The ACSS Newsletter

University of Minnesota
Twin Cities
November 1986

Statistics Packages

SAS on the VX

Bruce A. Center and Deb Felt

SAS (Statistical Analysis System) has long been the mainstay for statistical programming on IBM mainframes. It is comparable in power and scope to SPSS^X. Users who are experienced with SAS generally prefer it for its power, flexibility, file manipulation ability, and its unique user environment. SPSS aficionados, particularly on non-IBM systems, have been known to find SAS verbose and cumbersome. In any case, we have now installed SAS on our VAX VX, under the VMS operating system, to permit you to use it and arrive at your own conclusions.

How to Use SAS

SAS can be invoked in several ways: in interactive, batch, display-manager, and line-prompt mode, as show in the following sections.

Interactive SAS

The form is:

```
$ SAS filename
```

where ***filename*** contains SAS instructions. The output is placed in ***filename*.LIS**; the SAS logfile is placed in ***filename*.LOG** in your current directory. The data can be mixed in with the instructions or on an external file referenced with the SAS DATA command.

For example, if the file **EDUCAT.DAT** contains the last name, sex, age, height, and weight of a number of children, the SAS instructions in your SAS instruction file called **EDUCAT.SAS** might read:

```
FILENAME EDUCAT 'EDUCAT.DAT';
DATA TRY1;
        INFILE EDUCAT;
```

Continued on page 230

Continued from page 229

```
          INPUT NAME $ SEX AGE HEIGHT WEIGHT;
          YEARS = AGE/12.;
PROC PRINT;
PROC PLOT; PLOT WEIGHT * HEIGHT = SEX;
```

This would read the data from **EDUCAT.DAT,** compute age in years, print out the data, and plot weight by height separately for boys and girls.

To invoke SAS, type:

```
$ SAS EDUCAT.SAS
```

The output would go to **EDUCAT.LIS**, the SAS logfile to **EDUCAT.LOG.** You could then scan the listing (using EDT, for example), correct any errors (you surely didn't think this author would provide you with a working example) and then print the output. For instance:

```
$ PRINT   EDUCAT/NAME=site*bin
```

Batch SAS

Since large SAS jobs can run for substantial amounts of time and at low-rate times, you may wish to run SAS in the background as a **SUBMIT** job. To invoke SAS for batch execution, submit a command procedure that controls execution of the program and invokes SAS. In this case, you don't type in the SAS command at the terminal, as in interactive SAS. For example, to **SUBMIT** the example above for batch execution, you create a simple command file that looks like this:

```
SAS EDUCAT.SAS
PRINT EDUCAT.LIS/NAME=site*bin
```

If we called this two-line file **EDUCAT.COM,** then, to **SUBMIT** the job and have it run during VX low-rate hours, type:

```
$ SUBMIT EDUCAT.COM/AFTER=20:00
```

If we wanted VMS to tell us when the job had finished, we could use the **NOTIFY** qualifier:

```
$ SUBMIT/NOTIFY   EDUCAT.COM
```

This job produces three files: the VMS log, the SAS log, and the SAS procedure output file. The VMS log contains information on VMS commands you entered (or processed in your submit job), error messages for VMS commands, and processing statistics such as CPU time used. The VMS log goes to the printer by default.

The SAS log and the SAS procedure output file are written to files **EDUCAT.LOG** and **EDUCAT.LIS** respectively, as before. The **PRINT** command prints the output file and routes it to a specified site and bin.

SAS in Display-Manager Mode

SAS contains a full-screen, two-window editor called Display Manager, which enables the user to enter SAS instructions, execute SAS, and see the results on the log screen. If the PROC step produces printed results, Display Manager shows these results in the Procedure output window. Display Manager makes it virtually unnecessary to leave the SAS environment until all SAS runs are successfully completed.

To invoke Display Manager, type:

```
   $ SAS/FSD=terminal type
```

Replace **terminal type** with a specific type of terminal–i.e., VT100.

You can use Display Manager to enter the SAS instructions from the above example and **SUBMIT** it to SAS. Display Manager can also **SAVE** a set of SAS instructions in an external file, or **INCLUDE** an external file that already contains SAS commands.

Information on Display Manager is in the *SAS User's Guide: Basics*, Chapter 14 and Appendix 3, and in the *SAS Companion for the VMS Operating System*, chapter 7. These references are not tutorials, and require some experience with the VMS operating system and SAS.

Line-Prompt Mode

This is invoked by typing:

```
   $ SAS
```

SAS will prompt you with a line number to enter the SAS program line by line. The program executes one step at a time, as it receives the statements. Errors are noted immediately at the terminal, and are not executed. Line prompt mode is terminated by typing **ENDSAS;**. We do not recommend using line prompt mode for any but the most trivial SAS applications.

Documentation

SAS is blessed with a wide variety of manuals, many of which will tell you something about what you need to run SAS on VX. These may be available at the Williamson Bookstore, or can be ordered directly from SAS. (You can get help in putting together your order from the ACSS Computing Information Center at 128A Lind Hall.) Among the more useful manuals are:

- *SAS Users Guide: Basics,* Version 5, 1985 (1290 pages).
- *SAS Users Guide: Statistics,* Version 5, 1985 (956 pages).
- *SAS Introductory Guide:* 3rd Edition, 1985 (99 pages).
- *SAS Companion for the VMS Operating System,* 1986 Edition, 1986 (135 pages) .
- *SAS/GRAPH Users Guide*: Version 5, 1985 (596 pages).

All of these manuals are published by SAS Inc., Box 8000, Cary, North Carolina.

The Dark Side of Computing: Health Hazards and Toxic Waste

Lawrence Liddiard

Reading over my past articles, I detect a positive attitude toward computing. My articles about the Macintosh, Seymour Cray, ACSS rates, MIPS/dollar, computing in the twenty-first century, and even Space Wars were all intended as upbeat information. But for several months I have noticed articles in the press that showed a dark side to the manufacture, use, and disposal of computers and computer supplies. Since I like computing as an occupation, I would not be pleased to discover that the manufacture and use of computers and their supplies generated vast amounts of toxic waste, environmental pollutants, and other health hazards. This would imply that good computing could only be built upon bad (i.e., dangerous) foundations.

The Manufacture

To learn if the local computer and circuit board companies disposed of hazardous waste products properly, I called the Minnesota Pollution Control Agency. A spokesperson there stated that Minnesota computer companies were very good about properly recycling and/or shipping to authorized disposal plants the cyanide and heavy metal solutions produced in circuit board manufacture. This also included excess paint and lead solder used in computers.

Computer Terminals

All of us in computing have been pleased to stop using the noisy hard-copy terminals (especially those TTY 33's) that were the basic input and output devices of early computers. Since much of the paper generated by those terminals was thrown away, we should be saving several forests and sparing some streams additional pollution from paper mills. (We are also preserving our hearing ability.) The replacement for the hard-copy terminal was the hardwired or micro-based Video Display Terminal (VDT), alias CRT (for Cathode Ray Tube).

But there have been reports from several industrialized nations that women who use VDTs have experienced reproductive problems. Consider this quote from an Associated Press report on a University of Michigan survey of 728 state employees: "The results of our study make it very unlikely that VDTs have a large effect on the outcomes of pregnancies," Professor William Butler said. "But we cannot rule out the chance that working with VDTs 21 or more hours a week will increase the risk of miscarriages by 5 percent." (Quoted in the Minneapolis *Star and Tribune* on October 1, 1986.)

The University of Minnesota Job Evaluation Questionnaire (of July 1, 1986) has two remarks dealing specifically with CRTs. These occur in the Physical Effort category: "For example, you would count eye strain from using your eyes to read a computer terminal screen." And the following remark appears in the Risk Factors category: "Working at computer terminals

(CRT's) does NOT count as a risk because scientific studies do not currently support any other conclusion."

Thus it appears that the jury is still out on the safety of VDTs. Our eyes may strain and smart, but the scientists suggest that we should live many a day. Meanwhile, there are benefits as well as risks: Still trying to avoid bifocals, I compose these articles on my Macintosh in 12-point Geneva, before shipping them out to the Publications group in harder-to-read Times Roman.

Chips, Tapes, and Chemicals

Currently most of the components of a computer are made of common elements. The silicon wafer chip (sand) and iron oxide (rust) for disk surfaces and magnetic tape recording are biodegradable when the computer component is disposed of. Recently, in the search for a chip material with faster switching speed, some computer manufacturers have switched to gallium arsenide (GaAs) as an effective replacement for silicon. In addition, GaAs chips consume less power and have greater resistance to radiation. But GaAs chip manufacture presents more serious environmental problems than silicon because of the use of arsine, a poisonous gas.

There is more bad news. Recently Donna Raimondi quoted (in her "Hard Talk" columns in the June 9 and July 14, 1986 *ComputerWorld*) a letter from William Manly, a physicist warning about dangers from chromium dioxide-based cartridge tapes that would poison the environment and had reduced life compared to iron oxide tape.

Manley asserted that "the chromium dioxide used in IBM's 3480 and other vendors' cartridge tapes degrades in the presence of moisture and oxygen to chromium hexavalent compounds that are highly poisonous." He concluded: "The bottom line is that tape makers that use chromium oxide material for coating tapes have to add so much doping to minimize the poisonous nature of the material that they reduce the life of the substance."

Representatives of both IBM and Du Pont disagreed. "The chromium dioxide tapes are suitable for archival storage and safe for handling and use by our customers," claimed IBM. Du Pont's Dr. Bruce Karrh, VP of safety, health, and environmental affairs stated: "About the only way you could be exposed to an unacceptable amount of chromium dioxide is in the manufacturing process." In answer to Manly's assertion that chromium dioxide tapes would have to be disposed of as a toxic waste in a government-approved fashion, Karrh remarked: "Tapes that use chromium dioxide can be disposed of in the same manner as any other tape."

Computer Graveyard?

With some of the old ironware computer cabinets you might think that they could be recycled as boat anchors; certainly a ship on Lake Superior that drifted onto a sand bar during a storm last fall could have used a couple. In reading the computer press literature, I enjoy those short news releases describing a museum's rescue of a significant ancient computer (i.e., more than 20 years old or the first of its kind) from a dump or a government agency. Recently, for example, Professor John A. N. Lee, having found a person who knew where the machine was buried, dug up a MANIAC-3 from a ravine.

Continued on page 234

Continued from page 233

But, on reflection, where is the proper place to dispose of computers that are out-of-date, have met an untimely death, or cost more to maintain than a new version? Most manufacturers move out-of-date systems by fire sales or as gifts to educational institutions, rather than shipping it to a reclamation center. That leaves the customer to dispose of it properly or to judiciously offer it as a trade-in on the next system. In America the current methods to insure proper disposal of products are: to have a recycle rebate associated with the purchase (i.e., the aluminum can and the radial tire), to make the manufacturer label the product with disposal directions (the aerosol can), to have heavy fines for not correcting the problem (asbestos building material), and/or to offer government support to aid the owner (as is done for trees with Dutch elm disease). To this we may always add the actuality or threat of litigation. I believe that computer makers will eventually be forced to respond to the need for proper disposal of their machines.

As an example of a local computer company's concern with environment, I notice that the Minnesota Waste Management Board awarded a grant of $30,000 to Control Data to reduce further the volume of hazardous waste it generates. The October 12, 1986 Minneapolis *Star and Tribune* reported that "Control Data will use its grant to study a process for removing lead from scrap computer circuit boards. By removing the lead, a potential environment threat, the scrap can be disposed of more safely."

Gold Mining

Sometimes it pays in gold to recycle. Those who have gone through the Homestake Mine in Lead, South Dakota, know that less than an ounce of gold is obtained from a ton of ore in that mine. But several early computers held more gold than a ton of ore. When gold was relatively cheap, it was selected as plating for connectors and other critical parts of computers. For example, the old UNIVAC 1004 computers that ACSS used for Remote Job Entry stations had 1/2 to 1 1/2 ounces of gold plating, depending on the date of manufacture. The early CDC 6000's and CYBER 74's had up to 18 ounces of gold in their various components. This gold content allowed ACSS to receive $1000 each for the 30 UNIVAC 1004's when they were retired, which was slightly more than we paid for the last several of them. I am sorry to report that, when gold rose to over $800 per ounce, most manufacturers found ways to eliminate gold from their computers. (Thanks to Ed Edmundson of our staff for the information on the UNIVAC 1004 and to Ed Bontrager of Control Data for information on the CDC 6000.)

Conclusion

I have to conclude again on a note of optimism: The dark side of computing, concerning public and personal safety, is not as dark as we might be led to believe, although there is certainly room for improvement.

In past years we often have sought proper waste disposal methods only after the environment was already damaged by some toxic and long-lived substance. Computing is young and clever enough to correct past mistakes and avoid more serious mistakes in the future. I think it will.

CLEAN77 on the VAX 8600

Tingli Pan and Michael Frisch

CLEAN77 (version 2.20) is now available on the VAX 8600. CLEAN77 is designed to reformat programs written in FORTRAN so that the cleaned programs adhere as closely as possible to FORTRAN 77 standards (ANSI X3.9-1978). CLEAN77 will perform such services as resequencing statement numbers, converting Hollerith fields to quoted fields, indenting the contents of DO-loop, IF-THEN-ELSE, and WHILE blocks, and moving FORMAT statements to the end of the routine. Some of these and other actions are user controllable via commands. The details of those commands can be found in a WRITEUP on the CA and MD machines by typing

```
WRITEUP,CLEAN77.
```

The command to run CLEAN77 under VMS is:

```
$ CLEAN77 [/L*IST=lfile] [/C*OMMAND=cfile] pfile [nfile]
```

Where

File ***pfile*** contains the FORTRAN routines to be cleaned. If the file type is not given, then the file type is assumed to be **.FOR.**

File ***nfile*** contains the new routines resulting from the cleaning process. If this is not present then **pfile.CLN** is used.

File ***lfile*** contains the list output from CLEAN77 if any is produced. If this parameter is not present then **pfile.LIS** is used.

File ***cfile,*** if present, contains any commands to be processed prior to reading from the input file. If the file type is not given, then the file type is assumed to be **.DAT**. If this parameter is not present, no command file is used.

As an example of the above,

```
CLEAN77  SAM
```

is the same as

```
CLEAN77/L=SAM.LIS  SAM.FOR SAM.CLN
```

The parameters can be in almost any order, but ***pfile*** must appear before ***nfile.***

Note: CLEAN77 does not work with lowercase and tab characters.

Tools for Using VMS Magnetic Tapes: Part I

Bryan Senn

For many people, the most frustrating aspect of using a computer is trying to read and write files on magnetic tapes. In the University environment this is especially true. The wide variety of computers and operating systems, each with their own unique way of handling tapes, and the need to read and write tapes for colleagues at other universities using tapes with "foreign" or unknown formats may make the task seem impossible to the average user.

This is the first of two articles on using VMS magnetic tapes, in which we hope to clarify some issues of tape use with explanations, examples, and advice.

This is not a subject for the new user. To make good use of this article, you will require some previous knowledge of the VMS system, including an acquaintance with VMS command formats, file names, and directory structure.

In this first article, we will discuss these topics:

Using common VMS tape formats.
Deciphering "foreign" tape formats.
Preparing to use a tape (including mounting, labeling, and dismounting a tape).
Using some common commands.
Manipulating backup tapes and file-structured tapes.

VAX/VMS Tape Formats

There are a variety of tape formats available on the VAX, each having its advantages and disadvantages. The most common formats are discussed below. Which format you use depends on what you intend to do with the tape. If the tape is to be used exclusively on the VAX to backup files, to store seldom used files, or to exchange files with another VMS system, you have more possibilities for tape format. If, however, you intend to use this tape on a non-VMS system, your options are limited.

BACKUP Tape: This is a tape that has been created using the VMS BACKUP utility. BACKUP allows you to save entire directory trees on a magnetic tape and retain all the owner and protection information about files. The backup format is typically more compact than a file-structured tape created using the VMS COPY command. The more compact format of the backup tape allows you to store more files on one tape.

The BACKUP utility also allows you to use multiple-reel tapes and performs *checksumming* to verify the integrity of the tape (i.e., to verify the accuracy of the copy). There are disadvantages to using BACKUP: It is difficult to read a backup tape in any sensible way on a non-VMS system, and the utility cannot add files to a backup tape without writing a new saveset. (A saveset is a group of files written on a tape at one time.)

File-Structured Tape: A file-structured tape is an ANSI-labeled tape created using the VMS `COPY` command. (A label is a piece of identifying information written on a magnetic tape. An ANSI label conforms to the requirements of the American National Standards Institute.) A file-structured tape will preserve the names of files, but not their protection information, modification and backup dates, creation time, owner, or directory structure. In contrast to backup tapes, you can easily add additional files to a file-structured tape. You can also copy files selectively by specifying the files names. It is, however, impossible to remove a file from a file-structured tape without rewriting the entire tape. Also, it may be difficult to read file-structured tapes on a non-VMS system.

Non-File Structured, Block Format Tapes: A non-file structured, block format tape, hereafter called a blocked tape, is created using the VMS `COPY` command on a tape which has been mounted as a foreign volume. The blocked tape is far and above the most widely supported tape format. It is the recommended format for exchange of information with non-VMS systems, including the ACSS CYBER machines.

Blocked tapes will not retain any information about a file, even the file name. The user of a blocked format tape is responsible for keeping track of what files are on the tape and what order they are in. You will also need to keep track of other information about the tape, such as the blocking factor and record size (terms we will discuss later). You can easily add files to blocked tapes, provided you keep an accurate record of the tape contents. Blocked tapes should not be used for binary files.

Reading Foreign Tapes on VMS

Some tape formats that are difficult to read on a VMS system include:

UNIX TAR format tapes.
CYBER ARCHIVE tapes.
CYBER tapes not written with the TBLOCK or COPYCH utilities.
Tapes written using IBM's EBCDIC character set.
CYBER labeled tapes. (This is possible, but not recommended.)

Tapes in these formats may require you to write a special program to read the tape or, in the case of a CYBER-labeled tape, work from a privileged account. (It is possible to label a CYBER tape in such a way as to be usable under VMS; however, the default label written by the CYBER `BLANK` command makes a tape inaccessible to a non-privileged VMS user.) ACSS's Contract Services Group (625-2303) is available to help if you do not wish to write the program yourself. You can also call our HELP-Line for more information at 626-5592.

In sending tapes to or receiving tapes from another facility, you can avoid problems by careful preparation. You should begin by discussing the tape exchange in advance with our HELP-Line staff and someone at the other computing facility. You should also consult the HELP-Line in advance if you are planning to move information from the VX to another ACSS machine, or from another machine to the VX, via tape.

Preparing to Use a Tape

Before you can read or write a tape in VMS you must first tell the system some things about the tape. If the tape is new and you wish it to be an ANSI-labeled tape, you must also

Continued on page 238

Continued from page 237

initialize the tape with a magnetic tape label. The **GETTAPE** command (which ACSS added to the VMS system) will get a tape physically mounted on a tape drive. Type **HELP GETTAPE** and **MOREHELP TAPES REQUEST** for more information on the **GETTAPE** command.

After you've used **GETTAPE**, VMS knows only that you have a tape drive allocated to your process. To use the tape, you must now tell VMS something about how it should access the tape. To do this you must use the VMS **MOUNT** command. The command name **MOUNT** can be quite confusing because we often refer to running **GETTAPE** as "having the tape mounted." It is true that **GETTAPE** accomplishes the *physical* mount of a tape on a tape drive—the **MOUNT** command, however, accomplishes the *logical* mount of the tape within the VMS system, defining the tape for the system.

The MOUNT Command

Now we will look at the general form of the **MOUNT** command and the parameters that are independent of the tape format. Later, when we discuss how to write tapes, we will give you the specific forms of the **MOUNT** command necessary for each format.

The general form of **MOUNT** is:

```
$ MOUNT device label logical_name
```

In this command, the parameter ***device*** can be either the logical name given the tape on the **GETTAPE** command or a specific device name, like MUA1. The parameter **label** is the tape name as written in an ANSI label. (This field may be omitted if the tape is not ANSI-labeled.) The parameter ***logical_name*** is NOT the logical name as specified on the **GETTAPE** command, rather it is another opportunity to associate a different logical name as a synonym to the device name.

The qualifier **/FOREIGN** is used on the **MOUNT** command to tell VMS that the tape is a foreign volume; it may or may not be ANSI-labeled. If you don't use the **/FOREIGN** qualifier, the tape must be an ANSI-labeled tape.

Also, with the **MOUNT** command you can specify for the system the density at which to write the tape. The qualifier is **/DENSITY=*n*** where ***n*** is either 1600 or 6250 BPI (bits per inch). (Tapes with a density of 800 BPI cannot be read or written on the VAX 8600; neither can seven-track tapes.) If you do not specify a density, by default the system uses 6250 BPI, unless the tape was previously written; then the density defaults to that of the first record on the volume.

By default **MOUNT** attempts to mount a tape as *write-enabled* (i.e., the system may write information on the tape). If you did not use the **/WRITE** qualifier on the **GETTAPE** command, you will get the following informational message from **MOUNT**:

```
%MOUNT-I-WRITELOCK, volume is write locked
```

The **/NOWRITE** qualifier on the **MOUNT** command has two purposes. If the **/WRITE** qualifier was not used on the **GETTAPE** command, then **MOUNT/NOWRITE** will suppress the above message. If **/WRITE** was used with the **GETTAPE** command, then

MOUNT/NOWRITE will serve to "lock" the volume (prevent writing on it) for the remainder of the job.

The INITIALIZE Command

If the tape is new or blank from having been tested, and you want to write an ANSI label on it, you will have to use the VMS **INITIALIZE** command to write an ANSI label onto the tape before issuing the **MOUNT** command. The command has the form:

```
$ INITIALIZE/DENSITY=n device label
```

where ***device*** is the tape drive name or logical name. The parameter ***label*** is 1 to 6 uppercase alphanumeric characters that will be written to the tape volume name field of the label. Once written, this label will be needed any time the tape is accessed, so it is wise to make it something easy to remember. The value ***n*** is the density you wish the tape to have, 1600 or 6250. We recommend that you always specify the density when you are initializing a tape.

Dismounting Tapes

Under normal circumstances it is not necessary to use the VMS **DISMOUNT** command, as **DROPTAPE** will do a dismount for you. (While on the VX, type **HELP DROPTAPE** and **MOREHELP TAPES DROPTAPE** for more information.) If, however, you need to logically dismount the tape to try mounting it a different way, the command to use is:

```
$ DISMOUNT/NOUNLOAD device
```

By default, **DISMOUNT** will take the tape drive off line and it would take another **GETTAPE** command to get it back on line. The **/NOUNLOAD** qualifier specifies that the logical mount is to be removed, but the tape will be left on line thus allowing you to issue another **MOUNT** command. This command should be used very carefully. If you were to **DISMOUNT/NOUNLOAD** and then terminate your job without doing a **DROPTAPE**, the potential exists for someone else to allocate the drive and logically mount and use your tape.

Manipulating Backup Tapes

To use a backup format tape, the tape must be initialized using the VMS **INITIALIZE** command and be mounted as a foreign volume. The following form of the **MOUNT** command should be used:

```
$ MOUNT/FOREIGN/DENSITY=n device
```

where ***device*** is a tape drive name or logical name. The **BACKUP** command is then used to write the tape.

The following sample procedure uses **BACKUP** to backup an entire directory tree starting at the VMS directory [MYFILES] and including all files and subdirectories of [MYFILES] and all subdirectories' files on to a tape with VSN of TT1234 and an ID of MYTAPE in a saveset named MYSAVSET.BCK :

Continued on page 240

Continued from page 239

```
$ GETTAPE/WRITE  TT1234  MYTAPE  TUNIT
$! If the tape is already ANSI labeled omit the following line
$ INITIALIZE TUNIT BCKUP
$ MOUNT/FOREIGN/DENSITY=1600  TUNIT
$ BACKUP  [MYFILES...]  TAPE:MYSAVSET.BCK
$ EXIT
```

If you later wish to backup more files onto another saveset on the same tape it will be necessary to skip beyond any savesets you wish to keep before issuing the **BACKUP** command. The command to skip tape files is:

```
$ SET MAGTAPE/SKIP=FILES:n device
```

Where *n* is the number of savesets you wish to skip over. If you start the backup operation before issuing this command, you will overwrite the saveset MYSAVSET.BCK. The default for **BACKUP** is **/NOREWIND.**

To find out what files are on a backup format tape use the **/LIST** qualifier:

```
$ BACKUP/LIST MUA1:MYSAVSET.BCK
```

The above command tells **BACKUP** to search the tape mounted on MUA1 for a backup saveset called MYSAVSET.BCK and report the file names.

To get a list of all savesets on a backup tape type:

```
$ BACKUP/REWIND/LIST MUA1:*.*
```

BACKUP will first rewind the tape to ensure that it starts processing for the beginning of the tape. **BACKUP** will then list all files in all savesets.

To read all the files from a backup format tape, it is easiest to give the same command you used to create the tape with the parameters reversed. From the first example:

```
$ BACKUP TUNIT:MYSAVSET.BCK [MYFILES...]
```

will restore all files from MYSAVSET.BCK to its original directory tree. You may also restore selected files using the **/SELECT** qualifier. For more information and **BACKUP** and its qualifiers see the on-line document HELP BACKUP or the *VAX/VMS Backup Utility Reference Manual* (AA-Z407A-TE), available for reference in our Computing Information Center, 128A Lind Hall.

Occasionally **BACKUP** may have trouble finding a particular saveset by name, especially if two or more savesets have the same name. When this happens, first use the **BACKUP** command, which lists all files of all savesets and keep track of which position the needed saveset has on the tape (e.g., the fifth saveset). Then use the following two commands to position the tape before issuing the **BACKUP** command:

```
$ SET MAGTAPE/REWIND device
$ SET MAGTAPE/SKIP=FILES:4 device
```

This will position the tape at the beginning of the fifth saveset on the tape.

It is sometimes necessary to create a duplicate of a backup format tape for sending to another VMS site, for additional backup, etc. The following procedure will create an exact duplicate of a backup tape:

```
$ GETTAPE  TTOLD  BCKUPTAPE  OLD
$ GETTAPE/WRITE  TTOLD  TEMPTAPE  NEW
$ INITIALIZE/DENSITY=6250  NEW  BCKTP
$! Where BCKTP is same as the magnetic label on TTOLD
$ MOUNT/NOHDR3/BLOCK=8192/NOWRITE  OLD  BCKTP
$ MOUNT/NOHDR3/BLOCK=8192/DENSITY=6250  NEW  BCKTP
$ COPY  OLD:*.*  NEW:*.*
$ EXIT
```

The **MOUNT** qualifier **/NOHDR3** tells VMS not to process HDR3 labels. HDR3 labels are file labels. For more information on label types consult the *Guide to VAX/VMS Disk and Magnetic Tape Operations* (AA-Y506A-TE), available for reference in our Computing Information Center, 128A Lind Hall. Since the VMS system will not try to process or interpret HDR3 labels, the **COPY** command will treat them as files and copy them to the new tape, creating a functional equivalent of the old backup tape.

Manipulating File-Structured Tapes

To write a file-structured tape, the tape must be ANSI-labeled. If it is not already labeled, you need to use an **INITIALIZE** command before the **MOUNT** command.

The VMS **COPY** command is used to write files to a file-structured tape. The files you wish to copy may be specified using the standard rules for wild-card characters. The output specification should begin with the magnetic tape device name or logical name. The following procedure will initialize a tape and copy all files with a type of .DAT from your default directory to the file structure tape:

```
$ GETTAPE/WRITE  TT1111  SECRET  FILETAPE
$ INITIALIZE/DENSITY=1600  FILETAPE  FILETP
$ MOUNT  FILETAPE  FILETP
$ COPY  *.DAT  FILETAPE:*.*
$ EXIT
```

You can add files to a file structure tape at any time without worrying about the position of the tape because the files will always be added to the end of the tape. Take care that no files with the same name type and version are written to the tape because the second one will not be readable using standard methods.

To see what files are currently on a file-structured tape, use the VMS **DIRECTORY** command:

```
$ DIRECTORY device:
```

The files will be listed in the order they are on the tape, not in alphabetical order.

The following procedure will read a single file from a file-structured tape. This example copies the tape file LOGIN.COM to the user's default directory in a file called XLOGIN.COM :

Continued on page 242

Continued from page 241

```
$ GETTAPE  TT1111  SECRET  FILETAPE
$ MOUNT  FILETAPE  FILETP
$ COPY  FILETAPE:LOGIN.COM  XLOGIN.COM
$ EXIT
```

To copy all files from the tape to their original names use:

```
$ COPY FILETAPE:*.* *.*
```

Next Month

In Part II of this article we will resume with a section on manipulating block-formatted tapes, and also explain how to use the **BLOCK** and **UNBLOCK** utilities and how to determine formats of foreign tapes. Finally, we will describe some common problems in magnetic tape use and explain how to avoid them.

VAX FORTRAN Libraries

Michael Frisch

We have recently installed several libraries on the VAX 8600 (VX) under the VMS operating system. The library names and their descriptions are:

BESPACK - Bessel and Airy functions library
BSPLINE - B-spline library
EISPACK - Eigenvalue/eigenvector library
FUNPACK - Special function library
GRG2 - Constrained nonlinear optimization library
IMSL - International Mathematical and Statistical Library
LINPACK - Linear equations library
LSODE - Ordinary differential equations library
LSODI - Ordinary differential equations – implicit form – library
MINPACK - Nonlinear equations and nonlinear least squares library
YSMPLIB - Yale Sparse Matrix Library

For details on the contents of these libraries, see ACSS's *Guide to Subprogram Libraries,* which you can purchase at the Minnesota Book Center's Electronics Desk or consult at the Computing Information Center, 128A Lind Hall.

All of these are double precision versions compiled with the **/G_FLOATING** option of the FORTRAN compiler. This option gives nearly the same exponent range and slightly better precision than CYBER single precision.

/G_FLOATING is 52 bits of coefficient – about 15 digits – and values in the range 0.5*10**(-308) through 0.9*10*(+308) versus the CYBERs' 48 bits of coefficient – about 14 decimal digits – and values in the range 10**(-292) through 10*(+322). Thus, all user programs that call these libraries must use double precision for floating point constants, variables, and functions declared as DOUBLE PRECISION or REAL*8 (DOUBLE PRECISION COMPLEX or

COMPLEX*16 for complex items). Such programs must also be compiled with the **/G_FLOATING** option of the FORTRAN compiler.

The method for accessing the libraries is the new **LIBS** command described elsewhere in this issue. It should be used once at the beginning of each time-sharing session or batch submission. The library names for the **LIBS** command are the ones listed in the table above.

A complete example of a use of IMSL is:

```
$ TYPE SAM.FOR
      DOUBLE PRECISION DGAMMA
      PRINT *,DGAMMA(5.0D0)
      END
$ LIBS IMSL
** 1 LIBRARIES ADDED. **
$ FORTRAN/G_FLOATING SAM
$ LINK SAM
$ RUN SAM
    24.0000000000000
```

In the near future, we plan to add to VX the rest of the libraries that are on CA. Watch this newsletter for the announcement of their availability.

IMPORTANT NOTE: The IMSL library is not completely double precision (or double precision complex). In the WRITEUPs for the IMSL routines, the PRECISION/HARDWARE section for H32 machines will say SINGLE/ALL for routines that have single precision (or single precision complex) parameters and results. Those that say SINGLE/DOUBLE have double precision (or double precision complex) parameters and results.

User Libraries on VMS: New LIBS Command

Michael Frisch

The VMS operating system doesn't have a simple command to locate a system FORTRAN application library such as IMSL and identify it as a library for the LINK command. ACSS has written a new **LIBS** command to do this and some related tasks as well.

VMS has a table for libraries used with LINK called LNK$LIBRARY. It can contain up to 9 slots. During the linking operation, LINK starts with the first library and stops when it reaches an empty slot in the table.

The **LIBS** command adds libraries to the table in a controlled way: The next available slot is used (unless the identical name is already in a slot in which case it is ignored). For system libraries such as IMSL, **LIBS** changes the name internally to the full pathname. **LIBS** also tries to open each library to make sure the file exists and is accessible. Using a qualifier, **LIBS** can clear the table before adding libraries, something you might do if you previously added the wrong library or if you need a completely different set of libraries. With the other qualifier, **LIBS** can list the table so you can see what libraries are present or have been added in the current list.

Continued on page 244

Continued from page 243

The complete `LIBS` command with qualifiers and parameters is:

```
LIBS /CLEAR /LIST lib1,lib2,...,libN
```

where *N* is between 1 and 8. If `/CLEAR` is used, the LNK$LIBRARY table is cleared first. If library names appear, they are added to the table. If `/LIST` is used, the table is listed after any libraries have been added.

You only need to enter the `LIBS` command once per time-sharing session or batch job. After that, LINK knows about the libraries so they don't have to be listed anywhere. For example, to use an IMSL routine with the following file SAM.FOR whose contents are:

```
DOUBLE PRECISION DGAMMA
PRINT *,DGAMMA(5.0D0)
END
```

then enter the commands:

```
$ LIBS IMSL
$ FORTRAN/G_FLOATING SAM
$ LINK SAM
$ RUN SAM
```

If you want to create your own libraries, you can enter the command:

```
$ LIBRARY/CREATE libname.OLB libname.OBJ
```

where *libname* is your library name. The file `libname.OBJ` is the binary output from the FORTRAN compiler. The file `libname.OLB` is the resulting library from the `LIBRARY/CREATE` command. To use the library with `LIBS`, enter the command:

```
$ LIBS device:[directory]libname.OLB
```

For example, if file JOE.FOR contains the routines of your library on device USERA in directory ABC.DEF, enter these commands:

```
$ FORTRAN/G_FLOATING JOE
$ LIBRARY/CREATE JOE.OLB JOE.OBJ
$ LIBS USERA:[ABC.DEF]JOE.OLB
```

The following normal messages are issued by `LIBS`:

```
** All library selections cleared. **
```

This message is given when `/CLEAR` is used.

```
** 'count' libraries added. **
```

This message is given when all libraries in the `LIBS` command have been added. *count* is the number of names in the list.

```
"LNK$LIBRARY" = " 'libraryname' " (LNM$PROCESS_TABLE)
"LNK$LIBRARY_'num' " = " 'libraryname' " (LNM$PROCESS_TABLE)
```

```
      .
      .
** 'count' active libraries. **
```

These messages are given when **/LIST** is used. Each '*libraryname*' is the name of a library from the **LIBS** command. The value of '*num*' is 1 to 8. The number of libraries in the LNK$LIBRARY table is '*count*'.

The following error messages are issued by **LIBS**:

```
** More than 8 library names.  LIBS command ignored. **
```

This message is given when more than 8 names appeared in the **LIBS** command.

```
** Use /CLEAR or /LIST, not /'qual'.  LIBS command ignored. **
```

This message is given when the qualifier '*qual*' is not **CLEAR** or **LIST**.

```
** More than 9 libraries requested.  Remainder ignored. **
```

An attempt was made to have more than 9 active libraries. Use **LIBS/LIST** to see what is presently in the table.

```
** File 'filename' not found so it was ignored. **
```

The library name '*filename*' could not be opened so it was not added to the LNK$LIBRARY table.

```
** Null file name ignored. **
```

There was no name following a comma in the list of library names on the **LIBS** command (or there were adjacent commas) so the null name was ignored.

The following are examples of using **LIBS** and their corresponding output:

```
$ LIBS          {This is a do-nothing command.}

$ LIBS/CLEAR
** All library selections cleared. **

$ LIBS IMSL
** 1 libraries added. **

$ LIBS/LIST
  "LNK$LIBRARY" = "USERC:[ABC1234]IMSL.OLB" (LNM$PROCESS_TABLE)
** 1 active libraries. **

$ LIBS/CLEAR/LIST LINPACK
** All library selections cleared. **
** 1 libraries added. **
  "LNK$LIBRARY" = "USERC:[GHI4321]LINPACK.OLB" (LNM$PROCESS_TABLE)
** 1 active libraries. **

$ LIBS /CLEAR MINPACK /LIST        {Qualifiers can appear in any order.}
** All library selections cleared. **
```

Continued on page 246

Continued from page 245

```
** 1 libraries added. **
  "LNK$LIBRARY" = "USERC:[DEF5678]MINPACK.OLB" (LNM$PROCESS_TABLE)
** 1 active libraries. **

$ LIBS EISPACK,USERC[JKL8765]MTLIBD.OLB,MINPACK  {MINPACK was
  ** 2 libraries added. **                          already there.}

$ LIBS /LIST
  "LNK$LIBRARY" = "USERC:[DEF5678]MINPACK.OLB" (LNM$PROCESS_TABLE)
  "LNK$LIBRARY_1" = "USERC:[MNP3456]EISPACK.OLB"(LNM$PROCESS_TABLE)
  "LNK$LIBRARY_2" = "USERC:[JKL8765]MTLIBD.OLB" (LNM$PROCESS_TABLE)
** 3 active libraries. **
```

Instructional Computing

Project Assist Faculty Workshops

Renate Lippert

The fall calendar for workshops, brown bag lunches, and other activities has been updated.

- Introduction to PLATO – Every Thursday, 3 - 4 p.m., 33 Fraser.
- Design Considerations for Computer-Based Instruction – November 18, 9 -12, 335 Peik.
- Instructional Design Model for Computer-Based Instruction – November 25, 9 - 12, 335 Peik.

Project Assist has moved to their new facilities in 33 Fraser Hall. Please call 626-1090 to request the fall calendar, a registration form for the workshops, or to be included in the mailing list.

Text Analysis Programs

Tom Rindflesch

We have recently enhanced several programs on the CYBER CA that help you analyze texts. You can use these programs to determine the frequency and distribution of the letters, words, and phrases that occur in a text. You can also create concordances and indexes with them. The programs discussed here would be useful in literary analysis or philological studies, or in any research that requires manipulation or quantitative study of the linguistic content of a text.

The following programs are available:

GENCORD creates concordances and indexes and provides frequency distribution information for letters and words. With GENCORD you can specify which words in the text are processed, and determine the format of the concordance as well as the sorting order of the words that appear in the concordance and the index. For more information, log on to the CYBER and type the command `WRITEUP,GENCORD` to read the on-line documentation.

TEXTAL is a pattern matching and retrieval program that allows you to specify word sequences or patterns that you want to search for in a text. You can also attach additional information to each word in the text by adding "tags," labels that classify the word according to any category you create (grammatical, metrical, etymological, etc.) You can then specify both tags and words in the patterns you search for. TEXTAL produces output in concordance format. See the on-line document `WRITEUP,TEXTAL`. TEXTAL is also available from the Special Projects Group for MS-DOS machines for $25.00.

LTTRCNT, as the name suggests, counts letters in a text. In addition to overall frequency of occurrence for each letter, it keeps track of the frequency of occurrence of letters at the three major positions in a word: initial, medial, and final. See `WRITEUP,TEXHELP=LTTRCNT`.

WRDSTAT counts the number of words in a text, and then calculates the total number of word tokens, the total number of word types, the average length of the tokens, and the average length of the types. It also determines the number of words of each length occurring in the text. See `WRITEUP,TEXHELP=WRDSTAT`.

If you would like to discuss the possible use of these programs in a research project or if you would like further information about them, contact

Tom Rindflesch
Ron Zacharski
Special Projects Group
625-8332

ScholarNet

Tom Rindflesch

We have recently received information about a new electronic network called **ScholarNet**, which allows you to exchange files with other subscribers in the United States and in over fifty other countries. You can also access information provided by ScholarNet, such as course syllabuses, bibliographies, reviews of software and books, the Associated Press News, an on-line encyclopedia, and public domain software.

ScholarNet currently includes two divisions: PoliNet, which covers the disciplines of political science, public administration, and criminal justice studies, and HumanNet, which covers English, history, philosophy, and religious studies. When you join ScholarNet you can use both PoliNet and HumanNet.

A one-time membership fee for ScholarNet costs $29.95 for individuals. In addition, while you are logged on to ScholarNet you pay $16.00 an hour during prime time and $6.00 an hour at other times. Departments may subscribe for a one-time fee of $100.00. Department subscriptions allow any number of associated subaccounts, which pay the hourly rates but do not pay separate one-time fees.

For more information call

Tom Rindflesch
625-8332

or contact

Richard W. Slatta
ScholarNet
School of Humanities and Social Sciences
North Carolina State University
Box 8101
Raleigh, N.C. 27695
(919) 737-7908

For Sale

The University of Minnesota, Morris, has a DEC PDP-11/44 computer for sale. The system includes the PDP-11/44, 1 Mb of memory, FPA, 1 -DZII 8-port asynchronous controller, 3-RK07 disk drives (28 Mb each) and TU-58 cassette tape. The system **must** be sold within the University. For further information, contact Andy Lopez at (612) 589-2211.

Thanksgiving Hours

On Thursday, November 27, ACSS systems will cease operations after 4 a.m., in observance of Thanksgiving. We will resume normal service at 7 a.m. Friday, November 28.

Christmas, Other Hours

With University holidays approaching for Christmas, New Year's, and Martin Luther King Day, the following are the ACSS holiday operating hours as now planned. They may be subject to change in the future.

In observance of Christmas, ACSS systems will cease operations after 1 p.m. Wednesday, December 24, and will resume operations at 7 a.m. Friday, December 26.

In observance of New Year's, ACSS systems will cease operations on Wednesday, December 31, at 5 p.m. and resume operations at 7 a.m. on Friday, January 2.

ACSS systems will operate during its normal hours during Martin Luther King Day, Monday, January 19, a University holiday.

Any changes in the above holiday hours will be reported in the December and January issues of this *Newsletter* and will also be announced in system notes.

On the Move: ACSS Offices, Services Relocate

Steven Brehe

As announced in our October issue, many ACSS administrative offices and some consulting offices (including our Contract Services Group) have moved to the Lauderdale Facility. A few Wulling Hall offices will move to Fraser Hall sometime this quarter, but remain in Wulling Hall for the time being. These are:

- The Scientific Programs Librarian
- The Statistics Software Manager
- The Text Processing Consultant

For application forms for computing accounts and research computing grants, and for rate sheets and other information, talk to our staff in the Computing Information Center, 128A Lind Hall (625-7397).

The Help Page

General Consulting

Walk-in: 128C Lind Hall,10 a.m. to 4 p.m. weekdays and 7 to 9 p.m. on Tuesday and Thursday evenings.

HELP-Line: 626-5592, 8 a.m. to 5 p.m. weekdays.

Specialized HELP-Lines

Artificial Intelligence: 625-8332, 3 to 4 p.m. weekdays.

Data Bases: 626-1887, 10 to 11 a.m. weekdays.

Microcomputers: 626-4276, 9 a.m. to noon and 1:30 to 4 p.m., Monday, Tuesday, Friday; 9 to 4, Wednesday and Thursday.

Statistics: 626-1893, 11 a.m. to 2 p.m. weekdays.

Text Analysis: 625-8332, 3 to 4 p.m. weekdays.

Text Processing: 625-1391, 9:30 to 11:30 a.m., Monday, Wednesday, Friday; 1 to 2 p.m., Tuesday, Thursday, Friday.

Consultants for Faculty

Faculty may call Sharon Krmpotich (626-1085) for help in research computing on the CYBER and VAX computers.

Computing Information Center

At 128A Lind Hall, you can acquire application forms for computer accounts and grants. You can also enroll in short courses there, get help in ordering vendor documentation, examine reference copies of vendor and all other documentation for ACSS software, and pick up copies of ACSS's free documents.

PHONE NUMBERS

Information: 626-1600 **HELP-Line: 626-5592**

Access:	
CYBER (CA)–	626-1620
CYBER (MD)–	626-1622
UNIX (UX)–	626-1681
VAX (VX)–	626-1641
RJE (2400 baud)	626-1656
RJE (4800 baud)	626-1663
Accounts:	
CYBER, UNIX, 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	627-4177
Equipment Maintenance/Repair	627-4357
Graphics Software	626-5592
Information, Lauderdale	626-1600

Lauderdale Computer Room	626-0550
Lauderdale Services	626-1838
Magnetic Tape Librarian	626-1838
Math and Statistics Packages	626-5830
Micro Information	626-4276
Newsletter Subscription	625-7397
Permanent File Restoration	626-0595
Project Assist	626-1090
Public Labs	
14 Folwell Hall	625-4896
306B Lind Hall	625-9032
9 Walter Library	626-1899
Publications Information	626-1093
Remote Batch (RJE) Services	625-7850
Short Course Registration	625-7397
Shuttle Bus Service	625-9525
System Status (recorded message)	626-1819

OPERATING HOURS

	CYBER (CA), UNIX (UX), and VAX (VX)	CA and VX Low Rate	CYBER (MD)
M-F	7 a.m. - 4 a.m.	8 p.m. - 4 a.m., 7 a.m.- 8 a.m.	7 a.m. - 1 a.m.
Sat	4 a.m. - 7 p.m.	all operating hours	7 a.m. - 10 p.m.
Sun	6 p.m. - 4 a.m.	all operating hours	6 p.m. - 1 a.m.

PUBLIC LABS-TWIN CITIES CAMPUS

Location	Batch	Interactive	Micro
East Bank			
ApH			X
Arch 160			X
CenH		X	
ComH		X	
DiehlH 207		X	
EddyH Annex 54			X
EltH 121, 124	P	X	
FolH 14, 14a	P	X	X
FronH		X	
LindH 26		X	
LindH 128B	X	X	
LindH 306B			X
MechE 308		X	
MoosT 8-425			X
Phys 130		X	
PioH		X	
SanH			
TerrH		X	
VinH 4		X	
WaLib 9		X	X
West Bank			
AndH 170			X
BlegH 140	P	X	
MdbH		X	
OMWL 2	P	X	
St. Paul			
BaH		X	
CentLib B50			X
ClaOff 125 E	X	X	
ClaOff B22			X

P means Printer only.
For more information see WRITEUP,LABS.

Contents

***The* ACSS Newsletter**
November 1986
Volume 20, Number 11

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

The *ACSS Newsletter* is published monthly by Academic Computing Services and Systems (formerly the University Computing Center) 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 running Microsoft Word, MacPaint, MacDraw, and Aldus Pagemaker software, with camera-ready copy produced on the Apple LaserWriter.

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

The University of Minnesota adheres to the principle that all persons should have equal opportunity and access to facilities in any phase of University activity without regard to race, religion, color, sex, national origin, handicap, age, or veteran status.

The ACSS
Newsletter

**Academic
Computing
Services and
Systems**

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