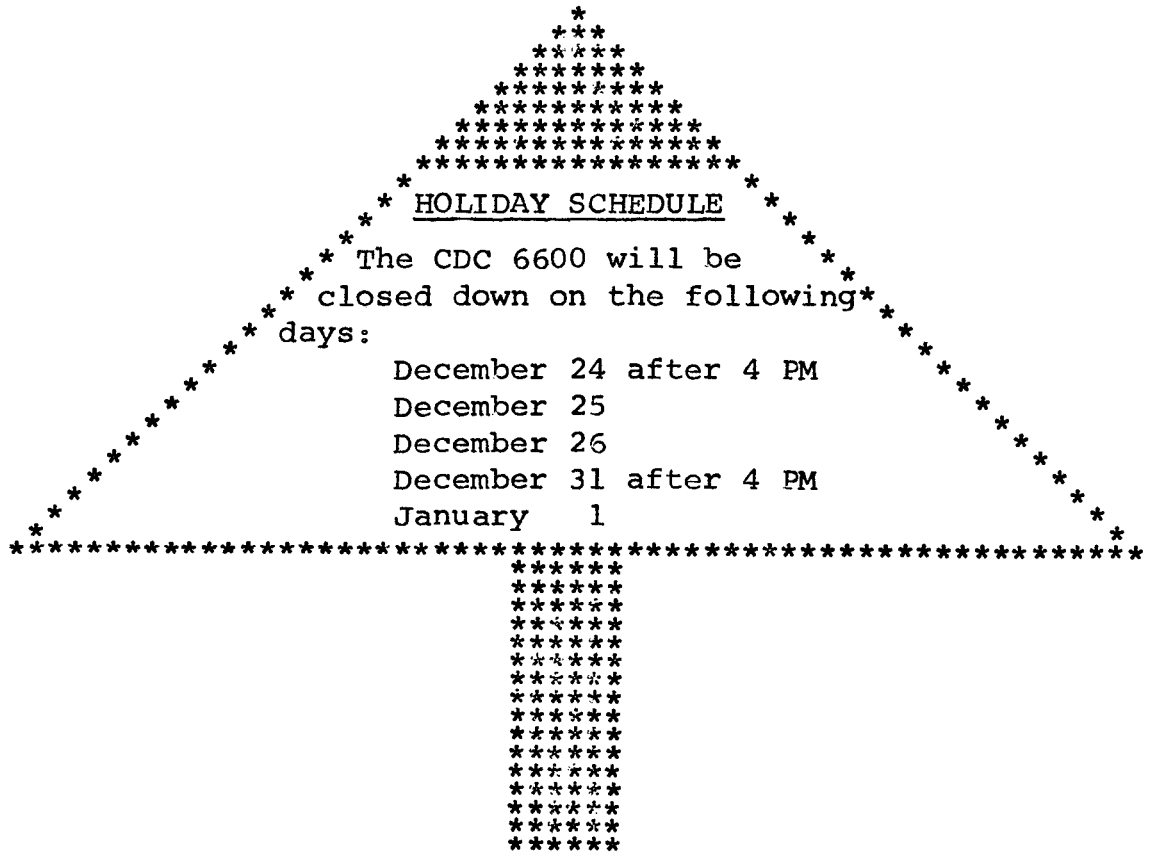
UNIVERSITY COMPUTER CENTER

NOTES & COMMENTS

Volume 4, Number 2
December, 1969

UNIVERSITY OF MINNESOTA
MINNEAPOLIS, MINNESOTA

TABLE OF CONTENTS



HOLIDAY SCHEDULE

The CDC 6600 will be closed down on the following days:

December 24 after 4 PM
December 25
December 26
December 31 after 4 PM
January 1

6600 NOTES

Fortran Compilers

We recommend the use of the FUN compiler since it produces better object code.

ECS

The direct access user of Extended Core Storage now has 300K words available for his use.

There is an apparent conflict in the use of the long tape driver with ECS. See Bill Franta (210 Exp. Eng.) with regard to any problems and questions.

Free Compilation & Execution Runs - MNF

As previously announced, Sunday, December 21, 1969 from 1 PM to 5 PM at the Lauderdale site, the UCC is giving free time for compilation and execution runs of under 5 minutes to test the new MNF compiler.

The greatest advantage to the user of this compiler will be in diagnostic information of more clarity and scope than that available on the current CDC Fortran compilers.

The purpose of this free time is to check out the compiler and give us added confidence in MNF's ability to compile and execute under standard operating conditions with normal jobs.

Exit Mode

Many programmers have noticed that the EXIT MODE occasionally changes in the EXCHANGE package without a MODE control card. This defficiency has been removed by a wire change in the exchange jump logic.

FORTRAN SHORT COURSE

The University Computer Center will be offering a short course in Fortran in January, 1970 No registration is required and no fees are charged. The days and hours are as outlined below:

DAYS : January 12-23 (2 weeks, Monday through Friday)
HOURS: 4:00 - 5:00 PM
ROOM : 109 Agricultural Engineering (St Paul Campus)

FROM THE WEST BANK

Effective Winter Quarter, 1970, an accounting system will be operational on the CDC 3200 computer. Any person wishing to make use of the 3200 system must possess a valid account number.

Application forms may be procurred from either the Social Science Research Facilities Center (25 Blegen Hall) or the Management Information Systems Research Center (93 Blegen Hall). Applications should then be filed with the Operations Manager, Phyllis Sympkins.

Users who wish to reserve output bins for the Winter Quarter must contact the West Bank Station I/O clerk, Barb Anderson Classes and research projects which will be generating many computer jobs throughout the quarter should reserve a bin for their particular use during the quarter.

LIBRARY CHANGES & ADDITIONS

BMD Changes

BMD06M - file name correction plus minor correction
BMD09S - file name correction
BMD10S - file name correction
BMD04V - file name correction plus minor correction
BMD05V - file name correction
BMD06V - file name correction

CONSULTANT'S CORNER

Described below are some of the more frequent problems brought to the consultant's desk:

1) Buffer Parameter Error
BUFFER PARAMETER ERROR is a dayfile message that has been occuring after LGO on a small percentage of jobs. In most cases the solution is to simply run the job again. This is an error in the loader and should disappear when the UCC implements the planned new loader.

2) Mode Mixing
Real and integer variables and constants may be mixed in almost any combination on the right-hand side of a replacement statement (see CDC Fortran Manual, Chapters 3 & 4). In the following cases, however, it is very unwise to mix modes:
   a) In data statements: DATA(X=1) will not be flagged as an error but it is not equivalent to DATA(X=1.0)
   b) In print statements: Integer variables should be printed under I fields and real variables should be printed under E, F, or G fields. Any other combination produces garbage.
   c) In subroutine calls: The parameter list in a CALL statement must agree in type with the corresponding parameter list of the subroutine. For example

```
CALL ABCDE(X1,A(N),I)
:
SUBROUTINE ABCDE(A1,A2,KK)
```

   If the types do not agree your results are so much trash. I suggest you discard them and try again.

3) Variable Dimension Statements
Variably dimensioned arrays in subprograms should have all except the last dimension the same as the corresponding dimensions in the calling program. (This applies only to doubly or triply dimensioned arrays.) For example, the following sequence of code will usually produce bad results

```
PROGRAM MAIN(INPUT,OUTPUT)
DIMENSION A(100,100)
N=50
CALL EVAL(A,N)
:
SUBROUTINE EVAL(B,N1)
DIMENSION B(N1,N1)
```

However, DIMENSION B(100,N1) will work properly.

JOHN NORSTAD