MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

Artificial Intelligence
Memo. No. 292

April 1974

U T

Donald E. Eastlake

UT is a user telnet program designed to run under the ITS time sharing system. It implements the relatively recent ARPA network negotiating protocol for telnet connections.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

1. INTRODUCTION

UT is a user (as opposed to server) telnet program designed to run under the ITS time sharing system. It implements the relatively recent ARPA network negotiating protocol for telnet connections.

The basic philosophy of UT is to have a number of "ports", as explained in section 2, each of which represents a complex external interface to a user's console, a network connection, or a file. The relatively simple internal interfaces of the "ports" can be interconnected in various ways.

A variety of commands can be detected by UT at each port as explained in section 3.

Using the basic facilities of UT is not as complicated as it may seem. If you just keep in mind that commands are preceded by a <CNTRL-UP-ARROW> and read the descriptions of the first few commands starting at subsection 3.1, it should be possible to carry on network conversations.

2. PORTS

All of UT's input and output is done through ports each of which is a logical bidirectional channel. Each port has both an input and an output side of which only one is used in some cases. In the simplest case of network use through UT, the user's console is on one port and a network connection to another computer system in on a second port. These are ports with both input and output open in the obvious way. They would be interconnected through UT so that the input on each port is directed to the output side of the other. Depending on how various echoing switches are set, the input of either, both, or neither of these two ports could also be connected to its own output side.

The maximum number of ports that can exist at one time in UT is seven. Each port has a unique name up to six characters long which is the only way the user can generally refer to ports in commands to UT. The primordial port on which the user's console is open has the name "TTY". In commands, the port by which the command is being executed can be referred to by the special name "self".

Among the flags kept for a port are the following:

a) If case shifting is in effect.

b) If the last character in was an effective <CNTRL-Q>.

c) If protocol messages should be typed out when the port has a teletype open for output on it.

d) The desired and actual believed local and remote states of echoing and go-ahead suppression.

e) If output on that port is clogged.

f) If input on that port is blocked because it would result in output to a clogged port.

g) The relative count of data marks and out of band interrupts received on the port.

There are actually two types of connections between ports, "in-connection" and "output-monitoring". If port A is in-connected to port B, input on port A is output on port B. If port A is output-monitored to port B, then all output on port A, except that from an output-monitor connection, is also output on port B. Thus the spread of an ordinary character input to a port is as follows: 1a) it may be echoed back out by UT; 1b) if echoed, it will also be output on all ports output-monitoring the original port; 2a) it is output on all ports in-connected from the original port, if any; 2b) for each in-connected port, it will also be output on all ports output-monitoring the in-connected port.

3. COMMANDS

Commands can be detected on input to any port (even several commands on several ports "simultaneously") if that port is "sensitive" to the particular command type. There are five types of commands as follows: 1) new telnet protocol messages (beginning with <IAC>), 2) old telnet protocol characters, 3) UT commands beginning with <CNTRL-UP-ARROW>, 4) <CNTRL-Q>, and 5) \ (backslash). The initial user console port is sensitive to all types of commands. The initial sensitivity of other ports is as described with the command that creates the port. (See also the sensitize and desensitize commands.)

Taking the above types of commands in reverse order, \ simply changes the UT case shift state for that port for the next character only (see also the shift and unshift commands). <CNTRL-Q> desensitizes the port for the following character only so that it will not be recognized if it would normally be a command. <CNTRL-UP-ARROW> indicates the beginning of a UT command; if the immediately following character is one listed in section 3.0 then the two characters together constitute the entirety of the command, otherwise a multicharacter command will be read as described in the next paragraph. Old telnet protocol characters, which mostly affect echo, basicly take effect by being passed through on input and noticed on output to a port with a teletype on it. New telnet protocol commands are parsed and, if understood (UT currently knows only about the echo and suppress-go-ahead options), agreed to. UT initially tries to negotiate remote echo and go-aheads suppressed both ways. Go-aheads received by UT have no effect. UT will transmit go-aheads by an idleness time-out mechanism to other systems that insist on it. Understood telnet protocol negotiation messages are acted on at the receiving port and not passed through although most other new telnet protocol signals are passed through. The desired propagation of echo changes occurs by the following mechanism. Whenever an echo negotiation message is received that requests a change, it is agreed to and a message of opposite polarity and sense (DO goes to WNT, DNT goes to WIL, etc.) is simulated as having been received on every port in-connected from the port on which the original request was received.

"Multicharacter" commands to UT are read, for each port, into a buffer until terminated by a carriage return or line feed. Characters can be rubbed out and <RUBOUT>, <CARRIAGE-RETURN>, etc. can be put in by preceeding them with <CNTRL-Q>. After the command is terminated, it is then parsed, usually entirely into syllables, but at least into an initial command word syllable and the remainder of the command. Syllables are terminated by ":", ";", "<", ">", or any character whose ASCII code is 40 or less. Syllables ending with ":" and ";" are usually recognized as devices and system names, those ending with ">" or "<" are normally recognized as ports, and all other syllables are called plain and may be files names, host names, etc. Any of the termination characters can be included in a syllable by preceeding it with <CNTRL-Q>. (<CNTRL-Q> always works in this context of a command regardless of the port's sensitivity to it.) The first syllable, regardless of its type, is looked up as a command in the table mostly listed by the help command. If it is found it then controls the parsing of the rest of the command. In general, the position of the device, system name, and port type syllables is not significant. Later ones override ones to their left and they are skipped when plain syllables are being looked for.

In most cases, syllables are looked up in tables in such a way that the unambiguous start of a table entry will be recognized and its completion typed out. The multicharacter commands in the headings of the parts of this section show by their initial capital letters how much is necessary to specify the command. In some cases, an attempt is made to parse a syllable, especially if otherwise unrecognized, as an octal number.

3.0 "Single Character" Commands

The following characters have the effects given when following a <CNTRL-UP-ARROW> on a sensitive port:

A Equivalent to typing an Are-You-There new protocol message.
B Equivalent to typing a Synch new protocol message (break).
C Equivalent to typing an Interrupt-Process new protocol message.
G Equivalent to typing a Go-Ahead new protocol message.
H Equivalent to typing an Erase-Character new protocol message.
N Directly turns off local echo (No echo).
S Equivalent to typing an Erase-Output new protocol message.
X Equivalent to typing an Erase-Line new protocol message.
Y Directly turns on local echo (Yes echo).
<CNTRL-UP-ARROW> Equivalent to typing this character on an unsensitive port.
? Prints a list of "single character" commands and also includes in the list the help command to tell you how to get a list of multicharacter commands.
\ Permanently complements UT's case shift state (as opposed to \ by itself which just complements it for the next character).
<UP-ARROW>
<BACK-ARROW> These two characters put UT input on the port into a mode where it reads successive sets of three octal digits as characters (values up to 377 are allowed for protocol characters). The only difference between them is that <UP-ARROW> puts an <IAC> in front of everything.

3.1 HElp
LISTCommands

These commands print out a list of most of the multi-letter commands available in UT. Each command listed is followed by its dispatch address in UT, then an asterisk in some cases, and finally a brief comment. If an ambiguous partial command is entered but only one of the possible completions has an asterisk after it, that one will be chosen.

The first command listed is "?" even though this is not a multi-letter command. It is there to remind the user that he can get a list of single character commands by typing "<CNTRL-UP-ARROW>?".

3.2 Connect [port] [host [socket]]

This command is the basic command for making connection over the network. If no port is specified, a port name will be gensymed by UT. If no host name or number is given, the last host designation used will be used again (this is initially ML for mathlab). If a host name or number is present, a socket number may also be specified which will be used, for that connect only, instead of the default socket number 1.

If all goes well, a network connection will be made on the port to the host [socket] specified. The network port will be initially sensitized only to old and new telnet protocol commands. This port and the port through which the connect command was entered (usually TTY) are then in-connected to each other.

In case of a serious error or a time out in trying to establish a network connection, the port is obliterated as if by a disconnect command. The time allowed for response by the logger of the system being connected to is the larger of 12. seconds and the UT's patience setting (see the patience command).

3.3 HOsts
LISTHosts

These commands list the host names recognized by the connect command. Each host name is followed by its network host number.

3.4 Disconnect [port]

This is the only command available to close out the file(s) or connection(s) open on a port and free up the port slot. In the process, all in-connection and output-monitor links from or to the port, are severed and the port name is obliterated.

If more than one port is entered after the disconnect command, only the last one will be disconnected. If no ports are listed after the command, all ports in-connected from the port on which the command is entered will be disconnected (ports just output-monitoring the command port are unaffected).

3.5 Listports
POrts

These commands list all active port slots and give the following information for each:

Port name;

Input type, if blocked, and file name;
List of ports in-connected to;
Output type, if blocked, and file name;
List of ports output-monitored to.

An input or output blocked state is inidicated by printing "<>".

(For debugging purposes the effect of these commands can be produced from a HACTRN with a UT as its currently selected inferior by typing "PUSHJ P,LP$X".)

3.6 MONitor [[port2] port1] [file]

All the characters being output on a particular port (except those coming from an output-monitor link) can be monitored by being output through another port to a specified file with this command. First a file is opened for output on a port determined as described below. If any part of the file description is not specified, it defaults to that last used with a monitor command from any port and is initally DSK: UT.OUT >. An output-monitor link is then made to the port on which the file has been opened from a port determined as described below.

If no port is specified with the command, the port on which the command is entered is output-monitored to a port with a gensymed name on which the file was opened. If exactly one port is specified, that port is output-monitored to a port with a gensymed name on which the file was opened. If two or more ports are specified the last is output-monitored to the file opened on the next to the last named port.

Since it is thought that it will be frequently desired to monitor telnet conversations spanning connection and disconnection with several remote hosts and to monitor parts of conversations with single remote hosts, the disconnect command defaults in such a way as to not automatically close ports output-monitored from the port on which the disconnect is entered.

3.7 Xfile [port] [file]

This command opens a file for input on a port sensitive to all types of commands except old and new telnet protocol commands. If no port is specified, a port name is gensymed. If any part of the file description is not specified, it defaults to that last used with an xfile command from any port and is initially set by the init file feature (see section 4.1).

3.8 As-if-from port command

This command enables multicharacter UT commands to be entered on one port but executed as though input on another port. It is unique in that the first port name after the command word is controlling and their must be a port specified. All of the characters after the ">" or "<" terminating the port name are transfered to the command buffer or port specified unless the buffer is already in use in which case "BUSY" is typed out. You should not put an additional <CNTL-UP-ARROW> in the as-if-from'ed command.

3.9 IBreak \ IMake \ OBreak \ OMake port1 port2

These commands respectively break and make in-connection links and break and make output-monitoring links. In each case a unidirectional link from port1 to port2 is affected. At least two ports must be specified and if more than two are specified, the rightmost two are used.

3.10 DEsensitize \ SEnsitize characters

These commands allow the sensitivity of a port to various command types to be set. All the characters after the character terminating the command word are scanned and the port for which the command is being executed is appropriately sensitized or desensitized if a \, <CNTRL-Q>, <CNTRL-UP-ARROW>, N (for new telnet protocol), or O (for old telnet protocol) is found. To affect a port other than the command entry port, use the as-if-from command.

3.11 IAc [port] protocol-words

This command allows arbitrary new telnet protocol messages to be sent out from a specified port, or, if none is specified, from every port in-connected from the command entry port. Octal numbers for protocol characters as well as the following symbols can follow the iac command: AO, AYT, BRK, DNT, DM, DO, EC, ECHO, EL, GA, IAC, IP, NOP, SB, SE, SGA, WIL, and WNT. The character sequence represented by the arguments to the iac command is then transmitted with an <IAC> in front of it.

3.12 IGnore \ UNIgnore characters

UT has a one bit flag for each character with a network code of less than 200. Initally these flags are off for all characters except the null character, code 0. Characters for which the flag is on are ignored on input unless they occur in a command. This ignore flag can be turned on or off for various characters with these commands. The characters affected are all those from just after the break character terminating the command word to the end of the command line. (Carriage return, line feed and <CNTRL-Q> can be included by preceeding them with <CNTRL-Q>.)

3.13 Interrupt [port]

This command causes a telnet interrupt to be output on a single port, if one is specified. Otherwise, an interrupt is sent on all ports that are in-connected from the port on which the interrupt command was entered. The interrupt signal referred to here is an out of band signal and a matching data mark in the data stream. They should cause most of the data sent over the telnet connection before the data mark to be skipped by the receiver.

3.14 MORe \ NOmore

The **MORE** feature of the ITS system is turned on or off for a port by entering these commands through the port. The **MORE** feature is initially off and when on it causes output to a display terminal to stop at the bottom of the screen until the user types a space.

3.15 NEw \ OLd

UT is somewhat compatible with both the new and old telnet protocols. It reponds properly to the receipt of either type of protocol message. The switch set by these commands sets the protocol used in transmitting some messages over telnet connections.

3.16 OPaque \ Transparent

The characters <CNTRL-Z> and <CNTRL-LEFT-ARROW> are normally interpreted by the ITS system and not let through to user programs. This is the opaque state. In the transparent state, these two special characters are transmitted through to user programs by ITS. These commands set user console connections on port on which they are entered to the state they name.

3.17 PAtience [number]

UT attempts to detect certain hung up states by timing then to see if they exceed a patience threshold. This is initially 5 seconds and can be set by a patience command followed by the octal number of second desired. Whether or not the patience command is followed by an argument, it types out the resulting patience setting as a decimal number of seconds.

3.18 PRInt arg

This command with no argument prints out "OFF", "ON", or "FANCY" as the protocol message flag for the port the command is entered on is set to inhibit print outs, print out in octal, or print out mnemonically. The command may be followed by any of the three print out words or 0, 1, or -1, respectively, to set the print out flag for the port.

3.19 PROceed \ Quit

These commands respectively valret a ":PRCED" and a ":KILL" to HACTRN.

3.20 SHift \ UNShift characters

These commands are similar to the ignore and unignore commands in that they set or clear a bit globally associated with each possible character code in UT. If the shiftable bit is on for a character it has 40 xored with it if UT is in the shifted state (see "<CNTRL-UP-ARROW>\" in section 3.0). Initially only the upper and lower case letters are shiftable.

3.21 TEst [args]

This command is for testing the parsing routines in UT and just prints its arguments back as parsed.

4. FEATURES

4.1 Init File

When UT is first started, it is as though the user had just typed "<CNTRL-UP-ARROW>XFILE _UT .INIT.". This will automaticly execute a file of that name if one exists in the user's directory. If not, "NO INIT FILE" is typed out as is always the case when a file is not found whose second name is .INIT.. The user may find it particularly convenient to set switches for the TTY port with AS-IF-FROM TTY> . . . commands in the init file. To see what is going on, you can put a "<CNTRL-UP-ARROW>OMAKE SELF> TTY>" at the start of the file to output-monitor the xfile port to your console.

4.2 Direct Protocol Negotiations

In some sense, the user has complete freedom to conduct protocol negotiations. You can make a telnet connection, turn on protocol printing for your console's port (see print command), and desensitize the network port to <IAC>. Then you can use the iac command to send direct protocol messages and the replies will just be passed through UT and typed out on your console.