[illegible]

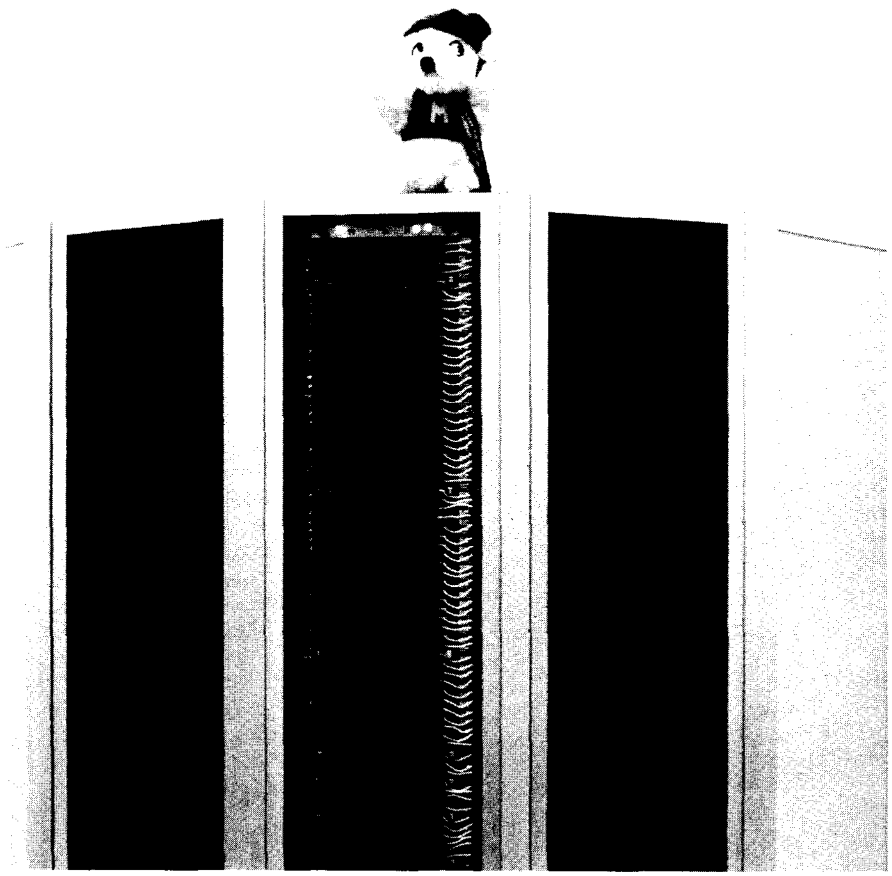
University Computer Center Newsletter

University of Minnesota
Twin Cities

November 1982
Volume 16, Number 11

THE CRAY—TODAY AND TOMORROW

Two years ago, with record high demand for resources at UCC, we went out on bid for a system that would provide cost-effective computing for our user community. Before the bid, we analyzed the systems that offered two to four times the computational growth of our current CYBERs, and found that they did not offer a comparable reduction in user charges. Since we did not have a granting agency or a government financial backer to provide us with an almost free computer, the system we selected would have to pay for itself out of our revenues. Thus our bid request for a replacement system for the CYBER 74 included a performance/cost ratio on a benchmark set of programs. The sole bidder was Cray Research Incorporated, with an eight million byte system that at half load (1000 hours per year) allowed us to charge one third as much as for CYBER 74 scalar jobs and one sixth as much for vector-limited jobs. As system use grows to a normal 2000 CP hours per year, the charges will be reduced to one sixth and one twelfth the cost of the CYBER. The CRAY was definitely cost-effective. It was purchased and installed at our Lauderdale facility one year ago.


"Goldie" Gopher is the official mascot of UCC's CRAY. (Photo: Dick Hotchkiss)

As a computer center serving the University and public service agencies, our mission is to provide more capability to our users. We do this in several ways: with more effective software that allows you to get quick results without spending excessive time learning the system; with robust and error-free hardware and software that give you confidence in your results; and with continuing cost reductions for processing and data storage so that you can do more with your current budget. The CRAY-1A system at Lauderdale provides a nucleus of cost-effective computing for both number crunching and graphics.

In addition to the benefits the CRAY offers, we continue to meet the needs of users as much as possible given recent, extremely severe, budget cutbacks. In the realm of microcomputers, our Micro Systems Group has encouraged standard micro systems for our users, and continues to develop better systems and software. We plan to introduce

CRAY to 119

Contents

Director: Peter C. Patton
Editor: Christine Mack Gordon

Comments about the content of this newsletter, or suggestions for changes may be directed to the editor, 235a Experimental Engineering, or call 612/376-9832.

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

CRAY from 117

broad-band and local-network packet switching to the computer systems at UCC. When fully implemented, this switching ability will extend your range beyond a single system. Compatible software, including compilers, statistical and data base packages, and graphics, will be available (as some are now) to increase your computing power. Our graphics services, for example, will substantially increase early this winter when the VAX/VMS system (currently located in Shepherd Labs) is united with the CRAY at Lauderdale.

Other changes we have planned at Lauderdale include an upgrade from CYBER 700 to CYBER 800 service on the main CRAY front end station, and the addition of a laser printing system to replace our current line printers.

Together these changes will improve our future services in those areas most in demand: economical, large scale computing; packet switching; microcomputing; and robust, portable, standard software. The CRAY alone will provide cost-effective large scale computing for years to come.

(Lawrence A. Liddiard)

YOUR FIRST CRAY JOB

With a year's experience on the CRAY and a beginner's mistakes behind us, we are now ready to assist you in using this powerful tool. It is not as formidable an undertaking as you might think.

Submitting Jobs

The CRAY Operating System (COS) control language has only about 70 commands, far fewer than the 450 of the Network Operating System (NOS) used on our CYBERs.

To begin, you must know a handful of NOS control statements, since the only way to access our CRAY is through the CYBER 730 or CYBER 74. Information also returns from the CRAY through the CYBER front-ends.

We are showing you the simplest form of a CRAY job, created interactively, stored in a file, and channeled to the CRAY with the interactive SUBMIT command. You will notice that your CRAY password is stored in the file as part of the ACCOUNT statement. This is generally regarded as bad practice, since the rare "user" who succeeds in breaking file security can discover more accounts and passwords as he or she riffles through your files.

It is better to use the SUBMIT command's so-called "reformatting directives" to, among other things, get the system to ask you interactively for your CRAY user number and password upon submission. The examples in the article on languages use these reformatting directives—the lines that begin with a slash (/) in column one.

Reformatting commands are described in WRITEUP(SUBMIT). Read this writeup *before* you use reformatting directives, since greater capabilities mean more complexity and also more pitfalls.

On an interactive terminal that accesses the CYBER 730, use any familiar editor (XEDIT is a favorite) to construct the following permanent file on the CYBER:

```
JOB.
ACCOUNT,AC=ABC0123,PW=CRAYPW.
CFT.
LDR.
-EOR-         (end-of-record mark)
  PROGRAM FORTRAN(INPUT,OUTPUT)
  DIMENSION A(1000)
  .
  .
  .
-EOF-         (end-of-file mark)
```

This file, which is a complete CRAY job, is composed of two records: record one consists of COS control statements, and record two is a source CRAY-1 FORTRAN (CFT, the ANSI X3.9-

1978 Standard CRAY-1 FORTRAN) program. The records are separated with an EOR (end-of-record) mark produced with the XEDIT WEOR command. You can send this permanent file to the CRAY as a CRAY job with the NOS command:

```
SUBMIT(CRAYJOB,MI=CR)
```

The CYBER systems respond to the submission request with a seven character jobname (the first four letters of which are always your user hash). For example, you might see:

```
AVGQXYZ.
```

When the CRAY finishes processing this job, the output file with the appended CRAY dayfile is sent to the CYBER 730 and stored as a permanent file under your user number on the default pack; this file is called AVGQXYZ.

It's as simple as that. You can look at your output with an editor or route it to a printer.

Compiling Programs

Suppose you have a CRAY Pascal program to compile, and it requires a record of input data at execution time. In this case, construct a permanent file with *three* records on the CYBER 730; the third record is your Pascal input data. (Note that on the loader statement, LDR, you must call for the Pascal library $PSLIB.)

```
JOB.
ACCOUNT,AC=ABC0123,PW=CRAYPW.
PASCAL.
LDR,LIB=$PSLIB.
-EOR-
  PROGRAM PASCAL(INPUT,OUTPUT);
  VAR SUBTOTAL,TOTAL,OPERAND :
    REAL;
  .
  .
  .
-EOR-
1.53E7
2.48E17
.
.
.
-EOF-
```

In fact, the CRAY Operating System is a simple batch operating system and, as you can see, it is quite easy to learn and use the control statements for typical number crunching tasks.

Inter-machine Transfer

Our last example concerns ACQUIRE, a rather complicated command for inter-machine transfer that calls upon the CRAY to acquire a permanent file from one of the CYBERs.

Suppose you have a large permanent data file named DATA1 on the CYBER 730 that is used as an input file for your FORTRAN program. We do not yet support nor guarantee the security of permanent datasets (the CRAY's name for files) on the CRAY, so they must be acquired from the CYBER 730 (or the CYBER 74), brought to the CRAY for use, then disposed to the CYBER permanent file families when you're done with them (if you've rewritten them). Many CRAY beginners are confused by the lengthy text parameters on the ACQUIRE and DISPOSE commands. A CRAY job that issues an ACQUIRE or DISPOSE is actually suspended, while the TEXT parameter, a lengthy character string, is run as a NOS job on the CYBER 730 to get and transmit (or receive and save) a file during the inter-machine transfer. This more complex, but very common CRAY job is structured as follows:

```
JOB.
ACCOUNT,AC=ABC0123,PW=CRAYPW.
ACQUIRE,DN=DATA1,UQ,ID=
  ABC0123, ^
TEXT='USER,ABC0123,CYBERPW.
  ATTACH,DATA1.'^
'CTASK.'.
DELETE,DN=DATA1,NA.
ASSIGN,DN=DATA1,A=FT01.
CFT.
LDR.
-EOR-
  PROGRAM FORTRAN(COMMENT,
    COMMENT,COMMENT)
  DATA R/1.E17/
  .
  .
  .
-EOF-
```

In this example, the TEXT parameter of the ACQUIRE statement is a character string two lines long, making the ACQUIRE command a total of three lines. The caret symbol (^) denotes line continuation in COS. CTASK is a new NOS command that does the actual CRAY-CYBER communication task. CFT treats all files listed in parentheses on the PROGRAM statement as comments, so you need the COS ASSIGN command to characterize the input data file, DATA1, as logical FORTRAN unit 1. Finally, the ACQUIRE command makes a dataset both local and permanent on the CRAY, so you must also use a DELETE command, lest the CRAY disks become saturated with old datasets acquired in forgotten runs.

The DISPOSE command, which sends CRAY datasets from the CRAY to the CYBERs, is almost identical to ACQUIRE.

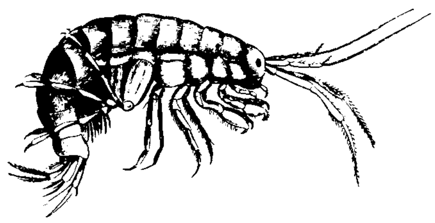
More details on these and other parameters and additional COS control statements are available in WRITEUP(CRAY=*). You may retrieve a printed copy of this writeup with these CYBER commands:

```
WRITEUP(CRAY=*/L=LIST,PT=AS)
ROUTE(LIST,EC=A9,DC=PR,TID=xx,
  BIN=xxxx)
```

The services we have established to assist you as you begin to explore the CRAY are delineated in the article *CRAY Related Services* in this issue of the *Newsletter* and in WRITEUP(CRAYUSE).

(Note: These CRAY job submission commands will change slightly when we convert our CYBER Operating System to NOS 1.4 at the end of fall quarter. See the September 1982 issue of this *Newsletter* for the command structure that will be implemented with this upgrade.)

(T. E. Kovarik)



CRAY RELATED SERVICES

As a result of our commitment to bring the latest in software and hardware to our diverse user community (some new to computing, others extremely sophisticated in the field), we have created a support structure of CRAY services to make this advanced technology available to everyone. We offer assistance in a variety of forms.

Consulting

We employ a professional staff with over a year of CRAY-1 software programming and consulting experience. This staff, trained at Cray Research Incorporated, participates in general consulting in 140 Experimental Engineering. See WRITEUP(CONSKED) for a schedule of hours and specialized consulting. Our HELP-line (376-5592) is also open for your questions from 9 a.m.-5 p.m. Monday through Friday.

Call on our CRAY consultants for any problems you cannot solve by referring to our documentation (see below). When you begin to use the CRAY, our consultants will help you with the simple problems we all stumble over as beginners, such as format errors on the job control statement. As your use of the CRAY becomes more sophisticated and your familiarity with the available software increases, your need for assistance will change: more difficult questions will arise, perhaps related to CRAY FORTRAN vectorization and optimization. Our consultants will still be able to save you CPU time, and thus money, when such questions occur. They can help you use this tool as its inventors intended.

Documentation

We offer CRAY software documentation, both in hard copy manuals and in machine retrievable form on the CYBERs, that runs the gamut of your needs. Some of our most popular CRAY documentation products include the following, but this list is by no means exhaustive.

—WRITEUP(CRAYUSE) Summarizes our CRAY services. This is the new name for WRITEUP (SERVICE=CRAY).

—WRITEUP(CRAY=INDEX) Lists all the CRAY job control statement articles in use at UCC by name, as well as additional articles on features of our CRAY system.

—WRITEUP(CRAY=EXAMPLES) Contains simple examples of CRAY access and use, intended for beginners.

—CRAY Ready Reference Manual A "CRAY Instant," intended for those who know COS job control language. (With its handsome brown vinyl binder, this is considered by many *the* chic item to carry around campus this quarter.)

Additional documentation is available for your specialized needs, including MINNLIB (the Minnesota program library), CFT (CRAY-1 FORTRAN ANSI X3.9-1978 standard) Reference Manual, and others. Reference copies of CRAY manuals are available in the UCC Reference Room, 140 Experimental Engineering. Talk to our consultants about the documentation you need to solve your specialized problems.

Classes

We offer CRAY software courses each quarter. These courses, held on the Minneapolis campus, are open to University students, faculty, and staff, and to outside users. Our fees for these courses are considerably lower than those for comparable courses offered elsewhere.

The schedule of classes is posted on campus at the beginning of each quarter and is also published in this *Newsletter*. Register by mail or in person at our Computer Store, 211 Experimental Engineering. The classes we offer at the present time are summarized below.

—Introduction to the CRAY-1A and COS Job Control Statements—Includes a slide show of CRAY hardware and software features, and a discussion and examples of CRAY job control statements in use at UCC. Students actually run jobs on the CRAY and move files from the CYBERs to and from the CRAY. This course is recommended for all beginners.

—CFT Vectorization and Optimization—Covers CRAY FORTRAN program writing techniques that go beyond the "automatic" vectorizing capability of the CFT compiler and allow the maximum vectorization possible in this higher level language.

—CAL (CRAY Assembly Language)—This course is intended for those who wish to write programs tailored to the hardware of the CRAY central processor.

Libraries & Applications Packages

Some libraries and applications packages are already available on the CRAY and more are being converted. Articles elsewhere in this issue detail these routines, many of them optimized for the CRAY vector hardware. To assist diverse academic disciplines, we can also acquire additional packages developed at other CRAY sites for research in areas such as structural analysis, hydrology, petroleum exploration, and nuclear engineering safety. Our Development Group (373-7745) can help you explore the availability and economic feasibility of such packages.

Contract Services

For departments or organizations that prefer to contract out complete or partial CRAY software projects, our Professional Services Division (PSD) can assist you. This division, which includes senior and experienced CRAY software professionals, is available for contract services for CRAY applications program development and all related services. Any or all parts of the entire software process—problem definition, systems analysis, systems design, applications programming, debugging, beta testing, documentation, user training, and systems integration—can be contracted for with our Professional Services Division. For more information, call PSD at 376-1764.

(*T. E. Kovarik*)

CRAY APPLICATIONS LIBRARY

We have received and installed the complete CRAY Applications Library (FORTRAN-callable only). Only binary code is available to you, either as as complete programs or as user libraries. The libraries and complete programs are listed in Exhibit 1.

We have one copy of the documentation for each of these packages and will make copies for you at cost. Documentation and further details can be obtained from Mike Frisch.

(Mike Frisch, 376-1636)

Exhibit 1.

Libraries:

MO1LIB	Linear systems routines.
MO3LIB	Integration and differentiation routines.
MO4LIB	Interpolation and approximation routines.
MO5LIB	Special functions routines.
MO6LIB	Fast Fourier Transform routines.
MO7LIB	Filter routines.
MO8LIB	Statistics routines.
ULIB	Utility routines for data manipulation (pack, unpack, convert, give machine constants).

Complete programs:

ARRIBA	Solve small integer programming problems.
BANNER	Put a banner page on a dataset.
BATCHLP	Solve linear programming problems.
BLOCK	Rewrite a dataset to have a blocked structure.
CANON	Preprocess problems for BATCHLP.
CATALOG	List size and contents of dataset.
CDCCONV	Convert CDC FTN 4.0 programs to CFT programs.
COPYSF	Shift dataset right one column (as COPYSBF does on the CYBERs).
CROSSRF	Produce cross reference map for CFT program.
FDSC	Show page and entry locations of files in a dataset catalog.
FILES	Show current assignment of all files known to a job.
GAUSIAN	Make Hartree-Fock molecular orbital calculations.
HONDO II	Make finite element calculations for large deformation dynamic response of axisymmetric solids.
HONDO 5	Make Hartree-Fock self-consistent field calculations.
INDEX	Produce sorted directory of statement numbers, variable names, routine names for CFT programs.
LINWOOD	Fit linear least squares curves.
MODECKS	Compare two CFT files and create UPDATE mod decks to make old into new.
PAGES	Add page headings to a print file.
PREUP	Create an UPDATE input file from CFT source.
QUERY	Show current status of all files known to the job.
ROFF	Produce text formatted output.
SAP4	Analyze structures.
SORTFIL	Perform alphabetic sort of a file.
SPICE2	Analyze electronic circuits.
TIDY	Clean up FORTRAN 66 programs.
UNBLOCK	Rewrite a dataset to have an unblocked structure.

SPSS ON THE CRAY

The SPSS conversion to the CRAY is proceeding very slowly due to a staff shortage. Our current projection is that there will be a CRAY version by December 1982 that includes all the general features of SPSS and four or five of the most heavily used procedures.

(E. Hinkley)

CRAY SCIENTIFIC APPLICATIONS PACKAGES

We have converted a number of user libraries from the CYBERs to the CRAY. (These and all other libraries and packages mentioned in this article are FORTRAN-callable only.) They are listed in Exhibit 2. To access these libraries on the CRAY, use the COS loader control statement:

LDR(LIB = lib1:lib2: ... :libN)

where "libI" is one of the library names. For example:

LDR(LIB = IMSL) or
LDR(LIB = MINNLIB:MNCORE)

For further information on these libraries, see WRITEUP(LIBNDEX) on the CYBERs.

We have installed the CRAY Applications Library, described elsewhere in this *Newsletter*. Several CYBER packages are available in this library, including:

SAP4 —Structural analysis package.
SPICE—Electrical engineering circuit analysis package.
TIDY —FORTRAN program cleanup package. (This version differs slightly from the CYBER version. Contact Mike Frisch for details.)

We will convert most of the remaining CYBER scientific applications packages, listed in Exhibit 3, to the CRAY by January 1, 1983.

We do not plan to convert the CYBER scientific applications packages listed in Exhibit 4. If you have suggestions for other scientific applications packages you would like to see on the CRAY, please contact us.

(Mike Frisch, 376-1636)

BESPACK	– Bessel function library.
BSPLINE	– B-spline interpolation library.
EPISODE	– Ordinary differential equation library.
FUNPACK	– Special functions library.
GRG2	– Constrained optimization library.
IMSL	– International Mathematics and Statistics Library.
LINPACK	– Linear equations library. These are the double precision and complex routines that are not on the CRAY SCILIB library.
MINNLIB	– UCC's library. (see WRITEUP(CRAY = MINNLIB) for conversion details.)
MINPACK	– Unconstrained minimization and least squares library.
MNCORE	– Core graphics library.
YSMPLIB	– Yale Sparse Matrix Package library.

Exhibit 2. CYBER user libraries now available on the CRAY.

DAREP	– Continuous simulation package.
FORSIM	– Partial differential equations package.
MPOS	– Linear programming package.
NONSAP	– Nonlinear structural analysis package.
TSP	– Time series package.

Exhibit 3. CYBER scientific applications packages to be available on the CRAY January 1, 1983.

ALMAP/FORPREP	– Algebraic manipulation library and FORTRAN preprocessor package. We are installing the REDUCE algebraic manipulation package on the CDC CYBERs which will replace this.
BOXJLIB	– Box-Jenkins methods library. We will soon be installing BMDP81 on the CYBERs which has the same features.
GPM	– Constrained optimization library. We feel that GRG2 is adequate for the CRAY for now.
PERT	– Program Evaluation and Review Technique package. Usage of this is too low to justify conversion now.
PERTCST	– PERT Cost package. Usage of this is also too low.
SIMPLX	– Linear programming library. We feel the IMSL linear programming routines are adequate.

Exhibit 4. CYBER scientific applications packages that will *not* be converted to the CRAY.

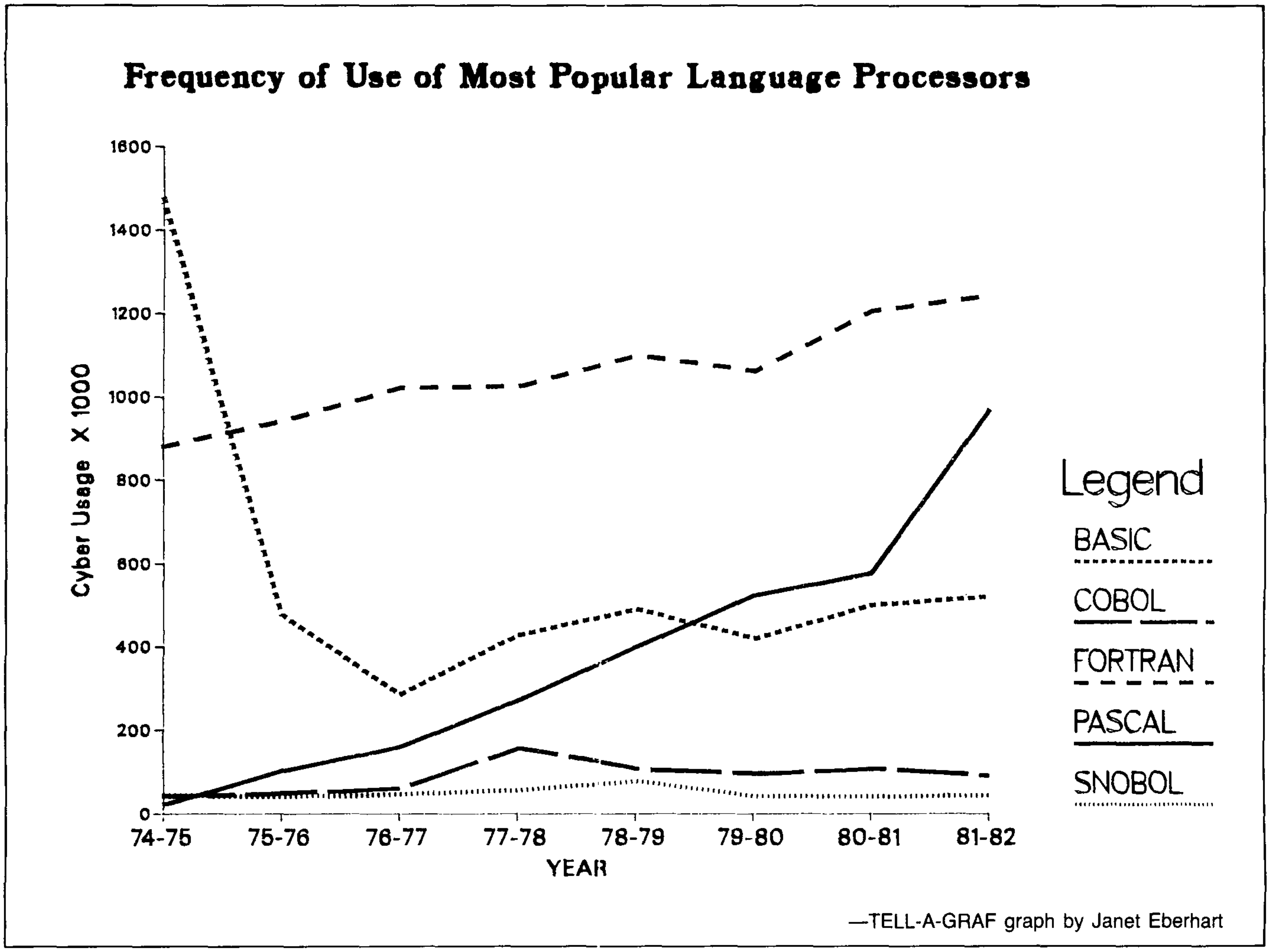
CRAY LANGUAGE PROCESSORS

We have three language processors available on the CRAY: CRAY-1 FORTRAN (CFT), CRAY Assembly Language (CAL), and Pascal. CFT and CAL are products developed and supported by Cray Research Incorporated (CRI). Pascal was developed at Los Alamos Scientific Laboratory and modified by CRI to run on the COS operating system; Pascal is not an official Cray product and is not supported by Cray personnel. Please let us know what other language processors you would like to see on the CRAY in the future.

CFT

We currently have three versions of the CRAY-1 FORTRAN compiler (CFT) available on the CRAY-1: past, current, and future. Past CFT is version 1.09 with bugfix 4; current CFT is version 1.09 with bugfix 5. Future CFT is the first release of CFT version 1.10, and it is in complete conformance with the 1977 standard. CFT version 1.09 conforms to the 1977 standard with the exception of CHARACTER type.

Three versions of the FORTRAN libraries correspond to the three versions of the CRAY-1 FORTRAN compiler (CFT) that we have available. They can be accessed as follows:



—TELL-A-GRAF graph by Janet Eberhart

For the past version of CFT (level 1.09 bugfix 4) use the following COS control statements:

```
CFTPAST.
LDR(LIB=$FTLIBP)
```

(Use past version of FORTRAN library.)

For the future version of CFT (CFT level 1.10), use:

```
CFTFUT.
LDRFUT(LIB=$FTLIBF:$SYSF)
```

(Use future version of libraries *and loader.*)

For the current version of CFT (level 1.09 bugfix 5) use:

```
CFT.
LDR.
```

(Access the default library $FTLIB.)

Here are a few sample jobs using CFT. They are files created interactively on the CYBER 730, and submitted to the CRAY with the interactive command SUBMIT(filename,MI=CR). The lines that begin with a slash (/) are "reformatting directives" used by the SUBMIT command to package the job for the CRAY. /ACCOUNT, for example, causes the CYBER to ask for your CRAY account and password. Read WRITEUP(SUBMIT) for an explanation of reformatting directives.

Example 1. This job uses the current (default) version of CFT to compile this sample program. After CFT compiles the program, the current (default) loader LDR loads the binary into memory for execution.

```
/JOB
/NOSEQ
JOB.
/ACCOUNT
CFT.
LDR.
/EOR
      PROGRAM TEST
      READ *,I,J
      WRITE *,I,J
      READ *,A,B
      C = A + B
      PRINT *,A,B,C
      WRITE (6,100) c
100   FORMAT (* A + B = *,F8.2)
      STOP
      END
/EOR
 1,   3
55.5, 44.4
```

Example 2. This job uses the future version of CFT (CFTFUT) to compile a FORTRAN program that uses CHARACTER type variables. CFTFUT (CFT version 1.10) creates a binary that must be loaded by the future version of the loader (LDRFUT); the loader satisfies externals from the future versions of the libraries $FTLIB ($FTLIBF) and $SYSLIB($SYSF).

```
/JOB
/NOSEQ
JOB.
/ACCOUNT
CFTFUT.
LDRFUT,LIBF=$FTLIBF:$SYSF.
/EOR
      PROGRAM TESTCH
      CHARACTER A*10,B*10
      A="MCGEE"
      B='TRAVIS'
```

```
    WRITE (6,100)A,B
100 FORMAT (1X,2A10)
    WRITE *,A,B
    WRITE(6,100) A(3:5),B(3:6)
    READ *,A,B
    PRINT *,A,B
    WRITE (6,100) A,B
    STOP
    END
/EOR
'TESTING',"CFT110"
```

For more information on CRAY-1 FORTRAN see WRITEUP (CRAY=CFT).

PASCAL

The Pascal compiler available on the CRAY implements nearly all the features of Pascal described in Jensen and Wirth's *Pascal User's Manual and Report*, as well as several additional predefined functions and procedures, including many of those available on the CYBERs. It does not conform to the new ISO DIS7185 Pascal standard.

To access the compiler, use the COS control statement:

```
PASCAL,INPUT=source,OUTPUT=
  listing,B=binary,LIST=ON/OFF.
```

The default for INPUT is $IN, the default for OUTPUT is $OUT, the default for B is $BLD (this is also the default input for the loader, LDR), and the default LIST is ON. INPUT and OUTPUT are implicitly declared in all programs, and specification of filenames in the program header is optional.

If you specify LIST=OFF, the only output from the compiler is error lines. If LIST=ON (or LIST is omitted), the entire compiler listing is produced.

Here are a few sample jobs using Pascal. Again, they assume the use of SUBMIT(filename, MI=CR).

Example 1: (using all defaults)

```
/JOB
/NOSEQ
JOB.
/ACCOUNT
PASCAL.
LDR,LIB=$PSLIB.
/EOR
PROGRAM TEST(INPUT,
  OUTPUT);
VAR
 I,J : INTEGER;
 ARR : ARRAY[1..10] OF CHAR;
BEGIN
 J := 8;
 FOR I := 1 TO J+2 DO
 BEGIN
  READ(ARR[I]);
  WRITE(ARR[I])
 END;
 WRITELN(OUTPUT)
END.
/EOR
1234567890123456789
```

This job compiles the second record of the submitted file and uses the third record as data for the run. The output goes to $OUT, which (along with $LOG, the "dayfile") is returned automatically as a permanent file on the default pack. The name of this file is the same as the jobname given by the SUBMIT command (the user hash followed by three random letters).

Example 2: (program is stored on CYBER file)

```
/JOB
/NOSEQ
JOB,T=10.
/ACCOUNT
ACQUIRE,DN=
  PROG1,UQ,TEXT=^
'USER,ABC1234,PASSWRD.
  ACQUIRE,PROG1.CTASK.'.
DELETE,DN=PROG1. *. WILL NOT
  WORK WITHOUT UQ
  PARAMETER ON ACQUIRE.
PASCAL,INPUT=PROG1.
LDR,LIB=$PSLIB.
```

Example 3:

```
/JOB
/NOSEQ
JOB.
/ACCOUNT
ACQUIRE,DN=
  PROG2,UQ,TEXT=^
'USER,ABC1234,PASSWRD.A,
  PROG2.CTASK.'.
DELETE,DN=PROG2.
*.
PASCAL,INPUT=PROG2,
  OUTPUT=OUT2,B=BIN2,
  LIST=OFF.
LDR,DN=BIN2,LIB=$PSLIB.
*.
DISPOSE,DN=OUT2,DC=ST,
  TEXT=^
'USER,ABC1234,PASSWRD.
  CTASK.RETAIN,OUT2.'.
```

In this example the compiler listing (error lines only) is saved (on the CYBER) on the default pack with filename OUT2. The loader output, execution output, and logfile are returned in the usual manner (filename=jobname on default pack). Note: the asterisks in Examples 2 and 3 indicate comments.

IMPLEMENTATION RESTRICTIONS:

1. Only files of type "text" are allowed; "file" is not a reserved word.
2. All files must be declared in the program block.
3. Files may not be passed as procedure parameters.
4. The type "file" is not implemented yet.
5. Read of "boolean" values is not supported yet.
6. Read of "real" values is not supported yet.
7. Procedures are not allowed as procedure parameters.
8. The target label of a "goto" must be in the same procedure body (a bug currently disallows jumps out of the body of a "with" statement).
9. All character strings are packed and the keyword "packed" is ignored.
10. The tag field must be specified in record variant declarations; the only accepted form is:

    ```
    case <identifier>:<type identifier>
      of
    ```

11. Only the first eight characters of identifiers are significant.
12. Data items longer than 64 words cannot be compared.
13. A semicolon preceding the "end" in a record declaration is flagged as a syntax error.
14. The following reserved words are not yet implemented:
    file
    pack/unpack
    round
    page

ENHANCEMENTS:

1. Octal constants of the form: **<octal digits>b**
2. Upper/lower case verbs (see implementation definitions below).
3. The following escape sequences within strings:

\\	=	\
\<B>	=	backspace
\<T>	=	tab
\<N>	=	newline
\<F>	=	formfeed
\<R>	=	carriage return

4. Sets may have up to 128 members. Set constructors of the form
   [<constant>..<constant>]
   are allowed, but the form
   [<expression>..<expression>]
   is not.
5. The following character substitutions are allowed:

REPORT		CRAY
and	=	& (ampersand)
or	=	| (vertical bar)
not	=	~ (tilde)
<>	=	=
[	=	(/
]	=	/)

For more information about non-standard predefined functions and procedures, as well as information on compiler options (pseudo-comments), see WRITEUP (CRAY = PASCAL).

IMPLEMENTATION DEFINITIONS

The range of integer arithmetic is -2**46...2**46. However, correct results are obtained for addition, subtraction, comparison, and reading/writing of integers in the range -2**63...2**63-I. Multiplication and division by constant powers of two are implemented with shifts and produce correct results in the same range.

The character set used is full ASCII. Predefined identifiers are recognized only when all characters are the same case (i.e., IF or if, but not If). User-defined identifiers (variable names, for instance) may be mixed case, but different case combinations are recognized as different identifiers (abc is different from ABC, which is different from Abc, etc.). This is also non-standard.

PLEASE NOTE:
This version of Pascal is not an official Cray release, and is not supported by any Cray personnel. Some work is being done on the Pascal system on an informal basis only, but no guarantee as to completeness, accuracy, or correctness of results is intended.

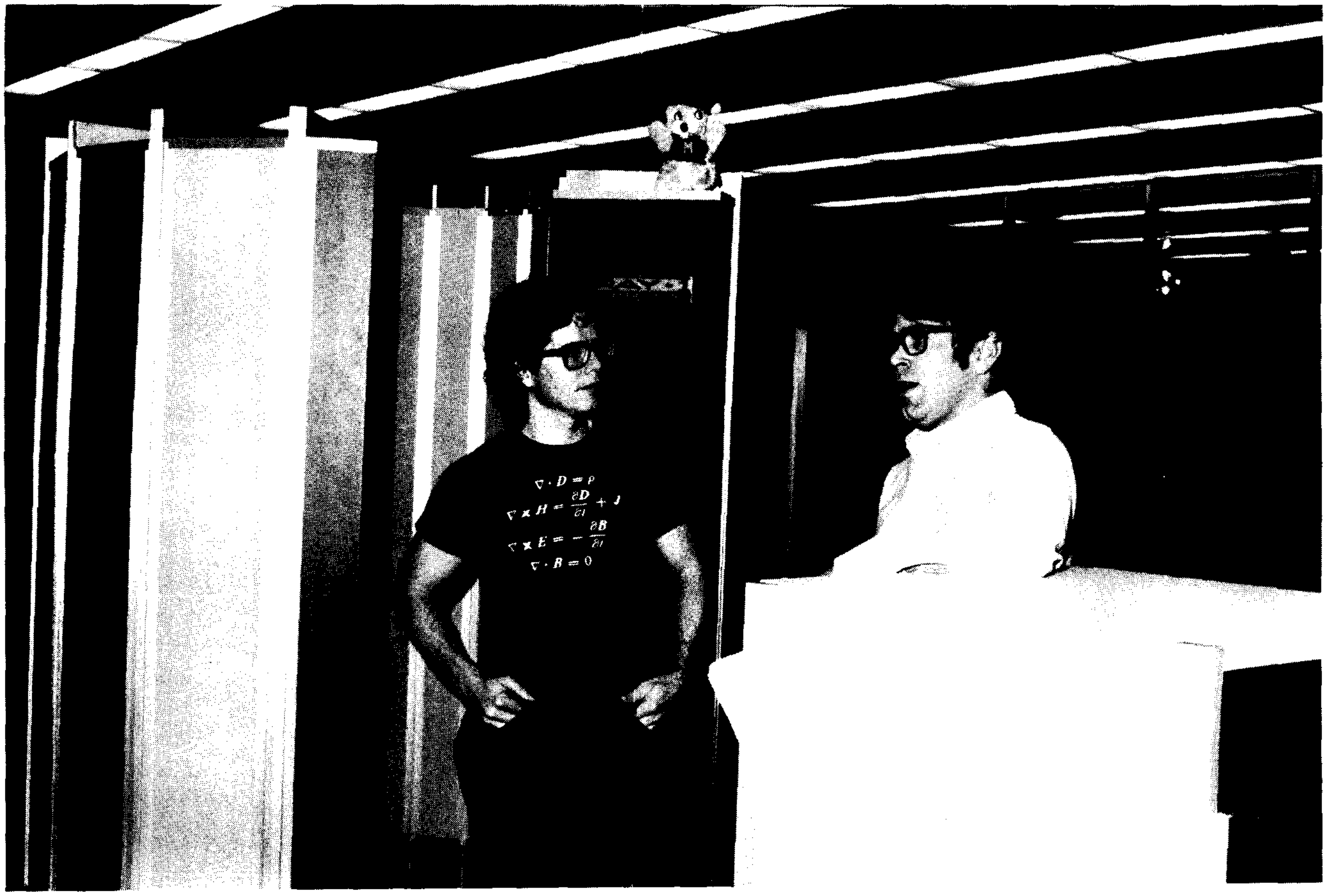
For more information see WRITEUP(CRAY = PASCAL).

CAL

CRAY-1 Assembly Language (CAL) is available for users who wish to write programs tailored to the hardware of the CRAY central processor. CAL consists of a set of instructions that symbolically express all hardware functions of the CPU and a set of pseudo-instructions for assembly options such as program block control, error control, data definition, macro and micro generation, and conditional assembly.

For more information on CRAY-1 Assembly Language see WRITEUP(CRAY = CAL).

(Susan Steffen)



Bill Sackett and K.C. Matthews are two of UCC's resident CRAY experts. (Photo: Dick Hotchkiss)

CRAY OPERATING SYSTEM NEWS

COS 1.11

We converted the CRAY-1 Operating System from 1.10 to bugfix 1 of COS 1.11 on October 25, 1982. COS 1.11 is now running on the CRAY along with the current 1.10 NOS station on the CYBER 730. When we go to NOS Release 5 on the CYBERs, we will run the NOS 1.11 station.

We made this change to take advantage of the large number of bugs that have been fixed in 1.11. The main difference between the two versions is the provision of magnetic tape support for CRAYs with an input-output processor (IOP) that the University's CRAY 1A does not have. Other than some added features, few differences should be apparent to current users of COS 1.10.

CRAY job deck structure will remain the way it is now until the CYBERs convert to NOS Release 5 in mid-December. This is contrary to the announcement in the September *Newsletter* (written in July when we thought that we would be running release 5 this fall). We will add up-to-date information to WRITEUP(CRAYINF) and WRITEUP(UPGRADE=CRAY). Our Computer Store (phone 373-4877) has some copies of the COS 1.11 manuals in stock and can order additional copies. They also have update packages for those of you with 1.10 manuals. If you have purchased a 1.10 manual from the store and have some verification of the purchase, the update packets are free of charge until December 1, at which time CRI is discontinuing its free update policy.

Running the 1.11 NOS station offers the following benefits:

—Datasets created on the CRAY by ACQUIRE will appear on AUDITs (they do not in 1.10 because the ACQUIRE had no way of knowing the owner of the dataset).

—Job cards with both a jobname and a time limit will be processed correctly (in 1.10 the time limit is ignored if a jobname is present).

—There are a number of problems with station instability and files never returning from the CRAY that we hope are corrected by the 1.11 station.

Additional features from COS 1.11 on the CRAY include:

—Station modifications allow users to check the status of their CRAY jobs from the CYBER. (Available with the December upgrade.)

—You need not specify your CYBER user number and password in the text fields of ACQUIREs and DISPOSEs. This may not be implemented at once in our version of COS 1.11 but we will do it as soon as possible for CYBER security. It may involve some change to the structure of your CRAY job decks.

—A symbolic interactive debugger, SID, that can also be used for batch jobs is provided. It is described in CRAY manual SG-0056, *Symbolic Interactive Debugger (SID) User's Guide*. We hope it will prove more useful than DEBUG in discovering the reasons for CFT problems.

—A text editor, TEDI, is available; it is described in CRAY manual SG-0055, *Text Editor (TEDI) User's Guide*. Its batch mode usability is uncertain at this time.

—An OPTION control statement has been added that may prove of use. You can specify a line page limit (LPP parameter) and turn on/off dataset statistics logfile (dayfile) information (STAT parameter).

—CHARGES has a new job name and user number, JNU, option.

—LDR can handle "relocatable overlays" (partially relocated modules) and has an SID="string" parameter for symbolic debugger loading.

—A SUBMIT control statement has been added that is intended to replace DC=IN on DISPOSEs.

—UPDATE has a new declared modifications option and inputs (DC).

—A CONTRPV macro has been added to continue normal job processing from within a reprieve subroutine.

—Other features affect tape usage; they will not be described here.

When we begin running COS 1.11 on the CRAY-1, we will update WRITEUP(CRAY) on the CYBERs to contain descriptions of the additional features mentioned above.

CTSS Evaluation

A July *Newsletter* article that described our evaluation of one means of CRAY interactive access (CTSS) generated only a negative response from some of our large CRAY accounts. UCC management has thus postponed any consideration of conversion from COS to CTSS until next summer at the earliest. A report of the evaluation committee's findings will be placed on WRITEUP(CTSS=EVAL) on November 15.

(*Bill Sackett*)

STREAMLINED GRAPHICS ON THE CRAY

We have installed the DISSPLA (version 8.6) and GRAFLIB graphics libraries on the CRAY, and they are easily accessible from the CYBER system. (This version of DISSPLA is not completely compatible with the VAX version and requires a different manual.) DISSPLA provides sophisticated graphing capabilities, while GRAFLIB offers simple, efficient capabilities. We recommend GRAFLIB, which includes routines from the MNCORE and PLOTPAC libraries, for special purpose or inexpensive two- and three-dimensional plots. (We plan to add CALCOMP routines early next year.)

We have written a CYBER procedure, GRAFCRA, that provides CRAY access to these general purpose graphing libraries. GRAFCRA asks for a program name, CRAY user number and password, and CYBER user number and password, then compiles and executes the program on the CRAY. With GRAFCRA you can run simple FORTRAN programs on the economical CRAY without learning a new operating system.

The graphics writeup CRAY describes two sample CRAY procedure files that show how to access the DISSPLA or GRAFLIB libraries, and shows how to use the simple GRAFCRA procedure and how to create a custom procedure by modifying the sample procedure files. To see this writeup on the VAX, TYPE the file:

usera:[writeups.graphics.general]cray

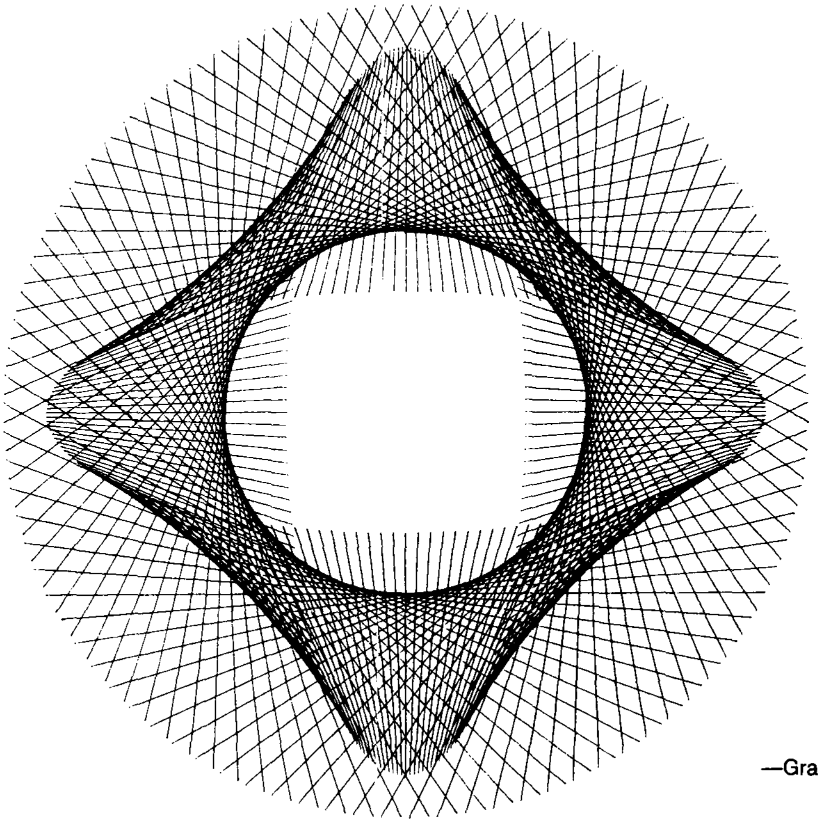
On the CYBERs, enter:

WRITEUP(GRAFGEN=CRAY)

The standard procedure (that run by GRAFCRA or an unmodified sample procedure file) clears any programs left over from a previous run, acquires your program from the CYBER, accesses the DISSPLA or GRAFLIB library (and package-dependent data files, if necessary), compiles and executes the program, and routes the resultant plot file back to the CYBER.

If you wish to execute a more complicated graphing task (chart a series of contour plots with data from the CYBERs, for example), take the DISSPLA sample procedure file, replace the program name and the CYBER user number and password with your own, then insert a command to acquire the data file(s) from the CYBER, and execute the procedure.

(John Cornelison)



—Graphic by Joe Cornell

All Systems Bulletins

GRANTS FOR RESEARCH

In our continuing effort to provide grant proposal assistance to our users and potential users, our Reference Room (140 Experimental Engineering) has material related to grants for your use. We will occasionally select information from these materials for this column. For more information or help with your proposal, please call Vicky Walsh, 373-5780.

From the *Federal Grants and Contracts Weekly*:

International Cultural Research —In the coming fiscal year, the Defense Department will be soliciting contracts for research on the cultures of countries in the Arabian Peninsula, Turkey, and Africa.

IBM will award $50 million in equipment and cash grants to update computer design and manufacturing curricula.

The Small Grants Program for pilot projects in Biotechnology will fund 10 to 20 pilot projects to study high technology and engineering advances in biomedical research.

Up to $150 million in university contracts will be awarded over the next five years to upgrade research instrumentation and diagnostic facilities (Department of Defense).

Under the Social and Economic Sciences Division of the National Science Foundation, the Measurement Methods and Data Resources program encourages proposals for research on surveys, quantitative analysis of social data, improving statistical social data, and developing and testing social indicators.

(*Vicky Walsh*)

NON-TRADITIONAL COMPUTING LECTURE SERIES

For those of you who have wondered what non-traditional computing is all about, UCC is sponsoring a series of lectures by University faculty and researchers that will discuss computing as a tool in a variety of disciplines. The scheduled lectures are:

November 17—"Word Frequency Distribution Studies in Medieval Literature," F.R.P. Akehurst, Department of French and Italian.
December 1—"The Old English Chronicle: Some Problems and Solutions," J. Lawrence Mitchell, Department of English.
January 19—"Computer Models of Race, Class, and Marriage in Old Mexico," Robert McCaa, Department of History.

The following talks will be given during the rest of the academic year, but dates have not yet been determined:

"Decipherment of the Mayan Hieroglyphs," Mary Ellen Kelly, Department of Anthropology.
"An Archaeological Data Base from Akhmim, Egypt," Sheila McNally, Department of Art History.
"Collection and Analysis of Set Phrases in Old Norse," Lewis Janus, Department of Scandinavian Languages.

For further information about this series, see this *Newsletter*, the Minnesota *Daily*, and the University's *Brief*, or call Vicky Walsh, 373-5780.

THANKSGIVING HOURS

Even our staff needs a break now and then—perhaps to join their families for a toothsome Thanksgiving dinner. Our operating schedule for the Thanksgiving holiday is as follows:

	Down	Up
Lauderdale	4 a.m. Thu 11/25	8 a.m. Fri 11/26
Experimental Engineering	Midnight Wed 11/24	4 p.m. Sun 11/28
Shepherd Labs (operator coverage)	5 p.m. Wed 11/24	8 a.m. Mon 11/29

FOR SALE

Texas Instruments Silent 700 data terminal (model 745). Portable, like-new condition, paper included. U of M department purchase only. $950. Call Stephanie Mitchell, 376-5297.

Fall Quarter Short Courses

INTRODUCTORY COURSES

PRICES: U. Student $10, U. Staff $20, Non-Univ. $30

015 Introduction to Computer Terms
Nov. 22-24 (MTW), 3:15-5 p.m.
035 Introduction to UCC
Dec. 2 (Th), 3:15-5 p.m.
120 Introduction to DBMS
Nov. 1 (M), 3:15-5 p.m.
130 XEDIT
Nov. 2-11 (TTh), 3:15-5 p.m.
150 Introduction to Programming
Nov. 2-18 (TTh), 3:15-5 p.m.
160 CP/M Software and UCC Systems
Dec. 2-9 (Th), 3:15-5 p.m.

ELECTIVES

PRICES: U. Student $20-30, U. Staff $30-50, Non-Univ. $50-100

510 Introduction to System 2000 (DBMS)
Nov. 1-17 (MWF), 3:15-5 p.m., $30-$40-$50
510 SIR (DBMS)
Nov. 2-18 (TTh), 3:15-5 p.m., $30-$40-$50
530 SPSS (Statistical Package)
Nov. 8-12 (MTWF), 3:15-4:30 p.m., $25-$40-$65
600 Beginning FORTRAN
Nov. 8-19 (MWF), 3:15-5 p.m., $25-$35-$60
620 Cray FORTRAN Features (CFT)
Nov. 9-18 (TTh), 3:15-5 p.m., $30-$40-$60
550 PROSE (Text Formatter)
Nov. 16-18 (TTh), 1:15-3 p.m., $20-$30-$50
560 Scribe (Text Formatter)
Nov. 15-19 (MWF), 3:15-4:30 p.m., $30-$40-$60
570 Text Processing on Micros
Nov. 22-30 (MT), 3:15-5 p.m., $25-$35-$60
580 Graphics
Nov. 29-Dec. 8 (MW), 3:15-5 p.m., $20-$30-$50
640 Introduction to Pascal
Nov. 2-18 (TTh), 3:15-5 p.m., $25-$40-$70
650 COBOL
Nov. 22-Dec. 10 (MWF), 3:15-5 p.m., $30-$45-$70
590 Cray Assembly Language (CAL)
Nov. 29-Dec. 3 (MWF), 3:15-5 p.m., $25-$45-$70

HOLIDAYS: Thursday and Friday, November 25-26, are University holidays. No classes will be held.

REGISTRATION

You can register for our short courses at the Computer Store, 211 Experimental Engineering (hours are 9 a.m.-3 p.m. Monday through Friday and 5 p.m.-7 p.m. Monday through Thursday). We'll accept mail registrations for an additional $1 fee per course.

A self-service registration terminal is located outside the store, and separate lines at the store window are designated for registrants and for regular customers. Course fees may be paid with cash, check, University journal voucher, or they may be charged to your non-instructional UCC user account.

If you have questions about our short courses or about registration, call Jerry Stearns, 376-8806.

PHONE NUMBERS

Budgets	373-2521
Computer-Aided Instruction	376-2975
Computer Hours (recorded message)	373-4927
Computer Store	373-4877
Consulting	
HELP-line	376-5592
9 a.m.-5 p.m., Monday-Friday	
Business Data Products	376-1761
1-3 p.m., Monday-Friday	
Statistics Packages	376-5062
1-2 p.m., Monday-Friday	
Data Bases	376-1761
1-3 p.m., Monday-Friday	
Microcomputers	376-4276
10-12 a.m. and 2-4 p.m., Monday-Friday	
Non-Traditional Computing	373-5780
10:30-11:30 a.m., Monday, Wednesday, Friday	
TELL-A-GRAF/DISSPLA	376-2663
1-3 p.m. Tuesday, Thursday	
Contract Programming	376-1764
Data Base Applications	376-1764
Educational Services	376-3963
EDUNET Liaison	373-7745
Engineering Services	376-1023, 376-8153
Equipment Purchase/Information	376-8153
Experimental Engineering I/O	373-4596
Field Engineering	376-7584
Graphics Software	376-5592
HELP-line	376-5592
9 a.m.-5 p.m., Monday-Friday	

HOURS-line (recorded message)	373-4927
Image Processing	376-2895
Information, Experimental Engineering	373-4360
Information, Lauderdale	373-4912
Information Systems	376-1764
Instructional Labs	376-2703
Job Status, ExpEng (recorded message)	373-4994
Lauderdale Computer Room	373-4940
Lauderdale Services	373-4995
Lauderdale Services Manager	373-7538
Lauderdale Users' Room	373-4921
MECC Liaison	373-4573
Newsletter Subscription	373-4912
Permanent File Restoration	376-5605
Professional Services Division (PSD)	376-1764
Project Assistance	376-1764
Program Librarian	376-1636
Programming Languages	376-7290
Reference Room	373-7744
Remote Batch (RJE) Services	376-2703
Short Courses	376-8806
Shuttle Bus Service	376-3068
System Status (recorded message)	373-4927
Tape Librarian: see Lauderdale Services	
Text Processing Services	376-2943
User Accounts	373-4548
User Services	376-3963

OPERATING HOURS

	Cyber 74/730	Cray	MERITSS (Cyber 172)	VAX
M-F	8 a.m. - 4 a.m.	8 a.m. - midnight	7:45 a.m. - 1:30 a.m.	8 a.m. - 6 a.m.
Sat	4 a.m. - 5:15 p.m.	8 a.m. - 5 p.m.	7:45 a.m. - 1:30 a.m.	24 hours
Sun	4 p.m. - 1 a.m.	4 p.m. - midnight	4 p.m. - midnight	24 hours

PUBLIC LABS—TWIN CITIES CAMPUS

Location	Batch	Interactive	Micro
East Bank			
Arch 160		X	X
CentH		X	
ComH		X	
DiehlH 270, 207		X	
EltH 121, 125		X	
EltH N640	X		
ExpE 130	?	*	
FolH 14, 14a		X*	X
LindH 25		X	
MasCan 39	X		
MechE 308		X	
Physics 69		*	
SanfH		X	
TerrH		X	
VincH 4		X	
WaLib 204		X	

Location	Batch	Interactive	Micro
West Bank			
BlegH 90	X		
BlegH 91T			X
BlegH 140		X	
MdbH		X	
OMWL 2		X	
SocSci 167		X*	
St. Paul			
BaH		X	
ClaOff 125	X	X	
NorH 24	X		
Lauderdale			
Users Room	X		

* Research cluster; access to Cyber 730 and VAX/VMS

X in interactive column indicates access to MERITSS

? Unknown at the present time

University
Computer
Center
Newsletter

User Services
227 Experimental Engineering
University of Minnesota
208 Union Street SE
Minneapolis, Minnesota 55455