UNIVERSITY COMPUTER CENTER

UNIVERSITY COMPUTER CENTER, UNIVERSITY OF MINNESOTA-TWIN CITIES, MINNEAPOLIS, MINNESOTA

CONTENTS

DEADLINE FOR THE OCTOBER ISSUE OF THIS NEWSLETTER IS SEPTEMBER 25.

BULLETINS

CYBER 172 HOURS EXTENDED:

For September, production hours on the Cyber 172 have been extended to 2400 hours Monday through Friday.

Beginning on October 1, Cyber 172 hours will be further extended to be the same as Cyber 74 hours.

##

System 2000 consulting:

Fall quarter 1978:

10:00 am -11:00 am Monday-Friday
1:00 pm - 2:00 pm Monday-Friday

Call 376-1761 for consulting assistance.

##

DON'T FORGET TO LOOK AT THE LATEST WRITEUP,UPGRADE DOCUMENTS:

```
WRITEUP,UPGRADE=CONTROL.
WRITEUP,UPGRADE=MMF.
WRITEUP,UPGRADE=CHAR.
```

UCC newsletter
VOLUME 12 NUMBER 9 SEPTEMBER, 1978

Director: Peter C. Patton
Editor : A. Koepke

Comments about the content of this newsletter, or suggestions for changes may be directed to the editor, 235a Experimental Engineering, 373-7744.

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, age, or national origin.

SYSTEMS & OPERATIONS SUMMARY

Mass Storage

Three additional double density disk drives arrived in late June and one more will arrive for September. At present, the final mass storage configuration for the multi-mainframe Cyber 74/172 will be as follows:

#	name	type	used for
Cyber 74:			
8	unnamed	DJ	User and staff direct and indirect files, temporary and rolled files.
4	unnamed	DJ	User and staff direct files, temporary and rolled files.
1	PF01	DJ	Large user and accounting files, temporary and rolled files.
1	removable	DI	Normal user or system packs of files.
Cyber 172:			
4	unnamed	DL	User and staff direct and indirect files, temporary and rolled files.
1	unnamed	DL	User and staff direct files, temporary and rolled files.
1	removable	DI	Named user or system packs of files.
Cyber 74/172:			
1	SPL	DJ	System files and libraries.
1	UCC	DJ	Scratch system pack; replacement for DL or DJ drives.
1	SHA	DJ	Shared user and staff files.
1	removable	DI	Named user or system packs of files; replacement for DI drives.

DI: single density drive; 110M 6-bit characters; 470,000 char/sec transfer rate.
DJ: Double density drive; 235M 6-bit characters; 470,000 char/sec transfer rate.
DL: Double density drive; 235M 6-bit characters; 940,000 char/sec transfer rate (172 only).

The total mass storage capacity is thus 21*235M + 3*110M or approximately 5.25 billion characters, of which approximately 4 billion is available for user files.

ASCII printers

Currently, a CDC 580-120 and a CDC 512-1 are available at Lauderdale to print the extended ASCII character set with upper and lower case characters. A CDC 512-1 printer will be delivered to Lauderdale this month; when it arrives, the CDC 580-120 will be transferred to Experimental Engineering to give ASCII character printing there. At this time, two CDC 501 printers that have given us good service for 10 years will be retired (scrapped).

Timesharing ports

With the addition of sixteen ports to the PDP11 front-end, approximately 50 to 60 ports will be available on the Cyber 74 and up to 68 will be available on the Cyber 172:

rotary	speed	hardware	ports	machine
376-5730	autobaud	2551	43	Cyber 74
Duluth	autobaud	PDP11	5	Cyber 74/172
Morris	autobaud	PDP11	1	Cyber 74/172
376-1850	1200	PDP11	2	Cyber 74/172
376-5706	1200	PDP11	3	Cyber 74/172
376-5930	autobaud	PDP11	36	Cyber 172
Laud	1200	PDP11	1	Cyber 74/172

During the next year, we should see only small equipment additions. We feel that we all deserve a quiet year of consolidation after the very interesting April through August period this year.

L. Liddiard, 373-5239

Multi-mainframe facility

A new section describing the multi-mainframe facility has been added to the UPGRADE writeup. You can get copies with the following control statement

```
WRITEUP,UPGRADE=MMF.
```

The MMF writeup describes the CDC multi-mainframe facility of the NOS operating system. This facility lets you save permanent files on the auxiliary permanent file device named SHA. These permanent files can then be accessed from either the Cyber 74 or the Cyber 172. Since the files on a shared device can be accessed from either machine, they are charged for on both machines. Thus, they cost twice as much as other permanent files.

Next month, a facility will be added to the system that will allow input and output queue files to be placed on a shared device on one machine and then processed on the other. Although this facility does not work well yet, a limited amount of shared I/O queue processing is provided by the TRANSIT subsystem, also described in the MMF writeup.

User ECS

The amount of available user ECS has been reduced from 300,000B words to 250,000B words. The other 30,000B words are used by the operating system and the shared disk management. The total amount of ECS used for storing frequently used programs was also reduced. These changes were necessary since all three computers share the ECS.

Output from submitted jobs

Those of you who use the SUBMIT control statement have had to take special pains to ensure that information from the submitted job was not lost. This task is now much easier than it used to be. Many submitted jobs had sequences like the following at the end of the control statement record:

```
GOTO,1D.
EXIT.
1D,DISPOSE.....
DAYFILE,DAY.
REPLACE,DAY.
```

This is now unnecessary. The dayfile from submitted jobs is now appended automatically to a file named DAYFILE. One logical record is appended to the file for each submitted job processed. If no permanent file named DAYFILE exists, one will be created. The file DAYFILE is a permanent file in the permanent file catalog belonging to the user number appearing in the submitted job. (the file may have been submitted from a different user number).

The file DAYFILE will not grow without limit as jobs are submitted, but will reach a maximum size of 192 sectors. Thus, users who wish to retrieve a job's dayfile must purge the file DAYFILE from time to time.

Remember that the last record on the file DAYFILE will be the most current job dayfile. The following control statements will, in the BATCH subsystem, copy the latest dayfile to a timesharing terminal:

```
    GET,DAYFILE.
    SKIPEI,DAYFILE.
    BKSP,DAYFILE.
    COPY,DAYFILE.
```

The deferred ROUTE facility can be used to ensure that the output from the job will be sent to the proper site at the end of the job. (Remember that ROUTE replaces DISPOSE!)

The deferred ROUTE specifies that a certain file is to be processed at a certain site when the job terminates. This processing will occur if the job terminates normally or if it aborts. The statement

```
    ROUTE,OUTPUT,DEF,DC=PR,TID=BC,BIN=999.
```

establishes OUTPUT as a deferred print file. The TID and BIN parameters are used to specify where the output will go. The DEF parameter specifies a deferred ROUTE. The file is created if it did not exist already. Then, any data written to OUTPUT will be properly processed when the job terminates. If a ROUTE statement is placed after the USER card on all submitted jobs, you will be assured of getting output from those jobs. Remember that any file name, not just OUTPUT, can be routed.

*** WARNING ***

You cannot yet do a deferred ROUTE from one machine to another. On the 172, this means that your deferred ROUTE files will print only at site BC (Lauderdale).

* * * * * *

Which computer am I on?

Users are beginning to develop procedure files that do different things, depending upon the computer system on which the job is running. How then, can you check to see which machine you are on?

The answer is the MID control language variable and the LIT control language function. For example

```
    IF(MID=LIT(74)) STATEMENT
```

will execute the STATEMENT if the job is running on the Cyber 74. Either 72 or 64 can be substituted to check for these machines.

Why this odd construction? Because the control language really 'thinks' only in terms of numbers, not characters. The machine ID is a 2-character field. Thus, MID is the character string "72" which is the octal number 4235. Thus, you could say

```
    IF(MID=4235B)....
```

but this looks ugly.

The LIT function was added to the control language several years ago for dealing with a similar problem with terminal IDs. (Terminal IDs are also 2-character fields that are 4-digit octal numbers to the control language.) The LIT function converts a readable string of from 1 to 3 characters into the corresponding number. Thus, LIT(72) is really 4235B, and that's why the whole statement works.

Here are the machine IDs:

Cyber 74/MIRJE	: ID is 74
Cyber 172	: ID is 72
6400/MERITSS	: ID is 64

K.C. Matthews, 376-5605

GRAPHICS PACKAGES CONVERTED

On August 20, when the character set change was implemented, most graphics packages that print or plot characters were updated to use the new character set. The only package that did not get converted is PASPLOT, which will be converted sometime in the middle of September. As a result of this change, some users may find out that their programs need to be modified so that the same characters will be printed or plotted under the new character set. A table showing both the new and old characters is printed below, and can be used to determine whether or not programs need to be changed.

OLD character	NEW character
equivalence	#
not equal	"
right arrow	underline
down caret (or)	!
up caret (and)	&
up arrow	'
down arrow	?
less than or equal	@
greater than or equal	\
triangle	^
question mark	;

This change affects users of PLOTPAC, PLOT3D, PLOT3S, TEKLIB, and HTEKLIB, though users of the TEKLIB or HTEKLIB packages should not need to modify their programs unless they use Hollerith constants containing characters that changed. In mid-September, when the PASPLOT package is updated to the new character set, Pascal programs that use

PASPLOT may also require modification so that the same characters are plotted.

VIEW and PLOT31 were also modified to plot the new characters.

Additionally, on September 25, the PAST version of TEKLIB will be removed from the system because there are no plans to update it to the new character set. All FTN3 TEKLIB users are advised to re-compile their programs with either FTN4 or MNF and use FETCH(TEKLIB).

D. Germann, 376-5262

SYSTEM 2000

The following is a suggested method for moving very large System 2000 data bases from the Cyber 74 to the Cyber 172.

bases.

Run the following SUBMIT job on the Cyber 74:

```
/JOB
MYJOB(CM55000,T100)
USER(usernum,passwrd)
LABEL(TAPE999,VSN=..........)
S2000(TP)
/EOR
USER,MYPASS:
DBN IS MYDB:
CONTROL:
SAVE DATA BASE ON TAPE999:
EXIT:
/EOR
/EOI
```

Then, run the following SUBMIT job on the Cyber 172:

```
/JOB
MYJOB(CM55000,T100)
USER(usernum,passwrd)
LABEL(TAPE999,VSN=..........  )
S2000(TP)
/EOR
USER,MYPASS:
RESTORE MYDB FROM TAPE999:
EXIT:
/EOR
/EOI
```

If you have any questions on this, please call

John Cosgrove, 376-1761

KWIC, TRIAL, S2KIND

Beginning Fall quarter, 1978, the procedure file, S2KIND, will be retired from the operating system. This procedure is no longer compatible with System 2000 Version 2.60 and the NOS operating system.

Also beginning Fall quarter, FUTURE(TRIAL) and FUTURE(KWIC) will become the current and only versions of these products. Questions regarding these products may be directed to

J. Cosgrove, 376-1761

NEW STATISTICS PACKAGE

Starting with the Fall quarter, there will be a new statistical package, MINITAB, available on the system. MINITAB, written at Penn State University, incorporates many of the characteristics of OMNITAB. It is designed to handle small to medium data sets, and is easy to use.

The package is accessed by this BATCH command:

```
MINITAB.
```

The version that we have is appropriate for interactive use. It is possible to change this to be more suitable for Batch processing by issuing the appropriate MINITAB commands immediately after accessing the package.

A MINITAB manual, published by UCC, will be available shortly. For information or questions on MINITAB, call

Betty Hinkley, 376-2773

NEW DEBUGGING PACKAGE

New program debugging package

Program DEBUG, a convenient, easy to use debugging package, is now available via FETCH,DEBUG. DEBUG provides methods for simulating or breakpointing CPU execution, changing the program's memory or registers while executing, tracing RA+1 requests, and tracing the usage of particular registers or instructions. Notably, DEBUG may be used with both absolute and relocatable programs without recompiling or modifying the programs in any way. DEBUG is an excellent tool for finding errors in COMPASS programs and for learning the COMPASS language. For more information on this package see the document available with the control statement:

```
WRITEUP(CPDEBUG).
```

Brian Hanson, 376-5262

WANT ADS

FOR SALE: One ASR-35 teletype, paper tape. Contact J. Doyle Britton, Department of Chemistry, 373-2382.

A NEW VERSION OF MANTRAP

MANTRAP AT A TERMINAL

The FUTURE versions of MNF and TSF on the Cyber computers now access a new version of Mantrap. The changes in this version do not affect batch jobs, but have a dramatic effect on MNF jobs run from a time sharing terminal. The time sharing version of Mantrap is now interactive - this enables the user to decide which variables and arrays to inspect in any desired subprogram. The need for "scrolling" the output is now removed for users of video screens because, if a value disappears off the top of the screen, it can easily be listed again. When a terminal job fails and enters the new Mantrap, the usual messages describing the cause and location of the error, and the items in current use, are first printed. This is normally the most useful part of the Mantrap output, and is not under interactive user control. After these messages "INTERACTIVE MANTRAP" will be printed, followed by "TYPE ROUTINE NAME OR A *COMMAND" and then a question mark. The question mark indicates that a response from the user is required.

Let us assume that the name of a Fortran subprogram in the job is typed in response. Mantrap will then print one of three messages:

(1) "SUBROUTINE (or FUNCTION) name WAS NEVER CALLED" (if the subprogram name given was never executed). This will be followed by the message "TYPE ROUTINE NAME OR A *COMMAND" because there is no point in trying to inspect the variables in a subprogram that was never called.

(2) "name IS THE MAIN PROGRAM" (if the subprogram name given was the name of the main program).

(3) "name WAS LAST CALLED FROM LINE NUMBER n OF caller" (where "caller" is the name of the subprogram which last called subprogram "name").

Unless message (1) was printed, Mantrap will then print "TYPE VARIABLE NAME OR A *COMMAND" followed by a question mark. At this point the name of any variable in the subprogram may be entered, and Mantrap will print the value of the variable and then repeat the message (or "prompt"). In this way, any variables or arrays in the subprogram may be listed.

Let us now assume that the variables in some other subprogram are to be inspected. In response to the prompt "TYPE VARIABLE NAME OR A *COMMAND", *ROUTINE must be entered (this is needed because Mantrap would otherwise think that the new routine name is the name of a variable in the current subprogram). If *ROUTINE is typed, Mantrap will again respond with the prompt "TYPE ROUTINE NAME OR A *COMMAND" and a question mark. The name of the new subprogram may then be entered, and its variables inspected as before.

When enough information has been obtained, *EXIT may be typed in response to any prompt - this will exit from Mantrap, and the terminal will be ready for the next user command. Note that *EXIT is the only way to get out of Mantrap - typing the "s" key while Mantrap is printing will simply suppress the rest of the information to be printed, and then repeat the previous prompt message. A description of valid responses ("type-ins") and their effects follows.

There are two "modes" when Mantrap is working interactively. These are "routine mode" and "variable mode", and correspond to the prompts:

```
    TYPE ROUTINE NAME OR A *COMMAND
and
    TYPE VARIABLE NAME OR A *COMMAND
```

With one exception, any *command is a valid response to either prompt. The *commands are:

*EXIT - this is THE ONLY WAY to exit from Mantrap

*HELP - print a 30 line note on the use of interactive Mantrap

*LONG - cause the prompt messages to be printed (reverse the effect of *SHORT)

*ROUTINE - return to routine mode from variable mode (required if a new subprogram is to be inspected)

*SHORT - suppress the two prompt messages (the question marks are not suppressed, nor are any error messages)

*TRACE - print a list of all of the user routines in the job (library routines are not included). The list contains the name of the routine and the location (source listing line number) from whence it was last called. If the routine was never executed (called), then this fact is stated.

*VARIABLES - list the values of all the variables in the current routine. The format is similar to the current Mantrap output at a terminal. Note that *VARIABLES is not a valid response when in routine mode - obviously, the routine name must be given before its variables can be listed.

Since Mantrap looks at only the * and the first letter of the commands, they may be abbreviated as desired - for example, *E is the same as *EXIT. The following are the valid responses to the prompt "TYPE ROUTINE NAME OR A *COMMAND":

(1) Any *command except *VARIABLES (*ROUTINE simply repeats the prompt).

(2) The name of a Fortran subprogram in the job (if the main program does not have a PROGRAM statement, then ST. will be the name of the main program). A correct subprogram name will lead to the second prompt:

"TYPE VARIABLE NAME OR A *COMMAND" - to which the valid responses are:

(1) Any *command.

(2) The name of any variable in the current routine - the name and value of the variable will be printed.

CONTINUED ON PAGE 72

MEMO FROM UNIVERSITY COMPUTER SERVICES

UCS INTRODUCES MEANS OF FUNDING FOR MICROCOMPUTERS

(The following is a copy of a memo from University Computer Services, dated July 5, 1978.)

There is a growing interest in the acquisition of microcomputers, particularly the Terak system, on which the staff of the Special Interactive Computation Laboratory (SICL) has done a considerable amount of development. Research users who are funded can make use of the University's contract for the purchase of Teraks in order to take advantage of the lower purchase price. Other users, particularly instructional users who have limited funding, are beginning to ask for a Terak computer with funding provided by University Computer Services.

Under current circumstances and policies, funding for the acquisition of microcomputers by departments and colleges is the responsibility of those units, utilizing supply, expense, and equipment budgets or other resources available to them. University Computer Services 0100 funding for equipment has to a large extent been limited to two categories: 1) purchase of major systems (including peripheral equipment) which have a general purpose function, that is, one which is available to university users generally; and 2) equipping of the instructional computing laboratories with terminals for time-sharing use and occasionally with an RJE terminal to the Cyber 74.

With regard to Terak computers for instructional use, some systems have been placed "on loan" in exchange for an understanding to do applications development of general University interest. The current interest is largely one of implementation; that is, actual instructional or research use of micro-computers. Purchase of such systems for transfer to dedicated department use lies outside the UCS "charter" by existing policy. It may be possible to have a policy change; that would require discussions both within the UCS Advisory Committee and with Vice-President Koffler.

In the meantime, it may be possible to meet some of the current needs in another way. UCS, chiefly through the University Computer Center, does have a rental arrangement. To date, it has been used primarily for Teletypes and RJE terminals. We can expand and modify the rental arrangement; expand it to include microcomputers or component equipment, and modify it to allow for rental accrual towards purchase (which present rental arrangements do not include).

As an initial proposal, which would be subject to later modification by mutual agreement of the user and UCS as we develop experience, an agreement would consist of the understandings listed below.

- A user could lease microcomputer equipment (or a graphics terminal, to give another example) at a monthly lease rate of 5% of the purchase price.
- Maintenance of the equipment could be provided at a monthly rate of 1.0% of the purchase price.
- The user may purchase the equipment at any time, with a discount equal to 100% of lease accrual.
- Upon completion of the leasing period or upon purchase, the system becomes departmental property. However, the user would be expected to sign a separate maintenance agreement, at the rate of 1.0% of the purchase price, or provide maintenance for itself.

Hopefully, the above plan will enable most users to get under way. Proposals may, as in the past, be submitted either to Bill Franta or Frank Verbrugge. Proposals will be reviewed primarily from the point of view of establishing priorities, because it is becoming evident that the demand for Terak computers is exceeding the supply and a waiting period following the submission of an order may develop.

The program will not be limited to Teraks; however, it will not be open-ended. It may be possible to add other systems or components to an approved list as time goes on. For example, one or more Tektronix terminals may be added in the near future. Another candidate might be the Plato terminal.

Frank Verbrugge, Director

The PTR documents that record problems with our computer systems are now being updated to include all recently discovered bugs.

Why don't you look at these PTR reports?

PTRNOS	*problems with the operating system*
PTRS2K	*problems with data base products*
PTRSTAT	*problems with the statistics packages*
PTRMISC	*any other reported problems*
PTRFORT	*problems with the Fortran compilers and the Fortran libraries*

PRODUCTION USAGE SUMMARIES

	July, 1978	July, 1977
CDC Cyber 74		
Number of Batch jobs and MIRJE sessions	80,478 (89,015)	71,402 (80,630)
Total Central processor hours inc. DELAY	207 (255)	160 (197)
DELAY queue processor hours	65 (72)	39 (41)
MIRJE terminal hours	10,077 ( )	6,905 (8,277)
Mass storage transfers (KPR)	231,050 (315,827)	200,921 (248,857)
Magnetic tape transfers (KPR)	5,931 (8,208)	5,494 (7,624)
Pages printed, charged from UCC	808,754 (895,328)	707,047 (804,594)
Cards punched	497,815 (514,393)	369,390 (385,881)
Microfilm frames produced	19,514 (276,162)	7,739 (225,724)
Statos plotting production (feet)	8,266	8,588
Tapes mounted	9,636	9,199
Average file storage (2347M available)	1,767.1 million char	1,121.8 million char
Mean time between failures	27.2 hours	26.3 hours
Available during scheduled hours	97.3 percent	98.2 percent
SUPIO uptime during available hours	96.5 percent	96.9
(totals in parentheses include staff development, accounting, and maintenance runs)		
CDC 6400		
Number of jobs run	92,064	112,615
Central processor hours	103	138
MERITSS terminal hours	11,253	14,693
Number of terminal sessions	21,859	27,853
Maximum number of simultaneous users	67	93
Average file storage	327.8 million char	267.0 million char
Mean time between failures	92.8 hours	57.1 hours
Available during scheduled hours	99.3 percent	99.6 percent

DOWNTIME SUMMARY: August, 1978 (Column 1, Cyber 74 : Column 2, Cyber 172)

	Monday-Friday 0800-1800		other		total	
Total possible scheduled uptime hours	230.0	230.0	310.0	137.0	540.0	367.0
Total downtime hours (see Schedule A)	11.1	7.6	3.5	1.5	14.6	9.1
Total uptime hours	218.9	222.4	306.5	135.5	525.4	357.9
Uptime (percent)	95.2	96.7	98.9	98.9	97.3	97.5
Average downtime per occurrence (minutes)	13.8	26.8	18.9	11.3	15.3	21.8
Mean time between failures (hours)	5.0	19.2	34.4	22.8	9.8	20.4
Subsystem failures						
SUPIO	22	-	9	-	31	-
TELEX	4	0	1	0	5*	0**
EXPORT	7	-	5	-	12	-

Schedule A: downtime hours

	Number		Total hours		Average minutes	
(1) Preventive maintenance over-runs	1	2	0.1	0.1	2.0	4.0
(2) Software related problems	47	8	11.8	1.1	15.1	8.4
(3) Hardware related problems	5	5	1.1	3.7	13.0	44.0
(4) Indeterminate software/hardware problems	2	3	0.5	1.2	14.0	24.7
(5) External Problems	2	7	1.1	3.0	34.5	26.7

*TELEX front-end failed once.

**TELEX front-end failed 3 times. On Tuesday, August 29, the front-end failed and was down all day.

SUBMISSION SITE USAGE SUMMARY: TELEX EXCLUDED : August, 1978

submitted from	total jobs	%	pages printed	%	cards read	%
Lauderdale	3,727	9.5	281,733	24.0	1,564,710	18.8
ExpEng	8,398	21.5	276,438	23.6	2,903,766	34.9
West Bank	3,601	9.2	80,433	6.9	923,573	11.1
6400	688	1.8				
SUPIO	22,630	58.0	534,614	45.6	2,926,136	35.2
TOTALS	39,044		1,173,218		8,318,185	

(3) The name of an array in the current routine typed without subscripts, the name and declared subscripts of the array will be printed.

(4) The name of an array in the current routine typed with subscripts - the name, subscripts, and value of the array element will be printed. The number of subscripts must be the same as the number of dimensions declared for the array, and each subscript must lie within the array dimensions. Note that the subscripts must be numbers - not names.

(5) A range of an array in the form:

    arrayref1...arrayref2

    This will print the name, subscripts, and values of all elements in the range. Note the use of ... to indicate an array range. The number and values of the subscripts must be correct as in (4), and arrayref1 must be nearer the start of the array (have a lower address) than arrayref2.

For convenience, as many names and *commands may be typed on one line as desired. Names and *commands must be separated by commas, for example:

```
*ROUTINE,BRITAIN,RULES,THE,WAVES(1)...WAVES(20)
```

would print the values of RULES and THE and the first 20 elements of WAVES as defined in routine BRITAIN.

Blanks are ignored in the responses, which may be up to 140 characters long. In addition, where one comma may appear, any number may be typed; and instead of ... any number of periods may be typed (even just one). Thus, it it possible to gain a lot of information just by typing a few characters. All real errors are detected by error messages - these are (believed to be) self explanatory, and are not listed here. Depending on the error, either the incorrect item will be ignored and the other items (if any) will be processed - or, if Mantrap cannot decode the line, the rest of the line will be ignored. Typing the "s" key while the terminal is printing will also usually cause the rest of the line to be ignored.

As described in WRITEUP(MANTRAP), the LGO(OP=F) load card parameter may still be used to over-ride the interactive features, and obtain the full Mantrap listing at a terminal.

Please address comments to C F Schofield at 227 Experimental Engineering.

RETURN TO:
UNIVERSITY COMPUTER CENTER
227 EXPERIMENTAL ENGINEERING
UNIVERSITY OF MINNESOTA
208 UNION STREET SE
MINNEAPOLIS, MN 55455