DEFENSE DATA NETWORK/TOPS-20 TUTORIAL
AN INTERACTIVE COMPUTER PROGRAM

by
Mark Klause Herman Herkert
and
Sheri Lynn Smith
December 1985

Thesis Advisor: Gary K. Poock

Approved for public release; distribution is unlimited.
DEFENSE DATA NETWORK/TOPS-20 TUTORIAL: AN INTERACTIVE COMPUTER PROGRAM

The DDNTUT program is an interactive tutorial, located on the ISIA host computer in Marina del Rey, California. The program is designed to first acquaint the new user on the Defense Data Network with the TOPS-20 operating system and its executive level commands. Following this, the user is taken step by step through nine different programs available on TOPS-20, knowledge of which is integral to effective use of the network. The user can, at any time, quit the tutorial, return to previously studied sections, and/or skip sections he may be familiar with.
Approved for public release; distribution is unlimited.

Defense Data Network/TOPS-20 Tutorial
an Interactive Computer Program

by

Mark K. H. Herkert
Lieutenant Commander, United States Navy
B.S., University of Wisconsin, 1974

Sheri L. Smith
Lieutenant, United States Navy
B.S., Texas A & M University, 1978

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Command, Control and Communications)

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

Author: Mark Klaus Herman-Herkert

Author: Sheri Lynn Smith

Approved by: Gary Kent Droock, Thesis Advisor

Thomas Jay Brown, Second Reader

Michael G. Sovereign, Chairman, Command, Control and Communications Academic Group

David A. Schrady, Academic Dean

2
The DDNTUT program is an interactive tutorial, located on the ISIA host computer in Marina del Rey, California. The program is designed to first acquaint the new user on the Defense Data Network with the TOPS-20 operating system and its executive level commands. Following this, the user is taken step by step through nine different programs available on TOPS-20, knowledge of which is integral to effective use of the network. The user can, at any time, quit the tutorial, return to previously studied sections, and/or skip sections he may be familiar with.
# Table of Contents

I. History of the Defense Data Network
   A. Introduction
   B. Packet Switching
   C. ARPANET
   D. AUTODIN II
   E. Formation of the DDN

II. An Overview of the Defense Data Network
    A. Introduction
    B. The Backbone
    C. The Access Network

Appendix A: User's Guide

Appendix B: Maintenance Manual

Initial Distribution List
I. HISTORY OF THE DEFENSE DATA NETWORK

A. INTRODUCTION

The Defense Data Network (DDN), an approaching worldwide military data communications networking system, was founded by and remains under the aegis of the Defense Communication Agency (DCA) for the Department of Defense (DoD). The DDN is not one, but rather several networks, with a wide variety of compatible hardware and software which allows interoperability and communications. The most commonly used of these networks are the ARPANET (Advanced Research Projects Agency Network), the MILNET (an unclassified operational Military Network) and the MINET (Movement Information Network, which is relatively new.

A look into how these networks and others of a similar nature came into existence must, perforce, start with the development of the concept of "packet switching."

B. PACKET SWITCHING

Packet switching is a means of handling data being transmitted through the various communication channels or linkages within a network. The theory is simple: local Interface Message Processors (IMPs) break up or subdivide the messages, files, programs, etc., being transmitted, into smaller "packets" which are then treated as individual
messages, all labeled and sequentially ordered as part of the larger message. Located throughout the network are "switching nodes", which receive each packet and check for and correct any errors or glitches which may have occurred during transmission. The node will then either collect and order all of the separate packets of a message before forwarding the completed whole to the intended recipient, or will simply forward each individual packet on as it is received and corrected. (See Figure 1.1.)

Packet switching is highly cost effective: switching nodes are completely automated, as well as reliable and inexpensive. Easy to install and keep up, there are now over a hundred of them scattered across the USA, and spreading rapidly throughout Europe and the Far East. Encryption of the system to permit vastly increased classified material handling capability has been budgeted for and is more slowly being incorporated throughout the system, with full online capabilities expected by the late 1980s.

C. ARPANET

The first of the packet switching networks was developed under a 1969 program run by the Defense Advanced Research Project Agency (DARPA). The experimental network thus conceived and put into operation was dubbed ARPANET. The network became a major success in the rapid, reliable,
and inexpensive sharing of data, ideas, theories, resources, and software. By 1975, a considerable number of operational users as well as researchers and experimenters had been authorized access to the net, and control of the asset was shifted to the Defense Communication Agency. Both military and civilian research projects are supported by ARPANET. The knowledge and experience, tools and services developed over the years have formed the basis for most if not all packet switching networks, both commercial and military, currently under development or in use today.

D. AUTODIN II

A DoD study initiated in 1974 predicted such imminent major advances and increases in the use of computer technology by the military services that a decision was made to develop a packet switching network dedicated to and internal to the DoD itself. This network was to have the full heterogeneous hardware and software capabilities already found in ARPANET, and was, additionally, to have the added features of standard military message precedence handling and encrypted transmission for classified material.

In 1976, the contract for this new network was awarded to Western Union and given the name AUTODIN II. (AUTODIN I is the Western Union message switching network currently leased by the government since the early 1960's). Between 1976 and 1982, however, improvements in technology and
increased cost consciousness on the part of the government, coupled with an added emphasis on redundancy and survivability, resulted in the government terminating the AUTODIN II contract and the subsequent development of the DDN under the DCA.

E. FORMATION OF THE DDN

DoD, in 1979, began interconnecting a number of its isolated computer networks through internet protocols resulting in a series of node computers linked by high-speed telephone lines. This was to become the "backbone" of the DDN. By 1983 ARPANET had grown to include over 300 computers, and was woefully inadequate in its ability to accommodate the increasing numbers of military would-be ARPANET users. Thus, in September 1984, ARPANET was officially divided into two separate unclassified networks, with military research and development remaining under the ARPANET heading, and control reverting to DARPA, while operational military communication uses were directed to the newly designated MILNET, which remained under DCA control. Electronic mail can still be sent between the two networks, but, as security safeguards are added to the MILNET, traffic will increasingly be limited and controlled by so-called "gateways".
II. AN OVERVIEW OF THE DEFENSE DATA NETWORK

A. INTRODUCTION

The DDN is envisioned as a worldwide unified packet switching network dedicated to meeting the data communication requirements of the DoD. The network is subdivided into two functional areas: (1) the network backbone, comprising the trunk circuits and switching nodes, and (2) the access network, comprising local circuits and interfacing equipment allowing individual authorized users to connect to the backbone. TOPS-20 is one of several timesharing operating system in current usage on the DDN. Additional information on TOPS-20 can be found in the <DOCUMENTATION> directory, as noted in the tutorial.

B. THE BACKBONE

The DDN backbone (IMPs and TACs) will eventually consist of some 200 nodes located at some 100 sites. Again, as mentioned earlier, redundancy and survivability are of key interest to DoD. Most of the transmission trunks will be via leased landline circuits, although some satellite transoceanic links are already in operation. (See Figure 2.1.)

C. THE ACCESS NETWORK

Individual user terminals normally access the DDN via a host computer, which is in turn connected to the
Figure 2.1
aforementioned IMP. It is also possible, however, to connect a hardwired or dial-up terminal to the DDN via a Terminal Access Controller (TAC). (See Figure 2.2.)

Each host computer has a hostname and address as its primary means of identification. Additionally, each network has a network address associated with it. While hostnames (usually mnemonic acronyms) are all that is necessary for sending electronic mail around the net, should you access the net via a TAC, you must use the equivalent numerical host address instead.

Examples of hostnames and addresses are:

<table>
<thead>
<tr>
<th>Host address</th>
<th>Host name</th>
<th>Location</th>
<th>Network type</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.1.0.7</td>
<td>MINET-LON-TAC</td>
<td>DCA London</td>
<td>MINET</td>
</tr>
<tr>
<td>26.3.0.16</td>
<td>AMES-VNSB</td>
<td>NASA, Moffet Fld, CA</td>
<td>MILNET</td>
</tr>
<tr>
<td>10.0.0.15</td>
<td>ROCHESTER</td>
<td>UNIV of Rochester, NY</td>
<td>ARPANET</td>
</tr>
</tbody>
</table>
APPENDIX A

USER'S GUIDE

TO THE

DEFENSE DATA NETWORK/TOPS-20 TUTORIAL
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>HOW TO ACCESS THE TUTORIAL/USE THE GUIDE</td>
<td>16</td>
</tr>
<tr>
<td>II.</td>
<td>EXECUTIVE LEVEL COMMANDS</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>A. CURSOR COMMANDS</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>B. DIRECTORY-RELATED COMMANDS</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>C. SYSTEM-RELATED COMMANDS</td>
<td>20</td>
</tr>
<tr>
<td>III.</td>
<td>MSG</td>
<td>21</td>
</tr>
<tr>
<td>IV.</td>
<td>XED</td>
<td>23</td>
</tr>
<tr>
<td>V.</td>
<td>FILE TRANSFER PROTOCOL (FTP)</td>
<td>25</td>
</tr>
<tr>
<td>VI.</td>
<td>TELNET (TN)</td>
<td>26</td>
</tr>
<tr>
<td>VII.</td>
<td>PHOTO</td>
<td>28</td>
</tr>
<tr>
<td>VIII.</td>
<td>REMIND</td>
<td>30</td>
</tr>
<tr>
<td>IX.</td>
<td>INQUIR</td>
<td>32</td>
</tr>
<tr>
<td>X.</td>
<td>FINGER</td>
<td>33</td>
</tr>
<tr>
<td>XI.</td>
<td>ELECTRONIC MAIL HOST (EMH)</td>
<td>34</td>
</tr>
<tr>
<td>XII.</td>
<td>EMACS</td>
<td>35</td>
</tr>
<tr>
<td>XIII.</td>
<td>GRAPH</td>
<td>35</td>
</tr>
<tr>
<td>XIV.</td>
<td>WHERE TO GO FROM HERE</td>
<td>36</td>
</tr>
</tbody>
</table>
USER'S GUIDE TO THE DDN/TOPS-20 TUTORIAL

This User's Guide to the Defense Data Network/TOPS-20 Interactive Tutorial presupposes that the user has worked or is working through the DDNTUT.FOR program. Access to the program is gained by entering the following at the "@" prompt:

EXECUTE <DECAIDS> DDNTUT.FOR

Compilation of the program, if required, will occur automatically. Once the program has been loaded by the system, you will see the opening line "WELCOME TO THE DDN", and you are off and running.

This guide is intended to serve strictly as a reference manual for users of the tutorial. It provides short descriptions of the Executive Level commands and the nine system level programs the user encounters while working through the tutorial.

Examples are not provided herein, as that is the purpose of the tutorial itself. Should the user remain uncertain regarding the correct procedure to follow in utilizing a system program, he/she is encouraged to refer to the tutorial.
At the end of this Guide is a recap of the "Where To Go From Here" portion of the tutorial. Users who wish to increase their knowledge of the TOPS-20 operating system and/or the Defense Data Network are urged to utilize these sources.

NOTE: Filename, Filetype, and Generation Number are abbreviated: FN.FT.NU respectively throughout this User's Guide and the tutorial.
EXECUTIVE LEVEL COMMANDS

The Executive Level (EXEC) of the TOPS-20 operating system is entered after you log on or after execution of system routines, and is noted by the display of the "@" prompt. Commands here fall into three categories:

1. Cursor Commands
2. Directory-Related Commands
3. System-Related Commands

Notes:

1. The "^" character refers to the control (CTRL) key.
2. ESC refers to the escape key.
3. <CR> refers to the carriage return or enter key.
4. DEL refers to the delete key.

CURSOR COMMANDS

<table>
<thead>
<tr>
<th>KEY</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC</td>
<td>Completes a command or Filename once you have typed enough for the system to recognize your intentions are.</td>
</tr>
<tr>
<td>CTRL</td>
<td>Used with another key to enter a command; must be held down while the other key is typed.</td>
</tr>
<tr>
<td>^C</td>
<td>Used to abort a partially completed command or to stop printing something you don't want to see. Will return you to the EXEC level if you are not already there.</td>
</tr>
<tr>
<td>^T</td>
<td>Used to show that the system is still running.</td>
</tr>
<tr>
<td>^S</td>
<td>Freezes typeout on the screen and acts as a no-scroll key.</td>
</tr>
<tr>
<td>^Q</td>
<td>Resumes typeout on screen after use of ^T command.</td>
</tr>
</tbody>
</table>
0 Prompt for the EXEC level. Must be typed twice to enter it into text while in XED and most other programs if accessing your host through a TAC.

DEL Deletes characters from right to left.

^O Stops terminal output.

^W Deletes words from right to left.

^Z End-of-file. Used whenever you have finished entering data, text, messages, etc.

<CR> Carriage return. Confirms a given command and permits execution of the command.

DIRECTORY-RELATED COMMANDS

A complete listing of these commands can be seen by typing "?" after the "0" prompt. The following is a listing of those most often used:

COMMAND USE(S)

DIRECTORY Or "DIR<CR>"); lists the names of all your files.

TYPE Types the contents of a file.

APPEND Appends one file to the end of another.

ARCHIVE Places a file in off-line storage.

DELETE Deletes a file from your directory.

UNDELETE Restores DELETEd files back to your directory, unless you have expunged or logged off or sometimes if the system crashes.

EXPUNGE Permanently erases all DELETEd files.

SET DIR PASS Allows you to change your password.

PROTECTION A six digit number which restricts access to any file in your directory. 775200 is the default. For more info, type HELP PROTECTION at the EXEC level.
# SYSTEM-RELATED COMMANDS

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTAT</td>
<td>Used to see everyone on a host. To see if a particular user is up on the host, type SY &lt;USERNAME&gt; &lt;CR&gt;. The system will respond with a job number, etc., if the user is on the host or with just a &quot;0&quot; if the user is not up.</td>
</tr>
<tr>
<td>FINGER</td>
<td>Used to find out information about a specific user and is invoked by typing FINGER &lt;USERNAME&gt; &lt;CR&gt; at the EXEC prompt.</td>
</tr>
<tr>
<td>ATTACH</td>
<td>Attaches a job to your current job.</td>
</tr>
<tr>
<td>TER NO RAISE</td>
<td>Makes your terminal accept both upper and lower case letters.</td>
</tr>
<tr>
<td>INFO MAIL</td>
<td>Asks system check for new messages.</td>
</tr>
<tr>
<td>INFO DISK</td>
<td>Lists information about your current page allocation.</td>
</tr>
<tr>
<td>TALK TO &lt; &gt;</td>
<td>Creates a communication link with another specified user, if on-line. If you are LINKed to, remember to type a ^C to place a hold on whatever it is you were doing before responding to the LINK. Use the symbol ; or ! to preface each line of type.</td>
</tr>
<tr>
<td>BREAK</td>
<td>Either individual LINKed can type this to terminate the link.</td>
</tr>
<tr>
<td>CONTINUE</td>
<td>Restores a user, following a LINK, to the place where a ^C was typed.</td>
</tr>
<tr>
<td>REFUSE</td>
<td>Type this in your LOGIN.CMD file if you do not wish to be interrupted by anyone LINKing to you.</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>Type this in your LOGIN.CMD file if you don’t mind being interrupted (receive is also the default).</td>
</tr>
</tbody>
</table>
MSG

MSG is a program which allows the user to create, send, file and read messages, referred to also as mail. A listing below provides the commands which may be used within MSG. They may also be seen from within MSG by typing a "?" after the MSG prompt, which appears as "<-". To enter MSG, type MSG at the EXEC prompt.

MSG COMMAND LEVEL COMMANDS

COMMAND USE(S)
A Answer a message
B Back up to previous message (also ^ or ^H)
C Current message
D Delete a message
E Exit and update message file
F Forward a message
G Go to message number < >
H Headers (HA: Headers ALL)
I Inclusion of length of header
J Jump into lower fork
K Koncise - provide shorter prompting
L List of messages
M Move message into a file (deletes it as a message)
N Next message number
O Overwrite old file
P Put a message into a file (does not delete it as a message)
Q Quit, return to EXEC level without update of MSG file
R Read in a file (specified)
S Send a message
U Undelete a message
V Verbose - provides more prompting
W Write file sorted by message arrival time
X XED - enter XED
Z Zap profile

Marks message as examined
Marks message as not examined
Display current time and date

Type with the command character for its description, alone for summary of commands available

Comment

Abort message

Message completion

COMMANDS WITHIN MESSAGE SEQUENCE

If you ask the system to type out a message (T - type) or output headers (H - headers), etc., the system will ask you for the message sequence. Usually, all you desire is one message, so you just type that message number followed by a carriage return. If you desire to have several messages printed you may use the following:

10:15 Prints messages 10 through 15
>15 Prints all message numbers greater than 15
6:3 Prints, in reverse order, messages 6 through 3
24,7 Prints message 24, then message 7

Other ways of responding to the message sequence request from the system are:

ESC Current message
^I Last sequence specified
A All message or headers
D Deleted messages or headers (prior to EX-PUNGE or EXIT)
E Examined messages
F String search of headers
I Inverse order
L Last message sequence
N Not examined messages
O Old messages/headers
R Recent messages only
S String search of subject
U Undeleted messages
XED

XED is a text editor; specifically a line editor. It allows you to both create and edit text, one line at a time. Entry into XED is by typing XED at the EXEC prompt. Some helpful definitions are:

- **Text Buffer**: Your current working space
- **Print Buffer**: Area where KILLED material is sent
- **Command Level**: Level you are at when entering XED, distinguished by the ";" prompt
- **Line Number**: Assigned by XED to each line in your text buffer; for your reference only, not stored in the file

**XED COMMAND LEVEL COMMANDS**

**COMMAND USE(S)**

A  APPEND, used to enter text after the current line
B  BACK UP, used to create a back up file
C  CHANGE, to modify current line(s)
E  EXIT, to leave XED, after a file save
F  FIND, to locate a specific letter, word or phrase in text
G  GROUP, to join current line with following line
I  INSERT, used to enter text before current line
J  JAM, puts contents of print buffer after current line
K  KILL, places current or specified line(s) into print buffer
L  LIST, Outputs entire text buffer without line numbers
P  PRINT BUFFER or DUMP, prints contents of print buffer
Q  QUIT, leaves XED prior to file save
R  READ, puts into XED the contents of specified file
S  SEARCH, like FIND, but locates all occurrences
T  TYPE, prints current line
V  VIEW, prints current and following 15 lines
W  WRITE, prints contents of text buffer to a file
X  EXCHANGE, used to search for and replace text
Z  ZAP, puts your entire file into the print buffer
"" SWITCH DUMP, swaps contents of the print and text buffers
,, TYPE CONTEXT, types current line and five lines before and after
/, TYPE, types current line
% SENDMSG, uses current text buffer as body of a message
$ LAST LINE, places you at end of file
? Displays available commands
^J TYPEs the next line
^Q ABORTs partially completed commands
_ FORMAT, right justifies a paragraph

COMMANDS WITHIN INSERT AND APPEND

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL KEY</td>
<td>Deletes letter to left of cursor</td>
</tr>
<tr>
<td>^R</td>
<td>RETYPES current line, leaving cursor at end</td>
</tr>
<tr>
<td>^X</td>
<td>KILLS LINE to left of cursor</td>
</tr>
<tr>
<td>^W</td>
<td>DELETEd WORD to left of cursor</td>
</tr>
<tr>
<td>^Z</td>
<td>RETURNs you to the command level</td>
</tr>
</tbody>
</table>
FILE TRANSFER PROTOCOL (FTP)

FTP is a program employed to transfer files around the net. This can be done from within a host or between hosts. It does necessitate that you have access to a directory on each host...the directory which has the material you desire and the directory to which you wish the material transferred. The following prompts are unique to FTP:

FTP> FTP command level prompt
USC-ISIE.ARPA> Sample foreign host command level prompt

FTP COMMAND LEVEL COMMANDS

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP</td>
<td>Invokes the FTP protocol</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Connects you from the local host you are logged onto, to the foreign host you are attempting to reach</td>
</tr>
<tr>
<td>SEND</td>
<td>Sends a file from your local directory to the foreign directory</td>
</tr>
<tr>
<td>GET</td>
<td>Retrieves a file from the foreign directory to your local directory</td>
</tr>
<tr>
<td>BYE</td>
<td>Disconnects you from the foreign host</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>Same as BYE</td>
</tr>
<tr>
<td>QUIT</td>
<td>Leaves FTP</td>
</tr>
<tr>
<td>EXIT</td>
<td>Same as QUIT</td>
</tr>
</tbody>
</table>
TELNET (TN)

TN is a program which allows a user on one host to access another host. It actually appears as though you are logged into the foreign host directly from a TAC rather than through your local host. The program also has the advantage of making your control characters do what you expect them to do, even if the control characters on the foreign host are different.

Using TN is actually quite simple. All you need do is type "TN", followed by the foreign host name, after the EXEC prompt. If a pathway to the foreign host is available and the host is up and running, you will connect to the foreign host ready to log into a directory. The standard "@" EXEC prompt will be displayed. Everything from then on is the same as logging into your own directory. When you are finished, simply log out of the foreign directory, and you're back in your own directory at your local host.

Note: If you access the foreign host by using the TN command, and then enter <HOSTNAME> <CR> at the TN> prompt, when you LOGOut of the foreign directory, you will be returned to the TN> prompt. Simply type EXIT <CR> and you’ll be back home.

One additional idiosyncrasy of TN is, since all control characters are meant to behave on the foreign host as they would on the local host, you cannot use "^C" to return to
your own EXEC level. There is a way around this, however.
You simply type "^C". The proper method of quitting TN,
however, is to LOGOut of the foreign host.
PHOTO

PHOTO is a program which allows you to record, or save, portions of a session on the net. It is invoked by typing "PHOTO <CR>" at the EXEC prompt. You will then be asked to supply a filename to which the PHOTO session is to be recorded. PHOTO will record what you type, as well as the responses from the computer. While you are in PHOTO, everything will work as normal. The control characters remain as before, and you still get the same prompts. What this means is, you cannot use ^C to exit from PHOTO. The normal means of exiting PHOTO is to type the command "POP <CR>" at any EXEC prompt.

There is one control character peculiar to PHOTO: "^Y" is used to suspend the PHOTO session. When you are ready to continue recording, simply input another ^Y.

If you check your directory after POPing out of PHOTO, you will see the new file. You can use the TYPE command or enter XED, READ in the file, and view it. While you are in XED, you can also edit the file.

While you are in PHOTO, the system beeps at you periodically to remind you that you are recording the session. The periodicity of the beeps is set at 30 seconds by default; you may change this, or eliminate them entirely through use of the following command:
@PHOTO/INTERVAL:0  Cancels the beeps
@PHOTO/INTERVAL:60  Gives beeps every 60 seconds, or whatever interval you prefer

And finally, a reminder that you must POP out of PHOTO before the system will allow you to log out.
REMIND

REMIND is a program which allows you to create reminders for yourself or others to be sent by the system at a later time, as specified by you. You may schedule a reminder to be sent only once, or as frequently as you like. You may also select whether the reminder will be MAILED (delivered via normal message service), SENT (delivered directly to addressee's screen, however, if the addressee is not logged on, the message is lost forever) or by BOTH methods (recommended).

To use the REMIND program, type "REMIND <CR>" at the EXEC prompt. Use the CREATE command to initiate a reminder message. A variety of prompts will be provided such as who the message is to be sent to, when, how often, how sent, etc. All of these prompts are fairly self-explanatory, but if you get stuck, simply type a question mark.

REMIND is the fastest way to send a message, as it doesn't force you to read the headers of any unread messages, etc., as with the MSG program.

REMIND COMMAND LEVEL COMMANDS

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>Used to CREATE a message</td>
</tr>
<tr>
<td>DAYTIME</td>
<td>Prints the exact date and EST time</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETEs remind messages</td>
</tr>
<tr>
<td>EXIT</td>
<td>EXITS out of REMIND</td>
</tr>
</tbody>
</table>

30
HEADERS
Lists subject line of all your REMIND messages
HELP
Gives info about all options; or a specific command
MODIFY
Changes the parameters (ex: time nag to be sent)
QUIT
Same as EXIT
SURVEY
Lists all recent remind messages you have sent
TYPE
TYPES a remind message already looked at
UNDELETE
UNDELETEs remind messages
INQUIR

INQUIR allows you to enter information into a file which can be read by yourself or others using the "FINGER / VERBOSE <USERNAME>" command. INQUIRE is entered by typing "INQUIR <CR>" after the EXEC prompt, following which you type the command "MODIFY <USERNAME> <CR>" where username is your own directory name. The system now will be at its entry level, and will either respond with a >> prompt, if data is already in your file, or with questions if there is no information in the file or in a specific field in the file.

While you are building or modifying an INQUIR file, the normal XED editing commands (DEL key, ^R, ^W, etc.) may be used. ^Z will signal the system that you have completed the file. The REMARKS section of the file allows you to leave short messages for people INQUIRING about you.

When you are at a >> prompt, there are a few commands available which can be useful:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Allows you to review/change the entire database.</td>
</tr>
<tr>
<td>&lt;FIELD NAME&gt;</td>
<td>Allows you to change a specific field</td>
</tr>
<tr>
<td>SHOW</td>
<td>Shows you the entire data base as it would appear with a FINGER/VERBOSE command</td>
</tr>
<tr>
<td>EXIT</td>
<td>Exits to the &gt; prompt from the &gt;&gt;, or to @ from &gt;.</td>
</tr>
</tbody>
</table>
FINGER

FINGER is a program to help locate and identify users on a system. It is called from the EXEC Level by typing "FINGER <CR>". Its main features are personal name and line location output. If you leave a message, called a "plan", in a FINGER.PLAN file, it will print out when someone FINGERs you and you are not logged on. This file is a free-form text file created in XED, and the file protection (SET PROTECTION command) should be at least 775252 to allow all other users to read it.

Calling sequences for FINGER include:

1. @FINGER <USERNAME>
   Prints whatever message user has in their .PLAN file; tells you if user is logged on.

2. @FINGER / <SWITCHES> <USERNAME> OR @FINGER <USERNAME> / <SWITCHES>
   If username is blank, all users at a switch will be summarized alphabetically. If username specified, info about that particular user on that switch is printed in more detailed fashion.

3. OTHER SEQUENCES:
   /DETACHED Displays detached jobs
   /DIAL-IN Displays dial-in jobs only
   /HELP Show this message
   /NO-DETACHED Suppress display of detached jobs
   /NO-OPERATOR Suppress display of operator jobs
   /OPERATOR Display operator jobs only
   /TERSE Output line job status only
   /VERBOSE Output plan and mail info, plus INQUIRE database, not including account. <USERNAME> must be specified.
   /WHOIS Useful for remote users; displays site dependent info about a specific user.
ELECTRONIC MAIL HOST (EMH)

An Electronic Mail Host, or EMH, is a special kind of host. The only capability an EMH has is sending and receiving messages. To use an EMH, you must be authorized access, and have a directory name and one or two passwords, as required. The passwords change frequently to prevent idle chitchat, user abuse, and large, inexplicable bills.

To access an EMH host, use the TN command. When the connection is established, log on and give passwords just as if you were logging into any directory. There are a number of command options in EMH useful in creating and sending messages, reading, deleting, and filing other messages, etc. They are largely self-explanatory, and bear a strong resemblance to the commands previously discussed for MSG, REMIND, and other programs found on the system. Remember to use a ? if you get stuck.

The following are some unique EMH commands:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>USE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPOSE</td>
<td>Used to create a message.</td>
</tr>
<tr>
<td>REPLY</td>
<td>Used to reply to a message.</td>
</tr>
<tr>
<td>.&lt;CR&gt;</td>
<td>Used to end a message when entered alone on a separate line.</td>
</tr>
</tbody>
</table>
EMACS

The EMACS text editor portion of the tutorial has not yet been written.

GRAPH

The GRAPH or graphics portion of the tutorial has not yet been written.
WHERE TO GO FROM HERE

The following materials serve as excellent sources of information on the TOPS-20 operating systems and sub-systems, including:


USER'S GUIDE TO TOPS-20
WAYNE TURNER
SEPT 1983 & APRIL 1984
USC INFORMATION SCIENCE INSTITUTE
4676 ADMIRALTY WAY
MARINA DEL REY, CALIFORNIA 90291
(213)822-1511 EXT 289
ACTION @ USC-ISI.ARPA

TOPS-20,
CHLOE SOMMERS HOLG
AUGUST 1983
USC INFORMATION SCIENCE INSTITUTE
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90291
(213) 822-1511 EXT 289
ACTION @ USC-ISI.ARPA

PROFESSOR GARY POOCK
CODE 55PK
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943
POOCK@USC-ISI.ARPA
The following are informative sources on the creation of the Defense Data Network, where it's at today and where it's going:

DDN NEW USER GUIDE (NIC 50001)
MARCH 1985
DDN NETWORK INFORMATION CENTER
DDN PROGRAM MANAGEMENT OFFICE
DEFENSE COMMUNICATION AGENCY
WASHINGTON, D.C.
(703) 285-5025

Useful addresses:

BOLT, BERANEK AND NEWMAN INC.
50 MOULTON STREET
CAMBRIDGE, MASSACHUSETTS 02238

computing analysis corporation
1400 WILSON BLVD, SUITE 1101
ARLINGTON, VA 22209

Digital equipment corporation
PO BOX CS-2008
NASUA, NEW HAMPSHIRE 03061

Document distribution,
usc information sciences institute
4676 ADMIRALTY WAY, SUITE 1100
Maria del rey, California 90291

Massachusetts institute of technology
artificial intelligence laboratory
atttn: publications
545 technology square
Cambridge, massachusetts 02139

uniologic, ltd
160 N. CRAIG
PITTSBURG, PENNSYLVANIA 15213
APPENDIX B

MAINTENANCE MANUAL

TO THE

DEFENSE DATA NETWORK/TOPS-20 TUTORIAL
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Program</td>
<td>41</td>
</tr>
<tr>
<td>Introduction Subroutine (Intro)</td>
<td>45</td>
</tr>
<tr>
<td>Basic Executive Level Commands Subroutine (ELC1)</td>
<td>49</td>
</tr>
<tr>
<td>Directory Sub-Subroutine</td>
<td>53</td>
</tr>
<tr>
<td>Type Sub-Subroutine</td>
<td>54</td>
</tr>
<tr>
<td>Append Sub-Subroutine</td>
<td>56</td>
</tr>
<tr>
<td>Rename Sub-Subroutine</td>
<td>57</td>
</tr>
<tr>
<td>Archive Sub-Subroutine</td>
<td>58</td>
</tr>
<tr>
<td>Delete Sub-Subroutine</td>
<td>59</td>
</tr>
<tr>
<td>Undelete Sub-Subroutine</td>
<td>60</td>
</tr>
<tr>
<td>Expunge Sub-Subroutine</td>
<td>61</td>
</tr>
<tr>
<td>Advanced Executive Level Command Subroutine (ELC2)</td>
<td>62</td>
</tr>
<tr>
<td>Talk Sub-Subroutine</td>
<td>66</td>
</tr>
<tr>
<td>Message Subroutine (AMSG)</td>
<td>68</td>
</tr>
<tr>
<td>Xed Subroutine (AXED)</td>
<td>80</td>
</tr>
<tr>
<td>Ftp Subroutine (FTP)</td>
<td>97</td>
</tr>
<tr>
<td>Tn Subroutine (ATN)</td>
<td>104</td>
</tr>
<tr>
<td>Photo Subroutine (APHOTO)</td>
<td>108</td>
</tr>
<tr>
<td>Remind Subroutine (ARMND)</td>
<td>110</td>
</tr>
<tr>
<td>Finger Subroutine (AFNGR)</td>
<td>114</td>
</tr>
<tr>
<td>Inquir Subroutine (AINQR)</td>
<td>118</td>
</tr>
<tr>
<td>Electronic Mail Host Subroutine (AEMH)</td>
<td>121</td>
</tr>
<tr>
<td>Screen Editor Subroutine (AEMACS)</td>
<td>124</td>
</tr>
</tbody>
</table>
------DDN TUTORIAL-----------------MAIN PROGRAM--------------------

AN INTERACTIVE TUTORIAL FOR THE DEFENSE DATA NETWORK

LCDR M. K. H. HERKERT, USN

LT S. L. SMITH-MOREAU, USN

US NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIFORNIA

------PURPOSE----------------MAINTENANCE--------------------------

THIS TUTORIAL IS DESIGNED TO INTRODUCE THE NEW USER TO THE
COMMANDS AND SYSTEMS IN USE ON THE TOPS-20 VERSION OF THE
DEFENSE DATA NETWORK (DDN).

THIS PROGRAM IS WRITTEN IN FORTRAN AND WILL PERFORM EQUALLY
WELL ON EITHER A FULL SCREEN VDT OR A SPOOL PAPER TERMINAL.

QUESTIONS ON THIS PROGRAM AS WELL AS REQUESTS FOR EXTERNAL
DOCUMENTATION AND MAINTENANCE SHOULD BE MADE TO PROFESSOR
GARY POOCK (POOCK@ISI.ARPA).

--------INSTALLATION OF PROGRAM-----------------------------

THE FOLLOWING ACTIONS ARE REQUIRED TO INSTALL THIS PROGRAM:

1. PLACE COPY OF "DDNTUT.FOR" IN THE DIRECTORY FROM WHICH IT
   IS TO BE RUN.
2. EXECUTE THE PROGRAM (EXECUTE DDNTUT.FOR) UNTIL YOU GET TO
   THE INITIAL INTRODUCTION. THIS COMPILES THE PROGRAM;
   CREATES A "DDNTUT.REL" FILE.
3. SET PROTECTION AS FOLLOWS:
   DIRECTORY    777740
   DDNTUT.FOR    777700
   DDNFOR.REL    777752

THE PROGRAM CAN NOW BE RUN FROM ANY ACCOUNT ON THE HOST BY
THE USER ENTERING:
"EXECUTE ⟨DIRECTORY WHERE INSTALLED⟩ DDNTUT.FOR"

---END OF TUTORIAL---

INTEGER INPUT
DIMENSION DEM1(1)
DUM=0

*** PROVIDE INITIAL INTRODUCTION

TYPE 1600
TYPE 100
READ(05,130,ERR=15) DEM1
IF(DEM1(1).EQ.'G'.OR.DEM1(1).EQ.'q')GO TO 520
15
16

*** PROVIDE FOR SELECTION OF DESIRED AREA OF INSTRUCTION

READ(05,*,ERR=16) INPUT
IF(INPUT.LT.1.OR.INPUT.GT.16)GO TO 16
IF(INPUT.EQ.16)GO TO 500
GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,17)INPUT
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO HELP, ?, AND ACTION

CALL INTRO(DUM)
GO TO 15

*** SUBROUTINE FOR BASIC EXECUTIVE LEVEL COMMANDS

CALL ELC1(DUM)
GO TO 15

*** SUBROUTINE FOR ADVANCED EXECUTIVE LEVEL COMMANDS

CALL ELC2(DUM)
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO MSG

CALL ANSG(DUM)
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO XED

CALL AXED(DUM)
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO FTP

CALL AFTP(DUM)
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO TN

CALL ATN(DUM)
GO TO 15

*** SUBROUTINE FOR INTRODUCTION TO PHOTO

42
CALL APHOTO(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO REMIND

CALL ARMND(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO INQUIRE

CALL AIMQR(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO FINGER

CALL AFNGR(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO ELECTRONIC MAIL HOSTS (EMH)

CALL AEMH(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO EMACS --CURRENTLY EMPTY--

CALL AEMACS(DUM)
GO TO 15

**** SUBROUTINE FOR INTRODUCTION TO GRAPH --CURRENTLY EMPTY--

CALL AGRAPH(DUM)
GO TO 15

**** SUBROUTINE FOR BIBLIOGRAPHY

CALL BILBO(DUM)
GO TO 15

**** SIGN-OFF ROUTINE

500 TYPE 1240
   TYPE 1370
   READ(05,130,ERR=510) DEM1
510 STOP
520 TYPE 1250
   TYPE 1370
   READ(05,130,ERR=530) DEM1
530 STOP

******************MAIN PROGRAM FORMATS**************************

**** INTRODUCTION
WELCOME TO THE DEFENSE DATA NETWORK (DDN).

THE FOLLOWING TUTORIAL IS DESIGNED TO ACCOMPLISH YOU WITH VARIOUS FEATURES OF THE NETWORK, PROVIDING INSTRUCTIONS AND ALLOWING YOU TO GAIN SOME DEGREE OF PROFICIENCY. THE TUTORIAL IS INTERACTIVE.

ADDITIONAL SOURCES OF INFORMATION ON THE DDN ARE LISTED IN THE WHERE TO GO NEXT SECTION OF THIS TUTORIAL. ENTER A "Q" TO QUIT, ANY OTHER CHARACTER TO CONTINUE.

**** SELECTION OF AREA OF INSTRUCTION

TO BEGIN THIS SESSION, SELECT ONE OF THE FOLLOWING TOPICS:

1. INTRODUCTION (HELP, ACTION)
2. TOPS-20 EXECUTIVE LEVEL COMMANDS (BASIC)
3. TOPS-20 EXECUTIVE LEVEL COMMANDS (ADVANCED)
4. MESSAGE PREPARATION, SENDING, READING (MSG)
5. LINE EDITOR (XED)
6. FILE TRANSFERRING (FTP)
7. CONNECTING TO ANOTHER COMPUTER (TN)
8. COPYING YOUR WORK (PHOTO)
9. REMINDERS TO YOURSELF/OTHERS (REMIND)
10. FINDING OUT ABOUT (INQUIR)
11. IDENTIFYING YOURSELF/OTHERS (FINGER)
12. ELECTRONIC MAIL HOSTS (EMH)
13. SCREEN EDITOR
14. GRAPHICS
15. WHERE TO GO FROM HERE
16. QUIT

ENTER THE NUMBER CORRESPONDING TO YOUR SELECTION, FOLLOWED BY A <CR>: 

This terminates the TOPS-20 system tutorial.

Sorry to see you quit so soon.

Enter any key: 

END
SUBROUTINE INTRO(DUM)

**INTRODUCTION**

* PROVIDES INTRODUCTION TO HELP, ? AND ACTION *

**INTRODUCTION TO SYSTEM HELP AIDS**

TYPE 1600

**INTRODUCTION TO HELP AND ?**

TYPE 100

READ(05,1400,ERR=10) INPUT
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520

**INTRODUCTION TO DOCUMENTATION AND ACTION**

TYPE 110

READ(05,1400,ERR=15) INPUT
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520

**DEMONSTRATION**

TYPE 120

READ(05,1410,ERR=20) SLCTN
IF(SLCTN(1).EQ.'Q'.OR.SLCTN(1).EQ.'q')GO TO 520

**DEMONSTRATION**

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15

GO TO 15
**SIGN-OFF ROUTINE**

```plaintext
500 TYPE 1240
   TYPE 1370
   READ(05,1400,ERR=510) INPUT
510 RETURN
520 TYPE 1250
   TYPE 1370
   READ(05,1400,ERR=530) INPUT
530 RETURN

**********FORMATS FOR INTRODUCTION SUBROUTINE**********

**INTRODUCTION**

100 FORMAT(12X,'HELP,ACTION,? AND DOCUMENTATION',//,
   'The DDN network, or more specifically the Tops-20 system,
   allows the user to receive help at almost all levels. The fol-
   lowing is an explanation and demonstration of commands to re-
   ceive help at the Executive Level, but the first two may be
   used at almost any time',//,
   'Notes':',//,
   1. You are in the executive level when you first log on or a
     fter you have finished with any of the system routines. You
     will know that you are in the Executive Level because the
     "@" prompt will be displayed by the system',//'9',
   2. Commands may be entered using either upper or lower case
     letters',',//,
   1. Commands below are presented within quotes. This is for
     emphasis only',//,'The quotes are not entered',//)

**HELP**

105 FORMAT('The HELP Command:',//,
   'This command, by itself, will give you guidance on how to get
   more in depth knowledge on a particular subject. It is invoked
   by typing "HELP" after the "@" prompt, followed by a carriage
   return',',//,
   'The ? command:',',//,'This command, when invoked after the "@" or
   "^" prompt, will give you a menu of appropriate commands from
   which to select',',//,
   'The HELP NAME command:',',//,'This command provides specific in-
   formation on the subject named. It is invoked',',//,'by typing "HELP
   NAME" followed by a carriage return after the "@" prompt',',//,'"N
   AME" is the subject on which you desire further information',//)

**DOCUMENTATION & ACTION**

110 FORMAT('DOCUMENTATION:',',//,'DOCUMENTATION provides additional in-
   formation about specific subjects. To see the available sub-
   jects, type "DIR <DOCUMENTATION>.*.*" after the "@" prompt',',//,'if

46
Followed by a carriage return. To see a specific documentation file, enter: "TYPE <DOCUMENTATION>FILENAME.FILETYPE", followed by a carriage return; FILENAME, and FILETYPE are those as shown in the directory.

ACTION: Having exhausted the above and still not satisfied, or if you have comments about the system, etc. you can send a message to ACTION. The procedure for, sending a message is covered in the MSG section of this tutorial.

**** DEMONSTRATION SELECTION

FORMAT: To see a demonstration of the above commands, type one of the following commands after the "G" prompt, followed by a carriage return.

HELP, HELP ACTION
QUIT

FORMAT: You have not selected an appropriate choice.

**** HELP

FORMAT: The HELP command prints helpful documentation on various system, features. The "G" is printed by the system. You type the rest of the line followed by a carriage return. HELP NAME, will look for, and print out information about the system feature, named in "NAME". If this is your first time using this system, please type HELP SYMESSAGES, will give a list of features for which HELP is available and, retype HELP to wait for any additional input. To see this text again type HELP followed by a carriage return.

**** AVAILABLE WITH HELP

FORMAT: Command, one of the following: ACCESS, ADVISE, APPEND, ARCHIVE, ASSIGN, ATTACH, BACKSPACE, BDIT, BLANK, BREAK, BUILD, CANCEL, CD, CL 30SE, COMPILE, CONNECT, CONTINUE, COPY, CREATE, CREF, CSAVE, DAYTIME, DDT, DEASSIGN, DEBU 5G, DECLARE, DEFINE, DELETE, DEPOSIT, DETACH, DIRECTORY, DISABLE, DISCARD, DISMOUNT, DO, ECHO, EDIT, ENABLE, END-ACCESS, EOF, ERUN, EXAMINE, EXECUTE, EXPUNGE, FDIRECTORY, FINGER, FORK, FREEZE, GET, HELP, IDDT, INFORMATION, KEEP, KKJOB, KNIC, LIST, LOAD, LOGIN, LOGOUT, MAIL, MAP, MERGE, MODIFY, MOUNT, NAME, NO, ORIGINAL, PLOT, POP, PRINT, PUNCH, PUSH, QDIRECTORY, R, RDIRECTORY, RECEIVE, REJECT, REFUSE, REMARK, REMAKE, REPLACE, RE SET, RETRIEVE, REWIND, RUN, SAVE, SET, SKI 6P, START, SUBMIT, SYSTAT, TAKE, TALK, TDIR 7ECTORY, TERMINAL, TRANSLATE, TYPE, UNATTACH, UNDECLARE, UNDELETE, UNKEEP, UNLOAD, UNMAP, VDI 9RECTORY, WDIRECTORY, XPRESS, or kept fork name, or system program name.

C C C...

DEMONSTRATION SELECTION

.. DEMONSTRATION SELECTION...

...DEMONSTRATION SELECTION...

C C C...DEMONSTRATION SELECTION...

...DEMONSTRATION SELECTION...

C C C...DEMONSTRATION SELECTION...

...DEMONSTRATION SELECTION...

C 47
The directory <ACTION> exists to receive user messages concerning questions, suggestions, documentation or problems with subsystem software, systems, operation, director 31esa, or terminals. <ACTION> message file is scanned constantly during the day, and all requests are acted upon by User Services personnel. The purpose of this centralized 6 mailbox is to insure that:

1. all comments are readily recognized and acted upon.
2. action requests are not delayed in personal message files due to unexpected abences.
3. all comments are readily recognized and acted upon.
4. action requests are not delayed in personal message files due to unexpected abences.
5. When in doubt as to where your question, suggestion, etc. should be sent, to <ACTION>. Your comments are also welcome. See MSG section of this tutorial for instructions on sending a message.

Whenever a user logs into TOPS-20, any new system messages are printed on their terminal. The purpose of these messages is to inform users of:

1. changes to the system.
2. If this is your first session, the system did not display the new system messages to you yet. On the next session you will see all the system messages.
3. There may be a lot of messages because we keep a lot of messages on line. On later sessions you will see only the new messages.

This concludes the introduction section.

Sorry to see you quit so soon.

ENTER ANY KEY TO RETURN TO THE MAIN MENU:

END
SUBROUTINE ELC1(DUN)

* ----* BASIC EXECUTIVE LEVEL COMMANDS SUBROUTINE************
* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES BASIC COMMANDS
* USEFUL AT THE EXECUTIVE LEVEL OF THE DDN/TOPS-20 SYSTEM.
*--------------------------------------------------------------------

DIMENSION INPUT(1)
DUN=0
NEXT=0

INTRO TO CURSOR COMMANDS

TYPE 1600
TYPE 1100;GO TO 200
15 TYPE 1110;GO TO 200
20 TYPE 1115;GO TO 200

INTRO TO DIRECTORY RELATED COMMANDS

25 TYPE 1120;GO TO 200
30 TYPE 1125
NEXT=NEXT+1
READ(05,1400,ERR=35) INPUT

QUERY IF USER DESIRES DEMONSTRATION

IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 100

DEMONSTRATION SUBROUTINES

35 CALL DIRECT(DUM)
GO TO 200
40 CALL TYPE(DUM)
GO TO 200
45 CALL APPEND(DUM)
GO TO 200
50 CALL RENAME(DUM)
GO TO 200
55 CALL ARCHIV(DUM)
GO TO 200
60 CALL DELETE(DUM)
GO TO 200
65 CALL UNDEL(DUM)
GO TO 200
70 CALL EXPUNG(DUM)
GO TO 100
100 NEXT+NEXT+12

INTRO TO SYSTEM RELATED COMMANDS

TYPE 1200
GO TO 200
```plaintext
105 TYPE 1205
   GO TO 500
   
   QUERY TO SEE IF USER DESIRES TO CONTINUE

200 TYPE 1300
   NEXT=NEXT+1
   READ(05,1400,ERR=205) INPUT
   TYPE 1600
   IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520
   
   GOTO (15,20,25,30,35,40,45,50,55,60,65,70,105) NEXT
   
   **** SIGN-OFF ROUTINE

500 TYPE 1240
   TYPE 1370
   READ(05,1400,ERR=510) INPUT
   RETURN

510 TYPE 1250
   TYPE 1370
   READ(05,1400,ERR=530) INPUT
   RETURN

   *****FORMATS FOR BASIC EXECUTIVE LEVEL COMMANDS SUBROUTINE******

   ***** LISTING OF EXECUTIVE LEVEL COMMANDS

1100 FORMAT(20X,'EXECUTIVE LEVEL COMMANDS (BASIC)').'//,
   '1' You are in the Executive (EXEC) Level of the TOPS-20 operating a
   2ystem after',//,
   '3' you log on or after execution of system routines; noted by the d
   3isplay of ',//,
   '4' the "@" prompt. Commands here fall into three categories:',//,
   '5' 1. Cursor Commands',//,
   '6' 2. Directory-Related Commands',//,
   '7' 3. System-Related Commands',//,
   '8' The more basic/most often used of these commands are presented i
   9n this section',//,' of the tutorial. The remainder are presented
   '*in section three',//,
   '1' Notes:',//,
   '2' 1. The "^" character refers to the control (CTRL) key',//,
   '3' 2. ESC refers to the escape (ESC) key',//,
   '4' 3. <CR> refers to the carriage return or enter key',//,
   '5' 4. DEL refers to the delete (DEL) key',//)

   ***** CURSOR COMMANDS

1110 FORMAT(20X,'BASIC CURSOR COMMANDS'.'//,
   '1' KEY (a) USE (a)'.'//,
   '2' ESC This key usually completes a command or FileName. On
   3es you haven't typed',//,' typed enough of a command or FileName
   4or the system to recognize',//,' it, type the Escape key
   5y to complete it. If you haven't typed',//,' enough, t
   6he system simply beeps and waits for you to type more',//,
```

50
7' CTRL Used with another key to enter a command. You may ho
8ld down the '/,' CTRL key as long as you want, but shou
9ld press the other key only once (the same as you
would use the shift key on a typewriter). '/,' , '/,' ,
1' 'C Used to abort a partially typed or partially complete
2d command '/,' (sometimes this may require several "C
3a). If you are in the middle '/,' of a program, typin
4g 'C will return you to the EXEC level. (This is '/,"
5ot usually the best way to leave a program). '/,' ,'/,' ,
C
1115 FORMAT(' 0 Prompt for the EXEC level. (To enter this cha
1rector into text).'/,' ,
2' while accessing the host through a TAC, type it twice
* -- the system'/,' ,
* will respond with a third "0")'/,' ,'/,' ,
6' "O Stops terminal output.' ,'/,' ,
6' "W Deletes words from right to left.' ,'/,' ,
1' "Z End-of-file. Used whenever you have finished enterin
2g data, text,' ,'/,' ,
3' <CR> Confirms a given command. Most commands are not exec
4uted until you '/,' type a carriage return.' ,'/,' ,
C
C LISTING OF DIRECTORY RELATED COMMANDS

1120 FORMAT(' DIRECTORY-RELATED COMMANDS'/,' ,
1' A complete listing of Directory-Related commands can be seen by
2typing "?" after the "0" prompt. The following is a listing
3of those most often used.' ,'/,' ,
4' Note: A filename within the TOPS-20 system is composed of a nam
5e, a type and'/,' a generation number. Thus, HOMEWORK.EXE.5 is a
6file with a name of "HOMEWORK",'/,' a type of "EXE" and is the fi
7th generation. In the following material, we'/,' abbreviate file
8name with "FN"; filetype with "FT"; generation number with "NU";',
9'/,' ,
5' Command Use (a)' ,'/,' ,
6' DIRECTORY Lists the names of all of your files.' ,'/,' ,
7' TYPE Types the contents of a file.' ,'/,' ,
8' APPEND Appends one file to another.' ,'/,' ,
C
1125 FORMAT(' RENAME Renames a file.' ,'/,' ,
1' ARCHIVE Places a file in off-line storage.' ,'/,' ,
2' DELETE Deletes a file from your directory.' ,'/,' ,
3' UNDELETE Puts "DELETED" files back in your directory unless yo
4u have'/,' "EXPUNGED" or logged off or occasionally,
5' if the system crashes'/,' ,
5' after you "DELETED" the files.' ,'/,' ,
6' EXPUNGE Permanently erases all "DELETED" files.' ,'/,' ,
7' WOULD YOU LIKE A DEMONSTRATION OF THE ABOVE COMMANDS (Y or N)?',
8a)
C
C LISTING OF SYSTEM RELATED COMMANDS

1200 FORMAT(' SYSTEM-RELATED COMMANDS (BASIC)' ,'/,' ,
1' System related commands include:' ,'/,' ,

51
2' FINGER  SYSTAT  TALK  REF',/,
3' REC  INFO DISK  INFO MAIL  TER NO RAISE',/,
4' FINGER and SYSTAT are discussed in the following. The others are
4e',/,' presented in the third section (TOPS-20 LEVEL COMMANDS - AD
5VENTAGED)',/,'/

C

1205 FORMAT(' FINGER:',/),
1' FINGER is used to find out information about a specific user, an
2d',/' is invoked by typing "FINGER USER <CR>" after the "G" prompt
3t (USER',/' refers to the account name)',/),
4' SYSTAT:',/),
5' SYSTAT is used to see everyone, or if a specific individual is
6' on a specific host. Typing SYSTAT after the "@" prompt followed by a
7' prompt will',/),
8' list everyone on the net. To see if a particular individual is
9' on the net',/),
8' type "SY USER <CR>" after the "@" prompt. The system will
9' answer with the',/),
1' Job Number, etc. for that individual, or with just an "@" prompt
*'. The latter',/),
"' indicates that the USER is',/,' not on the net',/)

C

**** SIGN-OFF ROUTINE

C

1240 FORMAT(' This concludes the "basic" Executive Level Command sectio
1n of the',/,' tutorial)',/)
1250 FORMAT(' SORRY TO SEE YOU LEAVE SO SOON',/)
1300 FORMAT(' DO YOU WISH TO CONTINUE (Y or N)?',/)
1370 FORMAT(' ENTER ANY KEY TO RETURN TO THE MAIN MENU:
1400 FORMAT(A1)
1600 FORMAT(///)

END

C
SUBROUTINE DIRECT(DUM)

********************************************************************************
*          DIRECTARY SUB-SUBROUTINE==============================================
*                      DEMONSTRATION OF VIEWING YOUR DIRECTORY OF FILES LISTING  *
********************************************************************************

DIMENSION DEMO(2),INPUT(1)
NUM=0
5 TYPE 100
10 READ(05,105,ERR=15) DEMO
   IF(DEMO(1).EQ.'Q'.OR.DEMO(1).EQ.'q')GO TO 40
   IF(DEMO(1).EQ.'DIREC'.AND.DEMO(2).EQ.'TORY'.OR.DEMO(1).EQ.'dir'.AND.DEMO(2).EQ.'tory'.OR.DE
   2MO(1).EQ.'dir'.AND.DEMO(2).EQ.'')GO TO 35
15 NUM=NUM+1
20 TYPE 110
25 TYPE 115
30 NUM=0
TYPE 120
35 TYPE 130
40 RETURN

C
C  *** FORMATS FOR DIRECTORY SUB-SUBROUTINE------------------------------
C
100 FORMAT(//' DIRECTORY:',//,' Typing in "DIRECTORY" or just "DIR" after the "@" prompt followe
12d by a <CR>',/,' will list the current contents of your directory.
2 Give it a try. '///,' @',@)
105 FORMAT(2AS)
110 FORMAT(//' NOPE! TRY AGAIN. '///,' @',@)
115 FORMAT(//' NOPE! TRY HARDER. '///,' @',@)
120 FORMAT(//' What you should have typed in after the "@" prompt was: ',
13d '/,' DIRECTORY or DIR',//,' CARE TO TRY AGAIN (Y or N)?',S)
125 FORMAT(A1)
130 FORMAT(//' PS:<YOUR ACCOUNT NAME>',//,' 1' EXED-FILES..1 (This file controls the XED editor',//,' 2' MSG.PROFILE).1 (This file controls the MSG system)',//,' 3' FINGER.PLAN.1',//,' 4' HAS.ANY.1',//,' 5' MAIL.TXT.1',//,' 6' REMIND.RMD.1',//,' 7' Total of 23 pages in 6 files',//,' 8' The above is what you would see, except for the notes to the rig
9nt. '///,' NOTE: NEVER DELETE ANY OF THESE CONTROL FILES UNTIL YOU
10 FAMILIAR WITH',//,' THE SYSTEM',//' END

C
SUBROUTINE TYPE(DUM)

C **********************************************************
C SUBROUTINE TYPE SUB-SUBROUTINE********************************************************
C DEMONSTRATION OF TYPE COMMAND
C **********************************************************
C
DIMENSION DEMO(3), INPUT(1)
NUM = 0
5 TYPE 100
10 READ (05,105,ERR=15) DEMO
IF (DEMO(1).EQ.'Q'.OR.DEMO(1).EQ.'q') GO TO 40
IF (DEMO(1).EQ.'TYPE '.AND.DEMO(2).EQ.'FACE'.AND.DEMO(3).EQ.'WHO.1
1'.OR.DEMO(1).EQ.'type '.AND.DEMO(2).EQ.'face'.AND.DEMO(3).EQ.'who
1.1') GO TO 35
15 NUM = NUM + 1
GO TO (20,25,30) NUM
20 TYPE 110
GO TO 10
25 TYPE 115
GO TO 10
30 NUM = 0
TYPE 120
READ (05,125,ERR=15) INPUT
IF (INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 5
GO TO 40
35 TYPE 130
40 RETURN
C
**** FORMATS FOR THE TYPE SUB-SUBROUTINE********************************************************
C
100 FORMAT(/' TYPE:',//,'**YPE-commend is invoked by typing "TYPE FN.FT.NU <CR>" aft
2er the "0",/'prompt. Try TYPing out the contents of the FACE.W
3HO file.'),/,'0'.,'#)
110 FORMAT('/' YOU CAN DO BETTER THAN THAT',//,'0'.,'#)
115 FORMAT('/' WHY DON'T YOU TAKE ANOTHER STAB AT IT',//,'0'.,'#)
120 FORMAT('/' What you should have typed in after the "0" prompt was:'
1',/,' TYPE FACE.WHO.1',//,' CARE TO TRY AGAIN (Y or N)?',#)
125 FORMAT(A1)
130 FORMAT(14A1)

1', XXXXXXXXXXXXXX ,/,' 2', XXXXXXXXXXXXXX ,/,' 3', XXXXXXXXXXXXXX ,/,' 5', XXX XXX ,/,' 6', XXX XXX ,/,' 7', XXX XXX XXXXX XXXX XXX ,/,' 8', XX X XX X X XX X XX ,/,' 9', XX X XX X X XX X XX ,/,' *', XXX XXX XXX XXXX XXX ,/,' 1', XXX XXX ,/,' 2', XXXX XXX XXXXX ,/,' 3', XXX XXX XXXXXX ,/,'
THE FATHER OF YOUR COUNTRY...GEORGE WASHINGTON
SUBROUTINE APPEND(DUM)

C-------------APPEND SUB-SUBROUTINE-----------------

C DEMONSTRATION OF APPENDING ONE FILE TO ANOTHER
C

DIMENSION DENO(6), INPUT(1)

NUN = 0

READ(OS, 105, ERR = 15) DENO

IF(DENO(1).EQ.'APPEND'.AND.DENO(2).EQ.'D TEM'.AND.DENO(3).EQ.'P2.FOR'
1'.AND.DENO(4).EQ.'R.1 T'.AND.DENO(5).EQ.'EMP1.'.AND.DENO(6).EQ.'FO'
2R.1'.OR.DENO(1).EQ.'append'.AND.DENO(2).EQ.'d tem'.AND.DENO(3).EQ.'E'
3'p2.so'.AND.DENO(4).EQ.'r.1 t'.AND.DENO(5).EQ.'empl.'.AND.DENO(6).
4EQ.'for.1') GO TO 35

NUN = NUM + 1

GO TO (20, 25, 30) NUM

READ(OS, 125, ERR = 15) INPUT

IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 5

GO TO 40

35

GO TO 130

RETURN

C------------FORMATS FOR THE APPEND SUB-SUBROUTINE-------------

C

100 FORMAT(//' APPEND:',//,
1' The "APPEND" command is typed after the "O" prompt, followed by
2the file to be appended, followed by the file to be appended
3to, followed by a <CR>. i.e. '"APPEND B.1.1 A.1.1<CR>" will cause the contents
4of the B file to be placed in the A file at the end. Try appending TEMP2.FOR.1 to
5TEMP1.FOR.1 by typing the command after the "O" prompt was given.'//,
6'CARE TO TRY AGAIN (Y or 2N)?',)

125 FORMAT(A1)

C

END
SUBROUTINE RENAME(DUN)

C

RENAME SUB-SUBROUTINE

DEMONSTRATION OF RENAMING A FILE

DIMENSION DEMO(4), INPUT(1)

 NUM=0

5 TYPE 100

10 READ(05,105,ERR=15) DEMO

IF(DEMO(1).EQ.'Q'.OR.DEMO(1).EQ.'q') GO TO 40

IF(DEMO(1).EQ.'RENAME'.AND.DEMO(2).EQ.'T T T'.AND.DEMO(3).EQ..'1 T'

AND.DEMO(4).EQ.'A T'.AND.DEMO(1).EQ.'renam'.AND.DEMO(2).EQ.'at'

AND.DEMO(3).EQ..'1 T'.AND.DEMO(4).EQ.'at.'GO TO 35

15 NUM=NUM+1

GO TO (20,25,30) NUM

20 TYPE 110

GO TO 10

25 TYPE 115

GO TO 10

30 NUM=0

TYPE 120

READ(05,125,ERR=15) INPUT

IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 5

GO TO 40

35 TYPE 130

RETURN

C

C

FORMATS FOR THE RENAME SUB-SUBROUTINE

100 FORMAT('RENAME:

1 The "RENAME" command is issued by typing "RENAME (old) FN.FT.NU

2(new) FN.FT./" after the "G" prompt. Let’s give it a try

3 by renaming TTT..1 to TAT..1./,"YES").

105 FORMAT('NOPE! TRY AGAIN./","YES")

110 FORMAT('NOPE! TRY HARDER./","YES")

115 FORMAT('What you should have typed in after the "G" prompt was:','

1/"RENAME TTT..1 TAT..1/","YES")

120 FORMAT('CARE TO TRY AGAIN (Y or N)?","YES")

125 FORMAT('THE RENAME IS COMPLETE./","YES")

END
SUBROUTINE ARCHIV(DUM)

C .. ........
ARCHIVE
SUB-SUBROUTINE
...

C = DEMONSTRATION OF HOW FILES ARE ARCHIVED (OFF-LINE STORAGE)

C

DIMENSION DEMO(3),INPUT(1)

NUM=0
5 TYPE 100
10 READ(05,105,ERR=15) DEMO
IF(DEMO(1).EQ.'Q'.OR.DEMO(1).EQ.'q') GO TO 40
IF(DEMO(1).EQ.'ARCHI'.AND.DEMO(2).EQ.'VE TA'.AND.DEMO(3).EQ.'T..t'
1.OR.DEMO(1).EQ.'archi'.AND.DEMO(2).EQ.'ve ta'.AND.DEMO(3).EQ.'t..t'
2') GO TO 35
15 NUM=NUM+1
GO TO (20,25,30)NUM
20 TYPE 110
GO TO 10
25 TYPE 115
GO TO 10
30 NUM=0
TYPE 120
READ(05,125,ERR=15) INPUT
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 5
GO TO 40
35 TYPE 130
40 RETURN

C

**** FORMATS FOR THE ARCHIVE SUB-SUBROUTINE

C

100 FORMAT(//,' ARCHIVE:',//,
1 The "ARCHIVE" command is typed after the "@" prompt followed by
2 the FN.FT.WU',//,' (of the file you wish saved off-line) followed
3 by a <CR>. Thus, "ARCHIVE ',//,' FINGER.PLAN.1 <CR>" would request
4 the Finger file be archived. Give it',//,' a try by ARCHIVEing th
5 e file ',//,' CARE TO TRY AGAIN (Y or N)?',N)
105 FORMAT(3A5)
110 FORMAT(//,' I'LL PRETEND I DIDN'T SEE THAT WHILE YOU TRY AGAIN
1N.,//,' @',@)
115 FORMAT(//,' NOPE! TRY HARDER.',//,' @',@)
120 FORMAT(//,' What you should have typed in after the "@" prompt was:
1/',//,' ARCHIVE TAT..t',//,' CARE TO TRY AGAIN (Y or N)?',@)
125 FORMAT(A)
130 FORMAT(//' TAT..t (REQUESTED)',//,' THE ABOVE LINE INDICATES THAT TH
1E FILE WILL BE ARCHIVED.',//)
END
SUBROUTINE DELETE (DUM)

******************************************************************************

DEMONSTRATION OF DELETING A FILE

******************************************************************************

DIMENSION DEMO(4), INPUT(1)
NUM=0

5 READ(05,105,ERR=15) DEMO
10 IF(DEMO(1).EQ.'G'.OR.DEMO(1).EQ.'g')GO TO 40
20 IF(DEMO(1).EQ.'DELETE'.AND.DEMO(2).EQ.'E HAS'.AND.DEMO(3).EQ.'ANY.
1'.AND.DEMO(4).EQ.'1'.OR.DEMO(1).EQ.'delet'.AND.DEMO(2).EQ.'e has'.
2AND.DEMO(3).EQ.'any'.AND.DEMO(4).EQ.'I')GO TO 35
15 NUM=NUM+1
20 GO TO (20,25,30) NUM
25 TYPE 110
30 TYPE 115
120 READ(05,125,ERR=15) INPUT
35 TYPE 130
40 RETURN

****** FORMATS FOR THE DELETE SUB-SUBROUTINE ******

100 FORMAT('DELETE:',//)
105 FORMAT(4A5)
110 FORMAT('TRY AGAIN, SWEETIE. I WON'T TELL. . . . ',//, 'Y',//)
115 FORMAT('ALRIGHT, YOU CAN HAVE ONE MORE CHANCE. . . . ',//, 'Y',//)
120 FORMAT('WHAT YOU SHOULD HAVE TYPED IN AFTER THE "G" PROMPT WAS: ',
11, 'DELETE HAS.ANY.1',//, 'CARE TO TRY AGAIN (Y or N)? ',//)
125 FORMAT(1A1)
130 FORMAT('HAS.ANY.1 [OK]',//, 'THE ABOVE LINE INDICATES THAT THE FILE HAS BEEN DELETED. ',//)

END
SUBROUTINE UNDEL(DUM)

--------------------------UNDELETE SUB-SUBROUTINE--------------------------
* DEMONSTRATION OF UNDELETING A FILE
--------------------------SUB-SUBROUTINE UNDELETE--------------------------

DIMENSION DEMO(3), INPUT(1)
NUM = 0
5 TYPE 100
10 READ(05, 105, ERR=15) DEMO
   IF(DEMO(1).EQ. 'O'. OR. DEMO(1).EQ. 'q') GO TO 40
   IF(DEMO(1).EQ. 'UNDEL'. AND. DEMO(2).EQ. 'ETE T'. AND. DEMO(3).EQ. 'AT.. 1'
1. OR. DEMO(1).EQ. 'undel'. AND. DEMO(2).EQ. 'ete t'. AND. DEMO(3).EQ. 'at. 2.1' ) GO TO 35
15 NUM = NUM + 1
   GO TO (20, 25, 30) NUM
20 TYPE 110
   GO TO 10
25 TYPE 115
   GO TO 10
30 TYPE 120
   READ(05, 125, ERR=15) INPUT
   IF(INPUT(1).EQ. 'Y'. OR. INPUT(1).EQ. 'y') GO TO 5
   GO TO 40
35 TYPE 130
40 RETURN

-----------FORMATS FOR UNDELETE SUB-SUBROUTINE--------------------------

100 FORMAT('/* UNDELETE: */
   1 The "UNDELETE" command is typed after the "O" prompt followed by
2 the FN.FT.NU',/' (of the file you wish returned to the directory
3) followed by a <CR>. That is', '/', "UNDELETE FINGER.PLAN.1 <CR>
4 would return the Finger file. Lets’'s give it', '/' a try by UNDEL
5 SETing the TAT..1 file', '/' ,' ', 'O', 's)
105 FORMAT(95S)
110 FORMAT('NOPE! TRY AGAIN', '/', ' ', 'O', 's)
115 FORMAT('ONCE MORE PLEASE', '/', ' ', 'O', 's)
120 FORMAT('What you should have typed in after the "O" prompt was:',
1/ , 'UNDELETE TAT..1', '/ , ' CARE TO TRY AGAIN (Y or N)?', 's)
125 FORMAT(91S)
130 FORMAT('TAT..1 [OK]', '/', ' THE ABOVE LINE INDICATES THAT THE FI
FILE HAS BEEN RETURNED', ' )

END
SUBROUTINE EXPUNG(DUM)

* DEMONSTRATION OF EXPUNGING AN ACCOUNT *

DIMENSION DEMO(2),INPUT(1)

5 TYPE 100
10 READ(OS,105,ERR=15) DEMO
   IF(DEMO(1).EQ.'Q'.OR.DEMO(1).EQ.'q')GO TO 40
   IF(DEMO(1).EQ.'EXPUN'.AND.DEMO(2).EQ.'GE'.OR.DEMO(1).EQ.'expun'.AN
1D.DEMO(2).EQ.'ge')GO TO 35
15 NUM=NUM-1
   GO TO (20,25,30)NUM
20 TYPE 110
   GO TO 10
25 TYPE 115
   GO TO 10
30 NUM=0
   TYPE 120
   READ(OS,125,ERR=15) INPUT
   IF(INPUT(1).EQ.'y'.OR.INPUT(1).EQ.'y') GO TO 5
   GO TO 40
35 TYPE 130
40 RETURN

FORMAT{//' EXPUNGE:',//,' This command is typed after the "@- prompt followed by a <CR>. 24
   FileName',/,' in required, as it permanently erases ALL files for which you had previously',/,' issued the "DELETE" command. Give it a try.',//,' @I,$)
100 FORMAT(2A5)
105 FORMAT(//' NOPE! TRY AGAIN.',//,' @I,*)
110 FORMAT(//' NOPE! TRY HARDER.',//,' @I,*)
115 FORMAT(//' What you should have typed in after the "@" prompt was:',//,' @I,* EXPUNGE',//,' CARE TO TRY AGAIN (Y or N)?',//,' @I,*)
120 FORMAT(//' PS:YOUR ACCOUNT NAME> 10 PAGES FREED!',//,' THE ABOVE LINE INDICATES THAT YOUR DIR HAS BEEN EXPUNGED.',//,')
125 FORMAT(A1)
130 FORMAT{//' PS:<YOUR ACCOUNT NAME> 10 PAGES FREED!',//,' THE ABOVE L
   LINE INDICATES THAT YOUR DIR HAS BEEN EXPUNGED.',//,' END
**SUBROUTINE ELC2(DUN)**

*ADVANCED EXECUTIVE LEVEL COMMANDS SUBROUTINE*

- This subroutine explains and demonstrates higher level commands at the executive level of the DDN/TOPS-20 system.
- It serves as a follow-on to the ELC1 subroutine.

**---**

**DIMENSION INPUT(1)**

**DUN=0**

****** Intro to Executive Level Related Commands**

**TYPE 1600**
**TYPE 1100**
**TYPE 1320**
**TYPE 1330**
**READ(05,1400,ERR=15) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

**15 TYPE 1110**
**TYPE 1330**
**READ(05,1400,ERR=25) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

****** Intro to Directory Related Commands**

**25 TYPE 1120**
**TYPE 1330**
**READ(05,1400,ERR=35) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

****** Intro to System Related Commands**

**35 TYPE 1200**
**TYPE 1330**
**READ(05,1400,ERR=40) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

**40 TYPE 1205**
**TYPE 1330**
**READ(05,1400,ERR=105) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

**105 TYPE 1210**
**TYPE 1330**
**READ(05,1400,ERR=110) INPUT**
**TYPE 1600**
**IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520**

**110 TYPE 1220**
C
**** QUERY TO SEE IF USER DESIRES DEMONSTRATION OF TALK COMMAND
C
READ(05,1400,ERR=499) INPUT
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n') GO TO 520
CALL TALK(DUM)
TYPE 1240
READ(05,1400,ERR=510) INPUT
RETURN
TYPE 1370
READ(05,1400,ERR=530) INPUT
RETURN
****FORMATS FOR ADVANCED EXECUTIVE LEVEL COMMANDS SUBROUTINE*****
**** DEMONSTRATION OF EXECUTIVE LEVEL COMMANDS
C
1100 FORMAT(20X,'EXECUTIVE LEVEL COMMANDS (ADVANCED)',//,
1 'You are in the Executive (EXEC) Level of the TOPS-20 operating a
2 system after',//,
3 you log on or after execution of system routines. You are in th
4 is level when',//,
5 the system prompts you with the "S" prompt. Commands at this le
6 vel fall into',//,
7 three categories:',//,
8 1. Cursor Commands',//,
9 2. Directory-Related Commands',//,
10 3. System-Related Commands',//,
11 The more basic/most often used of these commands were presented
12 in section 2',// of the tutorial. The remainder are presented
13 here',//,
14 Notes:',//,
15 1. The " " character refers to the control (CTRL) key',//,
16 2. ESC refers to the escape (ESC) key',//,
17 3. <CR> refers to the carriage return or enter key',//,
18 4. DEL refers to the delete (DEL) key',//)
C
C
1110 FORMAT(20X,'ADVANCED CURSOR COMMANDS',//,
1 'KEY (s) USE (s)',//,
2 'T' This command can be used to satisfy yourself that the
3 system is',//,
4 still running. For example, if the syst
5 em is responding slowly and',//,
6 you type "T", informati
7 on the system status and the amount of',//,
8 CPU time
9 you have used since you logged on is displayed',//,
10 'S' Freeze typeout on the screen and acts as a No Scroll
11 key',//,
12 'Q' Reauses typeout on a screen that was stopped by a 'S
13 command',//)
C
C
**** DEMONSTRATION OF DIRECTORY RELATED COMMANDS
C
1120 FORMAT(' DIRECTORY RELATED COMMANDS',//,
1 'A complete listing of Directory Related commands can be seen by
2 typing "?"',// after the "S" prompt. The following is a listing
3of those most often used.

**Note:** FN = FileName; FT = FileTy

4pe; MU = FileNumber

5 *Command*  Use (a)

6 *SET DIR PASS* This command changes your password.

7 *PROTECTION* The protection for each file in your directory consists of a six-digit number divided into three groups of two digits each. The first pair represents owner access to the file, the second pair represents group access, and the last pair represents world access to the file.

2 By setting the protection for files in your directory, you can control who can access and/or change your files. The default protection is 755200.

4 This allows you full access (77), members of your group partial access (52), and others no access (00).

7 For more info, type HELP PROTECTION.

---

**DEMONSTRATION OF SYSTEM RELATED COMMANDS**

1200 FORMAT

1 System related commands include:

2 FINGER SYSTAT TALK REF

3 REC INFO DISK INFO MAIL TER NO RAISE

4 FINGER and SYSTAT were previously discussed. The others are presented here.

1205 FORMAT (' ATTACH:

1 ATTACH Attaches a you to a job. For example, if you come somehow get detached from your directory before you get a chance to log out, log back in to the host and ATTACH yourself to the your old job number.

*If you don't remember what your old job number was, just SYS TAT yourself.*

4 TER NO RAISE:

5 TER NO RAISE Makes your terminal accept both upper and lower case letters. If, for some reason, it is only accepting upper case letters, you can use this command to switch to lower case.

8 INFO MAIL:

9 INFO MAIL Asks the system to list any unlooked-at messages in your mailbox.

1 INFO DISK:

2 INFO DISK Lists information about your current page allocation and for the system.

1210 FORMAT (' TALK:

1 TALK <user> Creates a communication link with another user. Use this in place of a message if the other user is logged on at the same time as you are.

3 If you suddenly see the phrase "LINK FROM <username>" appear on your screen,
type a "C", which puts a hold on what you were doing. You can then talk to the other person by prefacing your lines with the symbol ! or ;."

5' BREAK This command breaks the communication link with the other user.

1220 FORMAT(' CONTINUE: ".",
7' CONTINUE This command must be typed by the person who was interrupted and will take him/her back to the place of interruption.
8' REFUSE: ".",
9' REFUSE If you do not wish to be interrupted while you are on the net, type REFUSE in your LOGIN.CMD file.
10' RECEIVE: ".",
11' RECEIVE If you don't mind people talking to you while you are on the net, type RECEIVE in your LOGIN.CMD file.
12' WOULD YOU LIKE A DEMONSTRATION OF THE ABOVE COMMANDS (Y or N)?  

1240 FORMAT(' THIS CONCLUDES THE ADVANCED EXECUTIVE LEVEL COMMANDS SECTION 
130 FORMAT(' NOTE: YOU MAY QUIT THE FOLLOWING DEMONSTRATION AT ANY TIME SIMPLY BY TYPING A "Q <CR>" IN RESPONSE TO ANY PROMPT. 
1330 FORMAT(' ENTER A "Q" TO QUIT, ANY OTHER KEY TO CONTINUE: 
1400 FORMAT("/\)
1600 FORMAT(/\)
SUBROUTINE TALK(DUM)

**************************************************************************

DEMONSTRATE TALK/LINK COMMANDS

**************************************************************************

DIMENSION DEM1(3),INPUT(1)

25 TYPE 1600
30 READ(05,205,ERR=30) DEM1
35 IF(DEM1(1).EQ.'Q'.OR.DEQ1(1).EQ.'q')GO TO 60
40 IF(DEQ1(1).EQ.'TALK'.AND.DEQ1(2).EQ.'TO ED'.AND.DEQ1(3).EQ.'ZEL'
45 .OR.DEQ1(1).EQ.'talk'.AND.DEQ1(2).EQ.'to ed'.AND.DEQ1(3).EQ.'ze'
50 )GO TO 50
55 NUM=NUM+1
GO TO (35,40,45) NUM
60 TYPE 300
GO TO 30
65 GO TO 30
70 TYPE 140
75 TYPE 320
80 READ(05,210,ERR=45) INPUT
85 IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 60
90 IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 25
95 NUM=0

SIGN-OFF ROUTINE

100 TYPE 200
105 TYPE 350
110 READ(05,210,ERR=55) INPUT
115 RETURN
120 TYPE 340
125 RETURN

**************************************************************************

DEMONSTRATION OF TALK/LINK COMMANDS

**************************************************************************

120 FORMAT('To initiate a conversation with your coworker, Edzel, you
130   first type SY EDZEL. If the system responds with Edzel''s 30
140   number, you know he is on the net, and can now type TALK TO
150   EDZEL. Give it a try:','',' '','
160 50 FORMAT('What you should have typed was: TALK TO EDZEL <CR>','','
170 200 FORMAT('If you are Edzel, toiling diligently away at your VDT, yo
180   lu''s will see (and do) the following:','','...the derivative of
190   2aquarLINK FROM <YOUR FRIEND>','','"c",','"HI, WANNA GET TOGETHER
200   ABOUT 1630 TO DISCUSS TOMORROW''S BRIEFING?"','',';Sure...meet you
210   in the Rat''a Nest!'','',' I'llOG, SEE YOU THERE. BYE.','',' @ break','','
220   5/,'','
7' Did you notice that both parties typed either a ! or : to cause
*the system to' //,
8' send the line?', //,
9' Invoking the CONTINUE command at this point will place you back
'to where you', //,
1' were prior to the link. Such as: ', //,
2' @continue', //, 'the derivative of aquar', //,
3' You can then use a "R to get the entire line to reprint.' //,
4' This concludes the demonstration of TALK/LINK.', //,
5' Remember to use the "?" any time you are not quite sure what the
* system is', //, 'asking for.' //
205 FORMAT(3A5)
210 FORMAT(lAI)
300 FORMAT(/// NO. TRY AGAIN.', ///)
310 FORMAT(/// YOU STILL BLEW IT.', ///)
320 FORMAT(/// CARE TO TRY AGAIN (Y OR N)? ', &)
340 FORMAT(' Sorry to see you quit early. Goodbye.', ///)
350 FORMAT(' PRESS ANY KEY . . .', ///)
1600 FORMAT(///)
END
C
C
SUBROUTINE AMSG(DUM)

C THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "MSG" MESSAGE -
C PROCESSOR

DIMENSION INPUT(1),DEM1(1),DEM2(2),DEM3(3),DEM7(7)

**** INTRODUCTION TO MSG

TYPE 1600
TYPE 1000
TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** MSG COMMAND LEVEL COMMANDS

10 TYPE 1010
TYPE 1300
READ(05,1400,ERR=15) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520
TYPE 1020
TYPE 1300
READ(05,1400,ERR=20) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

20 TYPE 1030
TYPE 1300
READ(05,1400,ERR=21) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520
TYPE 1035
TYPE 1300
READ(05,1400,ERR=22) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** SEND, QUEUE OR ?

22 TYPE 1038
TYPE 1320
READ(05,1400,ERR=25) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** DEMO - ENTRY INTO MSG

68
C
25 NUM=0
30 TYPE 1040
32 READ(05,1410,ERR=30) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q') GO TO 520
IF(DEM1(1).EQ.'M'.OR.DEM1(1).EQ.'m') GO TO 50
NUM=NUM+1
GO TO (35,40,45) NUM
35 TYPE 1340
TYPE 1500
GO TO 32
40 TYPE 1350
TYPE 1500
GO TO 32
45 TYPE 1045
TYPE 1360
READ(05,1400,ERR=45) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 25
C
C **** DEMO - SEND
C
C
50 NUM=0
55 TYPE 1050
57 READ(05,1400,ERR=55) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF(INPUT(1).EQ.'S'.OR.INPUT(1).EQ.'s') GO TO 75
NUM=NUM+1
GO TO (60,65,70) NUM
60 TYPE 1340
TYPE 1505
GO TO 57
65 TYPE 1350
TYPE 1505
GO TO 57
70 TYPE 1055
TYPE 1360
READ(05,1400,ERR=70) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 50
C
C **** DEMO - CONFIRMATION
C
C
75 NUM=0
80 TYPE 1060
82 READ(05,1400,ERR=80) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 100
NUM=NUM+1
GO TO (85,90,95) NUM
85      TYPE 1340
         TYPE 1510
         GO TO 82
90      TYPE 1350
         TYPE 1510
         GO TO 82
95      TYPE 1065
         TYPE 1360
         READ(05,1450,ERR=95) INPUT
         TYPE 1600
         IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
         IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 75
C C     **** DEMO - MESSAGE PREPARATION "TO"
C C 100     NUM=0
105     TYPE 1070
107     READ(05,1450,ERR=105) DEM2
         TYPE 1600
         IF(DEM2(1).EQ.'O'.OR.DEM2(1).EQ.'q')GO TO 520
         IF(DEM2(1).EQ.'JONES'.AND.DEM2(2).EQ.'KF'.OR.DEM2(1).EQ.'jones'.AN
          D.DEM2(2).EQ.'kf')GO TO 125
         NUM=NUM+1
         GO TO (110,115,120) NUM
110     TYPE 1340
         TYPE 1515
         GO TO 107
115     TYPE 1350
         TYPE 1515
         GO TO 107
120     TYPE 1075
         TYPE 1360
         READ(05,1440,ERR=120) INPUT
         TYPE 1600
         IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
         IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 100
C C     **** DEMO - MESSAGE PREPARATION "cc"
C C 125     NUM=0
130     TYPE 1080
132     READ(05,1450,ERR=130) DEM2
         TYPE 1600
         IF(DEM2(1).EQ.'O'.OR.DEM2(1).EQ.'q')GO TO 520
         IF(DEM2(1).EQ.'WILLB'.AND.DEM2(2).EQ.'E'.OR.DEM2(1).EQ.'willb'.AND
          A.DEM2(2).EQ.'e')GO TO 150
         NUM=NUM+1
         GO TO (135,140,145) NUM
135     TYPE 1340
         TYPE 1520
         GO TO 132
140     TYPE 1350
         TYPE 1520
         GO TO 132
145     TYPE 1085

70
TYPE 1360
READ(05,1400,ERR=145) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'y'.OR.INPUT(1).EQ.'y')GO TO 125
C
C
**** DEMO - MESSAGE PREPARATION "SUBJECT"
C
150 NUM=0
155 TYPE 1090
157 READ(05,1450,ERR=155) DEM2
TYPE 1600
IF(DEM2(1).EQ.'Q'.OR.DEM2(1).EQ.'q')GO TO 520
IF(DEM2(1).EQ.'PASCA'.AND.DEM2(2).EQ.'L'.OR.DEM2(1).EQ.'space'.AND
DEM2(2).EQ.'1')GO TO 175
NUM=NUM+1
GO TO (160,165,170) NUM
160 TYPE 1340
TYPE 1525
GO TO 157
165 TYPE 1350
TYPE 1525
GO TO 157
170 TYPE 1095
TYPE 1360
TYPE 1600
READ(05,1400,ERR=170) INPUT
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'y'.OR.INPUT(1).EQ.'y')GO TO 150
C
C
**** DEMO - MESSAGE PREPARATION "MESSAGE"
C
175 NUM=0
180 TYPE 1100
182 READ(05,1430,ERR=180) DEM7
TYPE 1600
IF(DEM7(1).EQ.'Q'.OR.DEM7(1).EQ.'q')GO TO 520
IF(DEM7(1).EQ.'WHAT'.AND.DEM7(2).EQ.'PASCA'.AND.DEM7(3).EQ.'L'.AND.
DEM7(4).EQ.'GUAGE'.AND.DEM7(5).EQ.'S ARE'.AND.DEM7(6).EQ.'A'
AND.DEM7(7).EQ.'LABLE'.AND.DEM7(1).EQ.'WHAT'.AND.DEM7(2).EQ.'space'
AND.DEM7(3).EQ.'L lan'.AND.DEM7(4).EQ.'guage'.AND.DEM7(5).EQ.'S are'
AND.DEM7(6).EQ.'avai'.AND.DEM7(7).EQ.'lable')GO TO 200
NUM=NUM+1
GO TO (185,190,195) NUM
185 TYPE 1340
TYPE 1530
GO TO 182
190 TYPE 1350
TYPE 1530
GO TO 182
195 TYPE 1105
TYPE 1360
READ(05,1400,ERR=195) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520

71
IF INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 175
C
**** DEMO - MESSAGE PREPARATION "QUEUE"
C
200 NUM=0
205 TYPE 1110
207 READ(05,1400,ERR=205) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 225
NUM=NUM+1
GO TO (210,215,220) NUM
210 TYPE 1340
TYPE 1350
GO TO 207
215 TYPE 1350
TYPE 1350
GO TO 207
220 TYPE 1110
TYPE 1360
READ(05,1400,ERR=220) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF (INPUT(1).EQ.'y'.OR.INPUT(1).EQ.'Y')GO TO 200
C
**** DEMO T (TYPE) COMMAND
C
225 NUM=0
230 TYPE 1120
237 READ(05,1400,ERR=230) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF (INPUT(1).EQ.'t'.OR.INPUT(1).EQ.'T')GO TO 250
232 NUM=NUM+1
GO TO (235,240,245) NUM
235 TYPE 1340
TYPE 1350
GO TO 232
240 TYPE 1350
TYPE 1350
GO TO 232
245 TYPE 1125
TYPE 1360
READ(05,1400,ERR=245) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF (INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 225
C
**** DEMO - MESSAGE SEQUENCE COMPLETION (ONE MESSAGE)
C
250 NUM=0
255 TYPE 1130
257 READ(05,1440,ERR=255) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
IF(DEM1(1).EQ.'Y'.OR.DEM1(1).EQ.'y')GO TO 275

72
NUM=NUM+1
GO TO (260,265,270) NUM
260 TYPE 1340
TYPE 1540
GO TO 257
265 TYPE 1350
TYPE 1540
GO TO 257
270 TYPE 1135
TYPE 1360
READ(05,1400,ERR=270) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 250
C
**** DEMO - H (HEADERS) COMMAND
C
275 NUM=0
280 TYPE 1140
282 READ(05,1400,ERR=280) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'H'.OR.INPUT(1).EQ.'h')GO TO 300
NUM=NUM+1
GO TO (285,290,295) NUM
285 TYPE 1340
TYPE 1505
GO TO 282
290 TYPE 1350
TYPE 1505
GO TO 282
295 TYPE 1145
TYPE 1360
READ(05,1400,ERR=295) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'H'.OR.INPUT(1).EQ.'h')GO TO 275
C
**** DEMO - MESSAGE SEQUENCE (;)
C
300 NUM=0
305 TYPE 1150
307 READ(05,1460,ERR=305) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DE1M1(1).EQ.'q')GO TO 520
IF(DEM1(1).EQ.'H'.OR.DE1M1(1).EQ.'h')GO TO 326
NUM=NUM+1
GO TO (310,315,320) NUM
310 TYPE 1340
TYPE 1545
GO TO 307
315 TYPE 1350
TYPE 1545
GO TO 307
320 TYPE 1155
TYPE 1360
READ(05,1400,ERR=320) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'y'.OR.INPUT(1).EQ.'Y')GO TO 320
320 TYPE 1160
330 TYPE 1240
330 TYPE 1370
READ(05,1400,ERR=310) INPUT
510 RETURN
520 TYPE 1250
520 TYPE 1370
READ(05,1400,ERR=330) INPUT
530 RETURN

****FORMATS FOR MSG SUBROUTINE*******************************

**** INTRODUCTION

1000 FORMAT(' MSG',//,
1 ' MSG is a program which allows the user to create, send, file and
2 read',//,' messages (referred to also as mail). A listing below p
3 provides the',//,' commands which may be used within MSG. They may
4 also be seen from',//,' within MSG by typing a '? after the ''<'' a
5 msg prompt',//,
6 ' The following is a sample message',//,
7 ' (msg. # 34, 279 chars)',//,
8 ' Mail-From: JONESK@USC-ISI created at 16-Oct-85 19:13:01',//,
9 ' Date: 16 Oct 1985 19:13:01 EDT',//,
10 ' From: JONESK@USC-ISI.ARPA',//,
11 ' Subject: REQUESTED INFORMATION',//,
12 ' To: H1D0N3@USC-ISI.ARPA',//,
13 ' cc: FLEMMING@USC-ISI.ARPA, JONESK@USC-ISI.ARPA',//,
14 ' Return',//,
15 ' Thank you very much for your quick response',//,
16 ' The message itself is comprised of various fields, or parts. Th
17 ey are',//,' referred to as Date Field, To Field, From Field, Subject
18 Field, etc',//,' Each field is entered separately, allowing the
19 user to view the fields',//,' separately, such as just listing the
20 subjects of some or all messages',//)

**** MSG COMMAND LEVEL COMMANDS

1010 FORMAT(' MSG COMMAND LEVEL COMMANDS',//,
1 ' ** COMMAND Use',//,
2 ' A Answer a message',//,
3 ' B Back up to previous message',//,
4 ' C Current message',//,
5 ' D Delete a message',//,
6 ' F Forward a message',//,
7 ' G Go to message number',//,
8 ' H Back up to previous message',//,
1' H Headers',/
2' I Inclusion of length of header',/
3' J Jump into lower fork',/
4' K Koncis - provide shorter prompting',/
5' L List of messages',/
6' M Move a message into a file (deletes it as a message')',/
7' N Next message number',/
8' O Overwrite old file',/

1020 FORMAT(' COMMAND USE',//,
1' P Put a message into a file (leaves it as a message also)',/
2',/
3' Q Quit - return to operating system without update of mag file',/
4' R Read in a file',/
5' S Send a message',/
6' T Type a message',/
7' U Undelete a message',/
8' V Verbose - provides more prompting',/
9' W Write file sorted by message arrival time',/
=' X XED - enter XED',/
1' Z Zap profile',/
2' ' Marks message as examined',/
3' ' Marks message as not examined',/
4' (system tells you upon entry if you have "not-examined = Messages")',/
7' : display current time and date',/
8' ? Type the command character for its description, ? alone
9 for summary',/
=' : Comment',/
1' ' N Abort message',/
2' Z Message completion',/

1030 FORMAT(' COMMANDS WITHIN MESSAGE SEQUENCE',//,
1' If you ask the system to type out a message (T type) or output h
2' Headers',/
3' (H headers), etc. the system will ask you for the message sequen
4' ce',/', Usually, all you desire is one message, so you just type
5' that message',/', number followed by a carriage return. In some i
6' stances you may desire',/
7' to have more than one, possibly a string of message headers or t
8' wo',/', different messages. The following describes the procedur
9' 9es for various',/', means of answering the (Message Sequence) pro
=' apt.',/',
1' . 1. Any single number',/
2' 2. Message number range (ie number separator number, where se
3' operator',/', is one of the following',/
4' COMMAND USE',/
5' > greater than (>20 = output msg with number greater t
6' hen 20)',/',
7' : through (15:20 = msgs 15 through 20)',/
8' (20:15 = same msgs, but in inverse order
9')',/

1035 FORMAT(' COMMAND USE',//,

75
After you have completed typing the body of a message, which you have signaled to the system with a -V, the system will prompt you with a 'QUE: SEND or ?'. This is asking if you want the message sent (do it now while I wait), queued (do it at the system's convenience) or ? (provides help).

**NOTE:** YOU MAY QUIT THE FOLLOWING DEMONSTRATION AT ANY TIME. SIMPLY BY TYPING A -Q FOLLOWED BY A CARRIAGE RETURN, IN REPLY TO ANY OF THE PROMPTS.

To begin, we type an "S<CR>" after the prompt to enter the message send/ routine. Give it a try.

**What you should have typed, was: S<CR>**
1060 FORMAT(' SNDAEG (CONFIRMA', ' "Y<CR>" to confirm, or a "N<CR>" to ca
2ncell. "Sndmag (CONFIRMA', ' "Sndmag (CONFIRMA', ' "Sndmag (CONFIRMA', ' "Sndmag (CONFIRMATION"
1065 FORMAT(' What you should have typed, was: Y<CR>''
1070 FORMAT(' If you wish to abort back to MSG, type control-N', ' ', ', '
2' The system is now asking for the address of your
3message. 'To (? for help): ', 
1075 FORMAT(' What you should have typed, was: JONESKF<CR>''
1080 FORMAT(' cc (? for help): ', 
1085 FORMAT(' What you should have typed, was: WILLBE, (YOUR ACCOUNT NAME)
1090 FORMAT(' Subject:', 
1095 FORMAT(' What you should have typed, was: PASCAL<CR>''
1100 FORMAT(' Message (? for help): ', 
1105 FORMAT(' What you should have typed, was: ', 
1110 FORMAT(' G,S,T, carriage return: ', 
1115 FORMAT(' What you should have typed, was: G<CR>''
1120 FORMAT(' G,S,T, carriage return: ')}
V.

1. JONESKF -- queued'/', ' WILBEE -- queued'/', (your account)
2. -- queued'/', '
3. The system is now processing the message and has given you the M
   4SG command'/', ' level prompt; awaiting your next command. Let's
   5read a message. To do so'/', we enter the T (type) command.'/',
6. '(-', '

The system is now processing the message and has given you the
M "SG command', ' level prompt; awaiting your next command. Let's
read a message. To do so', ' we enter the T (type) command.', '(-', '

The system is now asking for which message(s) you wish to
read. Let's', ' read message 14', '(-', 'type (message
sequence) ', ')

If we had not remembered the exact message number
of this message,
we could', ' have found it with the HEADER command. We could
do this by looking at all', ' "headers", or if we knew its
approximate number we could specify a range of', ' message headers to
be outputed by the system. Let's give this a try. First', ' in
voke the H (header) command.', '(-', ')

From the above we find that the message of interest is number 14.
Note also', ' the other information presented. We can tell th
Sat we received messages 10, 11, and 13 and that we sent messages 9, 13 and 15.

**** SIGN-OFF ROUTINE

1240 FORMAT('This completes the tutorial on MSG. Remember to use the 
1"?" any time you', ' are not quite sure what the system is asking 
2 for.', ')
1250 FORMAT('Sorry to see you quit early. Good bye.', ')
1300 FORMAT('Would you like to continue (Y or N)?', ')
1320 FORMAT('Would you like a demonstration of the above commands (Y or 
1N)?', ')
1340 FORMAT('No. Try again.', ')
1350 FORMAT('You still don't have it right.', ')
1360 FORMAT('CARE TO TRY AGAIN (Y or N)?', ')
1370 FORMAT('ENTER ANY KEY TO RETURN TO THE MAIN MENU: ', ')
1400 FORMAT(1A1)
1410 FORMAT(1A2)
1420 FORMAT(3A5)
1430 FORMAT(1A2)
1440 FORMAT(1A2)
1450 FORMAT(2A5)
1460 FORMAT(1A5)
1500 FORMAT(1A5)
1505 FORMAT(1A5)
1510 FORMAT(1A5)
1515 FORMAT(1A5) TO (? for help): 
1520 FORMAT(1A5) CC (? for help): 
1525 FORMAT(1A5) SUBJECT: 
1530 FORMAT(1A5) Message (? for help): 
1535 FORMAT(1A5) Q.S.? carriage return 
1540 FORMAT(1A5) type (message sequence) 
1545 FORMAT(1A5) headers (message sequence) 
1600 FORMAT(1A5) END
SUBROUTINE AXED(DUM)

---XED SUBROUTINE---

*THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "XED" TEXT EDITOR*

DIMENSION INPUT(1),DEMI(1),DEM2(2),DEM3(3),DEM7(7)

**** INTRODUCTION

TYPE 1600
TYPE 1000
TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** XED COMMAND LEVEL COMMANDS

10 TYPE 1010
TYPE 1300
READ(05,1400,ERR=15) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

15 TYPE 1020
TYPE 1300
READ(05,1400,ERR=20) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** COMMANDS WITHIN INSERT AND APPEND

20 TYPE 1030
TYPE 1320
READ(05,1400,ERR=25) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** DEMONSTRATION OF THE XED COMMAND

25 NUM=0
30 TYPE 1040
32 READ(05,1410,ERR=30) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
IF(DEM1(1).EQ.'XED'.OR.DEM1(1).EQ.'xed')GO TO 50
NUM=NUM+1
GO TO (35,40,45) NUM

35 TYPE 1340
TYPE1500
GO TO 32

40 TYPE 1350
TYPE 1500
GO TO 32
45 TYPE 1045
TYPE 1360
TYPE 1600
READ (05, 1400, ERR=45) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF (INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 25
C
C **** DEMONSTRATION OF THE READ COMMAND
C
50 NUM=0
55 TYPE 1050
57 READ (05, 1400, ERR=55) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF (INPUT(1).EQ.'R'.OR.INPUT(1).EQ.'r') GO TO 75
NUM=NUM+1
GO TO (60, 65, 70) NUM
60 TYPE 1340
TYPE 1505
GO TO 57
65 TYPE 1350
TYPE 1505
GO TO 57
70 TYPE 1055
TYPE 1360
READ (05, 1400, ERR=70) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF (INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 50
C
C **** DEMONSTRATION OF READING IN A FILE
C
75 NUM=0
80 TYPE 1060
82 READ (05, 1420, ERR=80) DEM3
TYPE 1600
IF (DEM3(1).EQ.'Q'.OR.DEM3(1).EQ.'q') GO TO 520
IF (DEM3(1).EQ.'INVIS'.AND.DEM3(2).EQ.'IBLE'.AND.DEM3(3).EQ.'1'.AND.DEM3(1).EQ.'invis'.AND.DEM3(2).EQ.'ible'.AND.DEM3(3).EQ.'1') GO 2 TO 100
NUM=NUM+1
GO TO (85, 90, 95) NUM
85 TYPE 1340
TYPE 1510
GO TO 82
90 TYPE 1350
TYPE 1510
GO TO 82
95 TYPE 1065
TYPE 1360
READ (05, 1400, ERR=95) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUTC1).EQ.'y')GO TO 75

**** DEMONSTRATION OF THE LIST COMMAND

100 NUM=0
105 TYPE 1070
107 READ(05,1400,ERR=105) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'L'.OR.INPUT(1).EQ.'l')GO TO 125
NUM=NUM+1
GO TO (110,115,120) NUM

110 TYPE 1340
TYPE 1515
GO TO 107
115 TYPE 1350
TYPE 1515
GO TO 107
120 TYPE 1075
TYPE 1360
READ(05,1400,ERR=120) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 100

**** DEMONSTRATION OF THE VIEW COMMAND

125 NUM=0
130 TYPE 1080
132 READ(05,1400,ERR=130) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'V'.OR.INPUT(1).EQ.'v')GO TO 150
NUM=NUM+1
GO TO (135,140,145) NUM
135 TYPE 1340
TYPE 1515
GO TO 132
140 TYPE 1350
TYPE 1515
GO TO 132
145 TYPE 1085
TYPE 1360
READ(05,1400,ERR=145) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 125

**** DEMONSTRATION OF THE INSERT COMMAND

150 NUM=0
155 TYPE 1090
157 READ(05,1400,ERR=155) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'I'.OR.INPUT(1).EQ.'i')GO TO 175
NUM=NUM+1
GO TO (160,165,170) NUM

160 TYPE 1340
   TYPE 1520
   GO TO 157

165 TYPE 1350
   TYPE 1520
   GO TO 157

170 TYPE 1095
   TYPE 1360
   READ(05,1400,ERR=170) INPUT
   TYPE 1600
   IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
   IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 150

**** DEMONSTRATION OF INSERTING MATERIAL

175 NUM=0

180 TYPE 1100

182 READ(05,1430,ERR=180) DEM7
   TYPE 1600
   IF(DEM7(1).EQ.'Q'.OR.DEM7(1).EQ.'q')GO TO 520
   IF(DEM7(1).EQ.'S'.AND.DEM7(2).EQ.'S'.AND.DEM7(3).EQ.'E'
      AND.DEM7(4).EQ.'L'.AND.DEM7(5).EQ.'V'.AND.DEM7(6).EQ.'L'
      AND.DEM7(7).EQ.'S'.AND.'S')GO TO 200
   NUM=NUM+1
   GO TO (185,190,195) NUM

185 TYPE 1340
   TYPE 1525
   GO TO 182

190 TYPE 1350
   TYPE 1525
   GO TO 182

195 TYPE 1105
   TYPE 1360
   READ(05,1400,ERR=195) INPUT
   TYPE 1600
   IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
   IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 175

**** DEMONSTRATION OF END OF FILE COMMAND

200 NUM=0

205 TYPE 1110

207 READ(05,1440,ERR=205) INPUT
   TYPE 1600
   IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
   IF(INPUT(1).EQ.'Z'.OR.INPUT(1).EQ.'z')GO TO 225
   NUM=NUM+1
   GO TO (210,215,220) NUM

210 TYPE 1340
   TYPE 1530
   GO TO 207

215 TYPE 1350
DEMONSTRATION OF KILL COMMAND

NUM=0
READ(05,1400,ERR=220) INPUT
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 225
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 200

DEMONSTRATION OF THE WRITE COMMAND

NUM=0
READ(05,1400,ERR=255) INPUT
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 260
IF(INPUT(1).EQ.'W'.OR.INPUT(1).EQ.'w')GO TO 270
NUM=NUM+1
GO TO (260,265,270) NUM
Demonstration of Writing to a File

275 NUM=0
280 TYPE 1140
282 READ(05,1450,ERR=280) DEM2
285 TYPE 1340
286 TYPE 1540
287 GO TO 282
290 TYPE 1350
291 TYPE 1540
292 GO TO 282
295 TYPE 1145
296 TYPE 1360
297 READ(05,1400,ERR=295) INPUT
298 TYPE 1600
299 IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
300 IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 275
305 TYPE 1150
306 TYPE 1155
307 READ(05,1400,ERR=305) INPUT
310 TYPE 1340
311 TYPE 1505
312 GO TO 307
315 TYPE 1350
316 TYPE 1505
317 GO TO 307
320 TYPE 1155
321 TYPE 1360
322 READ(05,1400,ERR=320) INPUT
323 TYPE 1600
324 IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
325 IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 300
330 TYPE 1160

Demonstration of the Zap Command

335 NUM=0
340 TYPE 1140
341 TYPE 1150
342 TYPE 1155
343 READ(05,1400,ERR=335) INPUT
344 TYPE 1600
345 IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
346 IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 300
350 TYPE 1160

Demonstration of the Append Command

355 NUM=0
360 TYPE 1140
361 TYPE 1150
362 TYPE 1155
363 READ(05,1400,ERR=355) INPUT
364 TYPE 1600
365 IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
366 IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 300
370 TYPE 1160
332  READ(05,1400,ERR=330) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'A'.OR.INPUT(1).EQ.'a')GO TO 350
NUM=NUM+1
GO TO (335,340,345) NUM
335  TYPE 1340
TYPE 1515
GO TO 332
340  TYPE 1350
TYPE 1515
GO TO 332
345  TYPE 1165
TYPE 1360
READ(05,1400,ERR=345) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 325
C
C **** DEMONSTRATION OF KILLING MORE THAN ONE LINE
C
350  NUM=0
355  TYPE 1170
360  READ(05,1440,ERR=355) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
IF(DEM1(1).NE.'3K'.AND.DEM1(1).NE.'3k')GO TO 357
TYPE 1173
READ(05,1400,ERR=355) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 325
357  NUM=NUM+1
GO TO (360,365,370) NUM
360  TYPE 1340
TYPE 1515
GO TO 356
365  TYPE 13950
TYPE 1515
GO TO 356
370  TYPE 1175
TYPE 1360
READ(05,1400,ERR=370) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 350
C
C **** DEMONSTRATION OF THE JAM COMMAND
C
375  NUM=0
380  TYPE 1180
382  READ(05,1440,ERR=380) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
IF(DEM1(1).EQ.'3J'.OR.DEM1(1).EQ.'J'.OR.DEM1(1).EQ.'3j'.OR.DEM1(1).EQ.'J')
1.EQ.('J')GO TO 400
   NUM=NUM+1
   GO TO (385,390,395) NUM

385  TYPE 1240
     TYPE 1545
     GO TO 382

390  TYPE 1250
     TYPE 1545
     GO TO 382

395  TYPE 1185
     TYPE 1360.
     READ(05,1400,ERR=395) INPUT
     TYPE 1600
     IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
     IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 375

C  **** DEMONSTRATION OF TYPING OTHER THAN CURRENT LINE
C
400  NUM=0
405  TYPE 1190
407  READ(05,1440,ERR=405) DEM1
     TYPE 1600
     IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
     IF(DEM1(1).EQ.'Y'.OR.DEM1(1).EQ.'y')GO TO 425
     NUM=NUM+1
     GO TO (410,415,420) NUM

410  TYPE 1340
     TYPE 1515
     GO TO 407

415  TYPE 1350
     TYPE 1515
     GO TO 407

420  TYPE 1195
     TYPE 1360.
     READ(05,1400,ERR=420) INPUT
     TYPE 1600
     IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
     IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 400

C  **** DEMONSTRATION OF TYPING MORE THAN ONE LINE
C
425  NUM=0
430  TYPE 1200
432  READ(05,1400,ERR=430) INPUT
     TYPE 1600
     IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
     IF(INPUT(1).EQ.'2')GO TO 450
     NUM=NUM+1
     GO TO (435,440,445) NUM

435  TYPE 1340
     TYPE 1550
     GO TO 432

440  TYPE 1350
     TYPE 1550
     GO TO 432

87
445 TYPE 1205
TYPE 1360
READ(05,1400,ERR=445) INPUT
TYPE 1600
IF(INPUT(1).EQ..'Q'.OR.INPUT(1).EQ..'q')GO TO 520
IF(INPUT(1).EQ..'Y'.OR.INPUT(1).EQ..'y')GO TO 425
C
C **** DEMONSTRATION OF THE EXIT COMMAND
C
450 NUM=0
455 TYPE 1210
457 READ(05,1400,ERR=455) INPUT
TYPE 1600
IF(INPUT(1).EQ..'Q'.OR.INPUT(1).EQ..'q')GO TO 520
IF(INPUT(1).EQ..'E'.OR.INPUT(1).EQ..'e')GO TO 475
NUM=NUM+1
GO TO (460,465,470) NUM
460 TYPE 1340
TYPE 1505
GO TO 457
465 TYPE 1350
TYPE 1505
GO TO 457
470 TYPE 1215
TYPE 1360
READ(05,1400,ERR=470) INPUT
TYPE 1600
IF(INPUT(1).EQ..'Q'.OR.INPUT(1).EQ..'q')GO TO 520
IF(INPUT(1).EQ..'Y'.OR.INPUT(1).EQ..'y')GO TO 450
C
C **** DEMONSTRATION OF FILE STORAGE BEFORE EXITING
C
475 NUM=0
480 TYPE 1220
482 READ(05,1450,ERR=480) DEM2
TYPE 1600
IF(DEM2(1).EQ..'Q'.OR.DEEM2(1).EQ..'q')GO TO 520
IF(DEM2(1).EQ..'INFO'.AND.DEEM2(2).EQ..'FILE'.OR.DEEM2(1).EQ..'info'
1.AND.DEEM2(2).EQ..'file')GO TO 499
NUM=NUM+1
GO TO (485,490,495) NUM
485 TYPE 1340
TYPE 1540
GO TO 482
490 TYPE 1350
TYPE 1540
GO TO 482
495 TYPE 1225
TYPE 1360
READ(05,1400,ERR=495) INPUT
TYPE 1600
IF(INPUT(1).EQ..'Q'.OR.INPUT(1).EQ..'q')GO TO 520
IF(INPUT(1).EQ..'Y'.OR.INPUT(1).EQ..'y')GO TO 475
499 TYPE 1230
**** SIGN-OFF ROUTINE

C

500 TYPE 1240
TYPE 1370
READ(05,1400,ERR=510) INPUT

510 RETURN

520 TYPE 1250
TYPE 1370
READ(05,1400,ERR=530) INPUT

530 RETURN

***** FORMATS FOR XED SUBROUTINE

**** INTRODUCTION

1000 FORMAT('XED',/).
1' XED is a text editor; specifically a line editor. It allows you
to both create and edit text, one line at a time. Entry int
to XED is by typing XED',/,' after the "0" prompt, followed by a <C
4R>',/,,' Definitions:'/,,'5'
6,'/
7' Text Buffer - The text buffer is your current working space.'/
8,'/
7' Print Buffer - The area where material you have "KILL"ed is
placed. Note',/,' only the material from the L
9AST KILL operation is retained.',/,'*
Command Level - The level you are at when you enter XED; distin-
guishable by',/,'the ":" prompt',/,'.*
2' Line Number - Line numbers are assigned by XED to each line in
your text',/,'buffer, starting with line number
3r one. The numbers are for',/,'reference while
within XED and are not stored into the file',/,'*/

**** XED COMMAND LEVEL COMMANDS

1010 FORMAT('XED COMMAND LEVEL COMMANDS',/).
* COMMAND USES (e)',/,'*
1' A - APPEND To enter text after the current line.',/,'*
2' B - BACK UP Create back up file.',/,'*
3' C - CHANGE To modify current lines.',/,'*
4' E - EXIT Exits XED, after file save, back to system li-
vel.',/,'*
5' F - FIND Locates a specific letter, word or ph
phrase.',/,'*
7' G - GROUP Joins current line with following line.',/,'*
8' I - INSERT Insert text before current line.',/,'*
9' J - JAM Jams the contents of Print Dump after the cur-
rent line.',/,'*
1' K - KILL Places current or specified line (a) into Pri
2nt Dump.',/,'*
3' L - LIST Outputs entire text buffer without line num-
bering.',/,'*
5' P - PRINT DUMP Prints contents of Print Dump.',/,'*
6' R - READ Reads into XED the contents of a file.',/,'*
9' S  - SEARCH  Like Find except it looks for all occurrences
  
1020 FORMAT(' COMMAND
  1' T  - TYPE Types out current line.',//,
  2' V  - VIEW Displays the current and following 15 lines.',
  3'/,  
  4' W  - WRITE Writes current contents of text buffer to file
  5e',//,
  6' X  - EXCHANGE Searches for specific text and replaces it with
  7th another.',//,  2  - ZAP Puts your entire file into the
  8 Print Dump.',//,
  9' '  - SWITCH DUMP Swaps contents of the Print and Text Buffers
  '',//,
  1' ',  
  2after.',//,
  3'/,  
  4' /  - TYPE Types the current line.',//,
  5e',//,
  6' $  - LAST LINE Moves you to the last line of the file.',//,
  7' ?  - COMMANDS Displays available commands.'',//,
  8' 'J  - TYPE Types the next line.'',//,
  9' 'Q  - ABORT Aborts partially completed commands.'',//,
  1' ' - FORMAT Right justifies paragraph.'',//

CC **** COMMANDS WITHIN INSERT AND APPEND

1030 FORMAT(' COMMANDS WITHIN INSERT OR APPEND:',',//,
  1' COMMAND USE',//,
  2' (DEL KEY) Deletes the letter to the left of the cursor.
  3'/,  
  4' 'R  - RETYPE Retypes the current line, leaving cursor at the
  5e end.'',//,
  6' 'X  - KILL LINE Kills the line to the left of the cursor.',
  7' 'W  - DELETE WORD Deletes the word to the left of the cursor.',
  8'/,
  9' 'Z  - RETURN Returns you to the command level.'',//,
  1' 'COMANDS WITHIN CHANGE:',',//,
  1' S  - SKIP Skips to a specified character.',
  2' 'D  - DELETE Deletes the character over the cursor.',
  3' 'I  - INSERT Inserts characters.',
  4' 'E  - END Moves cursor to end of line.',
  5' 'D  - ABORT Aborts any changes you have made.',
  6' ?  - COMMANDS Displays available commands.'',//,
  7' 'B  - BREAK Breaks line in two at cursor position.'',//,
  8' <CR> - RETURN Return to XED command level.'',//

CC **** DEMO - ENTRY INTO XED

1040 FORMAT(' NOTE: YOU MAY QUIT THE FOLLOWING DEMONSTRATION AT ANY TIME, SIMPLY BY',', 'TYPING A "Q" FOLLOWED BY A CARRIAGE RETURN, IN REPPLY TO ANY OF THE PROMPTS.',',//,
  1'ME. SIMPLY BY',', 'TYPING A "Q" FOLLOWED BY A CARRIAGE RETURN, IN
  2REPLY TO ANY OF THE PROMPTS.',',//,
  3' 'OK. Here we go. First off, we need to get from the Executive Command level',', 'to the XED command level. To do this, we type XED after the "S" prompt',', 'followed by a <CR>. Give it
3a try.//,' @/,*)

1045 FORMAT(' What you should have typed, was: XED <CR>',//)

C
C **** DEMO - THE R (READ) COMMAND
C
1050 FORMAT(' XED Version 4.2 (30-Aug-82)',//,' :://,',
1' This puts us at the XED command level prompt.',//,',
2' Now we have a choice. We can begin entering text for a new f
3ile or we can'.//,',
4' edit text from an existing file. First let's try editing an
5existing file'.//,',
6' To bring in the existing file, we type an "R" after the prompt.
7' (Note that',//,',
8' during this simulation, you will need to type a <CR> after enter
9ing the first',//,',
* letter of each command. In XED itself, this is not required).
10' So now we type',//,',
11' "R <CR>" after the "":" prompt. The system will respond by
12completing the',//,',
13' command and then asking you what file you wish to insert. Le
14't's give it a',//,',
15' try.\'//,' ;:'+,$)

1055 FORMAT(' What you should have typed, was: R <CR>',//)

C
C **** READING IN A FILE
C
1060 FORMAT(' read',',//,' Input File',',//,
1' At this point you type in the "FileName.FileType.FileNumber" of
2the file',',//,' you wish to edit. Let's edit "INVISIBLE.1"',',//,',
3' Input File',',//)

1065 FORMAT(' What you should have typed, was: INVISIBLE.1 <CR>',//)

C
C **** DEMO - THE L (LIST) COMMAND
C
1070 FORMAT(' 8 LINES',',//,' 1:',',//,
1' From the above, we see that the file has been read into the text
2buffer, is'.//,',
3' eight lines in length and the current line is line 1. Now we wi
4sh to take a',//,',
5' look at this file. Two of the commands we can use to do this ar
6' the "VIEW",//,' *
* and "LIST" commands. Give the "LIST" command a try',',//,',
7' REMEMBER: ONLY TYPE IN THE FIRST LETTER FOLLOWED BY A <CR>',//,'/
8',',',//,' 1:',$)

1075 FORMAT(' What you should have typed, was: L <CR>',//)

C
C **** DEMO - THE V (VIEW) COMMAND
C
1080 FORMAT(' list',',//,' INVISIBLE',',//,
1' Files that have been marked requesting archival are said to be "
2INVISIBLE",',//,' meaning that these filenames will not show up in
3the usual @ Directory',',//,' listing unless a special subcommand is
4given: @invisible(files only).',',//,' Note: Although a file does
5not show up in the regular directory listing, ',',//,' it is still in
6your directory and will not really go away until the Archive',',//,'
7 has been run (on weekends). If you prefer that the filename remain visible, use the "SET FILE VISIBLE" command.

9 Reading the above file, we note a mistake in the last line. We know this is the last line because the file is only eight lines long (we could also count the lines, but this would be a very laborious task in a large file with a mistake somewhere in the middle). Now then are we to find and correct our mistake?

Let's try typing out the contents of the file with the "VIEW" command.

- This allows us a very easy way to determine the line number.

Now we are ready to retype the line. Remember the "R", "W", "2X and Del Key" commands. In fact, make a few mistakes of your own while typing in the correct line, and then use these commands to make corrections. Just insure that you have the line completely right before typing the <CR>.

Now we have corrected line eight, but have been left in position to enter material into line nine. Since we do not desire to enter material in this line, we type a "Z".

5 Use the "SET FILE VISIBLE" command.

DEMO - THE Z COMMAND

1105 FORMAT('What you should have typed, was: Z <CR>

**** DEMO - THE K (KILL) COMMAND

1115 FORMAT('What you should have typed, was: Z <CR>

**** DEMO - THE I (INSERT) COMMAND

1100 FORMAT('insert', '/ <CR>

**** DEMO - INSERTING TEXT

1105 FORMAT('What you should have typed, was: I <CR>

**** DEMO - THE Z COMMAND

1110 FORMAT(' 9<CR>

**** DEMO - THE K (KILL) COMMAND

92
Remember that what we have just done is insert a new line eight 2. That means, that the old line eight has become the new line nine. However, we no longer want this line at all. To get rid of it we use the KILL command 5. To KILL line 9, we start by ensuring that line nine is the 6 current line, which is fact it is. If we were not sure, we would type a "9 <CR>" which would make line 9 the current line. To KILL two lines, we type the K after the "9", "prompt, then type the number "1", which is indicating that we wish to KILL one line. Note: the default for the KILL command is "1", therefore it need not be typed. 0

Get rid of line 9. 1

What you should have typed was one of the following: K <CR> followed by a 1 or blank and a <CR> or 0

K <CR> followed by a 1 or blank and a <CR>.

Now that we are done with this file, we wish to save it, but do not wish to leave XED. Therefore, we use the WRITE command. 0

Go ahead.

What you should have typed was: W <CR>.

writing to a file

We are being asked for the file we wish the material in the text buffer to be written. Send it to INVIS.IBLE.

What you should have typed was: INVIS.IBLE <CR>.

DEMO - THE Z (ZAP) COMMAND

The system has filed the material for use, yet left us both within XED and within the file we were editing. Note that the system responds with "ill", and then type the number "1", which is indicating that we wish to KILL one line. Note: the default for the KILL command is "1", therefore it need not be typed. 0

Get rid of line 9. 1

What you should have typed was: Z <CR>.

DEMO - THE A (APPEND) COMMAND

The system has filed the material for use, yet left us both within XED and within the file we were editing. Note that the system responds with "ill", and then type the number "1", which is indicating that we wish to KILL one line. Note: the default for the KILL command is "1", therefore it need not be typed. 0

Get rid of line 9. 1

What you should have typed was: Z <CR>.

DEMO - THE A (APPEND) COMMAND

The system has filed the material for use, yet left us both within XED and within the file we were editing. Note that the system responds with "ill", and then type the number "1", which is indicating that we wish to KILL one line. Note: the default for the KILL command is "1", therefore it need not be typed. 0

Get rid of line 9. 1

What you should have typed was: Z <CR>.
Good! Now we have an empty text buffer. Let's try entering some original text. This time use APPEND to get into the text entry mode.

What you should have typed, was: A <CR>.

Now we are ready to enter text. Let me do you a favor and enter some for you.

Append i, now we are ready to enter text. Let me do you a favor and enter some for you. Note how at the end of a line, I type a <CR> a 3nd the system provides the next line number for us. This tutorial does not explicitly cover a few of the commands, such as FIND, GROUP, and SEARCH. You should be able to figure these out for yourself.

This is the contents of line three, into the Print Buffer and line four will become line three. Then we ensure that line 3 is the current line and JAM the contents of the Print Buffer back into the text buffer. Remember, the JAM command works like the APPEND command, it places the material in the line AFTER the current line. Go ahead and kill line 3.

Now go ahead and JAM in the material from the print buffer.

To type them both, we first specify the "current line number" (3) then issue the TYPE command (T) and then the number of lines we wish typed (2). Give it a try.

Now let's see if we got it right by typing out line 4a three and, four. This can be done by using the VIEW command 5, the LIST command or the TYPE command. Similarly, the TYPE command can be used to type out each line separately or both together. To type them both, we first specify the "current line number" (3) then issue the TYPE command (T) and then the number of lines we wish typed (2). Give it a try.
FORMAT('3*not, try the HELP command. That is, type H and the com
mand you need help',/,'4*with. Do this from the XED command leve
l as at the "?" prompt',/,'."
3' Now that we have things the way we want them, we're done. Note
4 quit',/,' though. If we issue the QUIT command at this point, we
5 will leave XED as',/,' desired, but we will also lose the materi
6al. We either need to first WRITE',/,'.
7' the contents into a file and then quit, or use the EXIT command.
8 The EXIT',/,' command allows us to write what is currently in th
9e text buffer into a file',/,' before releasing us from XED. (Not
as: You could also use the "E command',/,'.
1' This will take you to the Executive level, but you would lose th
2* material in',/,'.
3' the text buffer, unless your next command is CONTINUE. This wil
4*piece you',/,'."
5' back at the point you were at). To use the EXIT command, we typ
6* the "E",/,'.
3' at the XED command level prompt. Go ahead',/,' :'$(')
1215 FORMAT('What you should have typed, was: E <CR>',/,')
C 1220 FORMAT('exit',/,' Output File:',/,')
1' See! The system needs the name of the file to store the materia
21. Store',/,' it in the Info.file',/,' Output File:',$(')
1225 FORMAT('What you should have typed, was: INFO.FILE <CR>',/,')
C 1230 FORMAT('INFO.FILE.1 !New Generation!',/,'4 lines',/,' @',($(')
1' The system has stored the material for you and placed you back at
2* the Executive',/,' level',($(')
C 1240 FORMAT('This concludes the XED portion of this tutorial. Remembe
1r to use the "$?" to',/,' find the available command, and "HELP Com
2mand Name" to find out how to use a',/,' particular command',($(')
1250 FORMAT('Sorry you have a hot date and have to leave so soon',($(')
1300 FORMAT('Would you like to continue (Y or N)?$',$(')
1320 FORMAT('Would you like a demonstration of the above commands (Y o
1r N)?$', $(')
1340 FORMAT('No. Try again',($(')
1350 FORMAT('You still don't have it right',($(')
1360 FORMAT('Care to try again (Y or N)?$', $(')
1370 FORMAT('Enter any character to return to the main menu: ', $(')
1400 FORMAT('1A1)
1410 FORMAT('1A3)
1420 FORMAT('3A5)
1430 FORMAT('7A5)
1440 FORMAT('1A2)
1450 FORMAT('2A5)
1500 FORMAT('@@', $(')
1505 FORMAT(':::', $(')
1510 FORMAT(//, 'INPUT FILE:, $)
1515 FORMAT(//, '1:, $)
1520 FORMAT(//, '8:, $)
1525 FORMAT(//, '8*, $)
1530 FORMAT(//, '9*, $)
1535 FORMAT(//, '9!, $)
1540 FORMAT(//, 'OUTPUT FILE:, $)
1545 FORMAT(//, '3!, $)
1550 FORMAT(//, '1!TYPE '!, $)
1600 FORMAT(////)
   END

C
C
SUBROUTINE AFTP(DU)

* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "FTP" PROTOCOL *

DIMENSION INPUT(1),DEN1(1),DEN3(3),DEN4(4)

**** INTRODUCTION TO FTP

TYPE 1600
TYPE 1000
TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** FTP COMMANDS

10 TYPE 1010
TYPE 1300
READ(05,1400,ERR=25) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

**** DEMO - ENTRY INTO FTP

25 NUM=0
30 TYPE 1040
32 READ(05,1410,ERR=30) DEN1
TYPE 1600
IF(DEN1(1).EQ.'Q'.OR.DEN1(1).EQ.'q')GO TO 520
IF(DEN1(1).EQ.'FTP'.OR.DEN1(1).EQ.'ftp')GO TO 50
NUM=NUM+1
GO TO (35,40,45) NUM
35 TYPE 1340
TYPE 1500
GO TO 32
40 TYPE 1350
TYPE 1500
GO TO 32
45 TYPE 1045
TYPE 1360
READ(05,1400,ERR=45) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 25

**** DEMO - CONNECTING TO A FOREIGN HOST

50 NUM=0
55 TYPE 1050
57 READ(05,1420,ERR=55) DEN3

97
TYPE 1600
IF(DEM3(1).EQ.'Q'.OR.DE3(1).EQ.'q')GO TO 520
IF(DEM3(1).EQ.'CONNE'.AND.DE3(2).EQ.'CT IS'.AND.DE3(3).EQ.'IE'.O
1R.DE3(1).EQ.'conne'.AND.DE3(2).EQ.'ct is'.AND.DE3(3).EQ.'ie')GO
2 TO 75
NUN.NUN*1
GO TO (60,65,70) NUM

60 TYPE 1340
TYPE 1505
GO TO 57
65 TYPE 1250
TYPE 1505
GO TO 57
70 TYPE 1055
TYPE 1360
READ(O5,1400,ERR=70) INPUT
TYPE 1600
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO 70
50 C
C
**** DEMO - LOGGING INTO AN ACCOUNT ON A FOREIGN HOST

75 NUM=0
80 TYPE 1060
82 READ(O5,1420,ERR=80) DEM3
TYPE 1600
IF(DEM3(1).EQ.'Q'.OR.DE3(1).EQ.'q')GO TO 520
IF(DEM3(1).EQ.'LOG S'.AND.DE3(2).EQ.'REGNI'.AND.DE3(3).EQ.'F 666
1'.OR.DE3(1).EQ.'log s'.AND.DE3(2).EQ.'regni'.AND.DE3(3).EQ.'f 6
266'.OR.DE3(1).EQ.'LOGIN'.AND.DE3(2).EQ.'SREG'.AND.DE3(3).EQ.'N
3IF 6'.OR.DE3(1).EQ.'login'.AND.DE3(2).EQ.'SREG'.AND.DE3(3).EQ.'N
4'ni 6')GO 100
NUM=NUM+1
GO TO (85,90,95) NUM

85 TYPE 1340
TYPE 1510
GO TO 82
90 TYPE 1350
TYPE 1510
GO TO 82
95 TYPE 1065
TYPE 1360
READ(O5,1400,ERR=95) INPUT
TYPE 1600
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 75
C
C
**** DEMO - SENDING A FILE

100 NUM=0
105 TYPE 1070
107 READ(O5,1470,ERR=105) DEM4
TYPE 1600
IF(DEM4(1).EQ.'Q'.OR.DE4(1).EQ.'q')GO TO 520
1.'AND.DEM4(4).EQ.'XE'.OR.DEM4(1).EQ.'send'.AND.DEM4(2).EQ.'home'
2.'AND.DEM4(3).EQ.'ork'.AND.DEM4(4).EQ.'xe')GO TO 125
NUM=NUM+1
GO TO (110,115,120) NUM
110 TYPE 1340
TYPE 1510
GO TO 107
115 TYPE 1350
TYPE 1510
GO TO 107
120 TYPE 1075
TYPE 1360
READ(05,1400,ERR=120) INPUT
TYPE 1600
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 100
C
C DEMO - RETRIEVING A FILE WITH FTP
C
125 NUM=0
130 TYPE 1080
132 READ(05,1470,ERR=130) DEM4
TYPE 1600
IF(DEM4(1).EQ.'O'.OR.DE4(1).EQ.'q')GO TO 520
IF(DEM4(1).EQ.'GET'.AND.DE4(2).EQ.'TG'.AND.DE4(3).EQ.'1' TTG
2'.AND.DE4(3).EQ.'1 ttg'.AND.DE4(4).EQ.'g.1')GO TO 150
NUM=NUM+1
GO TO (135,140,145) NUM
135 TYPE 1340
TYPE 1510
GO TO 132
140 TYPE 1350
TYPE 1510
GO TO 132
145 TYPE 1085
TYPE 1360
READ(05,1400,ERR=145) INPUT
TYPE 1600
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 520
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 125
C
C **** DEMO - CLOSING A CONNECTION
C
150 NUM=0
155 TYPE 1090
157 READ(05,1410,ERR=155) DEM1
TYPE 1600
IF(DEM1(1).EQ.'O'.OR.DEM1(1).EQ.'q')GO TO 520
IF(DEM1(1).EQ.'BYE'.OR.DEM1(1).EQ.'bye')GO TO 175
NUM=NUM+1
GO TO (160,165,170) NUM
160 TYPE 1340
TYPE 1510
GO TO 157
165 TYPE 1250
    TYPE 1310
    GO TO 157
170 TYPE 1095
    TYPE 1360
    READ(05,1400,ERR=170) INPUT
    TYPE 1600
    IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
    IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 150

C
C    **** DEMO - LEAVING FTP
C
175 NUM=0
180 TYPE 1100
182 READ(05,1460,ERR=180) DEM1
    TYPE 1600
    IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q') GO TO 520
    IF(DEM1(1).EQ.'QUIT'.OR.DEM1(1).EQ.'quit') GO TO 200
    NUM=NUM+1
    GO TO (185,190,195) NUM
185 TYPE 1340
    TYPE 1505
    GO TO 182
190 TYPE 1350
    TYPE 1505
    GO TO 182
195 TYPE 1105
    TYPE 1360
    READ(05,1400,ERR=195) INPUT
    TYPE 1600
    IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520
    IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y') GO TO 175
200 NUM=0
205 TYPE 1110
C
C    SIGN-OFF ROUTINE
C
500 TYPE 1240
    TYPE 1370
    READ(05,1400,ERR=510) INPUT
510 RETURN
520 TYPE 1250
    TYPE 1370
    READ(05,1400,ERR=530) INPUT
530 RETURN

C
C    ********** FORMATS FOR FTP SUBROUTINE ****************
C
C
C    **** INTRODUCTION
C
1000 FORMAT('./
    FTP - FILE TRANSFER PROTOCOL',/, //
    'FTP is a program employed to transfer files around the net. This
    can be done'//,' from within the same host or between two differ-
    ent hosts. It does necessitate'//,' that you have access to a dir
4ectory on each host (the directory which has the'/.,' material you
5 desire and the directory you wish the material to be transferred'
6'/.,' to',,'/,'.
7' The following prompts are unique to FTP:',,'/
8' FTP>' FTP command level prompt:',,'/
9' USC-ISIE.ARPA> Foreign host command level prompt:',,'/)

**** FTP COMMANDS
1010 FORMAT(' The following commands are peculiar to FTP:',,'/
1' Command Use',','/
2' FTP Invoke the FTP protocol',','/
3' CONNECT Connects you from the local host (the one you logged
4 into) to the',','/
5' foreign host (the one you are trying to get to)',',,'/
6' SEND Sends a file from your local account to the remote a
7ccount',',,'/
8' GET Retrieves a file from the remote account to your loc
9al account',',,'/

* Disconnect you from the foreign host',',,'/
1' BYE
2' QUIT Leaves FTP',',,'/
3' EXIT Same as QUIT',',,'/

**** DEMO - ENTERING FTP
1040 FORMAT(' Note: you may quit the following demonstration at any ti
1NE, simply by',',,' typing a "Q" followed by a carriage return. In
2reply to any of the prompts',',,'/
3' Here we go. To begin with, we need to get from the Executive Le
4vel to the FTP',',,' Command Level. We do this by entering "FTP" a
5fter the "$" prompt. Go ahead',',,' /
6' What you should have typed, was: FTP<CR>',',,'/

**** DEMO - CONNECTING TO A FOREIGN HOST
1050 FORMAT(' USC-ISIE.ARPA FTP user process 6(405)-4',',,' FTP>',',,'/
1' Very good. Now we are at the FTP Command Level prompt. At this
2point, we',',,' want to connect to the desired foreign host. Let'
3's use ISIE',',,'/
4' FTP>',',,'/
5' What you should have typed, was: CONNECT ISIE<CR>',',,'/

**** DEMO - LOGIN AT FOREIGN HOST
1060 FORMAT(' Connection opened (Assuming TYPE L 96, MODE 3, STRU P)',',,'/
1',',,'< USC-ISIE.ARPA FTP Server Process 52(27)-7 at (today''s date &
2 time)',',,'/
3' USC-ISIE.ARPA>',',,'/
4' Our connection is now open and we are ready to log into an accou
5nt on',',,' this foreign host. Unlike the ISIA host, other hosts ma
6ny require the',',,' login to begin with either LOGIN or LOG. Let''
7's use the SREGNIF account',',,' which has a password of 666 (the nu
8eber of the beast)',',,'/
9' USC-ISIE.ARPA>',',,'/

101
**DEMO - SENDING A FILE**

1065 FORMAT(' What you should have typed, was:/',
1 'LOG SREGNIF 666 or LOGIN SREGNIF 666',//)

1070 FORMAT(' OK. Now we are ready to send a file to the foreign host. We do this with',/,' the SEND command followed by the filename we are sending followed by the',/', 'filename it is to have at the remote host. NOTE: If a file name for the', '/', 'foreign host is not specified, it will keep the same name. Let''s send', '/', 'HOMEWORK EXE', 'USC-ISIE.ARPA>',//,
1 'USC-ISIE.ARPA>',#)

1075 FORMAT(' What you should have typed, was:/',
1 'SEND HOMEWORK.EXE HOMEWORK.EXE<CR>',//)

**DEMO - RECEIVING A FILE**

1080 FORMAT(' What you should have typed, was:/',
1 'GET WTGGT.1 TTGG.1<CR>',//)

**DEMO - LEAVING FTP**

1090 FORMAT(' What you should have typed, was: BYE<CR>',//)

**SIGN-OFF ROUTINE**

1240 FORMAT(' This concludes the FTP portion of this tutorial. Remember 1r to use the "?" if you are not sure what the system is asking for',//)

1250 FORMAT(' Sorry to see you quit early. Good bye.',//)

1300 FORMAT(' Would you like to continue (Y or N)?',#)

1320 FORMAT(' Would you like a demonstration of the above commands (Y or N)?',#)

1340 FORMAT(' EGAD, Late night last night, or what?',//)

1350 FORMAT(' You still don''t have it right.',//)

1360 FORMAT(' Care to try again (Y or N)?',#)

102
1370 FORMAT(' ENTER ANY KEY TO RETURN TO THE MAIN MENU',$)
1400 FORMAT(1A1)
1410 FORMAT(1A3)
1420 FORMAT(3A5)
1460 FORMAT(1A5)
1470 FORMAT(4A5)
1500 FORMAT(/, ' $', $)
1505 FORMAT(/, ' FTP', $)
1510 FORMAT(/, ' USC-ISIE.ARPA>', $)
1500 FORMAT(/, $)
END
C
C
SUBROUTINE ATN(DUN)

* ------------------ TELNET (TN) SUBROUTINE ------------------ *
* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "TN" PROGRAM *
* ------------------ TELNET (TN) SUBROUTINE ------------------ *

DIMENSION INPUT(1), DEM1(1), DEM2(2)

INTRODUCTION TO TN

TYPE 1600
TYPE 1000
TYPE 1300
READ(05, 1400, ERR=10) INPUT
TYPE 1600
IF (INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n') GO TO 520

10 TYPE 1010
READ(05, 1400, ERR=25) INPUT
TYPE 1600
IF (INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n') GO TO 520

C **** DEMO - ENTRY INTO TN

25 NUM=0
30 TYPE 1040
32 READ(05, 1430, ERR=30) DEM2
TYPE 1600
IF (DEM2(1).EQ.'O'.OR.DEM2(1).EQ.'q') GO TO 520

40 TYPE 1340
45 TYPE 1350
TYPE 1350
GO TO 32

45 TYPE 1045
TYPE 1360
READ(05, 1440, ERR=45) INPUT
TYPE 1600
IF (INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q') GO TO 520

50 TYPE 1050
57 READ(05, 1430, ERR=55) DEM2
TYPE 1600
IF (DEM2(1).EQ.'O'.OR.DEM2(1).EQ.'q') GO TO 520

C **** DEMO - SHOW COMMAND

50 NUM=0
55 TYPE 1050
57 READ(05, 1430, ERR=55) DEM2
TYPE 1600
IF (DEM2(1).EQ.'O'.OR.DEM2(1).EQ.'q') GO TO 520

104
1D.DEM2(2).EQ.'me')GO TO 75
   NUM=NUM+1
   GO TO (60,65,70) NUM
60   TYPE 1340
       TYPE 1500
       GO TO 57
65   TYPE 1350
       TYPE 1500
       GO TO 57
70   TYPE 1085
       TYPE 1360
       READ(05,1400,ERR=70) INPUT
       TYPE 1600
       IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 520
          IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 50
       C
C      **** DEMO - LEAVING ">>" LEVEL OF TN
       C
75   NUM=0
80   TYPE 1060
82   READ(05,1460,ERR=80) DEM1
       TYPE 1600
       IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 520
          IF(DEM1(1).EQ.'LOGO'.OR.DEM1(1).EQ.'logo')GO TO 100
       NUM=NUM+1
       GO TO (85,90,95) NUM
85   TYPE 1340
       TYPE 1500
       GO TO 82
90   TYPE 1350
       TYPE 1500
       GO TO 82
95   TYPE 1065
       TYPE 1360
       READ(05,1400,ERR=95) INPUT
       TYPE 1600
       IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
          IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 75
       C
C      **** DEMO - LEAVING TN
       C
100  NUM=0
C
C      **** SIGN-OFF ROUTINE
105  TYPE 1240
      TYPE 1370
      READ(05,1400,ERR=510) INPUT
510  RETURN
520  TYPE 1250
      TYPE 1370
      READ(05,1400,ERR=530) INPUT
530  RETURN
C
INTRODUCTION

TN is a program or system which allows a user on one host to communicate with another host. It actually appears as though you logged into the foreign host directly from a "TAC" rather than through another host. The system also has the advantage of making your control characters do what you expect them to do, even if the control characters on the foreign host are different.

Using TN is actually quite simple. All you need do is type "TN <host>" after the Executive Level prompt followed by a carriage return. If a pathway to the foreign host is available and the host is up, you will connect to the foreign host ready to log into a directory. When you are finished, simply log out of the directory, then type "EXIT" after the "TN" prompt. You're back in your own directory.

TN COMMAND CHARACTERS

It should be noted, that since all control characters are meant to behave on the foreign host as they would on the local host, you cannot use "C" to return to your own Executive Level. There is a control character "C" followed by a C, but you should.

really log out of the foreign host properly, if possible.

Would you like a brief demonstration of TN (Y or N)?

OK. First type TN and the name of the host you want to go to. We'll use ISIE for this demonstration. @

First type TN and the name of the host you want to go to. We'll use ISIE for this demonstration. @

Trying... Open...

There are 37*10 jobs and the Load Average is 1.66.

After login, type "HELP ?" followed by a carriage return for a list of on-line help topics.

Looks familiar eh? Now we can log into a directory on this host.

the directory "SHOW" with the password "ME". Did you realize th
at you need',
8' not use the command "LOG" or "LOGIN" to log on to an account? A
9'll that is'.
* really required is the account name and the password. Try it he
10're.'//.'@'.8')

1055 FORMAT(' What you should have typed, was: SHOW ME<CR>',//)

1060 FORMAT(' Job 51 ON tty234 (date and time)',//,
1' Previous LOGIN: (date and time)',//,
2' PS: <SHOW>',//,
3' 37 Pages assigned',//,
4' 200 Working pages, 200 Permanent pages allowed',//,
5' 15638 Pages free on PS: 25354 pages used.',//,
6' No new mail exists',//,
7' With clothes new are the best, with friends the old are best.',//,
8' End of LOGIN.CMD.1', '//.'@'.8')
9' Very good. Well I promised this would be brief; brief it will b
*s. While',//,' you are on the foreign host, everything works just
10' as it would on your own',//,' host. OK. Let's go home. First we
11' log out.',//.'@'.8')

1065 FORMAT(' What you should have typed, was: LOGO<CR>',//)

1240 FORMAT(' @',//,' Welcome home. Note: If when you had entered TN
1' initially, you had entered',//,
2' "TN <CR>" you would have gotten the TN> prompt. This is fine. J
3' ust give the',//,
2' host name at this point. Upon logout you will be left at the TN>
4' prompt.',//,
3' Another EXIT here and you are back home.',//,
4' This concludes the TN portion of this tutorial. Just remember t
the "C",//,
5' control character if you get stranded while in TN.',//)

1250 FORMAT(' Sorry to see you quit early. Good bye.',//)
1300 FORMAT(' WOULD YOU LIKE TO CONTINUE (Y or N)? ',S)
1340 FORMAT(' What a serious case of fumble fingers you have.',//)
1350 FORMAT(' You still don't have it right.',//)
1360 FORMAT(' CARE TO TRY AGAIN (Y or N)? ',S)
1370 FORMAT(' ENTER ANY KEY TO CONTINUE',S)
1400 FORMAT(1A1)
1430 FORMAT(2A5)
1460 FORMAT(1A4)
1500 FORMAT(1A8)
1505 FORMAT(1A8)
1600 FORMAT(1A4)

END

SUBROUTINE APHOTO(DUM)

*C          ------PHOTO SUBROUTINE---------------------------
    *  THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "PHOTO" PROGRAM *

*DIMENSION INPUT(1)

**** INTRODUCTION TO PHOTO

TYPE 1600
TYPE 1000
TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 530

**** VIEWING / CHANGING / BEEPS

10 TYPE 1010

**** SIGN-OFF ROUTINE

500 TYPE 1240
TYPE 1370
READ(05,1400,ERR=510) INPUT
510 RETURN
520 TYPE 1250
TYPE 1370
READ(05,1400,ERR=530) INPUT
530 RETURN

**********FORMATS FOR PHOTO SUBROUTINE***********************

**** INTRODUCTION

1000 FORMAT('PHOTO',//,
1' PHOTO is a program which allows you to record, or save, portions
2 of a',//,
3 of a computer session. It is invoked by typing "PHOTO" <CR> at the "
4 40" or'//,
5 5 Executive Level prompt. You will then be asked to supply a file
6 name'//,
7 to which the PHOTO session is to be recorded. PHOTO will record
8 what'//,
9 you type, as well as the responses back from the computer. While
10 you'//,
11 are in PHOTO, everything will work as normal. The control char
12 ters'//,
13 still remain as before and you still get the same prompts. What
14 this'//,
15 means is, you cannot use 'C to exit from PHOTO. The normal see
6ns of', /
7' exiting PHOTO is to type "POP" followed by a <CR> at the "8" pro
8apt',/',
9' There is one control character peculiar to PHOTO; \Y (control Y
\*). It',/',
1' is used to suspend the PHOTO session. When you are ready to con
2tinue',/',
3' recording, input another \Y',/',
4' One additional note. You must first POP out of PHOTO before you
5 can',/',
6' log out',/',

C C
**** BEEPS ETC

1010 FORMAT( ' If you check your directory after POPing out of PHOTO, yo
1u will see the new', '/',
2' file. You can use the TYPE command or enter XED to view it. XE
3D will also', '/',
4' allow you to edit it', '/',
5' While in PHOTO the system beeps at you (every 30 seconds by defa
6ult). If you', '/',
7' desire to change the frequency of the beeps or eliminate them en
7tirely, you', '/
8' use \INTERVAL:(frequency in seconds)' following PHOTO', '/
9' \@PHOTO/INTERVAL:O - Removes the beeps', '/
*',
* \@PHOTO/INTERVAL:60 - Gives you beeps once a minute', '/

C C
SIGN-OFF ROUTINE

1240 FORMAT( ' This concludes the PHOTO section of this tutorial', '/
1250 FORMAT( ' Sorry you have to leave so soon. Hurry back', '/
1300 FORMAT( ' ENTER A 0C" TO QUIT. ANY OTHER KEY TO CONTINUE. ', '/
1370 FORMAT( ' ENTER ANY KEY TO RETURN TO THE MAIN MENU: ', '/
1400 FORMAT(1A1)
1600 FORMAT(////)
END

C C
SUBROUTINE ARND(DUM)

---------------------------------REMIND SUBROUTINE---------------------------------
* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE REMIND MESSAGE *
* TICKLER SYSTEM. *
---------------------------------REMIND SUBROUTINE---------------------------------

DIMENSION INPUT(1),DEM1(2),DEM2(1),DEM3(2)

***** INTRODUCTION TO REMIND

TYPE 1600
TYPE 1000

***** QUERY TO SEE IF USER WOULD LIKE TO CONTINUE

TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(.EQ.'N'.OR.INPUT(.EQ.'n')GO TO 300

***** DEMONSTRATION OF ENTERING REMIND PROGRAM

10 NUM=0
30 TYPE 1010
READ(05,1410,ERR=30) DEM1
TYPE 1600
IF(DEM(1).EQ.'Q'.OR.DEM1(.EQ.'q')GO TO 300
IF(DEM(1).EQ.'REMIN'.AND.DEM1(2).EQ.'D'.OR.DEM1(1).EQ.'rem1'.AN
1D.DEM1(2).EQ.'d')GO TO 50
NUM=NUM+1
GO TO (35,40,45) NUM
35 TYPE 1340
 GO TO 30
40 TYPE 1350
 GO TO 30
45 TYPE 1020
 TYPE 1360
READ(05,1400,ERR=45) INPUT
TYPE 1600
IF(INPUT(.EQ.'Q'.OR.INPUT(.EQ.'q')GO TO 300
IF(INPUT(.EQ.'Y'.OR.INPUT(.EQ.'y')GO TO 10

***** DEMONSTRATION OF REMIND OPTIONS

50 NUM=0
55 TYPE 1030
60 READ(05,1400,ERR=65) DEM2
TYPE 1600
IF(DEM2(.EQ.'Q'.OR.DEM2(.EQ.'q')GO TO 300
IF(DEM2(.EQ.'?'')GO TO 80
65 TYPE 1040
 GO TO 60
**** DEMONSTRATION OF CREATING A REMIND MESSAGE

90 TYPE 1060
READ(05,1410,ERR=90) DEM3
TYPE 1600
IF(DEM3(1).EQ.'Q'.OR.DEM3(1).EQ.'q')GO TO 300
IF(DEM3(1).EQ.'CREATE'.AND.DEM3(2).EQ.'E'.OR.DEM3(1).EQ.'creat'.AND.DEM3(2).EQ.'e')GO TO 150
NUM=NUM+1
GO TO (110,115,120) MUM

110 TYPE 1350
GO TO 90
115 TYPE 1340
GO TO 90
120 TYPE 1065
TYPE 1360
READ(05,1400,ERR=120) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 90

**** DEMONSTRATE WRITING A MESSAGE

150 NUM=0
200 TYPE 1080
210 TYPE 1370
READ(05,1400,ERR=220) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300
220 TYPE 1090

SIGN-OFF ROUTINE

C

240 TYPE 1100
TYPE 1380
READ(05,1400,ERR=260) INPUT
260 RETURN
300 TYPE 1200
320 TYPE 1380
READ(05,1400,ERR=340) INPUT
340 RETURN

**** EXPLANATION OF REMIND

1000 FORMAT(20X,'REMIND'/,
REMIND is a program which allows you to create reminders for yo
2. others) which will be sent by the system at a specified time.
3. schedule a reminder to be sent only once, or as frequently as
   you like.
4. You may also select whether the reminder will be mailed to your
   mailbox, or:
5. displayed on your terminal screen, or both.

1010 FORMAT: To use the REMIND program, type "REMIND <CR>" at the E
   executive Level prompt.
1020 FORMAT: What you should have typed was: REMIND <CR>.
1030 FORMAT: After you have entered "REMIND <CR>" the following will
   appear on your screen:
1040 FORMAT: What you should have typed was a question mark.
1050 FORMAT: CREATE
1055 FORMAT: The above list comprises the options available to you in
   REMIND. Most of them are self explanatory, however, it is helpful to know that DAYTIM
   E will tell you the current exact month, date and time, while SURVEY lists all R
   EMIND messages you have sent recently. Use the HELP command to find out about t
   he other.
1060 FORMAT: Let's try CREATEing a REMIND message. Type CREATE <CR> after
   the REMIND> prompt.
1065 FORMAT: What you should have typed was: CREATE <CR>.
1080 FORMAT: The REMIND message format has a number of fields for you
   to enter the:
112
A message will look like:\",\",
REMIND\textbackslash CREATE\textbackslash ,\,
First reminder at: 1234\textbackslash ,\,
First reminder will be sent Monday, November 25, 1985 12:34PM\textbackslash ,\,
Frequency of reminders: ONCE\textbackslash ,\,
Notify mode: MAIL\textbackslash ,\,
To: JONES\textbackslash DTW\textbackslash ,\,
Subject: MEETING AT 1400 TOMORROW\textbackslash ,\,
Mag:\textbackslash ,\,
\texttt{=> HEY, DAN!!\textbackslash ,\,
I FORGET TO TELL YOU WE HAVE A MEETING ON THE JCS CHANGES\textbackslash ,\,
PROPOSAL TOMORROW AFTERNOON. BE PREPARED TO SUPPORT YOUR VIEWS!\textbackslash ,\,
Notice you end the REMIND message with a "Z.\textbackslash ,\,
A word of advice: When prompted for mode of notification, you have three\textbackslash ,\,
options: SEND, MAIL or BOTH. SEND will result in a receipt only if the\textbackslash ,\,
recipient is logged on to the host when the REMIND is sent (it is sent directly\textbackslash ,\,
to his screen at send time). MAIL causes the REMIND to be sent only to the\textbackslash ,\,
recipient's mail box. BOTH will result in both SEND and MAIL\textbackslash ,\,
That's all there is to it! The message is automatically queued then\textbackslash ,\,
forwarded at the time indicated\textbackslash ,\,
The next time JONES\textbackslash DTW (Dan) logs onto the host after 12:34PM\textbackslash ,\,' on 25 November 1985, he will be told he has mail from the REMIND\textbackslash ,\,' Daemon in his MSG file. By going into MSG, finding out the number of\textbackslash ,\,' the message, and giving the "TYPE (number) < 4CR>" command, the REMIND \textbackslash ,\,' message will be printed out\textbackslash ,\,
This concludes the REMIND portion of this tutorial\textbackslash ,\,
Sorry to see you quit so soon. Goodbye\textbackslash ,\,
Would you like to continue (Y or N)? ',$\textbackslash 
No. Try again, please.\textbackslash ,\,
Oops! I'll pretend I didn't see that!!\textbackslash ,\,
Do you want another chance (Y or N)? ',$\textbackslash 
Enter a "Q" to quit, any other key to continue: ',$\textbackslash 
Enter any key to return to main menu: ',$\textbackslash 
\texttt{Enter any key to return to main menu: \textbackslash ,\,
C C

113
SUBROUTINE AFNGR(DUM)

*************************************************************************
* FINGER SUBROUTINE                                                      *
* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE FINGER LOCATE           *
* AND IDENTIFY USER SYSTEM                                              *
*************************************************************************

DIMENSION INPUT(1),DEM1(2),DEM2(3)

**** INTRODUCTION TO FINGER

TYPE 1600
TYPE 1000

**** QUERY TO SEE IF USER WOULD LIKE TO CONTINUE

TYPE 1370
READ(OS,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300

**** DEMONSTRATION OF ENTERING FINGER PROGRAM

10 NUM=0
30 TYPE 1010
READ(OS,1410,ERR=30) DEM1
TYPE 1600
IF(DEM1(1).EQ.'Q'.OR.DEM1(1).EQ.'q')GO TO 300
IF(DEM1(1).EQ.'FINGE'.AND.DEM1(2).EQ.'R'.OR.DEM1(2).EQ.'r')GO TO 70
1D.DEM1(2).EQ.'r')GO TO 70
NUM=NUM+1
GO TO (35,40,45) NUM
35 TYPE 1340
GO TO 30
40 TYPE 1250
GO TO 30
45 TYPE 1020
TYPE 1360
READ(OS,1400,ERR=45) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 10

**** DEMONSTRATION OF FINGERING YOUR HOST

70 NUM=0
80 TYPE 1050
TYPE 1370
READ(OS,1400,ERR=85) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300
85 TYPE 1055
TYPE 1370
READ(05,1400,ERR=88) INPUT  
88 TYPE 1600  
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300  
TYPE 1058  
TYPE 1370  
READ(05,1400,ERR=90) INPUT  
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 300

*DEMONSTRATION OF CREATING A FINGER.PLAN*

NUM=0  
90 TYPE 1060  
READ(05,1415,ERR=90) DEM2  
TYPE 1600  
IF(DEM2(1).EQ.'Q'.OR.DEM2(1).EQ.'q')GO TO 300  
TYPE 1370  
IF(DEM2(1).EQ.'FINGE'.AND.DEM2(2).EQ.'R JON'.AND.DEM2(3).EQ.'ESDTW 1'.OR.DEM2(1).EQ.'finge'.AND.DEM2(2).EQ.'r jon'.AND. DEM2(3).EQ.'es 2dtw')GO TO 150  
NUM=NUM+1  
GO TO (110,115,120) NUM  
110 TYPE 1350  
GO TO 90  
115 TYPE 1340  
GO TO 90  
120 TYPE 1065  
TYPE 1360  
READ(05,1400,ERR=120) INPUT  
TYPE 1600  
IF(INPUT(1).EQ.'O'.OR.INPUT(1).EQ.'q')GO TO 300  
121 TYPE 1370  
IF(INPUT(1).EQ.'Y'.OR.INPUT(1).EQ.'y')GO TO 89  

*DEMONSTRATION OF WHAT A FINGER.PLAN LOOKS LIKE*

NUM=0  
150 TYPE 1080  
C  

********** FORMATS FOR FINGER SUBROUTINE **********

C  

********** EXPLANATION OF FINGER **********

1000 FORMAT(20X,'FINGER',/,
1' FINGER is a program to help locate and identify users on a syst
em. It is',/,

115
2. Called from the Executive Level by the FINGER command. Its main features are:
3. Personal name and line location output. FINGER will also allow you to leave a "plan" in a file called FINGER.PLAN to be printed out if.
4. Somebody FINGERs you and you are not logged on. This file is a "free-form text.".
5. File and the file protection should be at least 775252 to allow other users to.
6. Read it.

1010 FORMAT('To FINGER users on a system, type "FINGER <CR>" at the Executive Level prompt.
1. Go ahead and try it (Remember, you can always quit by entering a "Q" at any prompt)."
2."

1020 FORMAT('What you should have typed was: FINGER <CR>"

1050 FORMAT('User Personal name Job Subsys Idle TTY Co
Insole location. CLYNN Charles Lynn 15 PICKLE
2 344 Batch job. HERBERTMK M.K. Herbert 37 TYPE
3 .212 NPS-TAC.ARPA#27. SMTP System Utility 6 S
4XTPSD .355 Job O, OPERATOR, SYSJOB. WEST TA WEST
5 28 XED 211 NPS-TAC.ARPA#13."

1055 FORMAT('Other calling sequences for FINGER.PLAN include:
1. S FINGER / <switches> <username>
2. S FINGER (USERNAME) / <SWITCHES>
3. (USERNAME) May be blank to mean all users, or a local user name
4. or. If the username is not specified, a terse summary
5. of all users is output ordered alphabetically by username.
6. If a username is specified the output is a more detailed, showing the normal one-line job status,
7. the last logout time and line if the user is not logged in, the last time the mail was read, and
8. whether there are any pending messages, and the user's current FINGER.PLAN.
9."

1058 FORMAT('<SWITCHES> If no SWITCHES are specified the default are:
1. /DETACHED Display detached jobs.
2. /HELP Show this message.
3. /NO-DETACHED Suppress display of detached jobs.
4. /NO-OPERATOR Suppress display of operator jobs.
5. /OPERATOR Display operator jobs.
6. /TERSE Don't output plan and mail information just the line job status.
7. /VERBOSE Output plan and mail info and whatever info the person has in the INQUIR database, except for the account. Ignore if no username given.
8. /WHOIS Useful for remote users. Site dependent info is played about the user. For any ISI site the INQUIR data exists the following is displayed:
9. personal name, network address, the home and phone, work address and phone, project
***** MAKING YOUR OWN FINGER.PLAN

1060 FORMAT(' To make your own FINGER.PLAN, simply make up a file in XE
10 ID, set '/,' file protection at 775252, and title it FINGER.PLAN. Y
2ou '/,' can update any time you wish, of course. Let's see what
3FINGERing', '/,' an individual looks like. Type FINGER JONESDTW <CR>
4 after ', '/,' the Executive Level prompt.', '/,' S', '/,'

1065 FORMAT(' What you should have typed was: FINGER JONESDTW <CR>', '/,')

1080 FORMAT(' JONES DANIEL THOMAS W., JONESDTW not logged in',
1 //,' Last logout Tue 26-Nov-85 14:01 from TTY233', '/,',
2 ' Netmail from WBA0MIT-XX.ARPA at Tue 26-Nov-85 17:31, last read'
3 //, ' on Tue 26-Nov-85 12:58', '/,',
4 ' Plan': '/,',
5 ' This is my plan... this is my dream... to hahahah..... ', '/,',
6 ' Sorry I''m not here to talk to you!! Leave a MSG; I''ll be in'
7 //,' touch mos skosh!!!', '/,',
8 ' To learn more about a person, type ''FINGER <username>/VERBOSE.'
9 'You will get', '/,',
9 ' the FINGER information as in the above example plus the informat
10 'ion from the', '/,',
11 ' person''s INQUIR file. (See INQUIRE portion of this tutorial).
12 //,'/

1100 FORMAT(' This concludes the FINGER portion of this tutorial.', '/,'/
1200 FORMAT(' Sorry to see you quit so soon. Goodbye.', '/,'/
1300 FORMAT(' Would you like to continue (Y or N)? ', ' S', /
1320 FORMAT(' Enter any key to continue: ', ' S', /
1340 FORMAT(' Sigh. I guess it''s just not your day.', '/,'/
1350 FORMAT(' Aaack!! Call the repairman! You''ve scrambled my innards
1360 FORMAT(' 1!!', '/,'/
1370 FORMAT(' Do you want another chance (Y or N)? ', ' S', /
1370 FORMAT(' Enter a ''Q'' to quit, any other key to continue: ', ' S', /
1380 FORMAT(' Enter any key to return to main menu: ', ' S', /
1400 FORMAT(' ', ' S', /
1410 FORMAT(' ', ' S', /
1415 FORMAT(' ', ' S', /
1420 FORMAT(' ', ' S', /
1600 FORMAT(' ', ' S', /
END
SUBROUTINE AINQR(DUM)

**************************INQUIR SUBROUTINE**************************

- THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE "INQUIR" PROGRAM -

**************************INQUIR SUBROUTINE**************************

DIMENSION INPUT(1)

**** INTRODUCTION TO INQUIR

TYPE 1600
TYPE 1000
TYPE 1030
READ(05,1400,ERR=100) INPUT
TYPE 1600
IF(INPUT(1).EQ.'Q'.OR.INPUT(1).EQ.'q')GO TO 520
100
TYPE 1010
TYPE 1015
READ(05,1400,ERR=120) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 520

c

**** EXAMPLE OF COMPLETED DATA BASE & SHOW COMMAND

120
TYPE 1020
TYPE 1030
READ(05,1400,ERR=130) INPUT
TYPE 1600
IF(INPUT(1).EQ.'G'.OR.INPUT(1).EQ.'g')GO TO 520
130
TYPE 1040

c

**** SIGN-OFF ROUTINE

500
TYPE 1240
TYPE 1370
READ(05,1400,ERR=510)
510
RETURN
520
TYPE 1250
TYPE 1370
READ(05,1400,ERR=530)
530
RETURN

**************************FORMATS FOR INQUIR SUBROUTINE**************************

**** INTRODUCTION TO INQUIR

1000 FORMAT(/,,) INQUIR,,/*,
1' INQUIR allows you to enter information into a file which can then be read'./
3' by yourself or others using the "FINGER /VERBOSE <username>" command. It'/
5' is entered by typing "INQUIR <CR>" after the "G" Executive Level
The system will respond with the ">>" prompt, following which you type the "MODIFY <username> <CR>", where user name is your account name. The system now will be at its entry level, and will either respond with the ">>" prompt (if date is already in the file), or with question: 4(if there's/5 is no information in the file). Please note that while you are answering these questions, or later making changes, the editing controls (DEL KEY, "R", "W etc.) may be used. The "Z" is used to complete the remarks section.

When you finish with the questions you will be placed at the ">>" prompt. The commands now available can be viewed by entering a "?", as usual. The more useful of these are:

- ALL: Allows you to review/change the entire data base.
- <field name>: Allows you to change a specific field.
- SHOW: Shows you the entire data base as it would appear with the "FINGER /VERBOSE" <username> command.
- EXIT: Exits to the ">>" prompt from the ">>" prompt, or exits to the "->" prompt from the "->" prompt.

Thus, once we are satisfied with the file, it takes two EXITs to get back to the Executive Level.

Would you like to see what a completed INQUIR file looks like (Y or N)? 

**** EXAMPLE OF INQUIR FILE / SHOW COMMAND

1 > MODIFY HERKERTMK
2 >> SHOW
3 Directory name: PS:<HERKERTMK>
4 Name: 
5 Nick name: BOOTS
6 Work address: C3 CURRICULUM, NPS MONTEREY, CA 93943-100
7 Work phone: (408) 646-2772
8 Home address: SMC 1001, NPS MONTEREY, CA 93943-1001
9 " Home phone: (408) 372-1665
1 Network address: HERKERTMK
2 Birthday: CLASSIFIED
3 Supervisor: GARY POOCK (NET: POOCK@ISTA)
4 Project: INTERACTIVE TUTORIAL ON THE DDN NETWORK.
5. Relation: Grad-Student
6. Remarks: God helps those which...
7. Account: 
8. Last logout: 
9. Last alteration: HERKERTMK: 25-Nov-85 00:31:42
1. >> EXIT
2. > EXIT

1030 FORMAT(' ENTER "O" TO QUIT, ANY OTHER KEY TO CONTINUE: ',#)

1040 FORMAT(' Notice in the above output, certain fields are blank: AC
1COUNT, NAME and ',
2 LAST LOGOUT. These fields are completed by the system when some
3 one FINGERS', 'the file with /VERBOSE.' ')

1240 FORMAT(' This completes the tutorial section on INQUIR.' ')
1250 FORMAT(' SORRY YOU HAVE TO RUSH OFF', ')
1400 FORMAT(' ENTER ANY KEY TO RETURN TO THE MAIN MENU: ',#)
1600 FORMAT(':///)

END
SUBROUTINE AEMH(DUM)

--------------- ELECTRONIC MAIL HOST SUBROUTINE ---------------------
* THIS SUBROUTINE EXPLAINS AND DEMONSTRATES THE ELECTRONIC        *
* MAIL HOST MESSAGE HANDLING SYSTEM.                              *
---------------------------------------------------------------------
DIMENSION INPUT(1)

**** INTRODUCTION TO EMH

TYPE 1600
TYPE 1000

**** QUERY TO SEE IF USER WOULD LIKE TO CONTINUE

TYPE 1300
READ(05,1400,ERR=10) INPUT
TYPE 1600
IF(INPUT(1).EQ.'N'.OR.INPUT(1).EQ.'n')GO TO 50

**** DEMONSTRATION OF USING EMH PROGRAM

10 NUM=0
30 TYPE 1010
35 TYPE 1320
READ(05,1400,ERR=35) INPUT
TYPE 1600
35 TYPE 1020
35 TYPE 1320
READ(05,1400,ERR=40) INPUT
40 TYPE 1030
40 TYPE 1320
READ(05,1400,ERR=45) INPUT

SIGN-OFF ROUTINE

45 TYPE 1100
45 TYPE 1380
READ(05,1400,ERR=48) INPUT
RETURN
50 TYPE 1200
60 TYPE 1380
READ(05,1400,ERR=70) INPUT
RETURN

------------- FORMAT FOR EMH SUBROUTINE -----------------------------

**** EXPLANATION OF EMH

1000 FORMAT(20X,'EMH',//,
1' ELECTRONIC MAIL HOST (EMH) is a special kind of host on the net.
The only capability an EMH has is sending and receiving messages. You will be using TN in this tutorial; if you’ve forgotten that particular system, you should stop here and go back for a review.

To use EMH, obviously you will have to know someone who is on that system, worse yet, you will have to know the correct password(s) to get into that. For example, Hawaii has an EMH. In this tutorial, we will login using "NPSCLASS" with a first password of FALLODOWN and a second password of SPRINGUP. EMH passwords change frequently to prevent user abuse and large, inexplicable bills. Unless you have a good reason to access an EMH, you will probably not be able to get in.

Here is a demonstration of accessing an EMH "Host" and a lending:

@TN HAWAII-EMH
Trying...
login: NPSCLASS
PASSWORD: <falldown>
BBNCC RELEASE 5.3.3. INSTALLED 4 SEP 85
HAWAII ELECTRONIC MAIL HOST
TERM = (TI) TTY
ERASE SET TO BACKSPACE
Welcome to INFONAIL -- VERSION 2.5.1 -- USED UNDER LICENSE FROM BBNCC
USER NAME: NPSCLASS
PASSWORD: <springup>
INBOX NOW OPENED (NB: This is messages you are in receipt of)

INBOX.
6 1 FROM: POOCK @ US /SUBJECT: THESIS QUESTIONS?? / 29 NOV 85 18:29 GMT

You have a number of options in EMH: DESK, INBOX, OUTBOX, FILES, FORMS, TRASH, PROFILE, CURRENT, PREVIOUS, #, X,
2 LAST, ALL,
Other options used to manipulate message formatting, etc are:
COMPOSE, COPY, CREATE, DESCRIBE, DISCARD, DISPLAY, EDIT, PEN,
EXAMPLE, EXIT, EXPORT, FILE, FORWARD, GET, IMPORT, KEYPAD, LOCKUP, MAIL,
MAKE, MOVE, NEXT, OPEN, PRINT, QUIT, READ, REMOVE, REPLY, RESTORE, SCAN,
SET, SHOW, SYSTEM, TERMINAL.
A goodly number of these are self-explanatory, but if you find one, it’s a bit confusing, a question mark typed after the option will elicit an explanation. Unfortunately, the ESC and CONTROL keys do not work well, if at all, in EMH. Other than reading your INBOX, you will probably find the COMPOSE option, use 3d to create a message, the most useful.

Since the control characters do not work in EMH, we cannot term...
"note the text"./
5' of a message with "Z. Instead, enter a PERIOD "," on a line by
itself followed="/.
5' by a carriage return."\n1100 FORMAT(' This concludes the ENH portion of this tutorial.'\n1200 FORMAT(' Sorry to see you quit so soon. Goodbye.'\n1300 FORMAT(' Would you like to continue (Y or N)? '\n1320 FORMAT(' Enter any key to continue: ',\n1380 FORMAT(' Enter any key to return to main menu: ',\n1400 FORMAT(1A1)\n1600 FORMAT('\\nEND
C
C
C
C
C

123
SUBROUTINE AENACS(DUM)
TYPE 100
RETURN
100 FORMAT(// "THE EXACS PART OF THIS TUTORIAL HAS YET TO BE WRITTEN.
1",//)
END
SUBROUTINE AGRAPH(DUM)
TYPE 100
RETURN
100 FORMAT(' THE GRAPH PORTION OF THIS TUTORIAL HAS YET TO BE WRITTEN.
1'//)
END
SUBROUTINE BILBO(DUM)

------------------------WHERE TO GO FROM HERE SUBROUTINE-----------------------
* THIS SUBROUTINE OUTPUTS ADDITIONAL SOURCES OF INFORMATION ON *
* THE DDN NETWORK AND THE TOPS-20 OPERATING SYSTEM *
------------------------

DIMENSION INPUT(1)
TYPE 1600
TYPE 1000
TYPE 1200
READ(05,1400,ERR=10) INPUT
10 TYPE 1005
TYPE 1200
READ(05,1400,ERR=20) INPUT
20 TYPE 1010
TYPE 1240
TYPE 1250
READ(05,1400,ERR=30) INPUT
30 RETURN

*****FORMATS FOR BIBLIOGRAPHY SUBROUTINE**********

1000 FORMAT(10X,'WHERE TO GO FROM HERE',//
1' The following materials serve as excellent sources of information
  on the',//
2' TOPS-20 operating systems and sub-systems, including HELP, ?, AC
  3TION',//
4' DOCUMENTATION, FTP, TN, FINGER, MK, HERMES, XED, EMACS, REMIND,
  SHOTO',//
6' WHOIS, ARCHIVE AND SCRIBE',//
7' USER'S GUIDE TO TOPS-20',/ TOPS-20',//
8' WAYNE TANNER CHLOE SOMMERS HOLL',//
9' SEPT 1983 & APRIL 1984 AUGUST 1983',//
* USC INFORMATION SCIENCE INSTITUTE USC INFORMATION SCIENCE
1 INSTITUTE',//
2' 4676 ADMIRALTY WAY IBID',//
3' MARINA DEL REY, CA 90291',//
4' (213) 822-1511 EXT 289',//
5' ACTION @ ISIE',//
6' PROFESSOR GARY POOCK',//
7' CODE 55PK',//
8' NAVAL POSTGRADUATE SCHOOL',//
9' MONTEREY, CA 93943',//
* POOCK@USC-ISIA',//)

1005 FORMAT(' The following are informative sources on the creation of
1 the Defense Data',//
2 Network, where it's at today and where it's going',//
3 DDN NEW USEP GUIDE (NIC 50001)',//
4 MARCH 1985',//,

126
5' DDN NETWORK INFORMATION CENTER',/
6' DDN PROGRAM MANAGEMENT OFFICE',/
7' DEFENSE COMMUNICATION AGENCY',/
8' WASHINGTON, D.C.',/
9' (703) 285-5025',/}
C
1010 FORMAT(' Useful addresses:',/,
  1' BOLT, BERANEK AND NEWMAN INC.
  2PORATION',/,
  3' 50 MOUTON STREET
  4TE 1101',/,
  5' CAMBRIDGE, MASSACHUSETTS 02238
  62209',/,
  7' DIGITAL EQUIPMENT CORPORATION
  8',/
  9' PO BOX CS-2008
  *ES INSTITUTE',/,
  1' NASHUA, NEW HAMPSHIRE 03061
  2SITE 1100',/,
  3'
  4RNIA 90291',/,
  5' MASSACHUSETTS INSTITUTE OF TECHNOLOGY',/,
  6' ARTIFICIAL INTELLIGENCE LABORATORY
  7' ATTN: PUBLICATIONS
  8' 545 TECHNOLOGY SQUARE
  9A 15213',/,
  9' CAMBRIDGE, MASSACHUSETTS 02139',/)
1200 FORMAT(' ENTER ANY KEY TO CONTINUE: ',$)
1240 FORMAT(' This concludes the bibliography section.',/)
1250 FORMAT(' TYPE ANY KEY TO RETURN TO THE MAIN MENU: ',$)
1400 FORMAT(1A1)
1600 FORMAT(///)
END
This Flowchart Describes The Main Program Of The Interactive Tutorial For The DDN.

- INTRODUCTION
- MENU OF INSTRUCTIONAL SUBROUTINES
- CASE
  - INTRODUCTION TO HELP, ACTION, ?
  - BASIC EXECUTIVE LEVEL COMMANDS
  - ADVANCED EXECUTIVE LEVEL COMMANDS
    - MSG
    - XED
    - FTP
    - TN
    - PHOTO
    - REMIND
    - INQUIRE
    - FINGER
    - EMH
    - EMACS*
    - GRAPH*
    - WHERE TO GO FROM HERE
  - QUIT

* DENOTES EMPTY SUBROUTINE
This Flowchart Describes The ELC1 Subroutine:

FROM MAIN PROGRAM

INTRODUCTION

BASIC CURSOR COMMANDS

DIRECTORY RELATED COMMANDS

SUBROUTINES TO DEMONSTRATE DIRECTORY RELATED COMMANDS
  DIRECTORY TYPE
  APPEND
  RENAME
  ARCHIVE
  DELETE
  UNDELETE
  EXPUNGE

SYSTEM RELATED COMMANDS

CLOSING INFORMATION

RETURN

TO MAIN PROGRAM
This Flowchart Describes The ELC2 Subroutine:

FROM MAIN PROGRAM

INTRODUCTION

ADVANCED CURSOR COMMANDS

DIRECTORY RELATED COMMANDS

SUBROUTINES TO DEMONSTRATE DIRECTORY RELATED COMMANDS

TALK

SYSTEM RELATED COMMANDS

CLOSING INFORMATION

RETURN

TO MAIN PROGRAM
This Flowchart Is Typical For The Following Subroutines:

INTRODUCTION
AMSG (MSG)
AXED (XED)
ATN (TN)
AFTP (FTP)
APHOTO (PHOTO)
ARMIND (REMIND)
AINQRI (INQUIRE)
AFNGR (FINGER)
AEMH (EMH)

FROM INTRODUCTION MAIN PROGRAM

INTRODUCTION
DEFINITIONS
COMMANDS & USES
DEMO

NO
RETURN

YES
SPECIFIC COMMAND
REQUEST USER INPUT
USER INPUT

CORRECT RESPONSE

YES
MORE TO DEMO

NO

ERROR MESSAGE

ERROR MESSAGE

FIRST TRY

SECOND TRY

PROPER RESPONSE

TRY AGAIN

RESET COUNTER FOR NUMBER OF TRIES (NUM)
This Flowchart is Typical For The Following Subroutines:

- DIRECT
- TYPE
- APPEND
- RENAME
- ARCHIVE
- UNDELETE
- EXPUNGE
- TALK

FROM SUBROUTINE ELC1 & 2

COMMAND & ITS USE

REQUEST FOR USER INPUT

READ USER INPUT

CORRECT RESPONSE

YES

RETURN

NO

TO ELC1 OR ELC2

RESET TRIES COUNTER

ERROR MESSAGE #2

YES

ERROR MESSAGE #1

FIRST TRY

NO

YES

SECOND TRY

NO

PROPER RESPONSE

NO

TRY AGAIN

YES
This Flowchart Describes The Where To Go From Here Or BILBO Subroutine:

FROM MAIN PROGRAM

INTRODUCTION

DEFINITIONS

COMMANDS/USES

HOW TO GET TO <BILBO>

DOCUMENTS

PERSONNEL

INSTITUTES/COMPANIES

RETURN
<table>
<thead>
<tr>
<th>No.</th>
<th>Name and Details</th>
</tr>
</thead>
</table>
| 1.  | Defense Technical Information Center  
      Cameron Station  
      Alexandria, Virginia 22304-6145 |
| 2.  | Library, Code 0142  
      Naval Postgraduate School  
      Monterey, California 93943-5100 |
| 3.  | Professor M. G. Sovereign, Code 74  
      Naval Postgraduate School  
      Monterey, California 93943-5100 |
| 4.  | Joint C3 Curricular Officer, Code 39  
      Naval Postgraduate School  
      Monterey, California 93943-5100 |
| 5.  | Professor Gary K. Poock, Code 55PK  
      Naval Postgraduate School  
      Monterey, California 93943-5100 |
      Naval Postgraduate School  
      Monterey, California 93943-5100 |
| 7.  | LCDR Mark K. H. Herkert, USN  
      c/o Commanding Officer  
      FAIRECONRON THREE  
      FPO San Francisco, California 96601-6517 |
| 8.  | LT Sheri L. Smith, USN  
      94 Montaelas Dr.  
      Monterey, California 93943 |
| 9.  | Chloe Sommers Holg  
      University of Southern California  
      Information Sciences Institute  
      4676 Admiralty Way  
      Marina del Rey, California 90292 |
| 10. | Joel Goldberger  
      University of Southern California  
      Information Sciences Institute  
      4676 Admiralty Way  
      Marina del Rey, California 90292 |
11. Sam Delatorre
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292
END

FILMED

3-86

DTIC