AD-A134 118   NAVY ELECTRONIC MAIL SERVICE USER'S GUIDE(U) DAVID W   1/1
                TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER BET..
                A V CINCOTTA AUG 83 DTNSRDC/CMLD-83-22

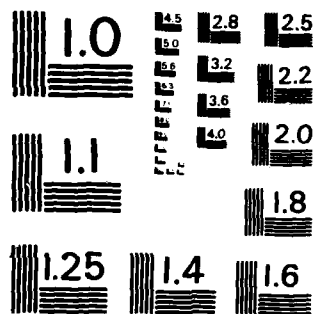UNCLASSIFIED   F/G 9/2   NL

END
DATE
FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART

CMLD-83-22

DAVID W. TAYLOR NAVAL SHIP
RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

NAVY ELECTRONIC MAIL SERVICE
USER'S GUIDE

By

Anthony V. Cincotta

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

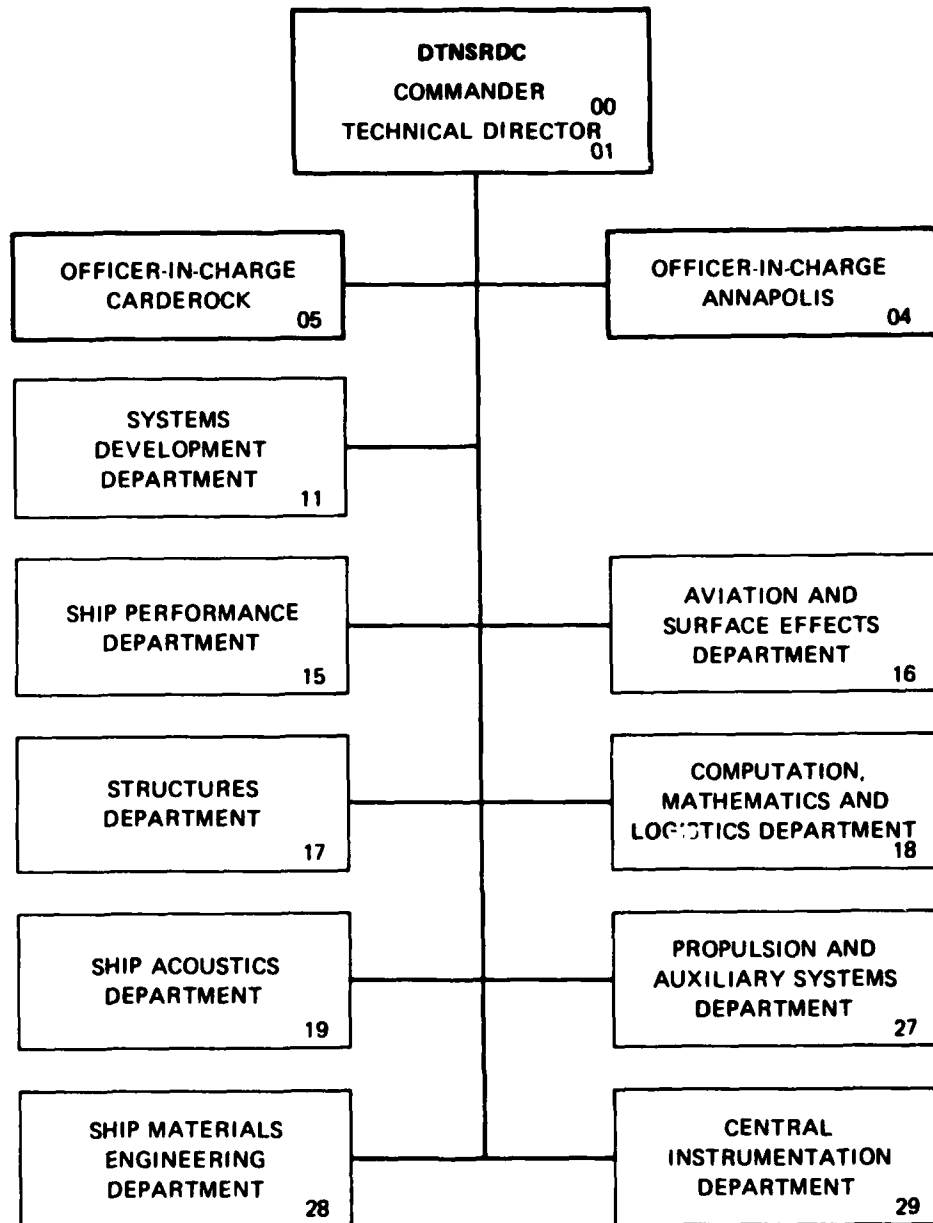Computation, Mathematics and Logistics Department
Departmental Report

August 1983

CMLD-83-22

83 10 93 026

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS

```
                        ┌─────────────────────┐
                        │       DTNSRDC        │
                        │   COMMANDER      00  │
                        │ TECHNICAL DIRECTOR   │
                        │                  01  │
                        └─────────────────────┘
                                   │
        ┌─────────────────────┐    │    ┌─────────────────────┐
        │  OFFICER-IN-CHARGE   │    │    │  OFFICER-IN-CHARGE   │
        │    CARDEROCK         │────┼────│    ANNAPOLIS         │
        │              05      │    │    │              04      │
        └─────────────────────┘    │    └─────────────────────┘
                                   │
        ┌─────────────────────┐    │
        │     SYSTEMS          │    │
        │   DEVELOPMENT        │────┤
        │   DEPARTMENT         │    │
        │              11      │    │
        └─────────────────────┘    │
                                   │
        ┌─────────────────────┐    │    ┌─────────────────────┐
        │  SHIP PERFORMANCE    │    │    │    AVIATION AND      │
        │   DEPARTMENT         │────┼────│  SURFACE EFFECTS     │
        │              15      │    │    │   DEPARTMENT     16  │
        └─────────────────────┘    │    └─────────────────────┘
                                   │
        ┌─────────────────────┐    │    ┌─────────────────────┐
        │    STRUCTURES        │    │    │   COMPUTATION,       │
        │   DEPARTMENT         │────┼────│  MATHEMATICS AND     │
        │              17      │    │    │ LOGISTICS DEPARTMENT │
        │                      │    │    │                  18  │
        └─────────────────────┘    │    └─────────────────────┘
                                   │
        ┌─────────────────────┐    │    ┌─────────────────────┐
        │   SHIP ACOUSTICS     │    │    │   PROPULSION AND     │
        │   DEPARTMENT         │────┼────│  AUXILIARY SYSTEMS   │
        │              19      │    │    │   DEPARTMENT     27  │
        └─────────────────────┘    │    └─────────────────────┘
                                   │
        ┌─────────────────────┐    │    ┌─────────────────────┐
        │  SHIP MATERIALS      │    │    │    CENTRAL           │
        │  ENGINEERING         │────┴────│  INSTRUMENTATION     │
        │  DEPARTMENT      28  │         │   DEPARTMENT     29  │
        └─────────────────────┘         └─────────────────────┘
```

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
CMLD-83-22	A134 118	

4. TITLE *(and Subtitle)*	5. TYPE OF REPORT & PERIOD COVERED
Navy Electronic Mail Service User's Guide	
	6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
Anthony V. Cincotta	

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
David Taylor Naval Ship R&D Center Systems Software, Code 1892, Bethesda, MD 20084	

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
Computation, Mathematics & Logistics Dept. Computer Facilities Division (189)	August 1983
	13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*
	Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release: Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report provides the basic information necessary for a user to obtain familiarity with the Navy Electronic Mail Service (NEMS). Specifically, this report provides information on obtaining a NEMS subscription, guidance on acquiring access to the NEMS computer resources, and detailed procedures for sending and receiving mail service.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73
S/N 0102-LF-014-6601

UNCLASSIFIED

Table of Contents

---

* UNIX is a Trademark of Bell Telephone Laboratories, Inc.

*** Abstract ***

This report provides the basic information necessary for a user to obtain familiarity with the NAVY Electronic Mail Service (NEMS). Specifically, this report provides information on obtaining a NEMS subscription, guidance on acquiring access to the NEMS computer resources, and detailed procedures for sending and receiving mail service.

*** Introduction ***

The NAVY Electronic Mail Service (NEMS) is an unclassified system which, for a prepaid annual subscription fee, will provide to the NAVY Department and their support personnel an electronic mail service. This service consists of the following:

    . Defense Data Network (DDN) access,
    . Mail facility,
    . DDN mail box,
    . File storage (50KB),
    . Utilities for generating and managing your mail,
    . User support services,
    . System support services,
    . File integrity.

The NEMS facility provides the following communication facilities:

    . DDN access,
    . Access to Internet networks available via DDN,
    . Access to NEMS via GTE Telenet (Value Added Network)
      service,

NEMS is located at the David W. Taylor Naval Ship Research and Development Center (DTNSRDC), and is managed by the Computer Facilities Division of the Computation, Mathematics & Logistics Dept.

*** NEMS User Policy ***

All NEMS subscribers must pay an annual charge per user subscription. The DTNSRDC Computer Center Business Office at (202) 227-1361 or Auto-von (8) 287-1361 will administer the addition of new users to NEMS. Because of Government regulations, no users will be allowed access to NEMS unless the user is a NAVY Department employee or support person (as verified by NAVY sponsor) and receipt of funds has been acknowledged by the DTNSRDC Disbursing office (Code 60).

Access to NEMS via GTE Telenet service is available. The yearly subscriber fee will cover nominal use of the service (up to three calls). High volume or extensive use of the GTE network will be charged directly to the user's individual account. Therefore, such users must have additional funds at DTNSRDC to cover these additional costs. Such funds not used will be refunded. GTE Telenet charges not paid will cause NEMS services for those users to be terminated with possible loss of the subscription fee.

Existing delinquent accounts will be removed ten days after start of the new charging period. User data associated with the delinquent account will be dumped to magnetic tape in a standard UNIX system tape format and held for thiry days. Delinquent account users may request data tapes be mailed to them. If after thirty days the user has not requested his/her data the data tape will be erased.

The operating system of NEMS is "4.1bsd". "4.1bsd" is a UNIX based system (see APPENDIX A) available from Berkeley University. Since the purpose of NEMS is to provide an electronic mail service, only a small subset of the "4.1bsd" commands are available.

*** DDN Access ***

NEMS is on the Military Network (MILNET) at address "26.0.0.81". MILNET is a subnet of DDN. NEMS users who access NEMS via the Terminal Access Controller (TAC) may contact the NEMS HOST Administrator [(202) 227-1361] to obtain DDN TAC registration. DDN requires all TAC users to be registered and will in December 83 require "login" to the TAC.

NEMS subscribers may contact the Network Information Center, SRI International, Menlo Park, California 94025, (415) 859-3695, for information on the availability of a DDN TAC in their locality and information on how to obtain authorization to use that DDN TAC.

---

* UNIX is a Trademark of Bell Telephone Laboratories, Inc.

*** NEMS Access via DTNSRDC TAC ***

Dial the DTNSRDC TAC and verify that a carrier signal is present on the line. Each port on the DTNSRDC TAC is ready to recognize the user terminal's speed from the first character typed to it. To detect User terminal baud rate (default baud rate is 300), the DTNSRDC TAC requires a single character, specifically a ^Q ("^" designates CONTROL key, "_" indicates a user typein). The initial ^Q will set the DTNSRDC TAC's input and output rate to the rate of the users terminal. A CR (CR designates Carriage-Return key) can be used instead of ^Q if 300 baud is desired. A typical TAC access session is as follows:

```
      User response              DTNSRDC TAC response

      Dial DTNSRDC TAC           (Carrier Detected)
      ^Q                         Defense Data Network - ...
                                 DAVID TAC 107 #:02
                                     where:
                                          107 is the TAC Version
                                          #:02 is the Port on the IMP

      @f o s                     (None)
                                 Enable flow control (XON/XOFF) from
                                 TAC to terminal.

      @Open 0/81                 TCP Trying...Open
                                 NAVY Electronic Mail Service (nems)
                                 (202) 227-1907 For User Support

                                 Login:

                                        or
                                 TCP Trying...Host down
```

*** GTE Telenet Access ***

Access to NEMS is also available thru the GTE Telenet public network. This network makes it possible for users to dial up a local GTE Telenet telephone number anywhere in the country to establish a connection to the GTE Telenet HOST located at DTNSRDC which will provide access to NEMS.

*** NEMS Access via GTE Telenet ***

A typical Telenet access session is as follows:

User response	Telenet response
Dial Local Telenet (See "man telenet" to ob- tain local phone number)	(Carrier Detected)
CR	
CR	Telenet Port-Number Address-Number
	TERMINAL=
CR	@
c 301160.01	301 160X Connected (Where: X = D or C or B or A depending on port obtained)
	NAVY Electronic Mail Service (nems) (202) 227-1907 For User Support
	Login:

While logging into NEMS via GTE Telenet you will notice that your "password" is echoed as you type it. The Telenet interface because of efficiency and cost considerations is setup to echo all characters typed while the NEMS system will supress echo on Telenet input lines.

GTE Telenet will transmit data to NEMS when a CR is typed or data has been buffered up and the user has been idle for four/tenths of a second. The actual time to transmit this data, coupled with the time required for NEMS to respond to the required action and then for GTE to transmit the output back to the user, can require one or more seconds. Because of these inherent timing constraints and normal user activity, the output from some of the NEMS packages may appear in error while in-reality the fault is the impatience of the user.

*** Logging-in ***

At this stage of the login procedure, you require a valid user account on NEMS. In reponse to NEMS "Login:" (quoted text is used to indicate NEMS responses) prompt, type in your user name followed by a CR. It is important that you type your name in lower case if possible; if you type upper-case letters the operating system will assume that your

- 4 -

terminal cannot generate lower-case letters and will translate all subsequent lower-case input to upper-case. The next NEMS prompt will be:

Password:

Type in your password. After both login name and password are validated, you will receive a salutation greeting followed by a "1-> " prompt. At this point a program called the Shell is listening to your terminal. The Shell reads typed-in lines, splits them up into a command name and arguments, and executes the command. A command is simply an executable program.

To protect your files from unauthorized access by others we require that you provide your own password during your first session on NEMS. If you fail to change your initially provided password within thirty (30) days, your subscription to NEMS will be invalidated.

The following example illustrates the user/machine correspondence required for changing ones password:

```
1-> passwd
Old password: _____CR
New password: _____CR
Retype new password: _____CR
Password changed
2->
```

NEMS will not echo your password as you type it. Your new password should contain six to ten alphanumeric characters. NEMS will complain if less than six characters are submitted.

*** Logging-out ***

If you entered NEMS from the DTNSRDC TAC, you may disconnect from NEMS by:

    .hanging up the phone,
    .executing the logout command,
    .executing the DTNSRDC TAC "close" command @c.

If you entered NEMS from Telenet, you can disconnect by executing logout. The logout command will terminate the NEMS session and provide the proper response to Telenet which will disconnect you from NEMS. Or, you may tell Telenet you want to disconnect by typing @CR. Telenet will respond with "@", you then type dCR.

To disconnect from Telenet you must hang up the phone. Charges for GTE Telenet are based on message units and time accumulated. Failure to close the Telenet connection will result in unnecessary charges.

*** Terminal Communication ***

The NEMS system can be tailored to most terminals available in the market-place. The default terminal of NEMS is the Digital Equipment Corporation (DEC) "VT100". If you are operating with a terminal other than a VT100 then you should inform the system as soon as possible what terminal you will be using. This is performed by the following command:

        setenv TERM terminal-type

Consult User Services for obtaining the correct "terminal-type" designation for the terminal you are using.

The NEMS system provides full-duplex terminal input/output. It has read-ahead, which means that you can type at any time. If you type during output, the output will have the input characters interspersed. However, whatever you type will be saved and interpreted in correct sequence. There is a limit to the amount of read-ahead, but it is generous and not likely to be exceeded unless the system is heavily loaded. When the read-ahead limit is exceeded, the system throws away all the saved characters.

*** Typing in UNIX Commands ***

If you make a typing mistake, and see it before CR has been typed, there are ways to recover. The backspace-character, denoted as ^H ("^" is denoted throughout this text as the "Control" key), erases the last character typed; in fact successive uses of the backspace-character will erase characters back to the beginning of the line (but not beyond). So if you type badly, you can correct as you go:

        dd^Hatte^H^He

is the same as date.

The ^U erases all of the characters typed so far on the current input line. If the line is irretrievably fouled up, type a ^U and start the line over.

The ^W erases the current word.

The ^R retypes the current line.

What if you must enter a ^H as part of the text? If you precede the ^H by a ^V (literal next character), it loses its special meaning. So to enter a ^H type ^V^H.

To erase a literal ^H character, you have to type ^H^H.

The ^V is used extensively to indicate that the following character is to be treated as a literal character.

The ^C character is not passed to programs but instead generates an interrupt. This signal generally causes whatever you are running to terminate. ^C can be used to stop a long printout.

The syntax !n, ("n" is the integer value contained in a previous "prompt" syntax) may be used to re-execute the command designated as "n-> ".

The syntax !! will re-execute the previous command.

Users obtaining NEMS access via the DTNSRDC TAC should keep in mind that all characters typed are passed to NEMS by the TAC. The TAC examines each character looking for a TAC command. The TAC commands start with an @ character. When an @ is typed, the TAC will echo back the @ and its associated command text. Typing a second @ , will instruct the TAC to pass the "@" and terminate the TAC command initiated by the first @. NEMS will echo back to the user the "@" received. Therefore; to get a single @ to NEMS , you type in two consecutive @ and see three "@".

*** Mail ***

A ,ostal system is available so you can send and receive mail from other users on NEMS or on DDN. The mail system is composed of two UNIX tools, "msg" reads the mail and "snd" sends the mail. These mail routines may be tailored to meet your hardware and specific desires. You are encouraged to read the complete up-to-date documentation for mail by typing:

        man msg snd

When you login, you may sometimes get the message:

        You have mail.

If, while logged-in, mail arrives for you, you will be notified on completion of your next commmand. Thereafter, you will be notified every ten (10) minutes that unseen mail has been posted until you read your mail. Each user on NEMS is provided a mailbox. The mailbox is actually a file named ".mail", residing in your login directory.

*** Reading mail ***

"msg" when executed will inform you of the total number of messages
you have in your mailbox, followed by the "msg" prompt as shown;

          nnn messages
          msg->

The "msg" utility has many commands to allow you to manage your mail-
box. A complete list of these commands is available by typing a ?. A
list of "msg" commands follow:

          List commands:
          b[ack up] .... List previous message.
          c[urrent message] ..... list current message number.
          h[eaders] <msg-seq> .. List one-line header for each message.
                                  "headers *" will list unread messages
                                  followed by all remaining messages.
          n[ext message] .... List next message on terminal.
                                  "msg" will starts with the first unread
                                  message.
          P[rint]   <msg-seq> .. Print messages on terminal's printer.
          H[eaders] <msg-seq> .. Print headers on terminal's printer.
          t[ype]   <msg-seq> .. List messages specified on terminal.

          Delete commands:
          d[elete]   <msg-seq> .. Mark messages for deletion (the "over-
                                  write" command is required for actual
                                  deletion)
          e[xit] .... Perform an "overwrite" then a "quit".
          o[verwrite] .... Remove deleted messages.
          u[ndelete] <msg-seq> .. Remove delete mark from messages.

          Save commands:
          l[ist]   <msg-seq> .. Copy messages for later printing.
          m[ove]   <msg-seq> .. Same as put plus delete.
          p[ut]   <msg-seq> .. Copy messages retaining message format.
          w[rite to] .... Sets default filename for l,p,m commands.

          Send commands (temporarily invoke "snd"):
          a[nswer] .... Answer "to" recipients only.
          A[nswer] .... Answer "to" and "cc" recipients.
          f[orward] .... Send copy of message to another user.

          Other commands:
          r[ead] .... Read from new message file.
          !command .... Pass command to "shell" for execution.

          Exit "msg" command:
          e[xit] .... Perform "overwrite" then a "quit".
          q[uit] .... Exit from "msg".

- 8 -

```
Examples of message sequences:
2,3,6                          messages 2, 3, and 6
11-15                          messages 11 through 15 (inclusive)
15-11                          messages 15 through 11 (inclusive)
3,3,3                          message 3 three times
15-$                           message 15 through the last message
*                              all messages
```

Any of the above commands are enacted by entering the initial charac-
ter of the command in response to the "msg" prompt (msg->). For ex-
ample, to read your next message enter n. This will result in your
next unread message being printed. To continue reading unread mes-
sages, continue entering n.

To quit "msg" you type q. The "shell" again takes command and pro-
vides you with the "shell" prompt "2-> ".

*** Recipient's address ***

Mail is posted by the "snd" utility. To post mail the sender must know
the recipients address. The address is a combination of the
recipient's login name and the HOST that contains the mailbox. When
the recipient and sender are on the same HOST, only the recipient's
login name is required. NEMS provides the utility "whois" for obtain-
ing a Directory for all NEMS subscribers.

A utility ("nicname") is available to provide netwide directory ser-
vice for NEMS users. This utility will access a server at the Network
Information Center at SRI International. SRI International maintains
an up-to-date directory service for DDN users. Documentation for
"nicname" is available by typing "man nicname". Executing:

        nicname strickland

returns:

Connection made to SRI-NIC ...

Strickland, Jeanne D. (JDS3)   uservice@NEMS
    David Taylor Naval Ship Research and Development Center
    Code 1892.1
    Bethesda, Maryland 20084
    Phone: (202) 227-1907 (AV) 287-1907

*** Sending mail ***

"snd" will mail a single message to each named recipient. "snd" will prompt for the "To:" and "cc:" (carbon copy) recipients address lists. After the "To" and "cc" lists have been entered, "snd" will ask for a subject. If it is required to have more than one line of text for the subject or either of the recipient lists, a CR must be preceded by a \.

After you have entered the subject and recipient lists, you will be prompted for the text of your message. While Entering your message, the following UNIX default characters are recognized:

```
            ^    Control key
           ^D    End of message
           ^H    Backspace
           ^U    Delete current line
           ^C    Abort "snd"
```

After you have finished entering and editing your message, type a ^D on a new line to signify "end-of-message". "snd" will validate the distribution lists and respond with the "snd" prompt "snd-> ". At this point, the user may execute any of the "snd" commands. A complete list of these commands is available by typing a ?. A partial list of "snd" commands follows:

```
    c[c]         ....... Modify "cc" field.
    d[isplay]    ....... Display text inputted.
    e[dit]       ....... Edit message with preselected editor.
    E[ditor]     ....... Selector editor to use.
    f[ile]       ....... Append specified file to end of text.
    g[roup]      ....... List "group" distributions
    i[nput]      ....... Continue to add more text.
    n[roff]      ....... Prepare text for printing.
    r[eceipt]    ....... Receive notification when message read by "to"
                         list (local NEMS user only).
    s[end]       ....... Send mail to all designated recipients.
    S[ubject]    ....... Modify "Subject" field.
    t[o]         ....... Modify "to" field.
    u[ser]       ....... User validation
    !command     ....... Pass command to the "shell" for execution.
    ?            ....... Provide help on using "snd".
```

The "snd" prompt will be displayed on completion of each command, until the "send" command is selected. The "s[end]" command will try to mail the message to each of the recipients. The name of each recipient will be printed out with the results of the attempt to post the message. "snd" will then terminate. The "shell" again takes command and provides you with the "shell" prompt.

*** Mail management ***

Managing ones mail is important. If the number of messages you re-
ceive daily encompases many projects and various activities you would
want to organize your mail into various files possibly even various
directories. The "msg" utility provides the options to easily set up
new mail files via the "move" and "put" options. The UNIX system pro-
vides the utilities to manage your mail. The reader is encouraged to
read Brian W. Kernighan's article "UNIX for Beginners". A selection
of material from this article has been included in this report as
APPENDIX A.

*** USER support ***

The NEMS system provides the following utilities for USER support:

  man ...... A utility to provide on-line documentation
             for all utilities available to users on NEMS.
             The command:

                  man man

             will provide documentation for the "man" utility. The
             command:

                  man utility-name

             will provide documentation for the utility of name
             "utility-name".

  unix .... A tool whereby the user can obtain a complete index
             of all tools available on NEMS and documentation on
             all utilities available.

  gripe .... A mechanism whereby the user may report problems and
             suggest enhancements for NEMS to those responsible
             for system integrity and enhancements. The user will
             get an acknowledgement when the "gripe" is read and
             will get results desired or a (hopefully) satisfac-
             tory response as soon as possible.

  whatsnew .. A utility which when executed will provide to the
             user the file names of all routines that have been
             enhanced or added to the NEMS system.

NEMS USERS GUIDE .. This manual (of approximately 30 pages) is available on-line. To have a copy of this manual displayed at your terminal execute the following:

unix NEMS

NEMS subscribers may contact the DTNSRDC User Services Group for answers to questions that require immediate response. This service is available on (202) 227-1907, or Autovon 8-287-1907. Requests for service may also be directed to the user "uservice@nems".

APPENDIX A - Extracts from "UNIX for Beginners"

Author:
Brian W. Kernighan

The Shell

When you have successfully logged in to UNIX a program called the
Shell is listening to your terminal. The Shell reads typed-in lines,
splits them up into a command name and arguments, and executes the
command. A command is simply an executable program. Normally, the
Shell looks first in your current directory for a program with the
given name, and if none is there, then in system directories. There
is nothing special about system-provided commands except that they are
kept in directories where the Shell can find them. The command name
is always the first word on an input line to the Shell; it and its ar-
guments are separated from ane another by space or tab characters.

When a program terminates, the Shell will ordinarily regain control
and type a "1-> " at you to indicate that it is ready for another com-
mand.

Executing Commands

If you make a mistake typing the command name, and refer to a non-
existent command, you will be told. For example, if you type

      whom

you will be told

      whom: not found

Of course, if you inadvertently type the name of some other command,
it will run, with more or less mysterious results.

Strange Terminal Behavior

Sometimes you can get into a state where your terminal acts strange-
ly. For example, each letter may be typed twice, or the CR may not
cause a line feed or a return to the left margin. You can often fix
this by logging out and logging back in. Or you can read the descrip-
tion of the command stty (man stty).

----
* UNIX is a Trademark of Bell Telephone Laboratories, Inc.

On-Line Manual

The UNIX Users Manual is available on-line. If you get stuck on some-
thing, and can't find assistance, you can print on your terminal any
manual page that might help. This is also useful for getting the most
up-to-date information on a command. To print a manual section, type
"man command-name". Thus to read up on the "ls" command, type

       man ls

and, of course,

       man man

tells all about the man command.

Creating Files - The Editors

If you have to type a paper or a letter or a program, how do you get
the information stored in the machine? Most of these tasks are done
with an editor. This system supports a line editor "ed" [1] and the
Berkeley screen editor "vi [2]" (Vee Eye). All we want it for right
now is to make some files. A file is just a collection of informa-
tion stored in the machine.

To create a file called junk with some text in it using "ed", do the
following:

       ed junk   (invokes the text editor)
       a   (command to "ed", to add text)
       now type text you want ...
       ...
       .   (signals the end of adding text)

The . that signals the end of adding text must be at the beginning of
a line by itself. Don't forget it, for until it is typed, no other
"ed" commands will be recognized. Everything you type will be treated
as text to be added. Finally, you must write the information you have
typed into a file with the editor command w:

       w

"ed" will respond with the number of characters it wrote into the file
junk.

Until the w command, nothing is stored permanently, so if disconnected
the information is lost. But after w if you disconnect while editing,
the information prior to the last w can be accessed at any time by
typing

       ed junk

Type a q command to quit the editor. (If you try to quit without writing, "ed" will print a "?" to remind you. A second q gets you out regardless.)

Now create a second file called "temp" in the same manner. You should now have two files "junk" and "temp".

What files are out there?

The ls (for list) command lists the names (not contents) of any of the files that UNIX knows about. If you type

        ls

the response will be

        junk
        temp

which are indeed the two files just created. The names are sorted into alphabetical order automatically, but other variations are possible. For example, the command

        ls -t

causes the files to be listed in the order in which they were last changed most recent first. The -l option gives a long listing:

        ls -l

will produce something like

        -rw-rw-rw- 1 bwk   41 Jul 22 2:56 junk
        -rw-rw-rw- 1 bwk   78 Jul 22 2:57 temp

The date and time are of the last change to the file. The 41 and 78 are the number of characters (which should agree with the numbers you got from ed). bwk is the owner of the file, that is, the person who created it. The -rw-rw-rw- tells who has permission to read and write the file, in this case everyone.

The use of optional arguments that begin with a minus sign, like -t and -lt, is a common convention for UNIX programs. In general, if a program accepts such optional arguments, they precede any filename arguments. It is also vital that you separate the various arguments with spaces, "ls-l" is not the same as "ls -l".

Printing Files

Now that you've got a file of text, how do you print it? There are a host of programs that do that, probably more than are needed. One simple thing is to use the editor, since printing is often done just before making changes anyway. You can say

```
ed junk
1,$p
```

ed will reply with the count of the characters in junk and then print all the lines in the file. After you learn how to use the editor, you can be selective about the parts you print.

There are times when it's not feasible to use the editor for printing. For example, there is a limit on how big a file ed can handle (several thousand lines). Secondly, it will only print one file at a time, and sometimes you want to print several, one after another. So here are a couple of alternatives.

First is "cat", the simplest of all the printing programs. cat simply prints on the terminal the contents of all the files named in a list. Thus

```
cat junk
```

prints one file, and

```
cat junk temp
```

prints two. The files are simply concatenated (hence the name "cat") onto the terminal.

"pr" produces formatted printouts of files. As with "cat", "pr" prints all the files named in a list. The difference is that it produces headings with date, time, page number and file name at the top of each page, and extra lines to skip over the fold in the paper. Thus,

```
pr junk temp
```

will print junk neatly, then skip to the top of a new page and print temp neatly. "pr" can also produce multi-column output:

```
pr -3 junk
```

prints junk in 3-column format. You can use any reasonable number in place of "3" and pr will do its best. "pr" has other capabilities as well (see "man pr").

Shuffling Files About

Now that you have some files in the file system and some experience in printing them, you can try bigger things. For example, you can move a file from one place to another (which amounts to giving it a new name), like this:

    mv junk precious

This means that what used to be "junk" is now "precious". If you do an ls command now, you will get

    precious
    temp

Beware that if you move a file to another one that already exists, the already existing contents are lost forever.

If you want to make a copy of a file (that is, to have two versions of something), you can use the "cp" command:

    cp precious templ

makes a duplicate copy of "precious" in "templ".

Finally, when you get tired of creating and moving files, there is a command to remove files from the file system, called "rm".

    rm temp templ

will remove both of the files named.

You will get a warning message if one of the named files wasn't there, but otherwise "rm", like most UNIX commands, does its work silently. There is no prompting or chatter, and error messages are occasionally curt. This terseness is sometimes disconcerting to newcomers, but experienced users find it desirable.

What's in a Filename

So far we have used filenames without ever saying what's a legal name, so it's time for a couple of rules. First, filenames are limited to 14 characters, which is enough to be descriptive. Second, although you can use almost any character in a filename, common sense says you should stick to ones that are visible, and that you should probably avoid characters that might be used with other meanings. We have already seen, for example, that in the "ls" command, ls -t means to list in time order. So if you had a file whose name was "-t", you would have a tough time listing it by name. Besides the minus sign, there are other characters which have special meaning. To avoid pitfalls, you would do well to use only letters, numbers and the period until you're familiar with the situation.

When you first made that file called junk, how did the system know that there wasn't another junk somewhere else, especially since the person in the next office is also reading this tutorial? The answer is that generally each user has a private directory, which contains only the files that belong to him. When you login, you are in your directory. Unless you take special action, when you create a new file, it is made in the directory that you are currently in; this is most often your own directory, and thus the file is unrelated to any other file of the same name that might exist in someone else's directory.

The set of all files is organized into a (usually big) tree, with your files located several branches into the tree. It is possible for you to "walk" around this tree, and to find any file in the system, by starting at the root of the tree and walking along the proper set of branches. Conversely, you can start where you are and walk toward the root.

Let's try the latter first. The basic tools is the command "pwd" (print working directory), which prints the name of the directory you are currently in.

Although the details will vary according to the system you are on, if you give the command pwd, it will print something like

    /usr01/your-organization/your-name

This says that you are currently in the directory your-name, which is in turn in the directory "your-organization", which is in the directory "/usr01", which is in turn in the root directory called by convention just "/".

If you now type

    ls /usr01/your-organization/your-name

you should get exactly the same list of file names as you get from a plain ls with no arguments, "ls" lists the contents of the current directory; given the name of a directory, it lists the contents of that directory. When the NEMS system administrator gave you a user name, the administrator also created a directory for you with the same name as your user name. When you login, any file name you type is by default assumed to be in this directory. Since you are the owner of this directory, you have full permissions to read, write, alter, or destroy its contents. Permissions to have your will with other directories and files will have been granted or denied to you by their respective owners.

  Next, try

    ls /usr01/your-organization

This should print a long series of names, among which is your own login name.

Next, try

    ls /usr01

This should list a series of organizations within the NAVY Department among which is your own organization.

The next step is to try

    ls /

You should get a response something like this (although again the details may be different):

    bin
    dev
    etc
    lib
    tmp
    usr
    usr01

This is a collection of the basic directories of files that the system knows about. We are at the root of the tree.
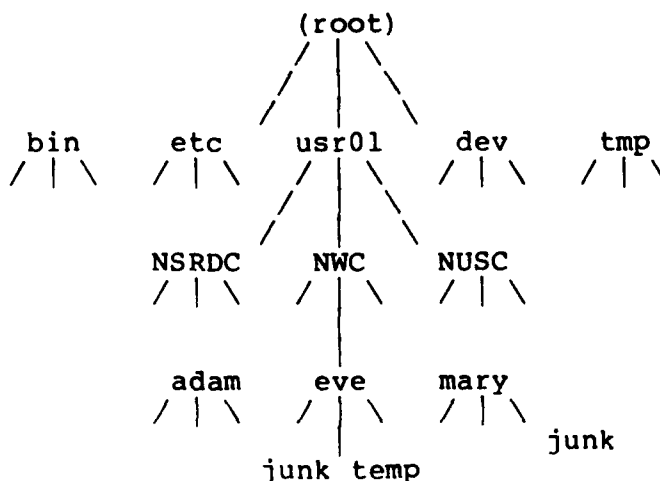
Now try

    cat /usr01/your-organization/your-name/junk

(if junk is still around in your directory). The name

    /usr01/your-organization/your-name/junk

is called the pathname of the file that you normally think of as "junk". Pathname has an obvious meaning: it represents the full name of the path you have to follow from the root through the tree of directories to get to a particular file. It is a universal rule in the UNIX system that anywhere you can use an ordinary filename, you can use a pathname.

Here is a picture which may make this clearer:

```
                           (root)
                          / |  \
                         /  |   \
                        /   |    \
    bin      etc     usr01      dev     tmp
   / | \    / | \    / | \     / | \    / | \
                    /  |  \
                   /   |   \
               NSRDC   NWC   NUSC
              / | \   / | \  / | \
                       |
             adam     eve    mary
            / | \    / | \   / | \
                      |           junk
                 junk temp
```

Notice that Mary's junk is unrelated to Eve's.

This isn't too exciting if all the files of interest are in your
own directory, but if you work with someone else or on several pro-
jects concurrently and have separted each project into a separate
directory, it becomes handy indeed. For example, your sponsor can
print a project by saying

    pr /usr01/your-organization/your-name/proj/*

Similarly, you can find out what files your neighbor has by saying

    ls /usr01/neighbor-organization/neighbor-name

or make your own copy of one of his files by

    cp /usr01/your-neighbor/his-file yourfile

If your neighbor doesn't want you poking around in his files, or vice
versa, privacy can be arranged. Each file and directory has read-
write-execute permissions for the owner, a group, and everyone else,
which can be set to control access. See "man ls" and "man chmod" for
details.

As a final experiment with pathnames, try

    ls /bin /usr/bin

Do some of the names look familiar? When you run a program, by typing
its name after the prompt character, the system simply looks for a
file of that name. It normally looks first in your directory (where
it typically doesn't find it), then in "/bin" and finally in
"/usr/bin". There is nothing magic about commands like "cat" or "ls",

except that they have been collected into a couple of places to be easy to find and administer.

What if you work regularly with someone else on common information in his directory? You could just log in as your friend each time you want to, but you can also say "I want to work on his files instead of my own". This is done by changing the directory that you are currently in:

    cd /usr01/friends-organization/your-friend

Now when you use a filename in something like "cat" or "pr", it refers to the file in your friend's directory. Changing directories doesn't affect any permissions associated with a file, if you couldn't access a file from your own directory, changing to another directory won't alter that fact. Of course, if you forget what directory you're in, type

    pwd

to find out.

It is usually convenient to arrange your own files so that all the files related to one thing are in a directory separate from other projects. For example, when you configure a project directory, you might want to keep all the mail associated with this project in a directory called project1_mail. So make one with

    mkdir project1_mail

then go to it with

    cd project1_mail

    /usr01/your-organization/your-name/project1_mail

To remove the directory project1_mail, type

    rm project1_mail/*
    rmdir project1_mail

The first command removes all files from the directory; the second removes the empty directory.

You can go up one level in the tree of files by saying

    cd ..

".." is the actual name of the parent of whatever directory you are currently in. For completeness, "." is an alternate name for the directory you are in.

Directing Files

Most of the commands we have seen so far produce output on the termi-
nal; some, like the editor, also take their input from the terminal.
It is universal in UNIX systems that the terminal can be replaced by
a file for either or both of input and output. As one example,

    ls

makes a list of files on your terminal. But if you say

    ls > filelist

a list of your files will be placed in the file filelist (which will
be created if it doesn't already exist, or overwritten if it does).
The symbol ">" means put the output on the following file, rather than
on the terminal. Nothing is produced on the terminal. As another ex-
ample, you could combine several files into one by capturing the out-
put of "cat" in a file:

    cat fl f2 f3 > temp

The symbol ">>" operates very much like ">" does, except that it means
add to the end of. That is,

    cat fl f2 f3 >> temp

means to concatenate "f1", "f2" and "f3" to the end of whatever is al-
ready in temp, instead of overwriting the existing contents. As with
">", if temp doesn't exist, it will be created for you.

In a similar way, the symbol "<" means to take the input for a program
from the following file, instead of from the terminal. Thus, you
could make up a script of commonly used editing commands and put them
into a file called script. Then you can run the script on a file by
saying

    ed file < script

Pipes

One of the novel contributions of the UNIX system is the idea of a
pipe. A pipe is simply a way to connect the output of one program to
the input of another program, so the two run as a sequence of
processes - a pipeline.

For example,

    pr f g h

will print the files "f", "g", and "h", beginning each on a new page.
Suppose you want them run together instead. You could say

```
cat f g h > temp
pr < temp
rm temp
```

but this is more work than necessary. Clearly what we want is to take the output of "cat" and connect it to the input of "pr". So let us use a pipe:

```
cat f g h | pr
```

The vertical bar "|" means to take the output from "cat", which would normally have gone to the terminal, and put it into "pr" to be neatly formatted.

There are many other examples of pipes. For example,

```
ls | pr -3
```

prints a list of your files in three columns. The program "wc" counts the number of lines, words and characters in its input, and as we saw earlier, "who" prints a list of currently-logged on people, one per line. Thus

```
who | wc
```

tells how many people are logged on. And of course

```
ls | wc
```

counts your files.

Any program that reads from the terminal can read from a pipe instead; any program that writes on the terminal can drive a pipe. You can have as many elements in a pipeline as you wish.

Many UNIX programs are written so that they will take their input from one or more files if file arguments are given; if no arguments are given they will read from the terminal, and thus can be used in pipe-lines. "pr" is one example:

```
pr -3 a b c
```

prints files "a", "b" and "c" in order in three columns. But in

```
cat a b c | pr -3
```

"pr" prints the information coming down the pipeline, still in three columns.

The Shell Revisited

We have already mentioned the "shell". The shell is the program that

interprets what you type as commands and arguments. It also looks after translating "*", into lists of filenames, and "<", ">", and "|" into changes of input and output streams. The "*" means anything at all, so

        pr mail*

translates into print all files whose names begin with mail, listed in alphabetical order.

This shorthand notation is not a property of the "pr" command. The "shell" is system-wide, a service of every program that interprets commands. The "*" is not limited to the last position in a filename, it can be anywhere and can occur several times. Thus

    rm *junk* *temp*

removes all files that contain "junk" or "temp" as any part of their name. As a special case, "*" by itself matches every filename, so

    pr *

prints all your files (alphabetical order), and

    rm *

removes all files. (You had better be very sure that's what you want-ed to say!)

The "*" is not the only pattern-matching feature available. Suppose you want to print only mailmsg 1 through 4 and 9. Then you can say

    pr mailmsg[12349]*

The "[...]" means to match any of the characters inside the brackets. A range of consecutive letters or digits can be abbreviated, so you can also do this with

    pr mailmsg[1-49]*

Letters can also be used within brackets: "[a-z]" matches any charac-ter in the range "a" through "z".

The "?" pattern matches any single character, so

    ls -l mailmsg?.1

lists information about the first file of each mailmsg (mailmsg.1, mailmsg1.1, etc.).

Of these niceties, "*" is certainly the most useful, and you should get used to it. The others are frills, but worth knowing.

If you should ever have to turn off the special meaning of "*", "?", etc., enclose the entire argument in single quotes, as in

    ls '?'

REFERENCES

1. Brian W. Kernighan, "A Tutorial Introduction to the UNIX Text Editor," UNIX PROGRAMMER'S MANUAL, January, 1979.

2. William Joy, "An Introduction to Display Editing with Vi", UNIX PROGRAMMERS MANUAL Vol 2C, Novenber 1980.

APPENDIX B - NEMS Hardware

```
Digital Equipment Corporation
11/750-UH                     VAX 11/750, 1MB ECC/MOS
                              LA-120 DECwriter
                              DZ11-A 8 line Asynch Ctr
                              DD11-DK 9-slot back-plane
Disk drives                   2 - RL02 10.4MB drives with disk controller

National Semiconductor Corporation
NS753                         1 MB ECC/MOS

System Industries Inc.
9800                          Disk drive controller

Control Data Corporation
9762                          Removable 80MB Disk Drives (RM02)

Fujitsu Ltd.
M2288                         Fujitisu 160MB Wichester Disk Drives

ABLE Computer Inc.
DMAX                          16 Line DH communications controller

Wester Peripherals
TC131P                        Magnetic tape controller

Associated Computer Consultants
LH/DH                         Communication controller providing 1822
                              interface to BBN IMP and DEC VAX processor

GTE Telenet
TP-3010                       Telenet host with four lines to NEMS

Racal Vadic
VA212A                        4 - Bell 212A compatible dial out modems
```

Initial Distribution

COPIES:

12   DIRECTOR
            DEFENSE DOCUMENTATION CENTER (TIMA)
            CAMERON STATION
            ALEXANDRIA, VIRGINIA   23314

                    CENTER DISTRIBUTION

COPIES:

250   1892.1   USER SERVICES
  1   18/1809   GLEISSNER, G. H.
  1   1808   WILDY, D.
  1   189   GRAY, G.
  1   1892.1   STRICKLAND, J.
  1   1892.2   SOMMER, D.
  1   1892.3   MINOR, L.
  5   189.3   MORRIS, J.
  5   1892.3   CINCOTTA, A.
  1   1804   AVRUNIN, I. L.
  1   1844   MATULA, P.
  1   522   LIBRARY, CARDEROCK

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECH-
NICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF
THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIM-
INARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE.
THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION
OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR IN-
TERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE
NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC
MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE
BASIS.

END

DATE
FILMED

/1 — 8

DTIC

MANENT TECH-
GARDLESS OF

OF A PRELIM-
SIGNIFICANCE.

CUMENTATION
NDED FOR IN-
TYPE AND THE
SIDE DTNSRDC
CASE-BY-CASE