The ACSS Newsletter

University of Minnesota
Twin Cities
July 1988

Communications

Using MAIL and NOTES on UX

Susan Hail
BITNET: SLH@UMNACUX

Electronic mail has become one of the most important and widely used applications of computers. With UNIX MAIL, you can send messages to any other UNIX user. Your correspondents do not have to be on the ENCORE UX at the time you send your message; when they log on to the UX, the system will inform them that new MAIL messages are waiting for them. You can also send electronic mail to users on other systems over Internet, and through gateways to other networks like BITNET. See **`man mail`** for more information on UNIX MAIL.

Beginning MAIL

To read your MAIL, simply type the command **`mail`** and press **RETURN** (indicated as <CR>). UNIX will respond with two possible messages. If you have no new MAIL, you will see:

```
Mail version 2.18  Type ? for help.
"/usr/spool/mail/doc":  0 messages
No Mail
```

If you have MAIL, you will be informed about the message(s):

```
Mail version 2.18  Type ? for help.
"/usr/spool/mail/doc":  2 messages 2 new
>N  1:huxley      Fri Jun 17:12:30  10/241 "Ordering manuals"
>N  2:slh         Fri Jun 17:12:47  11/281 "Newsletter"
&
```

The & character is the prompt for MAIL commands.

continued on page 135

In This Issue . . .

Many of this month's articles describe aspects of our **ENCORE UX** service and **electronic mail.**

In the Consulting Corner, Susan Hail describes **how to use the ENCORE's on-line documentation facilities.**

For UNIX statistics users, Bruce Center introduces the version of **SPSS^X on the UX**.

Another article describes **UX masteruser software** for maintaining subordinate accounts.

Elsewhere, Jerry Stearns explains **network mail addresses** for users of electronic mail on our CYBER CA, ENCORE UX, and VAX VX.

Using BITNET on UX.
(See page 138.)

Help Page

HELP-LINES

Mainframe (CA,NV,UX,VX):	
626-5592	8 am to 5 pm weekdays
Artificial Intelligence:	
625-8332	3 to 4 pm weekdays
Data Base:	
626-1887	10 to 11 am weekdays
Graphics:	
626-5592	8 am to 5 pm weekdays
Liberal Arts, Text Analysis:	
625-8332	3 to 4 pm weekdays
Microcomputer:	
626-4276	9 am to 4 pm, weekdays (Closed 12:00 to 1:30 pm, Monday, Tuesday, and Friday.)
Statistics:	
626-1887	1 to 3 pm weekdays
Text Processing:	
625-1391	10 am to noon, Tuesday, Wednesday, Friday

CONSULTING

Walk-In

East Bank:

128C Lind Hall; 10 am to noon, 1 to 4 pm weekdays

West Bank:

140 Blegen Hall; 1 to 3 pm, Tuesday, Wednesday, and Thursday

Microcomputer:

125 Shepherd Lab; 9 am to 4 pm, weekdays (Closed from 12:00 to 1:30, Monday, Tuesday, and Friday.)

Electronic Mail Consulting

Consulting is now available via the mail facility on all ACSS systems (the CA, NV, VX, and UX). Send mail to user name CONSULT for questions after hours and for low-priority questions that are not critical to your immediate computing work. Replies will be sent to your account through the mail facility on your system.

Instructional Computing Consultant

Department instructors may call 626-0200 for assistance in choosing ACSS systems (CYBER/NOS, CYBER/VE, ENCORE/UNIX, VAX/VMS), software, and for answers to any other inquiries on using computers for instructional computing.

COMPUTING INFORMATION CENTER

128A Lind Hall, 625-7397, YZE6075@UMNACCA, YZE6075@UMNACUX, or MAD@UMNACVX

Computing account and grant applications available for CYBER, ENCORE, and VAX computers.

Short course enrollment. Short course schedules and class descriptions available.

Assistance in ordering vendor documentation. Vendor documentation is not always available in the University bookstores and may be ordered directly from the company.

Complete documentation collection. Reference copies of vendor and all other documentation for ACSS software.

Free ACSS documentation. General information, and mainframe and microcomputer information available.

Computing Newsletters. Subscribe to the *ACSS Newsletter* and the *Microcomputer Newsletter.* Newsletters from other computing centers are also available for reference.

continued from page 133

Sending MAIL

To send MAIL to another UX user, you must know your correspondent's UX login name. (Also see the section on the Electronic Postmaster in the next article. The Postmaster can help you find mail addresses.

In response to the system prompt or the MAIL prompt (&), type the command **mail** followed by your correspondent's login name. MAIL will prompt you for the subject of your message:

```
&mail doc<CR>
Subject: Ordering manuals<CR>
Do Encore manuals have to be ordered by U.S. mail? <CR>
.<CR>
EOT
&
```

To find out what editing options are available to you while typing in your MAIL message, key in **~?**.

To send your MAIL message, type a **<CTRL>-d**, that is hold the CONTROL key while pressing **d**, or use a single period in the first column of a new line, followed with a carriage return (shown above).

To leave MAIL without sending your message, use **<CTRL>-c**. If you end the message by typing in **<CTRL>-c**, MAIL responds with `(Interrupt — one more to kill letter)`. Type in **<CTRL>-c** one more time and you will be back to the system prompt. The **~q** command will also allow you to exit MAIL without sending your message. The **~q** command must be on a line by itself, with the tilde (~) in the first character position.

If you make a mistake and want to correct it, the simplest way is to use the **DELETE** key on your terminal or the BACKSPACE key (depending on your terminal definition) to backspace over the error. You can correct an error on a line after you press **RETURN** by using the **~v** command to enter the vi editor. (The command **~e** accesses the ex editor.) Your message is written to a temporary file, which you can then edit. Exiting the editor will rewrite your MAIL message and return you to the next new line in your MAIL message. (Use **~p** to view the message as it will appear to the recipient.)

Reading Your Mail

To read your MAIL message, type **mail** at the UNIX prompt. Again, the system will tell you that you have MAIL waiting for you before you see the MAIL prompt. To read the waiting message, type **<CR>**:

```
Mail version 2.18  Type ? for help.
"/usr/spool/mail/doc":  1 message 1 new
>N  1:huxley                    Fri Jun 17:12:30  10/241 "Ordering manuals"
>N  2:slh                       Fri Jun 17:12:47  11/281 "Newsletter"
&<CR>
Message 1:
From Huxley Fri Jun 17 13:01:09 1988
Date: Fri, Jun 17 88 13:01:06 cst
From:  Huxley (Aldous Huxley)
To: doc
Subject: Ordering manuals

Do Encore manuals have to be ordered by U.S. mail?
&
```

continued on page 136

continued from page 135

Notice the format of the message you've received. MAIL numbers each unread message you have received, records the date and time it was sent, and provides the login name of the person who sent the message. After MAIL displays the message, it displays the `&` prompt, so you can enter more commands.

To exit MAIL, use the `quit` command (abbreviated `q`). MAIL will write any messages flagged as read to your file called **mbox**. These messages can be reread by using the `mail -f` command. If you do not want to have a MAIL message written to **mbox**, use the `delete` (abbreviated `d`) command or the `delete print` (abbreviated `dp`) command after reading the message. You can undelete a message with the `undelete` (abbreviated `u`) command.

Helpful MAIL Commands

The following are a few helpful commands. (For on-line information on MAIL commands, key in `?` while reading messages or `~?` when typing in a message.) In the following list, a ***messagelist*** consists of integers, ranges of integers, or user names separated by spaces. If omitted, MAIL uses the last message typed. A ***userlist*** consists of user names or distribution names separated by spaces. Distribution names are defined in **.sendrc** in your home directory.

`cd`	*directory*	change working directory to *directory* or **home** if none given
`d`	*messagelist*	delete messages
`dp`	*messagelist*	delete messages and print next one
`e`	*messagelist*	edit messages
`f`	*messagelist*	give header lines of messages
`h`		print out active message headers
`m`	*userlist*	mail to specific users
`n`		go to and display next message
`pre`	*messagelist*	make messages stay in system mail
`q`		quit, saving unresolved messages in mbox
`r`	*messagelist*	reply to messages and all recipients
`R`		reply to sender of messages only
`s`	*messagelist file*	append messages to *file*
`t`	*messagelist*	type messages
`u`	*messagelist*	undelete messages
`x`		quit, do not remove messages from system mail
`!`	*command*	shell escape to execute *command*

To send a reply to a UX user whose message you've just read, use the `reply` (abbreviated `r`) command. This saves you the trouble of typing in the user name and subject; UX will send your message to the sender of the last message you've read.

To re-read the last message you read, type a p. To go back one message, type a hyphen (-).

To end your UNIX MAIL session, type a `q` or `<CTRL>-d`. If you wish to terminate the mail session without modifying your mailbox, type an `x`.

Helpful Editing Commands

When using ~ as an escape character, it must be in the first column. Typing `~~` will allow you to use the tilde as a text character in the first character position of a line. Otherwise it will be interpreted as an escape character.

To use a file as a message, you can read the file into MAIL with `~r file.` (Replace *file* with the name of the file.) You can also write your message to a file with the `~w file` command. See the following summary:

~c	***users***	Add the given names to the 'cc' list (copy to ***users***)
~d		Read the file **dead.letter**
~e		Invoke the text edit
~f		Forward message
~h		Prompt for message header fields
~m	5	Read in message 5, right shifted by a tab
~p		Print the message on the terminal
~q		Abort the message to **dead.letter** in your home directory
~r	***file***	Read in ***file*** into the message barrier
~s	***string***	Change the subject field in the header to ***string***
~t	***users***	Add ***users*** to the To list
~v		Invoke the **vi** display editor on the message
~w	***file***	Write a copy of the message onto ***file***
~~		Quote a single tilde
~|	***command***	Pipe the message through the command
~!	***command***	Execute the shell command
~?		Display this command list

For More Information

UX's own HELP file also has information on the MAIL utility. Log on to UX and type **`help mail`**. More information is available through the UNIX on-line manual by using the command **`man mail`**. Within MAIL there is also a help file. Type **`help`** for a reference summary of MAIL commands.

One printed reference on UNIX MAIL is the section in the *UNIX Beginner's Guide*, which consists of selections from the Berkeley UNIX Reference Set, reprinted by ACSS with permission of the University of California, Berkeley. You can find a reference copy of this manual (and a complete Reference Set) in the ACSS Computing Information Center, 128A Lind Hall. You can purchase the manuals in the Student Book Store in Dinkytown and in the Minnesota Book Center in Williamson Hall.

Using notes

`notes` is a news system for computer-managed conferences. Many topics are available through UX's **`notes`** facility. You can see the conference topics by simply typing **`notes`**:

```
notes
Usage: notes [-s] [-t ttytype] [-f file] [-a seqname] topic [..]
Some Local Notesfiles:

announce                  | Announcements of lectures, seminars, doughnut hour.
general                   | General purpose notes.
notesfiles                | Descriptions of most of the notesfiles
problems                  | Problems with system software & hardware.
sysnews                   | System level software/hardware announcements
test                      | Test purposes.
nfgripes                  | Notesfile complaints
nfmaint                   | Notesfile maintenance
email                     | Electronic Mail news
sports                    | Sports-related notes
consultnotes              | Notes from 'consult'
maildaemon                | Contain DAEMON mail
linkfail                  | BITNET Linkfail messages

Other Network Notesfiles:

cyber-l         future-l        ibm-nets        mail-l
jnet-l          tcpip-l         big-lan         ibmtcp-l
bind-l          info-enc        cisco-l
```

continued on page 138

continued from page 137

In order to read the notes available type in **notes** followed by the topic name. **notes** will respond with a list of all notes in the topic:

```
notes linkfail
5/11/88    9*Syracuse Outages                  lsg
5/13      10*BAYLOR downtime                   lsg
          11*link instability in Chicago       lsg
          12*Downtime at UKCC                  lsg
5/18      13*Psuvax1/Shire switch              lsg
          14*Turkish nodes downtime            lsg
          15*WISCMAC3 downtime                 lsg
          16*CMU CC down 6/4                   lsg
          17*MAILER problems at HASARA5        lsg
          18*UKANVM disk problems              lsg
5/20      19*Akron nodes downtime              lsg
          20*SEARN down 6/4                    lsg

             **** End of Notes ****
```

To read a note, type in the note number (for example, to read about Syracuse Outages, type in **9**.) To advance to the next unread response to the note, or the next unread note if there are no unread responses, type in **j** for jump. A **<CR>** will skip responses and read the next unread note. Type in **q** to quit.

For more information, **notes** has an on-line HELP facility. To access, type a question mark. See also **man notes** for a detailed discussion of **notes**.

Using BITNET on UX

Susan Hail
BITNET: SLH@UMNACUX

With UNIX MAIL, you can send messages to colleagues who have an account on ACSS's ENCORE UNIX mainframe computer (the UX system). See the previous article, "Using MAIL and NOTES on UX," for more information on using MAIL. It is also possible to send mail from UX to other networks, including BITNET (the Because It's Time NETwork). ACSS's UNIX is connected to the BITNET network as node UMNACUX.

NOTE: BITNET on UX is a mail-only node. It can send and receive BITNET mail messages, but not data files, binary files, or interactive commands.

UX is connected to the Internet network as node UX.ACSS.UMN.EDU. If BITNET mail is not being delivered via the address UMNACUX, the UX.ACSS.UMN.EDU address may be more appropriate for the specific networks involved in the mail transfer.

Node names are public knowledge—it's a code for the particular computer your correspondent uses for his or her electronic mail. Some institutions have more than one computer serving as a BITNET or Internet node; make sure you know which node your recipient uses.

Your recipient's user name is *not* generally public knowledge, and without it you cannot send mail to a particular BITNET or Internet user. **Typically you will exchange BITNET or Internet addresses with your recipients by some other form of communication—letter, phone, etc.** Some scholarly journals and professional associations now publish the BITNET or Internet addresses of their contributors and members. A future edition of the *University Student and Staff Directory* will include electronic mail addresses of University staff and faculty.

For more information, see **`man bitnet`**.

Electronic Postmaster

ACSS provides a utility called the Electronic Postmaster (EP). EP provides electronic mail addresses in response to specific requests, information about mainframe computers or the name of a contact person who can help you with electronic mail services. EP can also supply general help on using its commands.

To request information from the EP, you send a mail message to the user named **Postmaster** using the appropriate commands. To request information from the EP data base, the commands available are:

```
TELLME ABOUT username
TELLME HOSTS
TELLME CONTACTS or
TELLME        {which defaults to TELLME ABOUT}
```

These commands can be the SUBJECT of the mail message to the EP, or can be the text of the mail message. ***username*** can be the full name, the login name, the first name or the last name of a user.

The EP can understand other words in your message. For example, EP will understand the following messages:

Who is DOC
CONTACTS
HOSTS
Who are your contacts
What is ux
Give me information about user DOC

There is also a **HELP** command within mail. If the EP does not understand your command, it will treat your message as though it received a **HELP** command.

Using epupdate

To add or change information from the Electronic Postmaster, use the utility program **epupdate**. The following are required by the EP:

Your login ID
Full name of your machine
Your real name

The proper full name of ACSS's UNIX computer is:

@ux.acss.umn.edu

continued on page 140

continued from page 139

Other information requested for the EP data base are:

Your work department
Your work telephone number
Your mailing address at work
Your title, or college attending

Optional information for the EP data base are:

Your home telephone number
Your home mailing address

epupdate will display any information currently associated with your login id, then prompt you for all the information listed above. **epupdate** queues the information for later verification and update. When the update is made, a mail message will be sent to your mail box notifying you of the update.

All updates to the EP data base from any computer will be forwarded to all university computers subscribing to the EP.

For more information on the EP, see **help postmaster**.

Sending BITNET or Internet Mail

A BITNET address, mailed from UX's MAIL utility, takes this form:

```
mail user@node
```

In this address, you have to replace ***user*** with the user name of the recipient of your message, and replace ***node*** with the "node name" for your recipient's BITNET connection. The `@` character separates the user name and node name.

Here's an example: To send a BITNET message to the University of Notre Dame, at the system prompt you would use this address:

```
mail phj@irishmvs
```

This address contains the computer name of your recipient (here **phj**, a fictional name for the purposes of this example) and the name of Notre Dame's node (**irishmvs**).

To send mail over the Internet network, you would use the following address:

```
mail szw@vx.acss.umn.edu
```

The computer name of your recipient here is **szw**. The Internet name **`vx.acss.umn.edu`** is for ACSS's VAX/VMS machine, VX.

The entire process is very much like sending a MAIL message to another UX user. Your message is sent immediately to node UMNACVX, where user **`szw`** can read it the next time she uses electronic mail on the VX.

If someone wants to send BITNET mail to *you*, they will have to know your address—that is, your MAIL name and UNIX node name, UMNACUX or UX.ACSS.UMN.EDU. Whenever you send a BITNET or Internet message, your complete address will automatically be included as your return address with the message.

Receiving BITNET or Internet Messages

To read messages, enter the MAIL utility and read the BITNET or Internet message just as you read mail from other UX users. By simply pressing RETURN you display the new message on your terminal screen. If there is more than one new mail message, in place of RETURN you can type in the message number of the message you want to read.

Example

```
You have mail.
ux.acss.umn.edu% mail<CR>
Mail version 2.18  Type ? for help.
"/usr/spool/mail/szw": 5 messages 5 new
>N  1 doc                  Fri Jun 17 15:42  11/308 "Summer Hours"
 N  2 phj                  Fri Jun 17 16:30  11/295 "Meeting"
 N  3 SLH@vx.acss.umn$     Fri Jun 17 16:45  12/420 "Beginning c Programming"
 N  4 Postmaster@vx.a$     Fri Jun 17 17:13  28/899 "Undeliverable mail"
 N  5 MAILER-DAEMON        Fri Jun 17 17:26  23/717 "Returned mail: User unknown"

& 3<CR>
Message  3:
From SLH@vx.acss.umn.edu Fri Jun 17 16:45:10 1988
Subject: Beginning c Programming
To: doc@ux.acss.umn.edu
X-Vms-To: IN%"doc@umnacux"

Will this short course be offered this Fall quarter?
& x<CR>
```

In this example, the mail facility is exited with an **x**. All messages that have been read or are still unread will remain in your mail buffer. To clear a message out of your mail buffer and write it to a permanent storage file, use the **q** command after reading your mail. Only messages flagged as having been read will be written to your file called **mbox**. See the previous chapter for more editing options.

When Things Go Wrong

If you type the node name incorrectly, or send to a non-existent node, MAIL will attempt to deliver the message. When it fails, it may return the message to you as incoming mail from the mail transfer agent called "Postmaster" or "MAILER-DAEMON"; it will be labeled "Undeliverable mail." If the user **szw** (referred to in the previous example) did not exist, your mail will include an appropriate message. The following message is an example:

```
Date: Fri, 17 Jun 88 17:17 cst
From: PMDF Mail Server <Postmaster@vx.acss.umn.edu>
Subject: Undeliverable mail
To: doc@ux.acss.umn.edu

The message could not be delivered to:

...more..
```

Retype **mail** to read more information on the transmission. The following is an example of an undeliverable letter.

```
You have mail.
```

continued on page 142

continued from page 141

```
ux.acss.umn.edu% mail<CR>
Mail version 2.18  Type ? for help.
"/usr/spool/mail/doc": 5 messages 5 new
>N  1 doc                   Fri Jun 17 15:42  11/308 "Easter Hours"
 N  2 phj                   Fri Jun 17 16:30  11/295 "Meeting"
 N  3 SLH@vx.acss.umn$      Fri Jun 17 16:45  12/420 "Beginning c Programming"
 N  4 Postmaster@vx.a$      Fri Jun 17 17:13  28/899 "Undeliverable mail"
 N  5 MAILER-DAEMON         Fri Jun 17 17:26  23/717 "Returned mail: User unknown"

& 4<CR>
Message  4:
From postmaster@vx.acss.umn.edu Fri Jun 17 17:13:58 1988
Date: Fri, 17 Jun 88 17:11 cst
From: PMDF Mail Server <Postmaster@vx.acss.umn.edu>
Subject: Undeliverable mail
To: doc@ux.acss.umn.edu

The message could not be delivered to:

Addressee: szw
Reason:
%MAIL-E-NOSUCHUSR, no such user SZW at node ACSSVX

____________________

Return-path: doc@ux.acss.UMN.EDU
Received: from ux.acss.umn.edu by UMNACVX.BITNET via TCP; Fri Jun 17 17:11 cst
Received: by ux.acss.umn.edu (4.12/4.7) id AA19691; Fri, 17 Jun 88 17:13:12 cst
Date: Fri, 17 Jun 88 17:13:12 cst
From: Tech Pubs <doc@ux.acss.UMN.EDU>
Subject: Operating hours
Message-Id: <8803182313.AA19691@ux.acss.umn.edu>
To: szw@UMNACVX.BITNET
Status: RO

What will be the operating hours for the computer center during the summer session?
```

Message 5 in this example returns an error message for an invalid Internet address, much the same as an invalid BITNET address.

If Mail Is Not Being Delivered

A message may have to pass through another network node or nodes before it reaches its destination. If one of these nodes goes down, you can check the **notes** topic called **linkfail** for status reports. See the previous article on "Using MAIL and NOTES on UX" for information on how to use **notes**. Your message will be held until the node goes back up.

Other Networks

You can communicate over a number of other networks if you have the required address in the necessary format. Some of these are:

***name@node*.bitnet**
***name@node*.csnet**
***name@node*.uucp**

See this issue's article on network addresses for more information.

For More Information

Our short course schedule often includes free courses on UNIX's operating system and on electronic mail. Consult our Computing Information Center, 128A Lind Hall, for information about the current schedule. If you're having problems with UNIX mail or BITNET, you can also call our HELP-Line at 626-5592 from 8 to 5 daily.

Network Mail Address Formats

Jerry Stearns
BITNET: CORDWAINER@UMNACVX
INTERNET: CORDWAINER@VX.ACSS.UMN.EDU

The University has recently begun to collect faculty and staff electronic mail addresses, in preparation for publishing them as part of the listings in a future edition of the *University Student and Staff Directory.* (The plan, as I understand it, is to begin including electronic mail addresses in the 1989-90 edition of the *Directory.*)

This article is intended to help you understand what addresses you have — assuming you use electronic mail from an ACSS mainframe — and what the correct formats of the addresses are. It is understandable that people are confused about network addresses. Perhaps we can make it a little clearer.

Networks

First of all, ACSS mainframes are connected to three networks:

BITNET This is a national and international network of educational institution computers. The ACSS VAX VX (UMNACVX) and CYBER CA (UMNACCA) are directly connected to this network. The ENCORE UX (UMNACUX) is also a member, but it is a "mail-only" node and receives only mail messages, not files. Other University mainframes, including those operated by Health Sciences Computing Services and St. Paul Computing Services, are also on BITNET.

Internet An *internet* is a collection of networks using common communication protocols that allow them to talk to each other. The University *Campus Internet* is a collection of campus networks (including mainframe and micro networks) using the TCP/IP protocols. The Campus Internet communicates with regional and national networks like the statewide computer network (MRnet) and national networks like ARPAnet, MILnet, and NSFnet.

DECnet The Digital Equipment Corporation, who builds VAX computers, encourages administrators of VAXes to network them together. Many University VAXes are connected in a campus DECnet, and are also connected to nationwide DECnet networks (SPAN and HEPnet, the Space Physics Analysis Network and the High Energy Physics network).

There are many other networks (USENET, CSNET, EARN, NETNORTH, etc.) to which ACSS does not subscribe, but to which we have access through "gateways" and other computers. Some of these other networks do not use the TCP/IP protocols operating on the campus Internet. This can make addressing of electronic mail even more confusing.

continued on page 144

continued from page 143

Addresses: How computers know where to send messages

Each computer user has an address to which his or her messages are sent.

There are three components to an electronic mail address:

1. The user's *user name*, or local address.
2. The *node*, or computer on which that user resides.
3. Routing components with one or more *delimiters*, or separators.

Usually the local address consists of the **user name** with which you log onto the computer. This is true on the ACSS VAX and ENCORE machines. This username should always be no more than eight characters long, for that is all that BITNET will accept. To use mail on the CYBER CA, you first have to explicitly join CYBER Mail. To send mail to another CA user, you have to send the message to whatever CYBER Mail name your recipient has chosen as his or her CA address.

The **node** is the computer on which each user receives his or her messages. The ACSS VAX, ENCORE, and CYBER are each mail nodes on BITNET, and on the Internet. (CA does have an Internet adress, but only accepts mail via its BITNET connection.)

Delimiters separate the parts of the network routing information from each other. An `@` (the "at" sign) usually separates the username from the nodename. Sometimes you will see a `%` (percent) between user name and nodename, with an `@` to separate these two from a further gateway designation. Other delimiters and syntax can become very cryptic.

Formats: What your electronic mail address should look like

The following are generic forms and examples of address formats. You use one of these formats when your mail program prompts you (with `TO:`) for the address to which you're sending the message. In the generic forms (on the left), the italicized portions are to be replaced by specific user names or other information, as shown in the examples on the right:

BITNET:	***Username*@*Node***	**JSTEARNS@UMNACVX**
Internet:	***Username*@*xx.yy.zzz*.EDU**	**JSTEARNS@VX.ACSS.UMN.EDU**
DECnet:	***DEC_node*::*Username***	**ACSSVX::JSTEARNS**

In references to the BITNET address format, you will sometimes see an additional extension:

JSTEARNS@UMNACVXJ.BITNET

But it is generally unnecessary to add this extension when you're sending BITNET mail.

You may need to change or alter these formats slightly, depending on which mai package you're working with. If you are sending mail from the VAX VX via BITNET or the Internet, you must enclose each address in quotes: **IN%*"address"***. The **IN%** lets VMS Mail know that this message is being sent outside of the host computer (the VX).

On the CA and UX machines it is *not* necessary to enclose the address in this fashion. Nor is it necessary to use it on the VX when your destination address is on the DECnet.

Address Tables and Lists

There are **tables** of addresses on the host computer; the mailing software searches these tables to validate the destination address. Sometimes mail you send is rejected because the destination computer ignores its own name, and **sometimes the tables on your host computer are different from the tables on the destination computer.** If so, you and your addressee should contact the administrators of your systems for clarification.

In Internet addresses for VX users, the node address **vx.acss** may work fine for others on the campus network. But chances are it will not be sufficient for users at another node who do not have that same abbreviation defined in their tables. As shown above, the VX's full Internet address is **vx.acss.umn.edu.** The CA's address is **ca.acss.umn.edu** and the UX's address is **ux.acss.umn.edu**.

Some ACSS mainframes have lists of BITNET node addresses. On VX you may search the file **NET$DOCS:BITNET.LINKS** to find out about a specific BITNET address. This file contains the name of the institution which operates the node, although the names are not consistent in the forms in which institutions are listed. (e.g., the University may be listed under *MN*, *MINN*, or *Minnesota*). The same list is available on the CYBER CA as **WRITEUP,BNODES.**

The VX file **NET$DOCS:BITNET.GATES** is a list of gateways available, which are links to other networks. For the same information on the CA, see **WRITEUP,BGATES.**

Another place you can look for an address is the Electronic Postmaster. If you include your Electronic Mail address in the *Student and Staff Directory,* you should also consider including it in the Postmaster database. All updates to the Postmaster database made from any ACSS machine are forwarded to all campus computers subscribing to the Postmaster system. See the Electronic Postmaster writeup on UX (**help postmaster**) or VX (**ACSS$WRITEUP:POSTMASTER**) or CA (**WRITEUP,EP**) for information on using the Electronic Postmaster.

Using SPSS^x on UX

Bruce A. Center
BITNET: BAC@UMNACUX

We have installed the UNIX version of SPSS^x (Version 2.2) on the ENCORE UX. In the following instructions for using SPSS^x on UX, we assume you have some basic knowledge of the operating system. If you are new to UNIX, see the ACSS manual *An Introduction to UNIX Computing*, available at Minnesota Book Center in Williamson Hall and the Student Book Store in Dinkytown. ACSS also offers free short courses on UNIX every quarter.

Beginning SPSS^x

You can begin SPSS^x by typing:

```
spssx <infile >outfile
```

where:

infile specifies the name of the file that contains the SPSS^x commands, and possibly, in-line data. Command lines and in-line data cannot exceed 80 columns. If omitted, standard input is assumed and terminal data entry of commands is expected. You can use any valid UNIX file specification (including path names).

outfile specifies the name of the file to receive the printed output including warnings and error messages, of the SPSS^x job. If omitted, the standard output is assumed and the output will be displayed at the terminal. In addition error messages are always directed to the terminal. You can use any valid UNIX file specification (including path names).

When invoking SPSS^x, you *must have WRITE access* to:

a. the current directory in which SPSS^x is running, since SPSS^x writes scratch files to this directory.

b. the directory in which you want to write the output file.

The scratch files that SPSS^x creates are automatically deleted at the end of the job.

Two command line switches are recognized by SPSS^x:

-s followed by a positive integer indicates the amount of workspace desired. The default is 200k bytes and the maximum allowed is 4000k bytes. *K* and *k* are recognized in the standard fashion, that is 1024 (200k = 204,800).

-b suppresses the printing of the banner information that is printed at the front of every SPSS-X job listing. For example, the command

```
spssx -b -s10k < itest/command.sps
```

suppresses printing of the banner and allocates 10240 bytes of workspace for the job run with the command file called **command.sps** in directory **ltest**.

Running SPSS^x from Your Terminal

If you run SPSS^x interactively and enter input at your terminal, execution is usually delayed by one line as SPSS^x is unable to detect that you have finished entering a command until a new command is encountered. We strongly recommend that you don't do this.

Instead you should create a file with your SPSS^x commands and direct this file to the SPSS^x program using the shell operator <. For example, the command

```
   spssx <test.sps >test.lis
```

will execute SPSS^x at your terminal, read SPSS^x commands from the file **test.sps**, and write the results to a file called **test.lis**. Warnings and error messages will be printed at your terminal (as well as in the output file, **test.lis**), as they occur.

Running SPSS^x as a Detached Process

Running SPSS^x as a detached process allows you to work on other things at your terminal while SPSS^x is running. It also allows you to queue up several SPSS^x jobs to run at another time when the system has lower rates. There are several ways that you can run SPSS^x as a detached process.

The simplest way to run SPSS^x as a detached process is to create a file that contains your SPSS^x commands and then execute the shell command **`spssx`** as a background process (using an **`&`** character at the end of the command line). For example:

```
   nice spssx <test.sps >test.lis  &
```

runs the SPSS^x job in the background using the command file **test.sps**. The output is redirected to **test.lis**.

Typing **`ps -u`** will monitor the progress of the job. SPSS^x will also issue an `END OF JOB` message when it's through.

Another UNIX command can be useful to run jobs when you are not currently logged in:

`nohup` The **`nohup`** command (with an **`&`** at the end of the command line), will run a detached process preventing termination of your job when you log out. For example,

```
      nohup nice spssx <test.sps >test.lis &
```

runs the same job as above but the job will not terminate if you log out.

Since SPSS^x writes its error messages to the terminal as well as to the output file, these messages can become annoying, when you are running a job in the background. You can redirect the error output as well, using the **2>** redirection operator, but you *must* be in the Bourne shell to do this. As an example, consider the three commands in the following sequence:

```
   % sh        {Enters the Bourne shell; you will probably get a $ prompt, as in the next
                line. }
```

continued on page 148

continued from page 147

```
$ nice spssx <test.sps >test.lis 2>test.err &

                    {This command runs SPSSX in the background.
                     The input comes from  test.sps.
                     The output goes to  test.lis.
                     The errors and log data are placed in test.err
                     and test.lis but not  at the terminal.}

$ exit              {Exits the Bourne shell; returns to the C shell. }
```

For more information on running processes in the background, consult the *UNIX Beginner's Reference* and other UNIX documentation available at the Computing Information Center in 128A Lind Hall.

Running Jobs Inexpensively

The `at` command will run a command file at whatever date and time you specify. ACSS rates are approximately one-third cheaper after 8:00 p.m. and on weekends. So, for example, if you create a one-line file, **test.sub**, that reads

```
nice spssx <test.sps >test.lis &
```

and then type

```
at 9pm test.sub
```

your SPSS^{x} job won't run until 9 p.m. that night.

Printing SPSS^{x} Output

SPSS^{x} output files contain FORTRAN forms control, (i.e., printer control characters in column 1) so you need to use **lpr -f** to print them. For instance:

```
lpr -f -Plind128b -Jea.450 outfile
```

would print ***outfile*** at Lind Hall and direct it to output bin 450.

SPSS^{x} on UX defaults to a page width of 80 columns. If you want more than 80 columns (using the `SET WIDTH` command in SPSS^{x}, for instance), printing it is a little trickier. Currently, you need to use an Apple LaserWriter as your output device and type:

```
fpr <outfile | enscript -rl -f Courier8 -Plind128b -Jea.450
```

which would rotate your output (i.e., print it in landscape mode), print it at Lind Hall, and direct it to output bin 450. For more information, see **man enscript.**

FILE HANDLE

You should use an SPSS^{x} **FILE HANDLE** command to define each file that you read or write. The **FILE HANDLE** command provides a handle for use in other SPSS^{x} commands. For the UNIX version, the only subcommand required for most files is **NAME**, which supplies the file specification (including pathname, if required) enclosed in single quotes. For example, if you are reading **year80.dat** with a **DATA LIST**, the **FILE HANDLE** is

```
FILE HANDLE  RAWDATA/ NAME='year80.dat'
```

where **RAWDATA** is the arbitrary handle assigned to **year80.dat**. Upper- and lowercase may be used interchangeably in SPSS^{X} commands, but *case is significant in the quoted string* designating the file name. There is a 44-character limit to file specifications.

You can use two additional subcommands on the **FILE HANDLE** command. The **LRECL** subcommand specifies the record length in bytes:

```
FILE HANDLE inpdat / name='/u2/data/inp.dat' / LRECL=100
```

If not specified, the maximum record lengths are 1024 for sequential input files, 80 for portable files, and 1024 for all other output files. If they are longer than this, you must use **LRECL**.

The **MODE** subcommand may have the values **CHAR** (the default), **BINARY**, or **IMAGE**.

```
FILE HANDLE bindat / name='binary.data' / mode=binary
```

A **BINARY** file has an additional word at the beginning and end of each record containing the byte count for that record, and is the mode normally written by f77 for unformatted records. An **IMAGE** file also contains binary data but has no record delimiters; the **LRECL** subcommand must always be present for **IMAGE** mode files.

Reserved Files

The only **FILE HANDLE** that is reserved is **INLINE**, which refers to data imbedded in the command stream. Do not attempt to use this handle for any other purpose.

Commands not Implemented

The following commands and facilities are not implemented in the UNIX version of SPSS^{X}:

```
USERPROC
USERGET
GET SAS
GET SCSS
SAVE SCSS
GET OSIRIS
GET BMDP
P, PK, and Z formats
```

You won't miss them.

Documentation

The SPSS^{X} manual is available for examination and reference at the Computing Information Center in 128A Lind Hall. You can buy this manual at the Student Book Store in Dinkytown, and the H.D. Smith Bookstore on West Bank.

Use the command `man spssx` to access further information about using SPSS^{X} at the University of Minnesota on UX.

Quintus Prolog Demonstration Library on VX

Ron Zacharski
BITNET: RAZ@UMNACVX

The Quintus Prolog Demonstration Library contains a range of programs that both illustrate the usefulness of Prolog in general and point out features of the Quintus implementation. The first set of demonstrations presents applications in which the use of Prolog is preferred over the use of Fortran, C, or Pascal.

An example is "Computer Algebra in Prolog," by Sergio Vaghi. In Vaghi's article and accompanying demonstration programs, a system for symbolic integration is developed. Another demonstration program, CHAT, by Fernando Pereira, illustrates the use of a left corner chart parser to construct a natural language interface to a geographical data base. This is by far the most ambitious of the demonstration programs and certainly the most entertaining. The program can handle English queries such as:

> What are the capitals of the countries bordering the Baltic?
> What is the total area of countries south of the Equator and not in Australasia?

and is impressively fast, responding to these queries in less than one-tenth of a second.

Other general Prolog demonstrations include:

person:	a simple data base example
parser:	a graphic parser
wafer:	a simple expert system dealing with silicon wafers.

In the second set of demonstration programs illustrating the implementation, there are a number of programs that show how Quintus Prolog can interface with other programming languages. Both GRAPHICS, a program that handles simple windowing, and MATH, a program that calculates a number of mathematical functions, show how Prolog can link to C. Other programs include interfaces to Fortran, Pascal, and Basic code.

All these programs are easy to use. Each has a README file that gives a brief description of the program, as well as detailed instructions on how to run it. Each demonstration is contained in its own subdirectory of **userb:[languages.prolog.quintus$2_0.demo]**.

For more information enter

```
$ type prolog_demos:[000000]readme
```

at the main VMS prompt.

If you have any questions about Quintus Prolog or the demonstration library contact:

Ron Zacharski
Special Projects Group
M141 Fraser (office)
128A Lind (mail)
625-8332

Using On-line Documents on UX

Susan Hail
BITNET: SLH@UMNACUX

Question: What is "on-line" documentation and how do I use it on the UX?

Answer: *On-line* documentation is information that is stored within a computer. You can use system commands to read on-line documentation at your terminal.

On the ENCORE UX we have several types of on-line documentation. UX users should know how to use all of them.

Using `man`

The UNIX *User's Reference Manual* is available on-line. You can search this manual for information on a command by typing:

```
man command-name
```

To find out more about the `man` command, type:

```
man man
```

For a quick summary of some of the resources and packages available to you, type:

```
man packages
```

If there is more information available than can be displayed at one time on your screen, `man` displays the first page of text. At the end of each page, you will be informed as to how much of the document has been displayed, as well as instructions to use the SPACE bar to continue or the `q` to quit. (A `<CR>` is not needed.) You will exit `man` at the end of the available information, or type `<CTRL>-c` or `q` to exit the on-line manual at any time. Type `h` for on-line help on other `man` command options.

You may find the following `man` pages helpful:

`man csh`	Information on the C shell command interpreter. (This is the default shell at ACSS.)
`man ex`	The EX text editor.
`man help`	How to use the help program.
`man labs`	General information about campus lab facilities.
`man lpr`	How to get printed output.
`man man`	Information from programmer's manual on how to use `man`.
`man more`	Information on how to examine a text file a screen at a time.
`man packages`	General information about local packages.
`man sh`	Information on an AT&T Bourne compatible shell.
`man stty`	More terminal function options.
`man termcap`	How to define functions and capabilities of a terminal.
`man vi`	The `vi`sual screen-oriented editor.

continued on page 152

continued from page 151

Using **help**

There is also an interactive **help** facility on the services and packages available to our users. By typing **help**, the topics available will be listed at your terminal:

```
HELP:
This is the new help facility. Please select one of the topics below.

Subtopics:


awk                 system_use
editors             utilities
find_file
general
languages
libraries
mail
postmaster
printers
statpacks

(type '*' for commands, '#' for topics)
Help Topic?
```

As with the **man** documentation, **help** will display one screen of information at a time. At the end of each page, you will be informed of how much of the document has been displayed, as well as instructions to use the SPACE bar to continue or the **q** to quit. (A <CR> is not needed.)

To exit **help**, type a <CR> in response to `Help Topic?` For more information, see **man help** or **help general help**.

To print out a hard copy of a **man** or **help** document, write a copy of the document to your current working directory by typing:

```
man vi > tempfile
help editors > tempfile<CR>
```

After listing **help** information to a file, **help** will be expecting input. Press <CR> several times to exit **help**. ***tempfile*** can then be routed to a printer. The following example will print ***tempfile*** on the Lind Hall LaserWriter printer and deliver it to open bin 425 in Lind Hall 128B:

```
lpr -P lind128a -C ea.425 tempfile
```

How to Use the Masteruser Program

Adapted from the on-line document **help system_use masteruser**

The **masteruser** program on the UX was designed for professors and teaching assistants who may need to access their students' files, change students' passwords, and locate and manage other information, without having to know the system superuser password. The masteruser routine is used in both the **su** and **passwd** programs.

Before using the masteruser program, you have to get a master account defined in the file **/etc/masteruser**. To do this, send an electronic mail message to CONSULT on the UX.

```
/etc/masteruser file format:

master1:subord1:subord2:subord3
master2:subord4:subord5:subord6:subord7:master1
```

master1 is a master user to subord1, subord2, and subord3.

master2 is a master user to subord4, subord5, subord6, subord7, and master1, and indirectly to subord1, subord2, and subord3.

The su Program

The masteruser may run the **su** program to actually become the subordinate account. To do this, the masteruser types:

```
su subord
```

where ***subord*** is the login name of the subordinate account. The masteruser will then be running a shell on the subordinate account. At this point, the masteruser can make any modifications to subordinate files and file permissions, just as if the subordinate himself were logged in. The masteruser can create files, delete them, and use **chmod** to change permissions. The masteruser can also kill currently running subordinate processes, including the login shell, by running the **ps** program and killing the subordinate account's process ID's. To exit, type **exit**.

Using passwd, chownw, chmodm

The masteruser must be logged in to his own account—not the subordinate account— when changing the subordinate passwords, ownerships, or file permissions.

passwd: To change the subordinate's password, type:

```
passwd subord
```

where ***subord*** is the login name of the subordinate account. The old password will not be prompted for, and the new password can be entered.

continued on page 154

continued from page 153

The **passwd** program has also been modified to accept input on the command line. This may be useful when changing many passwords for many accounts. For example:

```
passwd subord musical
```

will change the ***subord*** password to **musical**. Please note that this line will be stored with .history files. Be careful about using this feature if you use the **savehist** command.

The **passwd -f** and **-s** options to change finger and shell information will also accept input on the command line. When using this option, be sure the option (**-f** or **-s**) is the first argument specified. When using the **-f** option, the account name must be specified, as well as all five finger arguments, or an error message will occur. (See the on-line document **man passwd**).

For example, the following will change finger information for the username **fred**:

```
passwd -f fred ‘Freddie Jones’ 140 333-5577 922-9000 300
```

where Freddie Jones is Fred's full name, 140 is Fred's office, 333-5577 is Fred's office phone number, 922-9000 is his home phone, and 300 is the bin he uses. If an argument requires spaces, for example in Fred's full name, the name must be enclosed in quotation marks (‘Freddie Jones). For a null entry, enter **none**.

chownm: To change the ownership of a file or directory, a valid masteruser relationship must exist between both the master/current-file owner, and master/new-file owner. See **man chown** for **chown** information.

Example:

```
chownm -R fred /usr5/derf
```

will make **fred** the new owner of all files and directories found under **/usr5/derf** providing that the masteruser is master over both **fred** and **derf**.

chmodm: See **man chmod** for **chmod** information.

Example:

```
chmodm -R 755 /usr5/fred
```

will change permissions of all files and directories found under **/usr5/derf** providing that the masteruser is a master over **fred**.

Symbolic Algebra System REDUCE 3.3 on VX

Michael J. Frisch
BITNET: MJFRISCH@UMNACVX
CA,UX,VX: MJFRISCH

On June 13, we upgraded the REDUCE system for symbolic algebraic manipulation from Version 3.2 to Version 3.3 on our VAX VMS system. Many algorithms were improved and bugs were fixed. We believe the package is completely upward compatible although it is larger, going from a 3.2 megabyte to a 5.7 megabyte virtual memory partition and so it may require a larger page quota in your account.

New features are: LIST data structure with supporting operators; eigenvalues available in the Matrix package; rational, finite field, and complex numbers available; and output format in two dimensions: (x+y)/2 is printed as

$$\frac{x+y}{2}$$

There are six new contributed packages:

- Algint, an extension of the basic REDUCE integration package,
- Anum, an algebraic number package,
- Excalc, a differential geometry package,
- Gentran, a Fortran and C code generation package,
- Groebner, a Groebner basis computation package, and
- Spde, a package for finding symmetry groups of partial differential equations.

See the documentation below for details.

Running REDUCE

REDUCE 3.3 is run interactively with the VMS command:

```
$ reduce
```

and it can also be run in batch using the **SUBMIT** command.

HELP File and Documentation

A short HELP file is available using the command:

```
$ MOREHELP APPLICATIONS REDUCE
```

and a 13-page user's guide is available when you enter:

```
$ TYPE ACSS$WRITEUP:REDUCE
```

continued on page 156

continued from page 155

or

```
$ EDIT /READ_ONLY ACSS$WRITEUP:REDUCE.LIS
```

To obtain a printed copy on the VAX, enter:

```
$ PRINT /CHAR=PSTD /NAME=xx.nnn ACSS$WRITEUP:REDUCE
```

where ***xx*** is the site code and ***nnn*** is the bin number where the output should be delivered. For example, **EA.425** is a bin number in 128B Lind Hall. (A laser printer copy of the user's guide is available for the price of photocopying in the Computing Information Center, 128A Lind Hall.) The user's guide explains how to get the 155-page *REDUCE User's Manual.* It also tells how to access seven self-paced instructional "lessons" that you can run interactively.

Tektronix 4010 Device Driver Removed

Michele Lewis
BITNET: GRAPHICS@UMNACVX

On June 30, we removed the Tektronix 4010 device driver from Precision Visuals Inc.'s graphics software. This will not affect your ability to display plots on Tektronix-type terminals. We also support the Tektronix 4014 driver, which is completely compatible with the 4010. You can access the 4014 driver for running with the packages DI-3000, Grafmaker, and the Contour System with the command:

```
$ setdrv t14
```

PicSure will access the 4014 driver with the command:

```
$ picsure t14
```

If you have any questions about the removal of the Tektronix 4010 device driver, please call Michele Lewis at 626-0314.

ACSS Short Courses

Summer Session 1988

MAINFRAME COURSES

Mainframe courses are free. To register, call 625-7397.
Some classes have size limitations: Register early.

		Introductory		
UNIX Brief, Operating System	July 18-21	MTh	2:30-5:00 pm	Free
		Advanced		
SIR (Data Base Management)	July 25-29	MWF	2:30-4:30 pm	Free
Beginning C Programming	July 26-Aug. 11	TTh	2:30-4:30 pm	Free

SHORT COURSE INFORMATION

Prerequisites: Please note any prerequisites for the class you are interested in. Instructors will not be able to review any prerequisite information. If you need more information on short courses, call the Computing Information Center at 625-7397.

Limits: All microcomputer classes and many mainframe classes have limits to class size. Please try to register early to be sure of getting a place. If you decide to withdraw from a class, please do so as soon as possible, so that we may make the space available to others.

Registration: Registration is located at the ACSS Computing Information Center, 128A Lind Hall (hours: 8:00 a.m. to 4:30 p.m., Monday through Friday). Fees must accompany registration. Mail registrations will be accepted. Deadline for registering is 4:00 p.m. on the last working day before the class begins. Please call and give us your name if you plan to attend a free class; also call to cancel if you decide later not to attend, so we know how many to expect. We reserve the right to cancel a course if registration is insufficient. For registration information call 625-7397.

SHORT COURSE POLICY

Each quarter ACSS offers a large number of free computer short courses for the University community. Although theses are usually introductory topics, the instructors spend a lot of time preparing for them. We ask people to register for these classes by phone, mail, or in person, so instructors know how many people to expect if materials are to be distributed. When registration for each class reaches classroom capacity we keep a waiting list of those wanting to attend. We always have waiting lists. If you do not show up for a class, or call to cancel, you are depriving those who wish to attend.

ACSS avoids overbooking classes so that instructors and class participants are not inconvenienced. However, if the the large number of "no-shows" continues, we will be forced to begin charging for these free classes and we will terminate telephone registrations. Please be considerate. Think of those staff who organize and teach ACSS short courses. And, in particular, think of those who are on the waiting list. If you aren't able to attend a class, a simple phone call (625-7397) will enable us to contact someone who can.

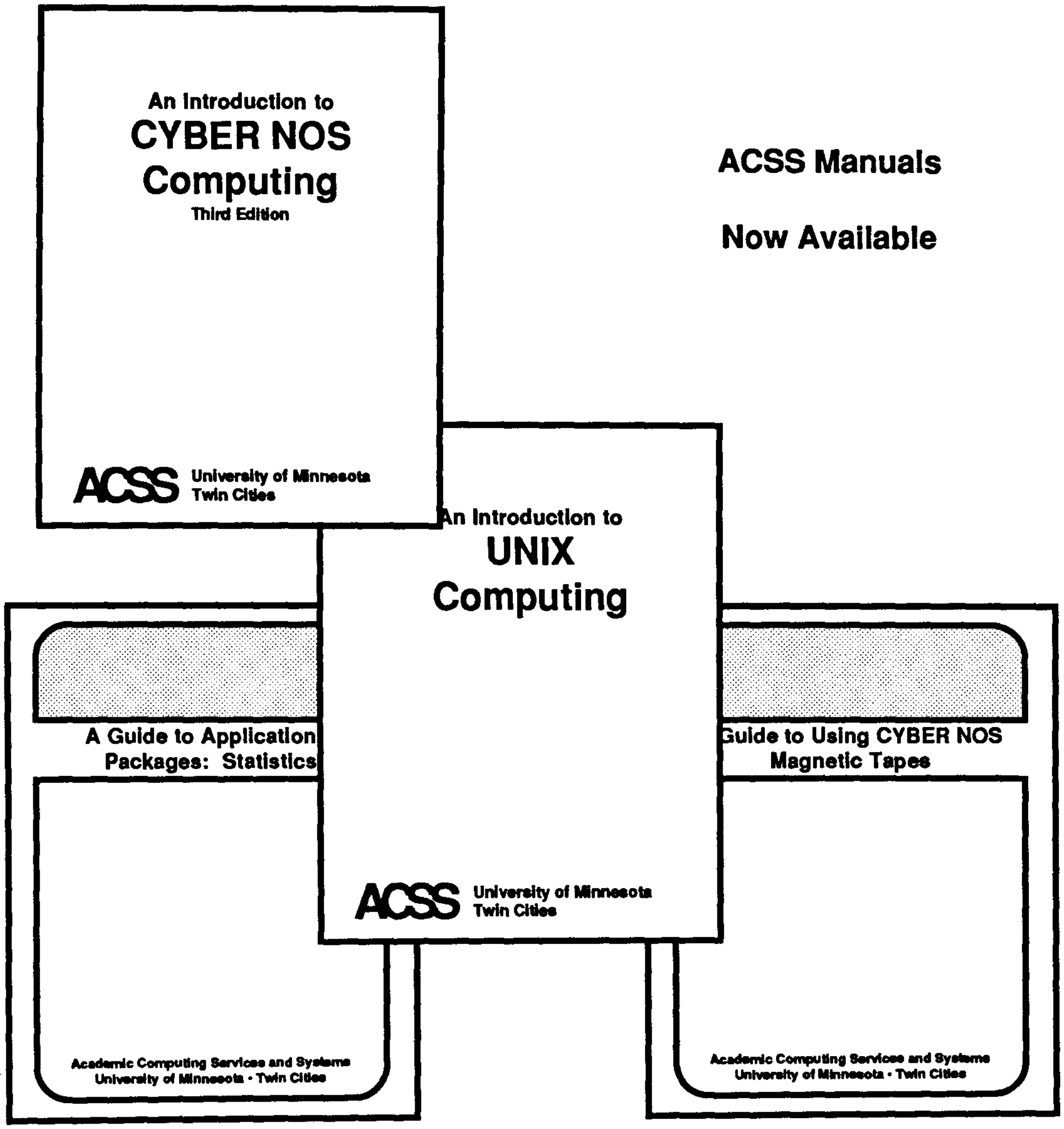
ACSS Manuals Available at University Bookstores

Paula Goblirsch
BITNET: PMG@UMNACVX

The Minnesota Book Center in Williamson Hall and H. D. Smith Bookstore on West Bank have these ACSS manuals in stock:

A Guide to Applications Packages: Statistics
A Guide to Using CYBER NOS Magnetic Tapes
An Introduction to CYBER NOS Computing
An Introduction to UNIX Computing

The Minnesota Book Center also carries other ACSS manuals. If you need an ACSS manual they don't have in stock, the bookstores will order it for you.


An Introduction to
CYBER NOS
Computing
Third Edition
ACSS University of Minnesota
Twin Cities
ACSS Manuals
Now Available
An Introduction to
UNIX
Computing
ACSS University of Minnesota
Twin Cities
A Guide to Application
Packages: Statistics
Academic Computing Services and Systems
University of Minnesota • Twin Cities
Guide to Using CYBER NOS
Magnetic Tapes
Academic Computing Services and Systems
University of Minnesota • Twin Cities

Phones/Hours/Labs

ACSS PHONE NUMBERS

Administrative Office: 626-1600

HELP-Line 626-5592

Access:	
CYBER (CA)	626-1620
CYBER (NV)	626-1622
ENCORE (UX)	626-1681
VAX (VX)	626-1641
RJE (2400 baud)	626-1656
RJE (4800 baud)	626-1663
Accounts:	
CYBERs, ENCORE, VAX	625-1511
Computer Hours (recorded message)	626-1819
Computing Information Center	625-7397
Contract Services	625-2303
East Bank I/O, 128B Lind Hall	625-5082
Engineering Services	627-4357
Equipment Maintenance/Repair	627-4357
Graphics Software	626-5592
Information, Lauderdale	626-1600
Lauderdale Computer Room	626-0550
Lauderdale Services	626-1838
LUMINA	626-2206
LUMINA Consultant	626-2272
Magnetic Tape Librarian	626-1838
Math and Engineering Packages	625-5830
Microcomputer HelpLine	626-4276
Newsletter Subscription	625-7397
Permanent File Restoration	626-0595
Project Assist	626-1090
Public Labs (Managed by ACSS)	
170 Anderson Hall	624-6526
140 Blegen Hall	624-5278
14 Folwell Hall	625-4896
306B Lind Hall	625-9032
32 OMWL	624-6526
130 Physics	625-6820
9 Walter Library	626-1899
Lab Manager (14 Folwell Hall)	625-7850
Publications Information	626-1093
Short Course Registration	625-7397
Shuttle Bus Service	625-9525
System Status (recorded message)	626-1819
West Bank Computing Services	624-0877

For the phone numbers of consulting services see the Help Page.

PUBLIC LABS
TWIN CITIES CAMPUS

Location	Mainframe Printing	Interactive	Micro
East Bank			
ApH 204	Closed for remodeling		
Arch 148			X
CenH		X	
ComH		X	
DiehlH 207, 270	2	X	X
EddyH Annex 54			X
EltH 121, 124	1	X	
FolH 14, 14a	1,2	X	X
FronH		X	
LindH 26		X	
LindH 128B	2,3	X	
LindH 306B			X
MasCanCtr M39		X	
MechE 308		X	
MoosT 8-425			X
Nich 109			X
Phys 130	2	X	X
PioH		X	
SanH		X	
TerrH		X	
VinH 4		X	
VinH 203			X
WaLib 9	2	X	X
West Bank			
AndH 170	2,3		X
BlegH 140	1	X	
MdbH		X	
OMWL B2			X
St. Paul			
BaH		X	
CentLib B50			X
ClaOff 125E	1	X	
ClaOff B22			X
McNH 69	2		X
Vet 43b			X

1 - Dot-matrix line printers allow printing from CYBER CA only.
2 - Laser printing available from ENCORE UX and VAX VX.
3 - Laser printing available from CYBER CA.

SYSTEM OPERATING HOURS

	CYBER (CA), VAX (VX)	ENCORE (UX)	CYBER (NV)	Low Rate
M-F	7 am - 4 am	7 am - 4 am	7 am - 1 am	8 pm - 4 am, 7 am - 8 am
Sat	4 am - midnight	4 am - 9 pm	7 am - midnight	all operating hours
Sun	6 pm - 4 am	6 pm - 4 am	6 pm - 1 am	all operating hours

Contents

Communications

Statistics

Artificial Intelligence

Consulting Corner

ENCORE News

Math and Engineering Packages

Graphics

Short Courses

Documentation News

The ACSS Newsletter
July 1988
Volume 22, Number 7

Acting Director of ACSS: *Michael M. Skow*
Editors: *Steven Brehe, Paula Goblirsch*

The *ACSS Newsletter* is published monthly by Academic Computing Services and Systems of the University of Minnesota, Twin Cities. Deadline for articles is the 10th of the month preceding publication; deadline for short announcements is the 15th. The *Newsletter* is produced with an Apple Macintosh Plus running Microsoft Word, FullPaint, MacDraw, Adobe Illustrator, and Aldus Pagemaker software, with camera-ready copy produced on the Apple LaserWriter Plus.

Direct comments, suggestions, articles, and announcements to the editors at the address below, or call (612) 626-1828 or 626-1093. For a free subscription call (612) 625-7397, or send your name and address to the Computing Information Center, 128A Lind Hall. Electronic Mail: YZE6075@UMNACCA, YZE6075UMNACUX, or MAD@UMNACVX. On-campus address changes *must* include your department's name and your *departmental* address.

The University of Minnesota is committed to the policy that all persons shall have equal access to its programs, facilities, and employment without regard to race, religion, color, sex, national origin, handicap, age, veteran status, or sexual orientation.

The ACSS
Newsletter

**Academic
Computing
Services and
Systems**

Technical Publications
100 Lauderdale CF
University of Minnesota
2520 Broadway Drive
Lauderdale, Minnesota 55113