7546

RFC 216
NIC 7546
Categories: D.3, G.3
Updates: None
Obsoletes: None

Telnet Access To UCSB's On-Line System

8 September 71

James E. White
Computer Research Lab
University of California
Santa Barbara

Contents

Figures

I. Motivation

A teletype-compatible interface to UCSB's On-Line System (OLS) has been implemented in accordance with the Telnet protocol adopted by the NWG. This Server Telnet is responsive to connection requests directed by User Telnet's to socket number 1, host address 3. Although OLS is not a teletype system and although much of its power as a mathematical tool rests in its graphical display capabilities, enough of the System survives the Telnet transformation to justify such an implementation.

II. Limitations

In this Telnet-style implementation of OLS, all curvilinear display generated by the user on Levels II and III, Real and Complex, is discarded by the System and hence not returned to the user through the Net. The same is true of the display of special, user-created characters. Although special characters may be constructed and stored, their display will be suppressed, both during the process of construction and later when they are invoked from the Type level. All other display generated by the System will be relayed to the user intact, in some cases with stylistic transformations having

first been applied. For example, Greek characters are displayed as lower-case a-z. All such transformations are described in detail in this document. Finally, those elements of the System (the operators which edit user programs are prime examples) which assume a fixed-screen display device function abnormally in a Telnet environment. For such a device, the System can 'remember' the position on the screen of a previously displayed segment of text and return to that position to, for example, underscore it. But when the 'screen' marches forward--relentlessly--through a continuous medium, as it does with Telnet's virtual teletype, that kind of strategy fails. Hence, the underscoring is not relocated, but rather appears on the current line, beginning in the next available character frame.

OLS assumes, normally, that the user is equipped with the specially-designed double keyboard depicted in Figure 1. Conventions are defined in this document which enable a Telnet user to simulate that keyboard; in particular, a means is provided for designating keys on the upper, or operator keyboard.

III. System Documentation

This document has three purposes:

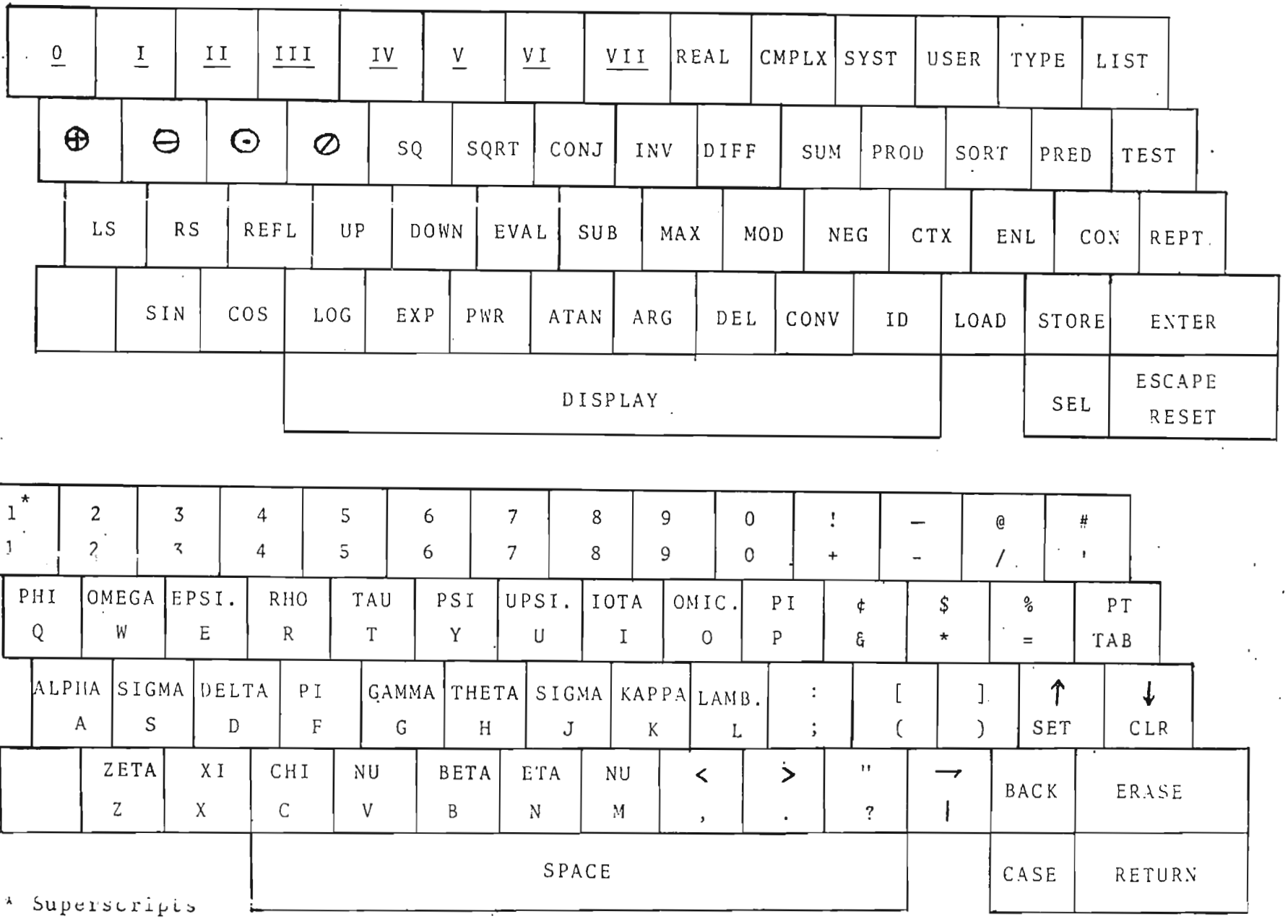

Figure 1. OLS Keyboard

(1) to describe the means by which a Telnet user simulates an OLS keyboard,

(2) to describe the transformations applied to output generated by the System, and

(3) to enumerate those aspects of the System which are unique to or behave differently for Network (Telnet) users.

In particular, this document is not a user's manual for OLS. Such a manual is available and on file with the NIC. In addition, a copy should exist at each Network site in its NIC collection; the user should consult his Station Agent. The document is titled 'UCSB On-Line System Manual' [NIC 5748]; its contents are current as of 1 January 71. A revision to the manual is currently in preparation and will be distributed when available. In addition, tutorial manuals for two of the subsystems available under OLS--MOLSF (Mathematically-Oriented Language Single-Precision Floating-Point) and COL (Card Oriented Language)--will soon be made available. The latter has already been published and is being transmitted to the NIC for distribution, while the former is nearing completion.

Documentation of the third subsystem of OLS--NET--has already been distributed through the NIC as two RFC's: 'Network On-Line Operators' [21 April 71, RFC 121, NIC 5833] and 'A User Telnet--Description of an Initial Implementation' [9 August 71, RFC 206, NIC 7176].

NET currently houses a set of operators for system-call-level interaction with UCSB's NCP, a User Telnet, and an operator (invoked by ID on Level II) which returns the status of Network hosts.

Staff members at the Computer Center will be happy to field questions about OLS from Network users. In particular, an OLS consultant is available for such purposes at (805) 961-4044 . Questions about OLS, including those specific to use of the System through the Network, may also be addressed to Jim White, UCSB's Technical Liaison, at (805) 961-3454 (if necessary, messages can be left at the Computer Center Office, (805) 961-2261).

IV. System Access

The Network user is encouraged to explore the System and is invited to do so with the following accounting parameters:

User Number: 196
Id Number: 57372
User Name: ARPA
Problem Name: (affiliation)-(name)
in 16 characters or less
(e.g., UCSB-WHITE)

Such use of the System will not be billed. Production users are asked to establish their own accounts with the

Computer Center ((805) 961-2261), the use of which will be billed in accordance with the then-current rate structure.

V. Software Structure

This document is the description of a Network front-end to the On-Line System, logically distinct from OLS itself. This front-end is hereafter referred to as NETOLS. NETOLS is always responsive to connection requests directed to socket 1. When contacted by a Network user, NETOLS performs the Network functions required to establish a duplex connection to him. The number of such duplex connections (and hence the number of Network users) is bounded by an assembly parameter whose current value is five.

Before the Network connection is established, NETOLS secures for the user a port into OLS. Sixty-four such ports exist and are shared by local, dial-up, and Network users. Should none be available, NETOLS will abort the connection sequence.

Once a port has been secured and a Network connection established, NETOLS will effectively push the SYST key for the user by transmitting to OLS the 8-bit code representing that key. A login sequence is thus initiated and the

user is transmitted the lines:

```
UCSB ON-LINE SYSTEM
ENTER USER NUMBER
```

to which he should respond with his user number. Beginning at this point in time and continuing for the life of the Network connection, NETOLS's sole function is that of interpreter--interpreting input from the user and making it meaningful to OLS, and interpreting output from OLS and making it meaningful to the user (it is at this point, for example, that curvilinear and special-character display are discarded).

When the user breaks his Network connection to NETOLS, if he hasn't logged out of OLS already, NETOLS performs that function for him by pushing SYST DOWN, just as it pushed the initial SYST. The OLS port acquired for the user is then released, and hence available for use by other users. It should be noted that the user can log out of OLS and back on again without the Network connection's being broken, since that action is transparent to NETOLS, who attaches no special significance to the key sequence which accomplishes it.

VI. Virtual OLS Keyboard

A major function of NETOLS is to provide a mapping between elements of the Telnet character set and the keys on an OLS keyboard (Figure 1). The lower, or operand portion of that keyboard is fairly easily represented, since it's similar to a standard typewriter keyboard. Most of the keys on the lower keyboard are mapped on a one-for-one basis from elements of the Telnet character set. Upper-case alphabetics are mapped into the alphabetics, lower-case alphabetics into the Greek characters, numerics into numerics, and a miscellany of punctuation into itself. All such one-for-one mappings are depicted in Figure 2. A line of that figure reads as follows:

> For '¬':
>
> The key labeled '¬' (meaning logical not) on the lower portion of an OLS keyboard is struck by causing the user's User Telnet to transmit '~' (tilde).

Those lower-keyboard keys not listed in Figure 2, and all the keys on the upper-keyboard (hereafter referred to collectively as non-standard keys), are represented by the Telnet user in the following manner. For each such key, a character string has been defined; the string is called the name of the key. In most cases, the name

of a key is identical to its label in Figure 1. The name of the SIN key, for example, is 'SIN' (in the On-Line System User's Manual, upper-keyboard keys are denoted by underscoring their labels, to distinguish, for example, the key SIN from the three keys 'SIN').

> Every non-standard key on the OLS keyboard is struck by typing its name (or any unique abbreviation thereof), preceded by a special prefix character and followed by a space.

NETOLS intercepts the prefix, name, and space and from them generates a single, 8-bit code which it forwards to OLS.

The default prefix character is semi-colon (';'), chosen simply because for touch typists it's one of the home keys. The prefix can be changed by the user to any character listed in Figure 2. The procedure for so doing is described in Section VII-B. To send the prefix character through NETOLS to OLS, type it twice in succession. Thus, if the default prefix is in effect, ';;' is mapped into a single semi-colon and relayed to OLS.

The names of all non-standard keys are listed in Figure 3. A line of that figure reads as follows:

For SIN:

> The key denoted SIN in the OLS User's Manual (the trigonometric function sine) is named 'SIN', and hence is struck by typing 'SIN', preceded by the prefix and followed by a space.

Assuming, then, that the default prefix ';' is in effect, SIN is struck by typing ';SIN_' ('_' is used here and in following examples to denote a space). Furthermore, if the user chooses, he may abbreviate that as ';SI_', since the key desired remains uniquely identified. Further abbreviation (to ';S_') is unsatisfactory and hence disallowed since the single character 'S' is insufficient to distinguish between a number of keys whose names begin with that character. Key names may be typed by the user in either upper- or lower-case.

As each character of a non-standard key's name is typed by the user, NETOLS consults its table of key names. If the character string so far specified cannot possibly lead to a valid name, the most recent character is ignored ('?' echoed). Hence, typing ';SJIN_' will be accepted as SIN, the erroneous 'J' being ignored (and a question mark echoed), and the subsequent 'IN_' accepted. If when the terminating space is typed, no single key is uniquely identified a '?' is echoed and the space ignored. Thus ';S_I_' will be recognized as SIN; the first space is

To Push	(OLS Explanation)	Send	(Telnet Explanation)
0-9	Decimal Digits	0-9	Decimal Digits
A-Z	Alphabetics	A-Z	UC Alphabetics
α-ζ	Greek Characters	a-z	LC Alphabetics
!	Exclamation Mark	!	Exclamation Mark
+	Plus Sign	+	Plus Sign
_	Underscore	_	Underscore
-	Minus Sign	-	Minus Sign
@	Commercial At	@	Commercial At
/	Slash	/	Slant
#	Number Sign	#	Number Sign
'	Apostrophe	'	Apostrophe
&	Ampersand	&	Ampersand
$	Dollar Sign	$	Dollar Sign
*	Asterisk	*	Asterisk
%	Percent	%	Percent
=	Equal Sign	=	Equal Sign
TAB	Horizontal Tab.	HT	Horiz. Tab. (↑I)
:	Colon	:	Colon
;	Semi-Colon	;	Semi-Colon
[	Left Bracket	[	Left Bracket
]	Right Bracket	]	Right Bracket
(	Left Parenthesis	(	Left Parenthesis
)	Right Parenthesis	)	Right Parenthesis
<	Less Than	<	Less Than
>	Greater Than	>	Greater Than
,	Comma	,	Comma
.	Period	.	Period
"	Quotation Marks	"	Quotation Marks
?	Question Mark	?	Question Mark
¬	Logical Not	~	Tilde
|	Logical Or	|	Vertical Line
BACK	Backspace	BS, DEL	Backspace/Rubout
RETURN	Carriage Return	CR	Carr. Return (↑M)
SPACE	Space	SP	Space

Figure 2. Keys With One-for-One Mappings

To Push	(OLS Explanation)	The Key Name Is	
⊙	Multiply	*	1
⊕	Add	+	
⊖	Subtract	-	
RETURN	Carriage Return	.	
⊘	Divide	/	
L0	Level 0	0	
L1	Level 1	1	
LII	Level II	2	
LIII	Level III	3	
LIV	Level IV	4	
LV	Level V	5	
LVI	Level VI	6	
LVII	Level VII	7	
⊕	Add	ADD	2
ARG	Argument	ARG	
ATAN	Arc Tangent	ATAN	
BACK	Backspace	BACK	3
CASE	Case	CASE	
¢	Cent Sign	CENT	
CLR	Clear Tab	CLEAR	
CMPLX	Complex	CMPLX	
CON	Contract	CON	
CONJ	Conjugate	CONJ	
CONV	Convolve	CONV	
COS	Cosine	COS	
CTX	Context	CTX	
DEL	Delta	DEL	
DIFF	Forward Difference	DIFF	
DISPLAY	Display	DISPLAY	
⊘	Divide	DIV	4
DWN	Down	DOWN	
ENL	Enlarge	ENL	
ENTER	Enter	ENTER	
ERASE	Erase	ERASE	
ESCAPE	Escape	ESCAPE	
EVAL	Evaluate	EVAL	
EXP	Exponentiate	EXP	
	NETOLS Command	FULLDUPLEX	5
	NETOLS Command	HALFDUPLEX	
	NETOLS Command	HELP	
ID	Identity	ID	
INV	Invert	INV	
↓	Line Feed Down	LFDN	
↑	Line Feed Up	LFUP	

Figure 3. Keys Represented As Strings

To Push	(OLS Explanation)	The Key Name Is	
LIST	List	LIST	
LOAD	Load	LOAD	
LOG	Logarithm	LOG	
	NETOLS Command	LOGOUT	
LS	Left Shift	LS	
MAX	Maximum	MAX	
MOD	Modulus	MOD	
⊙	Multiply	MULT	6
NEG	Negate	NEG	
¬	Logical Not	NOT	7
∣	Logical Or	OR	8
PRED	Predicate	PRED	
	NETOLS Command	PREFIX	
PROD	Running Product	PROD	
PT	Point	PT	
PWR	Power	PWR	
REAL	Real	REAL	
REFL	Reflect	REFL	
REPT	Repeat	REPT	
RESET	Reset	RESET	
RETURN	Carriage Return	RETURN	9
RS	Right Shift	RS	
0-9	Superscript 0-9	S0-S9	
SEL	Select	SELECT	
SET	Set Tab	SET	
	NETOLS Command	SHIFT	
SIN	Sine	SIN	
SORT	Sort	SORT	
SQ	Square	SQ	
SQRT	Square Root	SQRT	
	NETOLS Command	STATE	
STORE	Store	STORE	
SUB	Substitute	SUB	
⊖	Subtract	SUBTRACT	10
SUM	Running Sum	SUM	
SYST	System	SYST	
TEST	Test	TEST	
TYPE	Type	TYPE	
	NETOLS Command	UNSHIFT	
UP	Up	UP	
USER	User	USER	

Figure 3 (cont'd) Keys Represented As Strings

1. Alternate names for ⊙, ⊕, ⊖, RETURN, and ⊘ are 'MULT', 'ADD', 'SUBTRACT', 'RETURN', and 'DIV', respectively. RETURN can also be represented as the single character CR (carriage return), as indicated in Figure 2.

2. An alternate name for ⊕ is '+'.

3. Alternates for BACK are the single characters BS (backspace) and DEL (rubout), as indicated in Figure 2.

4. An alternate name for ⊘ is 'DIV'.

5. NETOLS commands are explained in Section VII.

6. An alternate name for ⊙ is 'MULT'.

7. An alternate for '→' is the single character '~' (tilde), as indicated in Figure 2.

8. An alternate for '|' is the single character '¦' (vertical line), as indicated in Figure 2.

9. An alternate name for RETURN is '.'. RETURN can also be represented as the single character CR (carriage return), as indicated in Figure 2.

10. An alternate name for ⊖ is '-'.

Notes for Figure 3.

ignored (and a '?' echoed, indicating that 'S' alone is ambiguous).

At any point in the entry of a key name, either Altmode (ESC) or '?' may be typed by the user. NETOLS will then determine whether a key has been uniquely specified by the characters already typed. If so, it will echo the remaining characters of the key's name, and consider them entered by the user. A subsequent space from the user will cause the indicated key to be pushed. If no single key is uniquely specified, NETOLS will echo Bel, causing a bell to be rung on many terminals. More of the key name is then expected from the user.

If after at least one character of the key name has been entered by the user and accepted by NETOLS (and before the terminating space is typed) the prefix is typed a second time, all already entered characters of the name are discarded by NETOLS. Thus ';CO;SIN_' is interpreted as SIN. If a carriage return is typed in the same context, the initial prefix will also be discarded. Hence, ';CO%S' ('%' denotes carriage return) is interpreted as the lower-keyboard key 'S'.

VII. NETOLS Commands

A number of commands to NETOLS are defined and

all are described in this section. The format for each such command is the same as that for non-standard keys, and hence the command keywords are included in Figure 3. All of the conventions of Section VI apply as well to the entry of commands. The user should understand, however, that such commands are processed by NETOLS, not OLS, and that they are defined only for Network users of OLS.

A. HELP

The HELP command (invoked with ';HELP_' if ';' is the prefix) reproduces for the user the third column of Figure 3; the names of all non-standard keys and the keywords for all defined NETOLS commands are listed in their collating sequence on the user's virtual teletype.

B. PREFIX

Issuing the PREFIX command causes the next character typed to become the prefix, provided it is one of those listed in Figure 2. Consequently, ';PREFIX_@' makes '@' the prefix, '@PREFIX_;' restores the default situation.

C. SHIFT and UNSHIFT

The SHIFT command causes a perturbation of lines 2 and 3 of Figure 2. After SHIFT is issued, all subsequent upper-case alphabetics are mapped into the Greek characters (rather than into the alphabetics), and lower-case alphabetics into alphabetics (rather than into the Greek characters). This convention change may be found convenient if the user's User Telnet sends lower-case alphabetics by default, and requires, for example, that a shift key be held down to send upper-case characters.

The UNSHIFT command nullifies the effect of SHIFT.

D. FULLDUPLEX and HALFDUPLEX

Issuing the FULLDUPLEX command causes all subsequent characters typed by the user to be echoed by NETOLS. HALFDUPLEX nullifies the effect of FULLDUPLEX, disabling echo by NETOLS. Half-duplex is the default situation.

E. STATE

The STATE command causes the current prefix, the mode of operation ('HALFDUPLEX' or 'FULLDUPLEX'),

and the case convention ('SHIFT IS ON' or 'SHIFT IS OFF') to be displayed on the user's virtual teletype in the following form:

```
PREFIX IS ;
HALFDUPLEX
SHIFT IS OFF
```

F. LOGOUT

Issuing the LOGOUT command causes the user to be logged out of OLS (i.e., SYST DOWN to be pushed) and his Network connection to NETOLS to be broken. About three seconds elapse between the two events.

VIII. OLS Display

NETOLS suppresses all but alphameric display before it reaches the user. Alphameric display is mapped into the Telnet character set according to Figures 4 and 5. Figure 4 lists all those OLS display characters which have one-for-one mappings. A line of that figure reads as follows:

For '→':

The character logical not, displayed as '¬' on an OLS terminal, is represented in Telnet as '~' (tilde).

Alphabetics are mapped into upper-case alphabetics and Greek characters into lower-case alphabetics. Numerics are mapped into numerics, and a miscellany of punctuation into itself. In addition a number of carriage control characters are appropriately mapped--line feed down into LF, TAB into HT, BACK into BS, etc.; line feed up is suppressed. ERASE is represented as Bel.

Figure 5 lists those OLS display characters which are mapped into strings of Telnet characters. In most cases, these character strings are stylistic representations of characters peculiar to OLS. For example, the ADD key is normally displayed in List mode as '⊕'. In this Telnet implementation, '(+)' is an attempt to represent that graphic. Superscripts are represented as underscored numerics. Carriage return is represented as CR LF. No attempt is made to effectively represent RS which, on an OLS display device, repositions the beam to the upper left corner of the screen; it is made equivalent to carriage return.

IX. Instructing a User Telnet

For local users, all echoing that's done at all

To Display	(OLS Explanation)	OLS Sends	(Telnet Explanation)
0-9	Decimal Digits	0-9	Decimal Digits
A-Z	Alphabetics	A-Z	UC Alphabetics
α-ʃ	Greek Characters	a-z	LC Alphabetics
!	Exclamation Mark	!	Exclamation Mark
+	Plus Sign	+	Plus Sign
_	Underscore	_	Underscore
-	Minus Sign	-	Minus Sign
@	Commercial At	@	Commercial At
/	Slash	/	Slant
#	Number Sign	#	Number Sign
'	Apostrophe	'	Apostrophe
&	Ampersand	&	Ampersand
$	Dollar Sign	$	Dollar Sign
*	Asterisk	*	Asterisk
%	Percent	%	Percent
=	Equal Sign	=	Equal Sign
TAB	Horizontal Tab.	HT.	Horiz. Tab. (↑I)
:	Colon	:	Colon
;	Semi-Colon	;	Semi-Colon
[	Left Bracket	[	Left Bracket
]	Right Bracket	]	Right Bracket
(	Left Parenthesis	(	Left Parenthesis
)	Right Parenthesis	)	Right Parenthesis
<	Less Than	<	Less Than
>	Greater Than	>	Greater Than
,	Comma	,	Comma
.	Period	.	Period
"	Quotation Marks	"	Quotation Marks
?	Question Mark	?	Question Mark
¬	Logical Not	~	Tilde
|	Logical Or	|	Vertical Line
BACK SPACE	Backspace	BS	Backspace
	Space	SP	Space
ENL/↑	Line Feed Up		
CON/↓	Line Feed Down	LF	Line Feed (↑J)
	List Mode Space		Underscore
▨	List Mode Rubout	X	Upper-case X
	List Mode Pointer		Underscore
BREAK	Break	SP	Space
ERASE	Erase	BEL	Bell (↑G)

Figure 4. Characters With One-for-One Mappings

To Display	(OLS Explanation)	OLS Sends
:	Post List	(:)
⊕	List Mode Add	(+)
⊖	List Mode Subtract	(-)
⊙	List Mode Multiply	(*)
⊘	List Mode Divide	(/)
↙	List Carriage Return	(¦)
RETURN	Carriage Return	CR LF
RS	Reset to Upper Left	CR LF
¢	Cent Sign	C BS ¦
0-9	Superscript 0-9	0 BS _ - 9 BS _

Figure 5. Characters Which Map Into Strings

is done by OLS; the terminal never echos. In general, OLS does not echo the user's input. There are exceptions to this rule, but they are relatively few in number and occur primarily on the SYST level. In particular, upper keyboard keys are never echoed except in List mode. The Network user is advised to instruct his Telnet to operate in full-duplex mode, i.e. to echo nothing. The FULLDUPLEX command provided by NETOLS is provided because is can be provided easily, but its use is not recommended.

OLS is meant to be used in character-at-a-time mode, and the user should so instruct his User Telnet. For those users provided with only a line-at-a-time mode, the end-of-line character should not be transmitted to NETOLS.

NETOLS flushes without comment all Telnet control characters it detects in the input stream. Characters in the Telnet character set which have no meaning to NETOLS are echoed as '?' and discarded. Exceptions are LF (line feed) and NUL, which are flushed without comment.

X. Examples

A. Logon

The dialogue which logs a user onto OLS, assuming the user number of Section IV, is as follows:

TELNET ENTRY	OLS QUERY/RESPONSE
	UCSB ON-LINE SYSTEM
	ENTER USER NUMBER (196)
196%	ID NUMBER=
57372%	USER NAME= (ARPA)
ARPA%	JOB NAME= (UCSB-WHITE)
UCSB-WHITE%	AUTOSAVE CODE= integer
MOLSF%	LOAD (MOLSF)
	FILE LOADED

In this and succeeding examples, '%' denotes CR (carriage return). Entries echoed by OLS are enclosed in parentheses above. The user should substitute for 'UCSB-WHITE' his own affiliation and name. The procedure above loads the math subsystem of OLS. To load instead either COL or NET, substitute its name for 'MOLSF'. To load a different subsystem (say COL) after logging in:

TELNET ENTRY	OLS QUERY/RESPONSE
;SYST_	WORK AREAS UPDATED
;LOAD_COL%	LOAD (COL)
	FILE LOADED

Again, '_' denotes a space, not an underscore.

B. NEWTON-RAPHSON SQUARE ROOT APPROXIMATION

A simple user program can be constructed to approximate the square root of a number N using the Newton-Raphson iteration procedure which derives the $(k+1)^{th}$ approximation from the k^{th} by the following algorithm:

$$x_{k+1} = (x_k + N/x_k)/2$$

The following entries construct the user program:

```
;LIST_;TYPE_% ENTER_N
;1_ ;REAL_;LOAD_;ENTER_;STORE_N
;TYPE_% ENTER_FIRST_GUESS
;1_ ;LOAD_;ENTER_;STORE_X
;TYPE_% #_OF_ITERATIONS?
;0_ ;LOAD_;ENTER_;STORE_N
;1_ ;REPT_(;LOAD_N ;/_X ;+_X ;/_2
     ;STORE_X ;DISP_%)N% ;USER_;SQRT_
;LIST_;STORE_;USER_;1_;SQRT_
```

To display the user program, enter:

```
;USER_;DISP_;SQRT_
```

When executed, the program obtains from the user the number N whose square root is sought, an initial guess, and the number of iterations to be performed. The program then computes and displays the result of each iteration, and then calls itself, permitting a second square root to be computed. The program is executed as follows:

TELNET ENTRY	OLS QUERY/RESPONSE
;USER_;1_;SQRT_	ENTER N
3 ;ENTER_	ENTER FIRST GUESS
1 ;ENTER_	# OF ITERATIONS?
4 ;ENTER_	2. +00 1.75 +00 1.73214+00 1.73205+00
	ENTER N

etc.

C. Remote Job Entry

A file of card images can be constructed with the help of the COL subsystem of OLS and submitted as a batch job. Assuming COL has been loaded, the following entries construct a card file which invokes the Fortran compiler:

```
;2_ //jobname_JOB_(acct#,name,,,,,,T) ;STORE_
//_EXEC_FORTGCLG ;STORE_
//FORT.SYSIN_DD_* ;STORE_
source-statement-1 ;STORE_
          ...
source-statement-N ;STORE_
/* ;STORE_
```

To display the completed file, type:

```
;3_;DISP_%
```

To submit the file, type:

```
;4_;SUB_%
```

To watch for it in execution, type:

```
;DISP_J%%%...
```

When execution is complete, 'printed' output can be retrieved with the following dialogue:

TELNET ENTRY	OLS QUERY/RESPONSE
;CMPLX_;LOAD_	UNIT= (2314)
2314%	VOL=SER= (MVT180)
MVT180%	DSNAME= (RJEOUT)
RJEOUT%	MEMBER= (jobname)
jobname%	NOW LOADING
	FILE LOADED

The output can then be examined by entering:

;2_;DISP_1%%%...