Telnet Access To UCSB’s Online System

Contents

I. Motivation ...................................................1
II. Limitations ..................................................2
III. System Documentation .........................................2
IV. System Access ................................................3
V. Software Structure ............................................3
VI. Virtual OLS Keyboard .........................................4
VII. NETOLS Commands ..........................................10
  A. HELP ..................................................10
  B. PREFIX .............................................10
  C. SHIFT and UNSHIFT .....................................10
  D. FULLDUPLEX and HALFDUPLEX ............................10
  E. STATE ...............................................11
  F. LOGOUT .............................................11
VIII. OLS Display ................................................11
IX. Instructing User Telnet ......................................12
X. Examples ....................................................14
  A. Logon ...............................................14
  B. Newton-Raphson Square Root Approximation ..............15
  C. Remote Job Entry ......................................16

Figures

Figure 1. OLS Keyboard ...........................................2
Figure 2. Keys With One-for-One Mappings ........................6
Figure 3. Keys Represented as Strings ............................7
Figure 4. Characters With One-for-One Mappings .....................12
Figure 5. Characters Which Map Into Strings .......................13

I. Motivation

A teletype-compatible interface to UCSB’s Online 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 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 disregarded 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 transformation 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. [Please view the PDF version of this RFC.]

(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. This 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 Online 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, NTCLS 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 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 eh 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, and 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 [‘half arrow’ - see the PDF version of this RFC]:
```
The key labeled ['half arrow'] (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 Online 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 interprets the prefix, name, and space from them generates a single, 8-bit code, which 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_:

They 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 ';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, NETOLVS 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 ‘JN_’ 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 ignored.

<table>
<thead>
<tr>
<th>To Push</th>
<th>(OLS Explanation)</th>
<th>Send</th>
<th>(Telnet Explanation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9</td>
<td>Decimal Digits</td>
<td>0-9</td>
<td>Decimal Digits</td>
</tr>
<tr>
<td>A-Z</td>
<td>Alphabetics</td>
<td>A-Z</td>
<td>UC Alphabetics</td>
</tr>
<tr>
<td>_-5</td>
<td>Greek Characters</td>
<td>a-z</td>
<td>LS Alphabetics</td>
</tr>
<tr>
<td>!</td>
<td>Exclamation Mark</td>
<td>!</td>
<td>Exclamation Mark</td>
</tr>
<tr>
<td>+</td>
<td>Plus Sign</td>
<td>+</td>
<td>Plus Sign</td>
</tr>
<tr>
<td>_</td>
<td>Underscore</td>
<td>_</td>
<td>Underscore</td>
</tr>
<tr>
<td>-</td>
<td>Minus Sign</td>
<td>-</td>
<td>Minus Sign</td>
</tr>
<tr>
<td>@</td>
<td>Commercial At</td>
<td>@</td>
<td>Commercial At</td>
</tr>
<tr>
<td>/</td>
<td>Slash</td>
<td>/</td>
<td>Slant</td>
</tr>
<tr>
<td>#</td>
<td>Number Sign</td>
<td>#</td>
<td>Number Sign</td>
</tr>
<tr>
<td>’</td>
<td>Apostrophe</td>
<td>’</td>
<td>Apostrophe</td>
</tr>
<tr>
<td>&amp;</td>
<td>Ampersand</td>
<td>&amp;</td>
<td>Ampersand</td>
</tr>
<tr>
<td>$</td>
<td>Dollar Sign</td>
<td>$</td>
<td>Dollar Sign</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>*</td>
<td>Asterisk</td>
</tr>
<tr>
<td>%</td>
<td>Percent</td>
<td>%</td>
<td>Percent</td>
</tr>
<tr>
<td>=</td>
<td>Equal Sign</td>
<td>=</td>
<td>Equal Sign</td>
</tr>
<tr>
<td>TAB</td>
<td>Horizontal Tab</td>
<td>HT</td>
<td>Horiz. Tab. ([)</td>
</tr>
<tr>
<td>:</td>
<td>Colon</td>
<td>:</td>
<td>Colon</td>
</tr>
<tr>
<td>;</td>
<td>Semi-Colon</td>
<td>;</td>
<td>Semi-Colon</td>
</tr>
<tr>
<td>[</td>
<td>Left Bracket</td>
<td>[</td>
<td>Left Bracket</td>
</tr>
<tr>
<td>]</td>
<td>Right Bracket</td>
<td>]</td>
<td>Right Bracket</td>
</tr>
<tr>
<td>(</td>
<td>Left Parenthesis</td>
<td>(</td>
<td>Left Parenthesis</td>
</tr>
<tr>
<td>)</td>
<td>Right Parenthesis</td>
<td>)</td>
<td>Right Parenthesis</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less Than</td>
<td>&lt;</td>
<td>Less Than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater Than</td>
<td>&gt;</td>
<td>Greater Than</td>
</tr>
<tr>
<td>.</td>
<td>Comma</td>
<td>.</td>
<td>Comma</td>
</tr>
<tr>
<td>&quot;</td>
<td>Quotation Marks</td>
<td>&quot;</td>
<td>Quotation Marks</td>
</tr>
<tr>
<td>?</td>
<td>Question Mark</td>
<td>?</td>
<td>Question Mark</td>
</tr>
<tr>
<td>[half arrow]</td>
<td>Logical Not</td>
<td>˜</td>
<td>Tilde</td>
</tr>
<tr>
<td></td>
<td>Logical Or</td>
<td></td>
<td>Vertical Line</td>
</tr>
<tr>
<td>BACK</td>
<td>Backspace</td>
<td>BS, DET</td>
<td>Backspace/Rubout</td>
</tr>
<tr>
<td>RETURN</td>
<td>Carriage Return</td>
<td>CR</td>
<td>Carr. Return (\M)</td>
</tr>
<tr>
<td>SPACE</td>
<td>Space</td>
<td>SP</td>
<td>Space</td>
</tr>
</tbody>
</table>

Figure 2. Keys With One-for-One Mappings
To Push | (OLS Explanation) | The Key Name Is
---|---|---
[center .] | Multiply | * 1
[center +] | Add | +
[center -] | Subtract | -
RETURN | Carriage Return | .
[center /] | Divide | /
L0 | Level 0 | 0
LI | Level I | 1
L II | Level II | 2
L III | Level III | 3
L IV | Level IV | 4
L V | Level V | 5
L VI | Level VI | 6
L VII | Level VII | 7
[center +] | Add | ADD 2
ARC | Argument | ARG
ATAN | Arc Tangent | ATAN
BACK | Backspace | BACK 3
CASE | Case | CASE
[center sign] | Cent | SignCENT
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
[center /] | 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
[down arrow] | Line Feed Down | LFDN
[up arrow] | Line Feed Up | LFUP

Figure 3. Keys Represented As Strings
<table>
<thead>
<tr>
<th>To Push</th>
<th>(OLS Explanation)</th>
<th>The Key Name Is</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>List</td>
<td>LIST</td>
</tr>
<tr>
<td>LOAD</td>
<td>Load</td>
<td>LOAD</td>
</tr>
<tr>
<td>LOG</td>
<td>Logarithm</td>
<td>LOG</td>
</tr>
<tr>
<td>LS</td>
<td>Left Shift</td>
<td>LS</td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum</td>
<td>MAX</td>
</tr>
<tr>
<td>MOD</td>
<td>Modulus</td>
<td>MOD</td>
</tr>
<tr>
<td>MULT</td>
<td>Multiply</td>
<td>MULT 6</td>
</tr>
<tr>
<td>NEG</td>
<td>Negate</td>
<td>NEG</td>
</tr>
<tr>
<td>OR</td>
<td>Logical Or</td>
<td>OR 8</td>
</tr>
<tr>
<td>PRED</td>
<td>Predicate</td>
<td>PRED</td>
</tr>
<tr>
<td>PROD</td>
<td>Running Product</td>
<td>PROD</td>
</tr>
<tr>
<td>PT</td>
<td>Point</td>
<td>PT</td>
</tr>
<tr>
<td>PWR</td>
<td>Power</td>
<td>PWR</td>
</tr>
<tr>
<td>REAL</td>
<td>Real</td>
<td>REAL</td>
</tr>
<tr>
<td>REFL</td>
<td>Reflect</td>
<td>REFL</td>
</tr>
<tr>
<td>REPT</td>
<td>Repeat</td>
<td>REPT</td>
</tr>
<tr>
<td>RESET</td>
<td>Reset</td>
<td>RESET</td>
</tr>
<tr>
<td>RETURN</td>
<td>Carriage Return</td>
<td>RETURN 9</td>
</tr>
<tr>
<td>RS</td>
<td>Right Shift</td>
<td>RS</td>
</tr>
<tr>
<td>S0-S9</td>
<td>Superscript 0-9</td>
<td>S0-S9</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select</td>
<td>SELECT</td>
</tr>
<tr>
<td>SET</td>
<td>Set Tab</td>
<td>SET</td>
</tr>
<tr>
<td>SHIFT</td>
<td>NETOLS Command</td>
<td>NETOLS Command</td>
</tr>
<tr>
<td>SIN</td>
<td>Sine</td>
<td>SIN</td>
</tr>
<tr>
<td>SORT</td>
<td>Sort</td>
<td>SORT</td>
</tr>
<tr>
<td>SQ</td>
<td>Square</td>
<td>SQ</td>
</tr>
<tr>
<td>SQRT</td>
<td>Square Root</td>
<td>SQRT</td>
</tr>
<tr>
<td>STATE</td>
<td>NETOLS Command</td>
<td>STATE</td>
</tr>
<tr>
<td>STORE</td>
<td>Store</td>
<td>STORE</td>
</tr>
<tr>
<td>SUB</td>
<td>Substitute</td>
<td>SUB</td>
</tr>
<tr>
<td>SUBTRACT</td>
<td>Subtract</td>
<td>SUBTRACT 10</td>
</tr>
<tr>
<td>SUM</td>
<td>Running Sum</td>
<td>SUM</td>
</tr>
<tr>
<td>SYST</td>
<td>System</td>
<td>SYST</td>
</tr>
<tr>
<td>TEST</td>
<td>Test</td>
<td>TEST</td>
</tr>
<tr>
<td>TYPE</td>
<td>Type</td>
<td>TYPE</td>
</tr>
<tr>
<td>UNSHIFT</td>
<td>NETOLS Command</td>
<td>UNSHIFT</td>
</tr>
<tr>
<td>UP</td>
<td>Up</td>
<td>UP</td>
</tr>
<tr>
<td>USER</td>
<td>User</td>
<td>USER</td>
</tr>
</tbody>
</table>

Figure 3 (cont’d) Keys Represented As Strings
1. Alternate names for [circle .], [circle +], [circle -], RETURN, and [circle /] 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 [circle +] is '++'

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

4. An alternate name for [circle /] is 'DIV'.

5. NETOLS commands are explained in Section VII.

6. An alternate name for [circle .] is 'MULT'.

7. An alternate for '[half arrow]' is the single character '˜' (tilde), as indicated in Figure 2.

8. An alternate for '|' is the single character '[2 vertical lines]' (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 [circle -] 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 run 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%$’ ('%' denotes carriage return) is interpreted as the lower-keyboard key ‘S’.

VII. NETOLS Commands

A number of commands to LETOLS are defined and all are described in this section. The format for each such command is the same as that for the 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 NEOLS 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 defaults 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 character, which have one-for-one mappings. A line of that figure reads as follows:

```
For '[half harrow]'  

The character logical not, displayed as '[half arrow]' on an OLS terminal, is represented in Telnet as '˜' (tilde).
```

Alphabets are mapped into upper-case alphabets and Greek characters into lower-case alphabets. 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 '[circle +]'. 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_

<table>
<thead>
<tr>
<th>To Display</th>
<th>(OLS Explanation)</th>
<th>OLS Sends</th>
<th>(Telnet Explanation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9</td>
<td>Decimal Digits</td>
<td>0-9</td>
<td>Decimal Digits</td>
</tr>
<tr>
<td><em>-</em></td>
<td>Alphabetics</td>
<td>A-ZUC</td>
<td>Alphabetics</td>
</tr>
<tr>
<td>_-5</td>
<td>Greek Characters</td>
<td>a-z</td>
<td>LC Alphabetics</td>
</tr>
<tr>
<td>!</td>
<td>Exclamation Mark</td>
<td>!</td>
<td>Exclamation Mark</td>
</tr>
<tr>
<td>+</td>
<td>Plus Sign</td>
<td>+</td>
<td>Plus Sign</td>
</tr>
<tr>
<td>_</td>
<td>Underscore</td>
<td>_</td>
<td>Underscore</td>
</tr>
<tr>
<td>-</td>
<td>Minus Sign</td>
<td>-</td>
<td>Minus Sign</td>
</tr>
<tr>
<td>@</td>
<td>Commercial At</td>
<td>@</td>
<td>Commercial At</td>
</tr>
<tr>
<td>/</td>
<td>Slash</td>
<td>/</td>
<td>Slant</td>
</tr>
<tr>
<td>#</td>
<td>Number Sign</td>
<td>#</td>
<td>Number Sign</td>
</tr>
<tr>
<td>'</td>
<td>Apostrophe</td>
<td>'</td>
<td>Apostrophe</td>
</tr>
<tr>
<td>&amp;</td>
<td>Ampersand</td>
<td>&amp;</td>
<td>Ampersand</td>
</tr>
<tr>
<td>$</td>
<td>Dollar Sign</td>
<td>$</td>
<td>Dollar Sign</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>*</td>
<td>Asterisk</td>
</tr>
<tr>
<td>%</td>
<td>Percent</td>
<td>%</td>
<td>Percent</td>
</tr>
<tr>
<td>=</td>
<td>Equal Sign</td>
<td>=</td>
<td>Equal Sign</td>
</tr>
<tr>
<td>TAB</td>
<td>Horizontal Tab.</td>
<td>HT.</td>
<td>Horiz. Tab (_I)</td>
</tr>
<tr>
<td>:</td>
<td>Colon</td>
<td>:</td>
<td>Colon</td>
</tr>
<tr>
<td>;</td>
<td>Semi-Colon</td>
<td>;</td>
<td>Semi-Colon</td>
</tr>
<tr>
<td>[</td>
<td>Left Bracket</td>
<td>[</td>
<td>Left Bracket</td>
</tr>
<tr>
<td>]</td>
<td>Right Bracket</td>
<td>]</td>
<td>Right Bracket</td>
</tr>
<tr>
<td>(</td>
<td>Left Parenthesis</td>
<td>(</td>
<td>Left Parenthesis</td>
</tr>
<tr>
<td>)</td>
<td>Right Parenthesis</td>
<td>)</td>
<td>Right Parenthesis</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less Than</td>
<td>&lt;</td>
<td>Less Than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater Than</td>
<td>&gt;</td>
<td>Greater Than</td>
</tr>
<tr>
<td>,</td>
<td>Comma</td>
<td>,</td>
<td>Comma</td>
</tr>
<tr>
<td>&quot;</td>
<td>Quotation Marks</td>
<td>&quot;</td>
<td>Quotation Marks</td>
</tr>
<tr>
<td>?</td>
<td>Question Mark</td>
<td>?</td>
<td>Question Mark</td>
</tr>
<tr>
<td>[half arrow]</td>
<td>Logical Not</td>
<td>[</td>
<td>Logical Not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logical Or</td>
<td></td>
</tr>
</tbody>
</table>

BACK      Backspace      BS       Backspace
SPACE     Space          SP       Space
ENL/[up arrow] Line Feed Up
CON/[down arrow] Line Feed Down LF Line Feed ([up arrow]J)
_          List Mode Space _     Underscore
[shaded rectangle] 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               (:)  
[circle +]         List Mode Add          (+)  
[circle -]         List Mode Subtract       (-)  
[circle .]          List Mode Multiply         (*)  
[circle /]          List Mode Divide          (/)  
[arrow]             List Carriage Return       ([2 vertical lines])  
RETURN            Carriage Return          CR LF  
RS                 Reset to Upper Left        CR LF  
[cent sign]         Cent Sign               C BS [2 vertical lines]  
0-9                Superscript 0-9            0 BS _  
                     -                       _  
                     9 BS _

Figure 5. Characters Which Map Into Strings

is done by OLS; the terminal never echoes. 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 it can be provided, 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 followings:

TELNET ENTRY                  OLS QUERY/RESPONSE

UCSB ONLINE 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. NEWTOWN-RAPSHON SQUARE ROOT APPROXIMATION

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

\[ X_{k+1} = \frac{x_k + n/k_k}{2} \]

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_INTERATIONS?
;0_;LOAD_;ENTER_;STORE_N
;1_;REPT_(;LOAD_N ;/_X;+_X ;/_2
   ;STORE_X ; DISP_%/_X ;+_X ;/_2
;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 results 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:

```plaintext
;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:

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

To submit the file, type:

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

To watch for it in execution, type:

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

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

<table>
<thead>
<tr>
<th>TELNET ENTRY</th>
<th>OLS QUERY/RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>;CMPLX_;LOAD_</td>
<td>UNIT = (2314)</td>
</tr>
<tr>
<td>2314%</td>
<td>VOL=SER= (MVT180)</td>
</tr>
<tr>
<td>MVT180%</td>
<td>DSNAME= (RJEOUT)</td>
</tr>
<tr>
<td>RJEOUT%</td>
<td>MEMBER= (jobname)</td>
</tr>
<tr>
<td>Jhostname%</td>
<td>NOW LOADING</td>
</tr>
<tr>
<td></td>
<td>FILE LOADED</td>
</tr>
</tbody>
</table>

The output can then be examined by entering:

```plaintext
;2_;DISP_1%%%
```

NOTE: Text within brackets describes non-ASCII characters that were part of the original document. Please see the PDF file for the original representation.