new character string;

<option> if present is one of the characters "P" or "L" which specifies that each line modified is to be printed at the terminal (preceded by line number if "P").

If only a single line address is given then that line alone is modified. If no line range is given, the current line is modified. If the column number is not specified then the column number last specified in a "V" command is used. If no string is specified then the string last specified in a "V" command is used.

4.6 Finding Text Strings

The "G" (go to string) command provides a fast method of finding occurrences of a nominated text string. The format is

<line range>G/string/<option>

or

<line range>G#/string/<option>

or

<line range>Gn/string/<option>

or

<line range>G#n/string/<option>

or

<line range>G*/string/<option>

or

<line range>G*#/string/<option>

or

<line range>G*n/string/<option>

or

<line range>G*#n/string/<option>

where

<line range>	is the range of lines to be searched - default is (.+1,$);
G	is the command mnemonic;
n	if present is a column number which specifies that the string match must occur at column "n" of a line;
/	is a string delimiter character, which may be any non-numeric character not present in string;
string	is the character string to be found;
<option>	if present is one of the characters "P" or "L" which specifies that the line containing the located string be printed at the terminal (preceded by line number if "P");
*	if present implies option "P" (if no option is present), and causes all lines (which lie within the given line range) containing the given string, to be printed at the terminal;

#	if present causes UNED to search for lines not containing the given string.

Note that if the command modifier * is not specified, the search terminates at the first line found to contain the string (at the specified column number).

If no line range is specified, the text file is searched from the line following the current line until the end of the text file.

4.7 Saving and Retrieving Text

Having created a file of text the user will probably want to save it for later use, and possibly for further editing sessions. UNED allows the text to be saved on bulk storage (usually disk) files which may optionally be organised as partitioned dataset libraries or as normal sequential datasets. Before the text file can be saved (or later retrieved), the dataset must somehow be connected (allocated for use) to UNED. The "OA" and "OD" commands provide a means of achieving this connection.

The "W" (write) command is used to save the text file on bulk storage. The "R" (read) and "E" (edit) commands are used to retrieve files from bulk storage.

4.7.1 Allocating datasets

The "OA" and "OD" commands provide a means for the user to allocate (connect) files on bulk storage for use by UNED. The format is

OA/ddname/datasetname/status/

or

OD/ddname/datasetname/status/

where

OA	is the command;
OD	is the command;
/	is a delimiter character used to delimit the argument fields, and may be any character not present within any of the arguments;
ddname	is a DDNAME (see MVS job control language specifications [IBM 1979]) by which the user will refer to the dataset from within UNED. This name consists of a character string of up to eight characters in length, legal characters being numeric digits and upper case alphabetic characters, the first character being alphabetic - alternatively a numeric value in the range 1 through 20 may be specified implying a ddname of form "SFXX" where "XX" is the specified number;
datasetname	is the actual name of the dataset, rules for which must follow MVS job control language specifications;

status	defines the type of access rights required; valid values for status are OLD for exclusive use, SHR for shared use, MOD for extending a sequential dataset, and NEW if a new dataset is to be created (MOD and NEW imply exclusive use) but for non-sequential datasets (libraries), only OLD, SHR, and NEW are valid.

The "OA" command is used to allocate the file (or library), and the "OD" command is used both to allocate a library of files and to convert the allocated library to the current library.

If a library has already been allocated in some fashion (probably by an earlier "OA" or "OD" command, then the "OD" command may be used to change the current library. An abbreviated format is used to do this:

OD/ddname/

This makes the library referred to by "ddname" the current library.

Another use of the "OD" command is to list all files and libraries currently allocated to UNED. The format is

OD ?

4.7.2 Saving text

The "W" command is used to save the edited text file permanently. The format is

<line range>W/member/<option>

or

<line range>W/member:comment/<option>

or

<line range>W/member/ddname/<option>

or

<line range>W/member:comment/ddname/<option>

or

<line range>W<option>

or

<line range>W//ddname/<option>

or

<line range>Wn<option>

or

<line range>W0

where

<line range> is the range of lines to be saved;
W is the command mnemonic;

/	is a string delimiter character, which may be any non-numeric not present in member, comment, or ddname;
member	is the library member name which will be used to refer to the saved text;
:	indicates that a comment, associated with member, is to be stored in the library directory;
ddname	is the name by which the user refers to the required library dataset (see Section 4.7.1);
<option>	if present is the character "P", which causes the output text lines to be sequentially numbered (in steps of one) in columns 73 through 80;
n	is the number of the sequential dataset where the text is to be saved ("n" is the number "XX" in ddnames of the form "SFXX"; see Section 4.7.1).

If only one line address is given, then that line alone is saved. If no line address is given, then the entire text file is saved. If member is not specified, then the "current member name" is used. The current member name is initially set to "????????", and is changed to the specified member name whenever an "E" command is issued, or whenever the entire text file is saved with the "W" command. The current member name reverts to "????????" whenever the text file is empty.

4.7.3 Retrieving text

The "E" and "R" commands are used to retrieve text files for further editing. The difference in the two commands is that the "E" command completely erases the current text buffer before retrieving the nominated file, whereas the "R" command appends the

nominated file after the nominated line of the current text buffer. The format is

E/member/<option>

or

E/member/ddname/<option>

or

E//ddname/<option>

or

E/:/<option>

or

E/:/ddname/<option>

or

En<option>

or

E0<option>

or

<line address>R/member/<option>

or

<line address>R/member/ddname/<option>

or

<line address>R//ddname/<option>

or

<line address>R/:/<option>

or

<line address>R/:/ddname/<option>

or

<line address>Rn<option>

or

<line address>R0<option>

where

E	is the command mnemonic;
/	is a string delimiter character, which may be any non-numeric character not present in member or ddname;
member	is the library member name of the file to be retrieved - if the specified member name is ":" then the entire contents of the library directory is retrieved and formatted;

ddname	is the name by which the user refers to the required library dataset (see Section 4.7.1);
<option>	if present is either the character "P" or "L", which causes the text file to be listed on the terminal while being retrieved (preceded by line numbers if "P") - the listing action may be terminated at any time by the user typing "?" at the terminal (or by pressing the "PA1" key if running UNED under TSO at a 3270 terminal);
:	specifies that library directory information is to be loaded into the text buffer;
n	is the unit number of a sequential dataset from which a text file is to be retrieved ("n" is the number "XX" in ddnames of the form "SFXX"; see Section 4.7.1);
0	specifies that the text of the user's profile is to be loaded into the text buffer;
R	is the command mnemonic.

A line address of zero is valid for the "R" command. If no line address is specified for the "R" command, the current line is assumed.

4.8 Listing Text

The "P" (print) and "L" (list) commands are used to list part or all of the current text buffer at the terminal. The format is

<line range>

or

<line range>P

or

<line range>L

where

<line range>	is the range of lines to be listed;
P	is the command mnemonic;
L	is the command mnemonic.

If the command mnemonic is omitted, the "P" command is assumed. If the line range is omitted, the current line is assumed. If only one line address is given, that line alone is listed. The special case of this command is the degenerate form in which nothing at all is specified (i.e. simply a carriage return is entered); it causes the "P" command to be assumed and the line after the current line to be listed.

4.9 Obtaining Hard Copy

The "H" (hard copy) command is used to obtain line printer listings of part or all of the current text buffer. The format is

<line range>H<option>

or

<line range>H/string/<option>

where

<line range>	is the range of lines to be listed;
H	is the command mnemonic;
<option>	if present is the character "P" which causes the line number to be listed with each line;
/	is a string delimiter character which may be any character not present in string;
string	is a text string which will be listed as a heading before the specified line range.

If no line range is specified, the entire text buffer is listed. If a null string is specified, a blank line is printed as the heading. If no string is specified, a heading consisting of the library dataset name and member name (or sequential dataset name) from which the current text buffer was loaded, and the current comment is printed. In this case, the string "***MODIFIED***" will be appended to the generated heading if the text buffer has been modified since loading, unless the entire text buffer has been saved using the "W" command (and not modified since).

4.10 Punched Card Output

The "U" command is used to unload the current text buffer to punched cards. The output is always the first 80 characters of each line of text (or 72 if sequence numbering is specified). The format is

<line range>U<option>

where

<line range>	is the range of lines to be punched;
U	is the command mnemonic;
<option>	if present is the character "P" which causes the punched cards to be sequence numbered in columns 73 through 80 (sequence numbers start at 1 and are incremented in steps of 1).

If no line range is specified, the entire text buffer is punched.

4.11 Submitting Jobs

The "J" (job submission) command allows the work file to be submitted to the IBM3033 as a batch job. The format is

J

or

J<option>

where

J	is the command mnemonic;
<option>	if present is the character "P", which causes the cards of the submitted job to be sequence numbered in columns 73 through 80 (the sequence numbers start at 1 and are incremented in steps of 1).

The entire text buffer is submitted. In addition, lines of the work file which begin with the character sequence "()()" and have

one of the following formats:

()()member

or

()()member ddname

or

()()member/ddname

cause that line to be replaced (in the submitted deck) by the indicated file, where

member is the member name of the file in a library;

ddname is the name by which the user refers to the required library dataset (see Section 4.7.1);

and the current library dataset is assumed if ddname is not present. This inclusion process may be nested to five levels (i.e. the replacement file may also contain lines beginning "()()" which are also interpreted in the same manner). If the "P" option was specified, included files are individually sequence numbered (in columns 74 through 80) and the inclusion nesting level is indicated by a digit placed in column 73 of each line of the file.

4.12 Sequence Numbering

The "N" (number) command allows general sequence numbering of lines in the text buffer. The sequence numbers are always placed in columns 73 through 80. The format is

<line range>Ni,s

where

<line range>	is the range of lines to be affected;
N	is the command mnemonic;
i	is the numbering increment (default 1);
s	is the starting number (default i).

If "I" (numbering increment) is zero, columns 73 through 80 of the given line range are replaced with the character blank.

4.13 Commenting Assembler Source

The "X" command is normally used to put comments into an ASSEMBLER language source file. The format is

<line range>Xm<option>

or

<line range>X/string/<option>

where

<line range>	is the range of lines to be commented;
X	is the command mnemonic;
m	is the column number at which the comment field begins;

<option>	if present is the character "P", which causes the line number of each statement to be listed;
/	is a string delimiter character, which may be any non-numeric character not present within "string";
string	may only be present if "m" is not specified, and causes the text "string" to determine the value "m" (for each individual line) by matching "string" in each line; "m" then becomes the column number of the first character of the matched string, and lines in which no match occurs are skipped.

For each line of the given range:

(a) Columns 1 through comment column minus 1 are printed at the terminal.

(b) The user may now type the new comment (this also destroys the contents of columns 73 through 80, i.e. sequence numbers are destroyed).
An immediate carriage return will leave the existing comment unchanged. Premature exit from the command may be effected at this point by typing the character "." followed by carriage return. This will not alter the existing comment.

If no line range is given, the current line is assumed. The default column number ("m") is initially 40 and thereafter is the last value specified.

4.14 Library Maintenance

The "F" (file control) command allows the user to rename and delete members from the current library dataset, as well as inspecting the directory contents. To list the contents of the library directory, the following formats are used:

F	will list the names of members, contained within the library, at the terminal;
FP	will list the names of members and corresponding comments at the terminal.

The following format is used to delete a member from the current library:

F/member/

where

F	is the command mnemonic;
/	is a string delimiter character, and may be any character not present within "member";
member	is the name of the member to be deleted.

UNED will prompt the user for verification of the deletion, at which time the user may decide whether to proceed with the deletion.

To rename a library member the following format is used:

F/old/new/

where

F	is the command mnemonic;
/	is a string delimiter character, and may be any character not present within "old" or "new";
old	is the current name of the member;
new	is the new name desired for the member.

UNED will prompt the user for verification of the rename, at which time the user may opt whether to proceed with the rename.

4.15 Date and Time

The "K" command is used to find the present date and time. The format is simply

K

The current date and time are printed at the terminal.

4.16 Help

The "?" (help) command is used to provide a memory jogging listing of command formats at the terminal. The listing is rather terse but gives the complete syntactic variations for each command. The format is

?	provides a summary of line range abbreviations, and all commands;
?c	provides a syntax summary of command "c".

4.17 Terminating the Edit Session

The "Q" (quit) command terminates the edit session. The format is simply

Q

If any modifications have been made to the text buffer since it was last saved (using the "W" command), the user will be prompted to verify the session termination.

5. USER PROFILE

The user may specify a standard set of UNED commands to be executed automatically each time the user logs into UNED. (UNED usually determines the user's identity from the system, but if this cannot be done the user is initially prompted for the account number by issuing an "ID:" prompt.) This set of commands is known as the user's profile. Each command is listed at the terminal (preceded by "++") as it is executed.

The user sets up the profile by simply creating a text file consisting of normal UNED commands, one command per line. Having created this file of commands, the user should save the profile by issuing the command

WO

If, at some future time, the user wishes to change the profile, it may be loaded into the text buffer by issuing the command

EO

and any UNED commands may be used to edit it. When satisfied with

this new profile, the user simply saves it again by issuing the command

WO

This is a useful feature which allows UNED to set up the user's favourite editing environment automatically; it includes working length, alphabetic case switch, and library allocation.

6. COMMENT COMMAND

Any input command line which has the character "#" in column 1 is treated as a comment line and thus ignored.

A special case exists while the user profile is being processed. Here, a command line which has the characters "##" in columns 1 and 2 is assumed to be a comment, and not printed at the terminal. This is useful for documenting the user profile, without the documentation being printed each time the user signs on to UNED.

7. DATASET FORMATS

UNED allows the user to operate upon datasets structured in a variety of standard IBM formats. In addition, a special compressed format is available for library datasets.

The following record formats are acceptable for both sequential and library files:

RECFM=FB, RECFM=FBA, RECFM=FBM, RECFM=F, RECFM=FA, RECFM=FM

See the IBM job control [IBM 1979], and data management services

[IBM 1976] manuals for descriptions of these record formats.

The special compressed library format is triggered by record format RECFM=VB. Use of this format results in considerable space saving for direct access devices (up to 70% for typical ASSEMBLER or FORTRAN source files).

Note: This prevents the use of RECFM=VB datasets without the compression option; in fact, UNED uses a non-standard blocking method and internally spans records across blocks. Thus UNED will not even correctly read RECFM=VB datasets created by IBM utilities.

7.1 Compressed Library Format

The compression algorithm restricts the character set to EBCDIC characters with codes greater than hexadecimal 40. This is quite acceptable for most text files as the restricted character set includes all the printable characters, only some of the more esoteric control characters being outlawed. Second, any outlawed character detected in the text is replaced with the space character in the output. Further, the algorithm assumes that the user is dealing with 80 byte card images, so compressed files are not applicable to other logical record sizes.

The character codes below hexadecimal 40 are used as repeat counts. Within a logical record (card image) a string consisting of a single character repeated any number of times is replaced by a single occurrence of that character followed by a repeat count indicating the number of times that the character is repeated. If more than 64 repeats are required, a repeat count of 63 (hexadecimal 3f) is used followed by a second repeat count.

This replacement technique usually results in a shorter output logical record, thus using less secondary storage. Of course the most common character string to be compressed in this way is a string of space characters.

The compression process is reversed on input, resulting in a sequence of 80 byte card images.

8. MACRO FACILITY

UNED provides a macro facility whereby a number of commands may be automatically executed in sequence. The list of commands may be executed any specified number of times, and the current line may be repositioned before the first execution. Normally, if a command (such as substitute) fails, the macro execution is terminated.

The "E" command format, consisting of the single character "E", has a special property within the context of macro execution. Instead of causing the last mentioned library file to be loaded into the text buffer, the issue of this command within a macro causes the next library member (determined by the EBCDIC collating sequence) from the current library to be loaded.

The "B" command is used to control macro definition and execution.

Macro command lists may be entered directly from the terminal, or loaded from library datasets. Each command line within a macro is limited to 72 columns, and continuation lines are not permitted. The individual commands within the macro appear as normal UNED commands.

8.1 Loading the Macro Buffer

Two methods of loading the macro buffer are available.

The command

B

is issued by the user, and UNED responds by prompting

-

The user then enters any UNED command, which is stored within the macro buffer but not immediately executed. UNED then issues the above prompt again. Commands entered in this way are stored consecutively within the macro buffer until the user responds to the "-" prompt with an immediate carriage return, which terminates macro input. The command

B/member/

or

B/member/ddname/

causes the specified "member" to be loaded into the macro buffer from either the current library or the library specified by "ddname" where

member	is the member name of the required library file;
ddname	is the name by which the user refers to the required library dataset.

Note: Only the first 72 columns of each line read from the library file are stored within the macro buffer. Comment lines are not loaded into the macro buffer.

8.2 Listing Contents of the Macro Buffer

The command

BP

causes the contents of the macro buffer to be printed at the user's terminal.

8.3 Macro Execution

The command

<line number>Bn<option>

causes the current macro to be executed, and the command

<line number>Bn/member/<option>

or

<line number>Bn/member/ddname/<option>

causes a macro buffer to be loaded from a library, and the macro to be executed, where

<line number>	is the line number to which the current line value is set before beginning macro execution;
B	is the command mnemonic;

n	is a number which specifies the number of times that the macro command list will be executed;
member	is the library member name of the file to be loaded into the macro buffer;
ddname	is the name by which the user refers to the required library dataset (see Section 4.7.1);
<option>	if present is either the character "P" or "L"; option "P" causes each command within the macro to be listed at the user's terminal immediately before execution; option "L" causes most errors encountered to be ignored (i.e. macro execution continues after such errors) and most error messages to be suppressed.

Macro execution will be terminated if the user enters an unsolicited character at the terminal, or if a "G", "S", or "Z" command fails. Macro execution is also terminated if the single character format of the "E" command encounters the end of the library directory.

8.4 Extended Command Set

A number of commands are available to enhance the programming capabilty of macros. These commands allow for branching within the macro, and the creation, manipulation, and use of string variables.

8.4.1 Labels and branching

A macro command line may be tagged with a label by the following command:

@:LABEL command

where

@:	indicates label definition;
LABEL	is the label, which should be alphanumeric, and a maximum of eight characters;
command	is any UNED command, including and extended command (but not another label definition); if no command is present, the label is defined only (i.e. the default "P" command is not implied).

A branch to a labelled line is effected via the "@GOTO" command as follows:

@GOTO:LABEL

where LABEL is the label used to identify the required command line.

A group of macro lines may be executed as a subroutine within the macro by issuing the command:

@CALL:LABEL

where

@CALL	is the command;
LABEL	is the label identifying the subroutine.

The following command is used to return from a subroutine:

@RETURN

8.4.2 String variables

A string variable is a named entity which contains a character string (the null string is valid). These variables are identified by unique names of up to eight characters. A maximum of 100 variables may exist at any time. Each variable may contain up to 72 characters. The commands used to define and manipulate these variables are as follows:

@SCLR	clears definitions and memory of all "string" variables;
@SD/name/text/	defines or redefines variable "name", and "text" becomes the value;
@SCAT/name1/name2/name3/..	the value of variable "name1" becomes the concatenation of the values of variables "name1", "name2", "name3", etc.;
@SSET/name1/name2/name3/..	the value of variable "name1" becomes the concatenation of the values of variables "name2", "name3", etc.;

@SQ/name/text/	the string "text" is output to the terminal as a query, and the user's reply becomes the value of the variable "name";
@SQS/name/text/	the string "text" is ouput to the terminal as a query, and the user's reply (not echoed on the terminal) becomes the value of the variable "name";
@SMEM/name/	the "current member" name becomes the value of the variable "name";
@SDSN/name/	the name of the library (dataset) from which the current member was loaded becomes the value of the variable "name";
@SLIB/name/	the DDNAME of the current library becomes the value of the variable "name";
@SCOM/name/	the "current member comment" becomes the value of the string variable "name";
@SLDOT/name/	the text representation of the current line number becomes the value of the variable "name";

@SLEND/name/	the text representation of the last line number ($) becomes the value of the variable "name";
<line>@SWRK/name/	the contents of the given work area line are copied to the string variable "name" (trailing blanks are removed);
@SM/name/	the value of the string "name" is written to the terminal as a message;
@SXCT/name/	the value of the variable "name" is executed as a UNED command.

8.4.3 Immediate message

The command "@MT" is used to issue a message to the terminal, and has the format

@MT/text/

where "text" is the message to be printed at the terminal.

8.4.4 Execute line of text

The command "@WXCT" is used to execute a line of the work file as a UNED command, and has the format

<line>@WXCT

where "<line>" is the line number to be executed. If no line

number is specified then the current line is assumed.

8.4.5 Query and act

The following commands allow immediate action as the result of the user's reply to a query:

@QBY/text/@:LABEL	the string "text" is written to the terminal as a prompt. If the user's reply begins with the character "y", a branch to LABEL is taken;
@QER/text/	the string "text" is written to the terminal as a prompt; the user's reply is executed as a UNED command;
@QECY/text/command/	the string "text" is written to the terminal as a prompt; if the user's reply begins with the character "y", "command" is executed; the "command" is any UNED command.

8.4.6 Conditional command execution

The following commands are available for testing various conditions and conditionally executing a command (which may be another condition testing command or a normal UNED command) if the condition is satisfied.

Note: it may be necessary to mask some or all of the assumed macro error termination conditions to use some of these condition testing commands. (See the "OB" commands described in Section

9.1.)

The condition testing commands are as follows:

@IF(S)command	execute "command" if the last "S" or "Z" command was successful;
@IF(!S)command	execute "command" if the last "S" or "Z" command failed;
@IF(G)command	execute "command" if the last "G" command was successful;
@IF(!G)command	execute "command" if the last "G" command failed;
@IF(M)command	execute "command" if the text file has been modified;
@IF(!M)command	execute "command" if the work file has not been modified;
@IF($)command	execute "command" if the last line exists (i.e. the work file contains some text);

@IF(!$)command	execute "command" if the last line does not exist (i.e. the work file is empty);
@IFSS(=/nm1/nm2/)command	execute "command" if the value of string variable "nm1" equals the value of string variable "nm2";
@IFSS(#/nm1/nm2/)command	execute "command" if the value of string variable "nm1" does not equal the value of string variable "nm2";
@IFSS(</nm1/nm2/)command	execute "command" if the value of string variable "nm1" is less than the value of string variable "nm2";
@IFSS(>/nm1/nm2/)command	execute "command" if the value of string variable "nm1" is greater than the value of string variable "nm2";
@IFST(=/nm/text/)command	execute "command" if the value of string variable "nm" equals the character string "text";

@IFST(#/nm/text/)command	execute "command" if the value of string variable "nm" does not equal the character string "text";
@IFST(</nm/text/)command	execute "command" if the value of string variable "nm" is less than the character string "text";
@IFST(>/nm/text/)command	execute "command" if the value of string variable "nm" is greater than the character string "text";
n@IFL(=exprsn)command	execute "command" if line number "n" equals the value of the numeric expression "exprsn";
n@IFL(#exprsn)command	execute "command" if line number "n" does not equal the value of the numeric expression "exprsn";
n@IFL(<exprsn)command	execute "command" if line number "n" is less than the value of the numeric expression "exprsn";
n@IFL(>exprsn)command	execute "command" if line number "n" is greater than the value of the numeric expression "exprsn";

n1,n2@IFR(=ex1,ex2)command	execute "command" if the line range (n1,n2) equals the line range (ex1,ex2) where "ex1" and "ex2" are numeric expressions;
n1,n2@IFR(#ex1,ex2)command	execute "command" if the line range (n1,n2) does not equal the line range (ex1,ex2) where "ex1" and "ex2" are numeric expressions.

8.4.7 Character translation

The following commands are used to convert text files from the EBCDIC code and either the ASCII or INIS code, and vice-versa.

n1,n2@TAE	translate text within the line range (n1,n2) from ASCII to EBCDIC;
n1,n2@TIE	translate text within the line range (n1,n2) from INIS to EBCDIC;
n1,n2@TEA	translate text within the line range (n1,n2) from EBCDIC to ASCII;
n1,n2@TEI	translate text within the line range (n1,n2) from EBCDIC to INIS.

8.4.8 Current line adjustment

The command

@NL

adds one to the value of the current line, without printing the new current line.

8.4.9 Macro termination

A macro terminates when the macro command list is exhausted or when the "@QM" command is executed. The format of the quit macro command is

@QM

9. EDITOR OPTIONS

Although default values are provided for all editor options, the user may sometimes need to set up a nonstandard editor environment, and the "O" command and its several derivatives are provided to allow the user to directly set any desired option values. Two uses of the "O" command have already been discussed, namely the "OA" and "OD" commands used to allocate datasets.

The format of the "O" command is

O<x><parameters>

where

O	is the command mnemonic;
<x>	is an optional character selected from the set (A B C D E F H L M P S T U W Y Z) which specifies which particular option is to be set;
<parameters>	is the parameter value (or values) used to set the selected option - this argument is only valid if the <x> argument is present.

Another form of the "O" command is

O?

which causes the current values of all options to be printed at the user's terminal.

When the command is issued using the single character form

O

UNED will prompt the user for values of each option in turn. An immediate carriage return reply to any of these prompts will leave the indicated option unchanged (if a valid value currently exists) and the user will be prompted for the next option. If the user replies to any of these prompts with the single character "S" followed by a carriage return, the indicated option will remain unchanged, and the remainder of the option prompts will be skipped.

As indicated in the "O" command format described above, the user may directly set values for a particular option by issuing a single line command, no prompting being done by UNED unless the

given option value (or values) is in error. The various options, and the single command forms of setting them, are described below (except for "OA" and "OD", allocating datasets, which have already been described).

9.1 Macro Execution Masks

The "OB" command is used to set and reset a number of masks which control conditions for premature termination of macro execution. These masks (each referred to by a single character) are as follows:

- G prevents macro termination on "G" command failure - initially not masked;
- S prevents macro termination on "S" or "Z" command failure - initially not masked;
- E prevents macro termination at end of library directory when using "E" command in single character format within a macro - initially not masked;
- M suppresses informational messages relating to "S","Z" and "G" commands - initially not masked.

The command format is

OBYmasks

or

OBNmasks

where

OB	is the command mnemonic;
Y	indicates mask set function;
N	indicates mask reset function;
masks	is a list of one character mask identifiers, as in the above table.

9.2 Alphabetic Case

By default, UNED folds all alphabetic input to upper case, which is desirable for editing ASSEMBLER or FORTRAN source files. The user may however wish to edit various types of text files in which lower case alphabetics are acceptable. The "OC" command is used to specify whether UNED will fold alphabetic input. The format is

OCL	specifies lower case input is permitted;
OCU	specifies that lower case input is to be folded to upper case.

9.3 Encryption

Two encryption methods are provided for the user to disguise text files. Both methods provide for decryption of input files, encryption of output files and encryption of the work file.

9.3.1 Simple encryption

This encryption technique provides a relatively simple (easy to attack) form of encrypting the user's data, and is only retained for historical reasons. The user specifies (or cancels) this technique by issuing the "OE" command. UNED then prompts with

INPUT WEAKLY ENCRYPTED (Y/N):

The user should now type "Y" (yes), or "N" (no) followed by carriage return. If the reply is "Y", the user is prompted with

ENTER KEY:

The user may now enter a character string up to 32 characters in length from which an encryption key is constructed. UNED then prompts with

OUTPUT WEAKLY ENCRYPTED (Y/N):

Again the user should type "Y" or "N", followed by carriage return. If the reply is "Y", the user is then prompted with

ENTER KEY:

The user may now enter a character string of up to 32 characters in length, as above. If an immediate carriage return is entered after this prompt, UNED checks if simple encryption has been specified for input and, if so, duplicates the input encryption key, otherwise the user is again prompted on whether output encryption is required.

UNED now issues the prompt

WORK FILE ENCRYPTED WEAKLY (Y/N):

This is followed by a similar dialogue to the above, except that a null string specification following the

ENTER KEY:

prompt causes UNED to check encryption specification for input and

output, and to duplicate the key given for input if valid, or else for output; if neither, it reissues the prompt for work file encryption specification.

9.3.2 Strong encryption

This is the recommended encryption method to disguise text files. Again, strings of up to 32 characters are entered to specify the encryption key. The specified character string is hashed to form an 8-byte key which is then used as the basis for a block encryption scheme; a key modification and chaining scheme similar to that used by the IBM Information Protection System (IPS) cryptographic programs [Konheim et al. 1980].

The user specifies this type of encryption by issuing the "OZ" command and answering the prompts in a manner similar to the dialogue following the "OE" command.

An alternative form of the "OZ" command (useful within macros) is

OZ/si/so/sw/

where

si	is the input encryption key:
so	is the output encryption key;
sw	is the work file encryption key.

9.4 Substitution and Search Range

The commands "OF" and "OL" are used to set the first and last substitute columns respectively. The effect of this is to restrict the range of text searched and possibly modified by the "G", "S" and "Z" commands to be only the text between the first

and last substitute column inclusive of each line inspected. The default values are column 1 for first substitute column, and column 72 for last substitute column. The command format is

OFn

or

OLn

where n is the column number required. Note that UNED will not allow the last substitute column to precede the first substitute column, nor will UNED allow either value to exceed the current working length (see Section 9.10).

9.5 Hard Copy Control

Controls are available to specify remote work station routing for hard copy print output, beginning a new output page for each "H" commissued, and to allow the user to provide ANSI (American National Standards Institute) printer control characters for the printer output, and to specify a specific print train image (UCS i.e. universal character set). The defaults are

Print destination	Local
Pagination	Enabled
ANSI control	Disabled
Print train	System default

The "OH" command is used to set these options. The format is

OH/dest/

or

OH/dest/optl/

or

OH/dest/opt1/opt2/

or

OH/dest/opt1/opt2/ucs/

or

OH//opt1/

or

OH///opt2/

or

OH////ucs/

where

OH	is the command mnemonic;
dest	is the remote destination specification (valid values at the LHRL are LOCAL, CENTRAL, PHYS, ADMIN, VIEW, PHYE, CEPD, PDP9, PDP1145, LIBR, NAL, and PHYC);
opt1	is either the character "Y" (yes) or "N" (no) which determines whether the pagination option is in effect;
opt2	is either the character "Y" (yes) or "N" (no) which determines whether ANSI printer control is in effect;
ucs	is the UCS desired.

9.6 Text Interpretation Mode

It is sometimes useful to be able to display (or input) text in hexadecimal format, in order to see any non-printable characters that may be present in the text file. UNED provides a "mode" facility which makes this possible. The command "OM" is used to switch to either EBCDIC (the default) or hexadecimal mode. When operating in hexadecimal mode, any input which would normally be interpreted as a character string must be encoded in its hexadecimal representation. Likewise, output lines of text will be displayed at the user's terminal in hexadecimal format.

Also occasionly the user will want to edit text datasets in which recorded using the INIS (International Nuclear Information Service) or ASCII character codes. The format of the command is

OMX	switch to hexadecimal mode;
OME	switch to EBCDIC mode;
OMI	switch to INIS mode;
OMA	switch to ASCII mode.

9.7 Terminal Page Size

The various types of terminal available for use at the LHRL have different screen sizes (both number of characters across the screen, and number of lines that may be displayed at one time), so a command is available to inform UNED of the user's terminal characteristics. This is the "OP" command, and the format is

OP1<,w>

where

OP	is the command mnemonic;
l	is the number of lines which may be displayed on the screen and is used to calculate the number of lines output as a result of the line range abbreviations & and ! etc.; the default value of 1 is 20 lines;
<,w>	specifies the number of characters to be displayed across the terminal screen; the default value is 86 characters.

9.8 Setting Tab Stops

Terminal tab stops are useful for formatting input text lines. The "OT" command allows the user to specify up to eight tab stops. The format is

OTn1,n2,n3,n4,n5,n6,n7,n8

where n1 to n8 inclusive are the tab stop values defined as column numbers, and the list may be prematurely terminated (resulting in 0 values being supplied for the missing stops).

Note: This facility is not available to TSO users.

9.9 Deallocating Datasets

The facility to allocate datasets dynamically using the "OA" or "OD" commands has been described above. The "OU" command provides the inverse capability, i.e. datasets may by dynamically deallocated (disconnected from UNED). The format is

OU/ddname/

where

OU	is the command mnemonic;
/	is a delimiter character which may be any non-numeric character not present within ddname;
ddname	is the ddname used to refer to the dataset; alternatively a numeric value in the range 1 to 20, which implies a ddname of form "SFXX", where "XX" is the numeric value.

9.10 Working Length

The working length is the number of columns per line of text which UNED will allow the user to display at the terminal. This is effectively the logical record size with which the user wishes to work. The first and last substitute columns are not permitted to lie outside the column range (1, working length). The default value is 72 columns. The maximum value permitted is 136 columns.

The "OW" command is used to alter the working length. The format is

OWn

where "n" is the number of columns to which the working length is set.

9.11 Automatic Query Reply

UNED normally queries the user for confirmation when about to perform an operation which will destroy all (or a significant portion) of the current text file, or which will replace an existing text file in a library or sequential dataset. The user may sometimes wish to reply always in the affirmative to such queries. A facility (known as "auto yes mode") is thus provided

to have UNED automatically generate the affirmative reply. The "OY" command is used either to enable or to disable this facility. The format is

OYY	to enable the facility;
OYN	to disable the facility.

10. EMERGENCY FILE SAVE

If, during the edit session, any unrecoverable error arises, UNED will immediately suspend processing and attempt to save the current text buffer (only if modifications have occurred). This action is taken under the following circumstances:

(a) An unrecoverable terminal input or output error occurs.

(b) The operating system decides to abend UNED for any reason.

(c) The computer operator cancels UNED.

When UNED decides to take this emergency action the following procedure is followed:

(i) If the current text file has not been altered, UNED exits immediately.

(ii) UNED attempts to allocate and catalogue a dataset named "UNED.XXX#SAVE". If this allocation fails (or if such a dataset already exists) UNED exits immediately.

(iii) The current text buffer is saved on the newly allocated dataset, plus information regarding the datasets which have been allocated during the session. Other

information such as working length, first and last substitute columns and alphabetic case setting is also saved.

The next time the user logs in to UNED, the catalogue is checked for the presence of the user's emergency save dataset. If it is not found, UNED continues normally. If found, UNED

(i) connects itself to the dataset;

(ii) loads the saved text buffer;

(iii) allocates all datasets that were allocated at the time of the catastrophe;

(iv) sets the working length, first and last substitute columns, and the alphabetic case switch to their state at the time of the catastrophe;

(v) uncatalogues and deletes the emergency save dataset;

(vi) bypasses processing of the user profile; and

(vii) at this time, allows the user to continue with the previous editing session.

Note: The tab stops are not restored.

11. UNED SYSTEM DATASETS

UNED requires a number of datasets to be able to execute. These are as follows:

(a) The news dataset, from which the current user news is read.

(b) The printer output dataset, where any hard copy produced by the "H" command will be placed.

(c) The punch output dataset, where any punched card output produced by the "U" command will be placed.

(d) The work dataset, used to store the current text buffer.

(e) The profile dataset, where user profiles are stored.

(f) The macro work dataset, where the current macro text is stored.

It is not necessary for the user to use any job control language statements to provide these datasets provided that the installation defaults are satisfactory. Upon initial entry, UNED checks if the user has explicitly specified any of these datasets (by providing DDCARDS in the JCL or, if TSO, by issuing ALLOC commands to allocate the datasets). Such datasets not specified by the user are then allocated by UNED using the dynamic allocation feature of the operating system.

The defaults used when UNED allocates its system datasets are:

DATASET	DDNAME	DATASET ALLOCATED
news dataset	SF00	UNED.NEWS(UNEDNEWS)
work dataset	DA01	a temporary dataset
macro work dataset	DA02	a temporary dataset
printer dataset	PRINT	SYSOUT=A
punch dataset	PUNCH	SYSOUT=B
profiles dataset	PROFILE	UNED.PROFILES

The profile dataset is not strictly necessary and, if not provided

and UNED cannot dynamically allocate it, execution will continue after a warning message is issued.

If the user chooses to supply a work dataset, the dataset should have the following characteristics:

RECFM	fixed (F);
BLKSIZE	preferably 18496, however a smaller blocksize may be used provided that it is an exact divisor of 18496.

11.1 Profile Dataset

The "profile dataset" is used to store the user profile commands. A partitioned dataset is used, the member names being the user's initials, as known to the accounting system. This dataset may be set up with either RECFM=VB or RECFM=FB; however RECFM=VB is normally used since this results in considerable space savings due to the UNED text compression algorithm.

It does not affect UNED if a member is not present for a particular user; that user simply does not have a profile.

11.2 Work Dataset

The "work dataset" is a direct access dataset used to store the current text buffer. Logically, each line of the text buffer has a corresponding disk address within the work dataset, and each line reference causes that line either to be stored or retrieved from the work dataset. In practice, lines of text are buffered (currently 130 lines per buffer) to reduce the amount of input and output processing.

A "link list" is maintained (within computer memory) which keeps the logical disk address of each line of the user's text buffer. This list also maintains a record of available (free) unused logical disk addresses (within the work dataset).

The list is of fixed size, each entry being a pair of half word integers. This limits the size of the user's text buffer to 32768 lines of text. The current implementation in fact limits the size to 30000 lines of text.

12. RESERVED DDNAMES

For reasons of integrity, UNED prevents the user from dynamically allocating or deallocating (using the "OA", "OD", or "OU" commands) datasets having certain restricted DDNAMES. These DDNAMES are

PRINT, PUNCH, DA01, DA02, UNEDSV, AEINTRDR, and any DDNAME beginning with the characters "SYS".

13. DYNAMIC INVOCATION OF UNED

A facility is provided to allow user programs to call upon the services of UNED. To do so, the user program should first load UNED from the dataset "UNED.LINKLIB" using the operating system "LOAD" macro, which will return the normal entry point of UNED in general register zero. The ASSEMBLER language calling sequence is then

```
LOAD   EP=UNED,DCB=UNDCB    LOAD UNED INTO STORAGE
LR     R15,R0               NORMAL ENTRY ADDRESS TO R15
LA     R15,20(,R15)         POINT TO SPECIAL ENTRY POINT
LA     R1,VECPTR            POINTER TO ARGUMENT LIST
BALR   R14,R15              CALL UNED
continue                    UNED HAS FINISHED
```

where UNDCB is the address of an open data control block (DCB) for the dataset "UNED.LINKLIB", and VECPTR is the address of a vector list of arguments of the following ASSEMBLER language format:

```
VECLST  DC    A(0)          1  A ZERO WORD
  DC    A(TSTRT)          2  ADDRESS OF TERMINAL START ROUTINE
  DC    A(TEND)           3  ADDRESS OF TERMINAL END ROUTINE
  DC    A(TWRITE)         4  ADDRESS OF TERMINAL WRITE ROUTINE
  DC    A(TWRITN)         5  ADDRESS OF TERMINAL WRITE ROUTINE
  DC    A(TREAD)          6  ADDRESS OF TERMINAL READ ROUTINE
  DC    A(TREAS)          7  ADDRESS OF TERMINAL READ ROUTINE
  DC    A(TTEST)          8  ADDRESS OF TERMINAL TEST ROUTINE
  DC    A(TTABS)          9  ADDRESS OF TERMINAL TAB SET ROUTINE
  DC    A(TCASE)          10 ADDRESS OF TERMINAL ALPHA CASE SET
  DC    A(TTERM)          11 ADDRESS OF STRING TERMINATOR ROUTINE
  DC    A(EXITLIST)       12 ADDRESS OF EXIT LIST VECTOR
  DC    A(SFRW)           13 ADDRESS OF SEQ FILE REWIND ROUTINE
  DC    A(SFRD)           14 ADDRESS OF SEQ FILE READ ROUTINE
  DC    A(SFWR)           15 ADDRESS OF SEQ FILE WRITE ROUTINE
```

Upon entry, UNED will set the contents of word 1 in the above list to the address of an emergency wind-up routine which the user supplied routines should call if they detect an unrecoverable error during terminal input or output operations. The routines supplied in the above list are called by UNED for the following functions:

TSTRT no arguments - is called to initialise and set up terminal (or pseudo-terminal) communication.

TEND no arguments - is called to terminate terminal communication.

TWRITE one or two arguments - is called to write a line of text at the terminal followed by a carriage return and line feed. The first argument is the text string to be written, and the second argument is the number of bytes in the text string. If the second argument is missing, the length is implicitly defined by the first occurrence of the current terminating character within the string. (The current terminating character is set by the TTERM routine.)

TWRITN one or two arguments - is called to write a line of text at the terminal leaving the cursor positioned at the end of the written text. This routine is called to issue prompts and ask questions of the user. The arguments are identical to those of the TWRITE routine.

TREAD two arguments - is called to read an input line of up to 80 bytes from the terminal. The first argument is a buffer which upon return contains the input line of text, the second argument upon return contains the number of bytes of text read.

TREAS two arguments - is called to read an input line of up to 80 bytes from the terminal, but not have the input text echoed on the terminal. The arguments are identical to those of the TREAD routine.

TTEST no arguments - is called to test whether an attention interrupt has been received from the terminal. A return code of zero indicates no attention interrupt, and a return code of 4 indicates an attention interrupt has occurred. (These are equivalent to FORTRAN "RETURN" and "RETURN 1" statements respectively.) The interrupt condition should be cleared before return.

TTABS up to nine arguments - is called to set tab stops for terminal input.

TCASE one argument - is called to set the alphabetic case input mode. The single argument is either the character "U" or "L" specifying upper or lower case input respectively. If lower case is specified, the TREAD and TREAS routines should return all input characters unmodified. If upper case is specified, the TREAD and TREAS routines should translate any lower case input characters to upper case before returning to UNED.

TTERM one argument - is called to set the text string termination character. The single argument is the character which is to become the current text string termination character. Whenever the routines TWRITE or TWRITN are called with a single argument (a text string), the end of the text string is marked by the occurrence of the current terminating character.

EXITLIST supplies the address of an array of vectors which specify various exit routines for specified UNED commands. One 6-word vector is supplied for each UNED command, and a vector must be specified for each command. The vectors follow one another in consecutive storage, beginning with the vector for the "A" command and terminating with the vector for the "@" command. The specific command order is:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, :, ?, @

Each vector takes the following ASSEMBLER language format:

```
DC      A(PRECMD)       1 ADDRESS PRECOMMAND ROUTINE
DC      A(CWRITE)       2 COMMAND TERMINAL OUTPUT ROUTINE
DC      A(CWRITN)       3 COMMAND TERMINAL OUTPUT ROUTINE
DC      A(CREAD)        4 COMMAND TERMINAL READ ROUTINE
DC      A(CREAS)        5 COMMAND TERMINAL READ ROUTINE
DC      A(ENDCMD)       6 COMMAND FINISHED ROUTINE
```

Each entry in the vector specifies either the address of a routine which is called to perform the specified function or zero if a routine is not supplied, except that a value of -1 specified for the precommand exit routine (word 1) causes UNED to ignore (treat as a comment) the command. If exit routines are not specified in words 2, 3, 4, or 5, the TWRITE, TWRITN, TREAD, or TREAS routines specified when UNED was initially called are used to perform the function. The specific function of each of the exit routines is:

PRECMD three arguments - is called before any action on the command. The three arguments are the length of the command line (integer), the current scan column (integer), and the address of the 80-byte command buffer. The command length may actually be one byte less as UNED will have already stripped off any trailing "P" or "L" option specified. The command buffer is provided primarily to permit inspection; however the user may (using extreme care) modify the contents of the buffer provided that the length change and scan column repositioning are reflected by modifying the values of the first two arguments accordingly. When this exit is called, UNED has determined the line range to be operated upon and the subroutine to be used to process the command. The exit routine may opt to allow the command to be processed by

returning with a return code of 4 (in register 15), or to have the command ignored by returning with a return code of 0 (in register 15). (These are equivalent to FORTRAN "RETURN 1" and "RETURN" statements respectively.)

CWRITE one or two arguments - performs the identical function to the TWRITE routine, but only for the duration of the current command. The arguments are identical to those of the TWRITE routine.

CWRITN one or two arguments - performs the identical function to the TWRITN routine, but again only for the duration of the current command. The arguments are identical to those of the TWRITN routine.

CREAD two arguments - performs the identical function to the TREAD routine, but again only for the duration of the current command. The arguments are identical to those of the TREAD routine.

CREAS two arguments - performs the identical function to the TREAS routine, but again only for the duration of the current command. The arguments are identical to those of the TREAS routine.

ENDCMD no arguments - is called after the command action is completed.

SFRW, SFRD, and SFWR are the addresses of subroutines to be used for sequential file input and output. These addresses are optional, but if any one is provided then so must the other two. If not provided, binary zeros must be placed in the corresponding vector list slots. When these routines are provided, all sequential file input and output ("En", "Rn", "Wn" commands) are performed via these routines. This provides a facility for in-core data transfer from invoking programs.

SFRW one argument - is called to effect a rewind of the unit. The argument is the integer unit number being referred to.

SFRD three arguments - is called to fetch the next line of input from the sequential file. The arguments are the integer unit number, address of an 80-byte area to receive the line of input, and an integer flag word which should be set to zero if a line of input is returned, or set to non-zero if end of file has been reached.

SFWR two arguments - is called to output a line of text. The arguments are the integer unit number, and the address of the 80-byte line to be output.

When UNED finally returns to the calling routine, register 15 will contain one of the following return codes:

0	Normal completion.
4	Hard copy output (printer or punch) has been produced.
12	UNED failed to initialise itself. This may be due to a variety of reasons such as allocation failure for the work dataset, or user already logged on to another UNED session.

13.1 FORTRAN Interface to UNED

The following subroutines are provided in the dataset "UNED.LINKLIB" to allow FORTRAN programmers to invoke UNED dynamically:

ZPUNED, STUNED, BRUNED, ERUNED

These routines are grouped as a load module named "CALLUNED", and the user includes them into the program at linkage edit time by using the linkage editor [IBM 1978] input statement:

```
INCLUDE LIB(CALLUNED)
```

where LIB is the DDNAME of the JCL card used to allocate the dataset "UNED.LINKLIB" for the linkage editor.

The use of these subroutines is

```
CALL ZPUNED
```

which clears the list of exit vectors for all commands.

```
CALL STUNED (CMD, PRECMD, CWRITE, CWRITN, CREAD,
CREAS, ENDCMD, &ERR)
```

is used to set up an exit vector for a given command, where CMD is the command character and PRECMD, CWRITE, CWRITN, CREAD, CREAS, and ENDCMD are the addresses of the routines described above, and should be declared as EXTERNAL subroutines in the FORTRAN routine calling STUNED. These routine addresses may be replaced by the integer zero in the CALL statement if the function is not desired, or by the integer -1 in the case of PRECMD to remove the

corresponding command from UNED. ERR is the FORTRAN statement number to which control is returned if an unknown command character is specified.

The above routine should be called once for each command for which the user wishes to set exits.

```
CALL BRUNED (TSTRT, TEND, TWRITE, TWRITN, TREAD,
             TREAS, TTEST, TTABS, TCASE, TTERM,
             ICODE, &ERROR)
```

or

```
CALL BRUNED (TSTRT, TEND, TWRITE, TWRITN, TREAD,
             TREAS, TTEST, TTABS, TCASE, TTERM,
             ICODE, SFRW, SFRD, SFWR, &ERROR)
```

This subroutine actually loads and calls UNED for the calling routine. TSTRT, TEND, TWRITE, TWRITN, TREAD, TREAS, TTEST, TTABS, TCASE, TTERM, SFRW, SFRD, and SFWR are the addresses of the routines described above and should be declared as EXTERNAL subroutines within the calling routine. ICODE is an integer variable which upon return will contain the UNED return code. ERROR is the FORTRAN statement number to which control will be passed if BRUNED is unable to branch to UNED.

```
CALL ERUNED
```

This routine should be called by any of the supplied terminal (or pseudo-terminal) input or output routines if an unrecoverable error condition is encountered. This will cause UNED to enter its emergency save routine, to save the current workfile, and then terminate.

Note: this action should only be taken in dire emergencies.

14. REFERENCES

IBM [1976] - OS/VS2 MVS data management services guide. IBM systems manual GC26-3875-0.

IBM [1978] - OS/VS linkage editor and loader. IBM systems manual GC26-3813-5.

IBM [1979] - OS/VS2 MVS JCL. IBM systems manual GC28-0692-4.

Konheim, A.G., Mack, M.H., McNeill, R.K., Tuckerman, B., Waldbaum, G. [1980] - The IPS cryptographic programs. IBM Systems J., 19(2)253-283.

Thompson, K., Ritchie, D.M. [1975] - UNIX programmers manual (sixth edition). Bell Telephone Laboratories (Western Electric Company), New York.

APPENDIX A - COMMAND SUMMARY

An alphabetic list of UNED commands is given. Any quantity within the first column which is enclosed in brackets (e.g. (n)) is to be considered as an optional command argument. Likewise, the construct (P:L) is to be interpreted as an optional specification of either the "P" or the "L" option.

A	Insert text after given line
(n)A(P)	"n" line number (default .); "P" causes input to be prompted with line numbers.
(n)A/string/(P)	immediate form - adds a new line consisting of "string", after line "n".
B	**Macro command**
B	allows commands to be loaded directly into macro buffer.
BP	lists current contents of macro buffer.
B/member/(ddname/)	reads a text file from a library into the macro buffer.
(n)Bm(P:L)	executes the current macro - "n" if present sets the current line to "n" before executing the macro; "m" is the number of times the macro is executed;

"P" if present traces macro execution by typing each command before its execution;
"L" if present causes a certain amount of silence, and ignorance of error conditions during macro execution.

(n)Bm/member/(ddname/)(P:L) causes the specified text file to be read from a library into the macro buffer, and then executes the new macro.

C — Replace given lines

(n1,n2)C(P)
"n1" - first line to be replaced (default .);
"n2" - last line to be replaced (default .);
"P" - causes input to be prompted with line numbers.

(n1,n2)C/string/(P) immediate form where lines "n1" through "n2" are replaced by a new line consisting of string.

D — Delete given lines

(n1,n2)D
"n1" - first line to be deleted (default .);
"n2" - last line to be deleted (default .).

E	Edit given file
E(/member/)(P:L)	"member" - member name of library file to be edited (default is the last member name specified for an "E" command, or last member name specified for a "W" command which outputs the entire text file); "P" - lists member while loading, each line preceded by line number; "L" - lists member while loading.
E/(member)/(ddname/)(P:L)	"ddname" - ddname of library from which to load member.
Em(P:L)	"m" - unit number of sequential file to be edited (m in range 1 to 20).
E/:/(ddname/)(P:L)	edit library directory.
EO(P:L)	edit the current user profile.
F	Library file management
F(P)	lists the names of all members in the library (comments stored in the directory are listed if "P" option present).
F/member/	"member" - name of a member to be deleted from the library.
F/member/newmember/	member "member" is renamed to "newmember".

G — Go to a given string

(n1,n2)G(*)(#)(m)(/string/)(P:L)

"n1" - first line to be searched (default .+1);
"n2" - last line to be searched (default $);
"*" - if given finds all occurrences of "string"; implies "P" option if "L" not specified;
"#" - if given causes lines not containing "string" to be found;
"m" - if present restricts the search to column "m" of each line
"string" - the character string to be located (default is the last string given for a "G" command);
"P:L" - if present causes matched lines to be listed (preceded by line number if "P").

H — Produce hard copy

(n1,n2)H(/head/)(P)

"n1" - first line to be printed (default 1);
"n2" - last line to be printed (default $);
"head" - if present is printed above the output text (default is current member name and comment and name of dataset from which loaded);
"P" - if present causes the printed lines to be numbered.

I	Insert text before given line
(n)I(P)	"n" - line number (default .); "P" - causes input to be prompted with line numbers.
(n)I/string/(P:L)	immediate form - adds new line consisting of "string", before line "n", and if "P" or "L" option present, prints the added line.
J	Job submission
J(P)	submits entire text file to the JES internal reader; "P" - lines of text sequentially numbered during input in columns 73 through 80; text lines of format ()()member or ()()member ddname cause the indicated library file to be included in place during the job submission; trailing "*" on such member names causes inclusion of all members, beginning with string "member" from the indicated library.

The user's account number should not be present on any job cards within the file. The string "********" should be used in the account number field of the jobcard, and is replaced with the user's account number during job submission.

K — Date and time

K — prints current date and time at the terminal.

L — List lines of text

(n1,n2)L(m) — "n1" - first line to be listed (default .);
"n2" - last line to be listed (default .);
"m" - first column to be listed (default 1);
the indicated range is listed without line numbers.

M — Move lines of text

(n1,n2)M(n3)(P:L) — "n1" - first line to be moved (default .);
"n2" - last line to be moved (default .);
"n3" - line after which moved lines are to be placed (default .);

"P:L" - if present the moved lines are listed (preceded by new line numbers if "P").

N — Number lines of text in columns 73 through 80

(n1,n2)N(i)(,s)

"n1" - first line to be numbered (default 1);
"n2" - last line to be numbered (default $);
"i" - numbering increment (default 1);
"s" - starting number (default i);
if "i" is zero, columns 73 through 80 are blanked.

O — Option setting

OA/ddname/dsname(/status)

allocate a library, or a sequential dataset;
"ddname" - symbolic name used to refer to the library (or sequential dataset);
"dsname" - the library (or dataset) name as stored in the MVS system catalogue;

	"status" - access status required; valid values are OLD for exclusive use, SHR for shared use, NEW to create a new library (or sequential dataset), and MOD if extending a sequential dataset.
OCk	set alphabetic case switch; "k" is either the character "U" for upper case, or the character "L" for lower case.
OD/ddname	the current library becomes the library referred to by "ddname".
OD/ddname/dsname(/status)	allocate a library, and make it the current library; "ddname" - symbolic name used to refer to the library (or sequential dataset); "dsname" - the library (or dataset) name as stored in the MVS system catalogue; "status" - access status required, valid values are OLD for exclusive use, SHR for shared use, NEW to create a new library (or sequential dataset), and MOD if extending a sequential dataset. This form of the "OD" command combines the functions of the first form and the "OA" command.

OE	specify encryption options and keywords; the user is queried whether input, output, and work files are to be encrypted and, if so, for encryption keys.
OFn	set the first substitute column "n" - is the column number.
OH/dest	set hard copy route code; "dest" - is the remote work station route code.
OH/(dest)/(op1)/(op2)/(ucs)/	set hard copy destination and pagination options; "dest" - is the remote work station route code; "op1" - is either the character "Y" or "N" specifying yes or no for starting a new page on the hard copy dataset each time an "H" command is issued; "op2" - is either the character "Y" or "N" specifying yes or no for use of the first column of hard copy output as the ANSI printer control character; "ucs" - is a print train name.
OLn	set last substitute column to column "n".
OME	set EBCDIC mode.
OMX	set hexadecimal mode.

OP(n)(,m)	set page width and length; "n" - is page length in lines; "m" - is page width in columns.
OT(n1)(,n2)...	set tab stops to columns "n1", "n2", etc.; up to nine tab settings may be specified.
OU/ddname	deallocate a library or sequential file; "ddname" - is the name used to refer to the library (or sequential file).
OWn	set working length to "n" columns.
OYY	set "auto yes" mode.
OYN	turn off "auto yes" mode.
OZ	specify strong encryption options and keywords; the user is queried whether input, output, and work files are to be encrypted and, if so, for encryption keys.
OZ/si/so/sw/	specify strong encryption options directly; "si" is input file encryption key, or null if the input file is not encrypted; "so" is output file encryption key, or null if the output file is not to be encrypted;

	"sw" is the work file encryption key, or null if the work file is not to be encrypted.
O?	list the current option settings at the terminal.
P	List lines of text preceded by line numbers
(n1,n2)P(m)	"n1" - first line to be listed (default .); "n2" - last line to be listed (default .); "m" - first column to be listed (default 1).
Q	Quit edit session
R	Read a text file from library (or sequential file)
(n)R(/member/)(P:L)	"n" - line after which the file read is to be appended (default .); "member" - member name of library file to be read (default is the last member name specified for an "E" command, or last member name specified for a "W" command which output the entire text file); "P" - lists member while reading, each line preceded by line number;

	"L" - lists member while reading.
(n)R/(member)/(ddname/)(P:L)	"ddname" - ddname of library from which to read member.
(n)Rm(P:L)	"m" - unit number of sequential file to be read (m in range 1 to 20).
(n)R/:/(ddname/)(P:L)	read library directory.
(n)R0(P:L)	read the current user profile.

S — String substitution

(n1,n2)S(m)(/s1/s2/)(P:L)	"n1" - first line to be tested and modified (default .); "n2" - last line to be tested and modified (default .); "m" - optional column number which constrains the string to be substituted ("s1") to begin in column "m" of each line tested; "s1" - text string to be replaced (default is the last "s1" string specified for an "S" or "Z" command); "s2" - replacement text string (default is the last "s2" string specified for an "S" or "Z" command); "P" - print modified lines preceded by line number; "L" - print modified lines.

(n1,n2)S(m)/s1/s2/s3/(P:L)

"s3" - a text string used as a condition for modification (this form of the "S" command does not allow defaults to be used for any of the text strings "s1", "s2", or "s3"); on any given line "s2" only replaces "s1" if string "s3" is also present on the line; in this case, the column number "m" condition migrates to the "s3" string.

T

Duplicate given lines

(n1,n2)T(m)(P:L)

"n1" - first line to be duplicated (default .);
"n2" - last line to be duplicated (default .);
"m" - line after which duplicated text will be placed (default .);
"P" - print duplicated lines preceded by line numbers;
"L" - print duplicated lines of text.

U

Unload (produce punched cards)

(n1,n2)U(P)

"n1" - first line to be punched (default .);
"n2" - last line to be punched (default $);
"P" - if present causes punched cards to be sequence numbered in columns 73 to 80 inclusive.

V	Overwrite text in given lines
(n1,n2)V(m)(/string/)(P:L)	"n1" - first line to be overwritten (default .); "n2" - last line to be overwritten (default .); "m" - starting column for overwriting (default is last specified for this command); "string" - text string with which to overwrite (default is last specified for this command); "P" - print modified lines preceded by line number; "L" - print modified lines.
W	Write to library (or sequential dataset)
(n1,n2)W(/mem/)(P)	"n1" - first line to be written (default 1); "n2" - last line to be written (default $); "mem" - name of library member to write to (default is the last member name specified in an "E" command or the last member name specified in a "W" command which output the entire text file to a library). "P" - causes the output text lines to be sequence numbered in columns 73 to 80 inclusive.

(n1,n2)W(/mem:com/)(P)

"com" - a comment text string (up to 60 characters long) which is stored in the library directory along with the member name (default is the comment loaded with the member when the "E" command was issued, or the comment specified in the last "W" command if it postdates the "E" command).

(n1,n2)W/(mem:com)/ddname/(P)

"ddname" - the symbolic name referring to a library.

(n1,n2)Wm(P)

"m" - unit number of the sequential file to which output is directed.

(n1,n2)W0

"0" - the lines of text are written to the user's profile.

X

ASSEMBLER comment given lines

For each line of the given range, columns 1 to comment column minus 1 are listed, and then a new comment may be typed in.
An immediate carriage return leaves the existing comment unchanged.
Premature exit from the command is achieved by typing the character "." followed by carriage return (this does not alter the existing comment).

(n1,n2)X(m)(P)	"n1" - first line to be commented (default .); "n2" - last line to be commented (default .); "m" - starting column for comment (default initially 40, thereafter the last value of "m" specified for this command); "P" - precede lines listed with line number.
(n1,n2)X/string/(P)	"string" - may only be present if "m" is not specified, and uses the text "string" to determine the value of "m" by matching "string" in a given line; "m" becomes the column number of the first character of located string; lines with no match are skipped.
Y	Suspend editor The format of this command depends on how UNED is entered.
Y$string	This format is only valid at the LHRL when UNED is entered by the user typing $UNED at a Dataway terminal.

"string" - is the name of any other "Dataway program" the user wishes to enter; when the called program terminates, the user is prompted by "=", and may now enter $ followed by the name of another "Dataway program", or an immediate carriage return if the user wishes to continue editing.

Y

this form is valid for all methods of UNED entry. if UNED is entered by the user typing $UNED at a Dataway terminal, the user is prompted by "=" and may continue as above (at Y$string).

If UNED was entered as an "I" class job, the user is returned to the terminal mode that existed before entering UNED, and needs to repeat the startup procedure to re-enter UNED.

If UNED is entered as a TSO command, the user is prompted to enter a TSO command or carriage return, the carriage return allowing editing to proceed.

Z	Global string substitution
(n1,n2)Z(m)(/s1/s2/)(P:L)	"n1" - first line to be tested and modified (default .); "n2" - last line to be tested and modified (default .); "m" - optional column number which constrains the string to be substituted ("s1") to begin in column "m" of each line tested; "s1" - text string to be replaced (default is the last "s1" string specified for an "S" or "Z" command); "s2" - replacement text string (default is the last "s2" string specified for an "S" or "Z" command); "P" - print modified lines preceded by line number; "L" - print modified lines.
(n1,n2)Z(m)/s1/s2/s3/(P:L)	"s3" - a text string used as a condition for modification (this form of the "Z" command does not allow defaults to be used for any of the text strings "s1", "s2", or "s3"). On any given line, "s2" only replaces "s1" if string "s3" is also present on the line; in this case, the column number "m" condition migrates to the "s3" string.

:	Print comment
:	prints current member name and comment.
:/member/	prints comment for the given member in the current library.
?	Help
?	prints a listing of line number abbreviations and command syntax for all commands.
?c	prints command syntax information about command "c".
@	select an extended command (see Section 8.4).

APPENDIX B - USING UNED FROM TSO

UNED is available as a TSO (IBM Time Sharing Option) command, so after logging onto TSO, the user has merely to issue the command

UNED

to enter UNED.

B1. DIFFERENCES TO STANDARD UNED

A number of differences to the standard UNED, which is designed for LHRL Dataway use, are to be noted:

(a) Tab stops are not available.

(b) Some line range abbreviations are not available, because the IBM terminals do not provide the appropriate characters.

(c) The "Y" command allows access to other TSO commands as subcommands of UNED. CLIST processing is supported.

(d) The user may pre-allocate some or all of the UNED system datasets using the TSO ALLOC command.

APPENDIX C - UNED AS AN INTERACTIVE BATCH JOB

At the LHRL it is possible to run certain types of batch jobs in an interactive mode. These jobs must be submitted as "CLASS=I" jobs, and use a set of locally written subroutines to communicate with the user's terminal. UNED can be run in this pseudo-batch mode. To do so, the user invokes UNED with the following job control statements:

```
//       EXEC  PGM=UNED
//STEPLIB DD  DSN=UNED.LINKLIB,DISP=SHR
```

Of course an appropriate job card, any optional output destination routing, and JOBPARM control statements must precede the above two statements. The user then submits this job as a CLASS=I job, and then starts the program "$RUN" to establish communication with UNED. The terminal dialogue required follows:

- User types $RUN

- Computer prints JOB: USR
  where USR is the user's initials.

- The user then completes the jobname by typing additional characters (up to 5) immediately after the displayed initials.

- The computer then types TESTING JOB USRXXXXX
  where USRXXXXX is the jobname of the required job.

- After a short delay the computer types JOB USRXXXXX STARTING; the remainder of the dialogue is similar to a normal UNED session.

C1. DIFFERENCES TO STANDARD UNED

The following differences to normal UNED usage should be noted:

(a) The "Y" command allows escape from UNED to permit the user to call other interactive programs; however communication with UNED (using the $RUN program as above) must be re-established within five minutes or UNED will terminate.

(b) The user may supply DDCARDS to define some or all of the UNED system datasets. This is most useful for overriding the "PRINT" (hard-copy output) dataset to specify esoteric print trains.