

UNIVERSITY OF MINNESOTA COMPUTER CENTER

Deadstart Systems Newsletter

13 July 1976

Vol. 2, No. 13

Send all comments, criticisms and contributions to the editor: T. W. Lanzatella
University Computer Center, 2520 Broadway Drive, Lauderdale, MN 55113

TABLE OF CONTENTS

NOTICE OF CHANGES TO THE SYSTEM1

PROPOSED CHANGES TO THE SYSTEM.....3

 NOTICE/NOTIFY - T. W. Lanzatella.....3

 DON'T DROP NODROP - T. W. Lanzatella.....4

 MACHINE IDENTIFICATION - T. W. Lanzatella4

 SYSTEM DIFFERENCES - K. C. Matthews.....4

 PROCEDURE FILE CALLS - J. Larson.....5

 PERMANENT FILE PROTECTION - T. Salo.....6

SYSTEM MAINTENANCE7

 LAST WEEKS SYSTEM GROUP MEETING - T. W. Lanzatella.....7

 MULTI USER/MULTI THREAD - S. P. Nachtsheim7

 CALLPRG NEWS - M. Riviere.....10

 COBOL5 - H. Kurs.....10

 LIBRARY TAPE ORGANIZATION - M. Riviere.....12

 CALLPRG INDEX ORGANIZATION - M. Riviere.....17

 MORE CONSOLIDATION - B. Williams.....17

NOTICE OF CHANGES TO THE SYSTEM

Jim Mundstock finished installing the CALLPRG XMIT option giving CALLPRG the capability of transferring files from the CYBER 74 to the 6400 (see DSN 2, 11p2). An unfortunate consequence of this modification is that CALLPRG now has a user number and password assembled into it. To repair this situation, XMIT must be changed to allow SUBMITting of system origin jobs (based, of course, on the SSJness of the XMITing job). The code review committee felt that this breach of security was lamentable but not lamentable enough to warrant postponing the feature installation until XMIT is changed because the validations associated with the user number in question are inconsequential.

Hats off to Bruce Johnson. After three years as an integral part of the system, SUPPIO, ISU and PDP have finally made it to our system program library MPL. Along with these additions, we also have a new common deck COMPDEF. This common deck defines a broad collection of PP macros which represent frequent PP code sequences.

Don Hamnes completed work on DIVERT (see DSN 2, 10p2). His portion of the DIVERT implementation will be installed with this tape. The DIVERT feature cannot be enabled until Bill Sackett finishes work on the DIVERT validation file (VENUS) utilities.

Tom Lanzatella repaired a bug in SYSEDIT which caused a rather inelegant rollout in the event that a SYSEDIT was in progress prior to the SYSEDIT request.

Bob Zalusky installed a portion of the DELAY queue feature (see DSN 2, 11p5). This portion enables the system to determine when LOW RATE is in effect and provides the capability to disable or enable the entire LOW RATE feature. The remainder of the DELAY feature entails the actual manipulation of DELAY jobs and the queue in which they reside. The second part is being written by Bruce Johnson.

Alan Johnston repaired a bug in the TELEX rotary check shortly before he departed. We ran with the rotary check disabled for 3 days before the bug was repaired.

Kevin Matthews installed a new version of the program PDUMP, a program used by the operators to perform permanent file dumps. The new version was necessary because of our recent change in permanent file dumping procedures. Kevin also finished converting the MERITSS ECS usage modifications. In addition, Kevin discovered an elusive bug in CIO which, under certain circumstances, destroyed a direct access permanent file if the device on which the file was located had encountered a track limit and CIO was attempting a rewrite on the file. The result being a EOI written at the beginning of the file. The bug was discovered because several S2K users with a propensity for large data bases filled up the disks causing S2K to attempt a unique recovery procedure of rewriting one of the six direct access files it uses with recovery information.

Tim Salo installed the following collection of modifications:

- 1) The program COST was modified to reflect the new \$1.00/KPR charging rate for MT PRUs transferred.
- 2) Account file messages issued by QDUMP and QLOAD were modified.
- 3) Program OPS was modified to include the number of sectors per track in the file size permanent file accounting message.
- 4) The following modifications were converted from the 6400.
 - a) The DDF command was added to DIS. Entering DDF under DIS automatically drops the display channel and calls DDF.
 - b) The ROL and ROL,XX commands were added to DIS. These commands are identical to the ROLLOUT and ROLLOUT,XXX commands.
 - c) Program ISF now runs with a CP priority of 50B rather than 2.
 - d) The fast attach file used by the TALK utility is now entered by ISF.
 - e) A new parameter NP has been added to the ISF command. The new parameter signifies that no procedure should be executed after the call.
 - f) Users without CSPF in their access word may now replace permanent files to an alternate user number.
 - g) Program PFM now checks the length of permit sectors.

- h) A new parameter TT has been added to TDUMP. The new parameter causes TTY style listing regardless of the device type associated with the file.
- i) Program MODIFY now treats *0 as an abbreviation for *OPLFILE.
- j) Program PFILES has a new parameter SS which allows the user to set the subsystem associated with a permanent file.

N. L. Reddy modified LAJ to store portions of the SSTL word from CMR in the job communication area (RA+67). These bytes contain information about system status like LOW RATE and ROTARY CHECK. The appropriate documentation was added to CPCOM. This change was brought about because of the Delay Queue installation.

PROPOSED CHANGES TO THE SYSTEM

NOTICE/NOTIFY - by T. W. Lanzatella

Among the several handy utilities brought to our system from the 6400 is NOTICE/NOTIFY written by Bill Wells. This utility offers the capability of distributing a collection of messages over various groups of users. Messages are entered and deleted from a fast attach file called NOTICE with the NOTIFY portion of the utility. Each message on the NOTICE file is identified as to which users are to receive the message. An individual user number can be specified or a group can be specified by using astrisks in the specified user number. A person may enter a message on the NOTICE file if his user number is validated. Currently, a valid user number must be assembled into NOTIFY. In the future, this will be handled differently. The user number ***** has permission to send messages to all users. The user number ABC**** may send messages to all users with user numbers of the form ABCnnnn. When a user logs in, TELEX calls the program NOTICE with the user's user number. NOTICE then searches the NOTICE file and if a message exists for the user, it is printed.

Several other features add to the attractiveness of the utility.

- a) Operations personnel need no longer worry about the MIRJE introductory message.
- b) Messages are automatically deleted from the NOTICE file three days after they are entered. This prevents the NOTICE file from filling up and forces the master users to perform regular maintenance.
- c) The program NOTIFY produces various reports based on the contents of the NOTICE file describing messages and their destinations.
- d) An easy enhancement to NOTICE/NOTIFY would be the ability to execute procedure files as well as send messages.

Assuming that everyone agrees that NOTICE/NOTIFY is good for MIRJE, I propose that the use of NOTICE/NOTIFY be extended to include batch jobs as well. With batch jobs however, messages would appear in the dayfile rather than on the output file.

This proposal was precipitated by two user suggestions:

- 1) For several months, 1004 users have complained that printing sysnotes is an irritating waste of time and paper. Hence, most 1004 users do not use BIN cards and thus never see sysnotes.

- 2) Users have suggested that short messages in the dayfile describing the contents of sysnotes would be just as useful as sysnotes.

After extending the use of NOTICE/NOTIFY to batch jobs, I propose that BIN be modified to send sysnotes only to jobs with batch or export origin. The 1004 and 200 UT users would then rely on the messages in the dayfile to determine when to run WRITEUP, NOTE.

//////////

DON'T DROP NODROP - by T. W. Lanzatella

In DSN 2, 9p2, John Larson proposed that we invent a type of local file which remained local despite the effects of the OLD, NEW, LIB or CLEAR commands. The proposal was rejected as being too complicated for such a trivial feature. After some study, John has discovered that such a feature could be installed in only five lines of code in PFILES (independent of cracking of NODROP parameters). Such an easy installation however, requires the use of the file ID field of the FNT which is currently reserved for use by SUPIO. I propose that a small interval of possible file ID values, say 60B to 67B, be reserved for use by utilities like NODROP. K. C. Matthews suggested that if we install NODROP, we must also install DROP.

//////////

Machine Identification - by T. W. Lanzatella

Since we will be running a common system on the CYBER 74 and the 6400 we must soon decide on the method by which the operating system will identify each machine. Fortunately, several sites have already worked out a consistent scheme for this purpose on a system very much like our own. First a new CMRDECK entry must be defined, MID=XX. The symbol MID stands for machine identification and XX is the actual ID. I suggest that MID=AA be used on the CYBER 74 while MID=AB be used on the 6400. Second, a KCL mnemonic would be useful for determining within the control card record, which machine a job is running on. I suggest the mnemonic be called MID. Third, so that a program can internally check the machine identification, a CPM function should be provided which returns MID to a user program. I suggest that the macro installed for this purpose be called a MACHID. Fourth, N. L. Reddy suggested that a MID field somewhere in RA to RA+100 would be useful and that a MID field in the system sector of output files would also prove useful.

//////////

SYSTEM DIFFERENCES - by K. C. Matthews

The need has arisen to have certain system programs do one thing on the 6400 and another on the CYBER 74.

The most recent case I encountered is the CIO track limit problem. On the CYBER, when CIO requests tracks (from the CPU monitor) and cannot get them, it displays a message in the B-display and tries the request again. This is bad since it keeps CPU monitor so busy that very little else can be accomplished by the system.

My proposal for CIO is given later, but it essentially involves having CIO wait for a few seconds before re-trying the request. On the 6400, however, there is seldom an operator watching the B display. They have long had a mod

in CIO which simply aborts on track limits. Note that aborting the job always is a good idea on a machine running mostly time-sharing jobs, but not a good idea on a batch machine.

Hence, the obvious thing to do is to have CIO abort on the 6400, and do something else on the CYBER.

One way to do this would be for CIO to check the machine ID field in low core. I don't like this solution, since it allows no flexibility for undoing the effect of the mod and no good way out if another machine ID appears some day.

I think a better solution is to define a new 60 bit word in low core called INPW. (INPW will be word 11 in the installation area). INPW will contain installation processing option bits. These bits can be set and cleared in two ways.

(1) At deadstart time values for these bits can be set in the IPRDECK.

(2) From the console, bits can be changed with the ENABLE/DISABLE commands.

The nicest way to handle the enabling process would be to give each new bit a name, and handle it like the other enable/disable commands.

TRACKLIMITABORT.

would toggle the bit value at deadstart time, while commands like

ENABLE, TRACK LIMIT ABORT.

would be used from the console.

I'm afraid this would tie up a lot of cells in DSD overlays and in deadstart. Instead, I'd rather specify the bit number.

INP00.

would toggle bit 0 in INPW at deadstart time.

ENABLE, INP00.

would enable the bit from the console.

I think these solutions make sense since it will only be in rare circumstances that bits are enabled/disabled from the console. All bits will, of course, be documented in SYSTEXT.

//////////

PROCEDURE FILE CALLS - by J. Larson

I wish to make the following proposal for calling procedure files.

1. Change the sequence of search from local, system, users PF catalog to local, users PF catalog, system. Because there are no user oriented procedure files on the system I believe this will cut down on the overhead of searching the system and will be more useful, allowing any name for user procedure files.

2. Add a parameter to the call card which will force the search of system files before the users own catalog.

If in the future the system group wishes to make procedure files available to the average user this scheme will allow for that. Also the CALLPRG program will be able to handle any system procedure files the group would want the user to have available.

Adding a parameter and changing some lines of code in CTL2 is trivial.

//////////

PERMANENT FILE PROTECTION - by T. Salo

KRONOS provides means of restricting and validating the access to permanent files from alternate user numbers. However, KRONOS provides no means of limiting or validating the access to a permanent file from another user running under the same user number. It is highly desirable that users be able to protect their permanent files against being purged, altered, or read by another user with the same account number. It is also desirable that alternate user numbers be able to access these protected files.

I propose that protection be granted to permanent files by means of a bit (the file-protect bit) in the catalog entry associated with each file. The bit set indicates that the file's password must be specified to access the file from the owning user number. This file-protect bit could be set by the user at the time of file creation (SAVE or DEFINE) or altered in a already existing file (CHANGE). The creation of protected files would be validated via a bit in the user's access word.

Specifying the unequivallenced *FP* (file-protect) parameter after the slash on the SAVE, DEFINE, CHANGE control card would set the file-protect bit. Specifying the unequivallenced *NP* (negate protection, no protection) parameter on the CHANGE control card would unset the file-protect bit on an existing file. Specifying both the *FP* and *NP* parameters would return * PFM ILLEGAL REQUEST.*.

A password (possibly null) is associated with all files at the time of file creation. All accesses to a protected file, including those made by the owning user number, would be required to specify the correct password. The GET, PURGE, REPLACE, APPEND, ATTACH, and CHANGE commands will return *(FILE NAME) NOT FOUND.* if the correct password is not specified when accessing a protected file, even when accessing the file from the owning user number.

Access to a protected file from an alternate user number would be unaffected by the status of the file-protect bit. The file-protect bit would be independent of all checks performed currently to validate alternate users for access to a specific file. A file's catalog type and permit entries would be unaffected by the implementation of the file-protect bit. Only access to a file from the owning user number would be affected by the file-protect bit.

An asterisk would be set in the catalog list by COMCFCE indicating that the file-protect bit is set for a specific file. The password of a protected file would be returned only to a catlist requested by a star number over the user number owning the file. The password would be replaced by an identifying sequence of characters to a catlist requested by the owning user number. The catlist returned when requested by an alternate user number would be unaffected by the status of the file-protect bit as the password is not returned for any file to an alternate

number. This scheme protects the password of a protected file, yet allows a star number above the owning user number to learn the password to the file and account to the owning user number and access the file.

The CHANGE control card will change certain attributes of a file regardless of the status of the file-protect bit. However, the user would not be able to change the password of a file while it is protected, as the password is used to validate access to the file.

A file with permit information could be changed to a protected ifle with the CHANGE control card. However, the PERMIT control card or macro could bot add permits to an existing protected file as the PERMIT control card and macro do not allow a password parameter. It would be possible to add a password to permit, but it would be inconsistent with the current use of PERMIT.

SYSTEM MAINTENANCE: People and Procedures

LAST WEEKS SYSTEMS GROUP MEETING - by T. W. Lanzatella

1. Regarding DRL's suggested SUPPRESS utility, we decided that the installation would be easy but there would be definite problems with EXPORT. DRL and NLR will meet to determine the best way to install SUPPRESS.
2. N. L. Reddy promised a new banner page by the end of summer (1976).
3. Regarding DRL's QUEUE proposal, after an hour of tense discussion we decided to relagate the matter to a special committee.
4. Barry Fox' proposal changes to AFD, DFD, and ELD (DSN 2, 12p3) were approved.
5. K. C. Matthews' proposed Busy File utiltiy (DSN 2, 12p3) was approved.
6. Regarding Bob Williams' Master User Revisited:
 - a) We decided to validate the use of the USER/ACCOUNT command with a bit in the access word. All users on the CYBER 74 will have the bit. Most users on the 6400 will not have the bit.
 - b) We decided to validate the use of the ACCESS subsystem with a bit in the access word.
7. Regarding K. C. Matthews comment on B1=1:

We all agreed that to assemble the common decks to assume that B1 is 1 on entry was the standard way to do things within UCC. But, since the programming practices of our users are not so standardized, we shall, in the future, if the labor is available, assemble the common decks to assume nothing about B1.

//////////

Multi-User/Multi-Thread edition of S2K - by S. P. Nachtsheim

We are currently running Version 2.40 of MRI's SYSTEM 2000 (S2K). The next feature release of S2K labeled 2.60, is due out around December of 1976. In addition to normal new features, this release has some optional extra cost items that the University may wish to acquire. These new items are the Sequential

File feature and a Multi-User/Multi-Thread edition of S2K. The former gives users the ability to store and access data directly from magnetic tape and has no impact on the system. The latter potentially has a significant impact on the system. Hence I am presenting a synopsis of Multi-User/Multi-Thread in hopes of generating discussion and comment as to the potential impact and problems.

From the users viewpoint, Multi-User/Multi-Thread (MUMT) (pronounced "DVRT") offers three significant advantages:

1. The ability for more than one user to update a data base at the same time;
2. Increased throughput and faster execution;
3. The ability to use the Programming Language Interface from TELEX without excessive user field lengths overlays and resultant thrashing due to rollout.

The Multi-User/Multi-Thread version operates as a subsystem under KRONOS, residing, when active, at control point 2. As such, MUMT becomes a secondary operating system managing its own sub-control points similar to the KRONOS product TRANEX or the CDC data base product TRANEX/TOTAL. (Refer to the KRONOS Workshop manual: pp. 5-19 thru 5-27; 5-44 thru 5-47/KRONOS Installation Handbook: III-4-6 thru III-4-7;). As such, MUMT manages up to 7 subcontrol points allowing up to 63 users to retrieve or update data bases, via Natural Language or PLI, from batch or timesharing.

The potential installation of MUMT has the following implication with regard to the operating system interface:

1. MUMT must process all RA+1 calls, executing some and passing the remainder to KRONOS:
2. All interface with KRONOS is relating to MUMT, so MUMT must relocate addresses before passing to KRONOS;
3. MUMT maintains its own psuedo FNT which maps to KRONOS FNT's.
4. The following restrictions apply to RA+1 calls:
 - a) CIO - FET must be at least 7 words
 - b) MSG - Message goes to MUMT dayfile
 - c) PFM - FET must be 11 words and SAVE or DEFINE use MUMT's user number.
 - d) SFM - Code = 0 is SUBMIT
 - e) LFM - Code = 12 is STATUS
13 is STATUS
15 is REQUEST but device DE or DP only
17 is SETID
 - f) CPM - Code = 2 is MODE
 - g) LDR - User must specify either:
 1. LWA/FWA or
 2. Run the BUILD utility prior to issuing LDR calls (L1 - primary overlay may never be zero)
 - h) LDV - Same as LDR

5. Magnetic Tapes are not allowed.
6. Permanent fiel creation uses MUMT's User Number; all other file creation requires the UN parameter.

In its current form MUMT requires the following modification to the operating system:

1. A PP routine to imitate MUMT at operator (console) command
2. Mods to DSD to recognize in initialization command
3. CIO must be modified to honor the error processing (EP) bit for I/O requests
4. A fix to KRONOS so CPU time used by each subcontrol point is accounted properly.

The current memory usage for S2K, Version 2.40 is from 47100 to 55000, depending on the option used. In PLI, the field length is 47100 plus the user program; this generally results in a field length of 70000 + words.

The projected memory map for MUMT is:

Field Length in Octal	Usage	Description
10000	Executive + Tables	Executive
100	Shared Common	SYSTEM 2000 (Subcontrol point 0)
22000 (variable)	Data Base and Scratch File Buffers	
5000	Serially Reusable Code	
3000	S2K (0,0)	
	S2K (P,0)	
	S2K (P,S)	
200 * Number of Data Bases	ID Files for all Open data bases	
n	Core for PLI programs (Max of 7)	Subcontrol Points 1-7

//////////

CALLPRG NEWS - By M. Riviere

In the last two weeks a few modifications were made to FT3LIB by request of J. Mundstock. James added the entry DOTD\$ for MNF and corrected a problem in UCLOAD that was not using the right registers for UNIT. H. Kurs introduced COBOL 5 and several of its associated products. In the Library Tape Howard placed a new System Text, CB5TEXT and replaced RMERTXT (Record Manager Text) by its level 12 version.

In CALLPRG Howard placed the CDC COBOL 5 under the name of COBOL5, its library, FETCH type under the name of CV5LIB, a COBOL 5 sort library called SORTLIB and three conversion aids programs to convert from COBOL 4 to COBOL 5. These programs are CVLCOP, the convert libraries, CPYSORC to convert sources and LCS and LCST to convert large tables. CVLCOP, CPYSORC, LCS and LSCT are control card callable.

The COBOL 5 package was not proposed in a previous DSN, as it should have been, but I did not delay its introduction due to the announcement in the June UCC Newsletter the CDC assistance to install and test it that was available for only a few days and the common knowledge that COBOL 5 was going to be installed. Howard is willing to modify the set up of the COBOL 5 products if the actual arrangement is objected to by the System's group.

Kevin Fjelsted replaced APLUM (University of Massachusetts APL interpreter). Kevin is replacing A. Mickel in maintaining APLUM. A. Mickel changed current MIXAL by its future version and deleted the one that was past, that stopped working. M. Frisch moved TXTEDT to FETCH type and introduced FMT, a text processing program previously announced in this newsletter. I replaced the future version of SYSIO by updating it with Record Manager and 8BIT subroutines sections level 12, submitted by H. Kurs and I placed SYMPL in a CALLPRG library tape.

Next weekend (July 17) J. Mundstock will be replacing several coded I/O routines of FT3LIB in the library tape with new ones (INPUT\$, OUTPUT\$, IOU\$, TRACE\$, and TRACSET). James will also be changing DOTD\$ and RTOR\$, also in FT3LIB to correct a problem with MNF error message output. I will be replacing RELCSUR in SYSLIB on the Library Tape with a version reassembled without assuming that B1 is equal two one. RELCSUR is a debugging routine used by the SNAPSHOT package. The assuming of a value for a given register is defeating somehow the purpose of the SNAPSHOT function.

//////////

COBOL 5 and its Products - H. Kurs

This article provides general information concerning the installation and status of the COBOL 5 compiler and the conversion aids program. Comments are welcomed and should be directed to Howard Kurs.

The COBOL 5 compiler and the conversion aids program are currently classified as experimental software until September 15, 1976. UCC is cooperating with Control Data in extensive testing of this compiler.

The COBOL 5 compiler (COBOL5) is designed to conform to the ANSI 1974 COBOL standard. This compiler was completely rewritten and is not compatible with COBOL 4. The conversion aids program (LCS) assists users in converting their COBOL 4 program to COBOL 5.

All products concerning COBOL5 and LCS have been placed on CALLPRG. The following list describes how users would access them:

- COBOL5 - The COBOL 5 compiler.
- CV5LIB - The COBOL 5 library
- SRTLIB - The SORT library used by COBOL 5.
- LCS - The conversion aids program with the following features:
 1. Allows the conversion of COBOL 4 source programs containing "COPY FROM LIBRARY" statements.
 2. Has a name table size of 3200 words (decimal).
 3. Requires 121K field length to execute.
- CBL COP - The copy utility for LCS that processes "COPY FROM LIBRARY" statements.
- LCST - The same as LCS except that the name table size is 6500 words (decimal) and requires 131K field length to execute.
- CPYSORC - The independent COPY utility that processes "COPY FROM SOURCE" COBOL statements.

NOTE: The names CV5LIB, LCST, and CPYSORC are not referred to in the CDC manuals. Currently, there is no documentation on CPYSORC. If COBOL 5 becomes a permanent product on our system, then CPYSORC documentation will be made available.

CV5LIB, SRTLIB, and CPYSORC are FETCH type called. The remaining products are accessed as current products by using their names. The actual file names and types are as follows:

<u>PRODUCT</u>	<u>FILE NAME</u>	<u>TYPE</u>	<u>DESCRIPTION</u>
COBOL5	COBL5	DA	COBOL 5 compiler
	CB5LIB	DA	COBOL 5 library
	SI012	DA	Record Manager library
	SRTLIB	IA	Sort library
LCS	C4C5CPY	DA	Conversion aids program
	CTAPE10	IA	File containing Syntax description
	COPIN	IA	COPY utility for processing "COPY FROM LIBRARY" statements.
CB5LIB	CB5LIB	DA	COBOL 5 Library
	SI012	DA	Record Manager
SRTLIB	SRTLIB	IA	Sort Library
	SI012	DA	Record Manager Library
CBL COP	COPIN	IA	COPY utility for processing "COPY FROM LIBRARY" statements.
CPYSORC	CSOR	IA	COPY utility for processing "COPY FROM SOURCE" statement.
LCST	LCSTAB	DA	Conversion Aids program (large table size)
	CTAPE10	IA	File containing syntax descriptions
	COPIN	IA	COPY utility for processing "COPY FROM LIBRARY" statement.

During this testing period corrective code will frequently be added to these products. It is for this reason that I have made these products current on CALLPRG.

For your general information COBOL 5 takes approximately 90 minutes of CP time to install. The COBOL 5 compiler source listing file contains over 31 million characters and produces two microfilms of over 6700 frames. The OPL (release tape) produced by this installation is just short of 10 million characters. The COBOL 5 compiler is composed of 1 absolute type record and 32 individual overlays and is about 1.5 million characters in length. The COBOL 5 library (CV5LIB) is composed of 134 records with a length of 167,040 characters.

The conversion aids program takes about 15 CP minutes to install and produces a source listing file of over 2 million characters. The OPL (release tape) produced is almost 1.2 million characters in length. This program consists of 8 overlays having a size of 243,616 characters.

Since COBOL 5 has been installed, 8 PSRs have been sent to CDC. As of this report, timing comparisons have not been made, but COBOL5 does have significantly slower compile time. Reference manuals for COBOL 5 and the conversion aids program are available in the UCC library.

//////////

LIBRARY TAPE ORGANIZATION FOR CYBER 74 - CDC 6400 - by M. Riviere

I will present here a comparative description of the products contained in the CYBER 74 Library Tape and the products contained in the section of the 6400 Dead Start Tape that could be considered equivalent to a 6400 Library Tape.

After the description I will outline my proposal for the format and the contents of the future Library Tape that will be common to both computers. My proposal is based partially in the conclusions taken in the June 11 System meeting and partially in considerations taken by myself after discussing several of the different products with J. Mundstock and B. Williams.

Contents of 6400 Library Tape Section

SYSLIB - Library for RUN and Run 23. This library is equivalent to the CYBER 74 PAST SYSLIB but not identical.

FT3LIB - MNF library built from the CYBER 74 FT3LIB and complemented by routines from the CYBER 74 PAST SYSLIB.

RUN2P3 - Not available.

COBLIB - 6400 COBOL Library

TSLIB - Not available

FORTTRAN - Not available

PSCLIB - Pascal library. This library is identical in both machines.

Contents of CYBER 74 Library Tape

SYSLIB - Relocatable version of common decks and Loader routines.

FT3LIB- PAST FTN and MNF library. This library should contain all the necessary routines used by PAST FTN and MNF.

RUN2P3 - RUN and RUN23 dummy library.

COBLIB - COBOL 3.0 library used by Time Share COBOL. PAST, COBOL (COBOL 3.0)

TSLIB - Time Share COBOL Library uses an identical version from CALLPRG. This identical version is available under the name of SYSMISC.

FORTTRAN - FTN 4 Library

PSCLIB - Same as 6400 Library.

ALGLIB - ALGOL 2.0 library this library is not identical to the CYBER library associated with PAST, ALGOL (also ALGOL 2) that is in CALLPRG.

SYSMISC - Old version of a SYSMISC library, built following the CDC convention to keep all the 3.0 products libraries in only one. Many of the contents of the 6400 SYSMISC library are duplication of other 3.0 libraries.

BASLIB - BASIC library Level 2.

MNF - 6400 MNF

MNFOLD - Previously used MNF.

FTN - Not available

BASIC - BASIC level 2.

XBASIC - A version of BASIC that is almost not used.

PASCAL - Same as CYBER 74 PASCAL.

LISP - Same as CYBER 74 LISP.

SNOBOLC - Fast SNOBOL subject

RUN - Similar to CYBER RUN but a newer version. The 6400 RUN was modified not to require the existance of RUN 2P3 as a Library.

ALGOL - ALGOL 2 adapted for Time Share.

APLSYS -

APLUM -

COPYCL - Old

ALGLIB - Not available in the Library Tape. There is a CALLPRG version associated with PAST, ALGOL (2.0) and another associated with current ALGOL (3.0).

SYSMISC - Contains only one section for BASIC level 11.

BASLIB - Not available

MNF - CYBER 74 MNF
MNF will not be identical in both machines. the 6400 version will be using less CM than what the CYBER version uses.

MNFOLD - Previously used MNF. It will be different in both machines.

FTN - FTN 4

BASIC - BASIC level 11

XBASIC - Not available

PASCAL - Not available from library tape. PASCAL is a CALLPRG product in the CYBER 74.

LISP - Not available in Library Tape. LISP is a CALLPRG product in the CYBER 74.

SNOBOLC - Not available in Library Tape. SNOBOLC is a CYBER CALLPRG product.

RUN - A version of RUN older than the the 6400 RUN.

ALGOL - Not available in Library Tape. ALGOL 2 and ALGOL 3 are CALLPRG PAST and current products of the CYBER.

APLSYS -

APLUM - Not available in Library Tape. APLUM is a CALLPRG package in the CYBER.

COPYCL - Not available in Library Tape. FUTURE and FETCH COPYCL are CALLPRG products.

SMTEXT - Sort/Merge text, OLD.

SMTEXT - Not available in Library Tape.
FUTURE and FETCH SMTEXT are CALLPRG products.

TXT6RM - Old version

TXT6RM - Text associated with Record Manager,
level 11.

CB5TEXT - Not available

CB5TEXT - COBOL 5 text

COS Type Records - Part of TSRUN
Library .

COS Type Records - Not available

Old Compilers Complimentary Products -
Not available

Old Compilers Complimentary Products -
ALGOL 2 overlays used by PAST, ALGOL
FTN 3 overlays used by PAST, FTN SYSTEX20
COMPASS 2 text used by PAST, FTN COMPI\$
COMPASS 2 overlay used by PAST, FTN COMP2\$
COMPASS 3 overlay used by MNF.

SORTMRG - SORT MERGE 4, old version

SORTMRG - Level 11, SORT/MERGE 4, not
available in Library Tape. SORTMRG
current and FUTURE are CALLPRG products.

FILE - Old version

FILE - Record Manager utility, Level 11.

PRECOB - COBOL source conversion program.

PRECOB -Not available in the CYBER 74.

COBOL - UCC Time Share COBOL.

COBOL - COBOL 4, level 11, not available
in Library Tape. Current and FUTURE COBOL
are CALLPRG products.

SORTMR3 - Not available in 6400.

SORTMR3 - SORT/MERGE 3.

RMERTXT - Not available

RMERTXT - Record Manager text, level 12.

COBERTXT - Not available

COBERTXT - COBOL 4 text, level 12.

From the above description it is easy to see that for the time being and perhaps never will both libraries be merged into one tape and have the full contents of that tape sysedited in both machines.

I suggest to create a common library tape containing three different files. The first file should contain CYBER 74 only products, the second file should contain products common to both machines and the third file should contain 6400 only products. Two copies will be made of each tape, using labeled tapes with associated VSN's, such as LF000X for the original tape and LF001X for the duplicates. One tape will be the back up of the other. It will be possible to use either tape in either machine. The sections of the tape that will be sysedit for each machine will be selected at Dead Start Time.

My suggestion about how the three files of the future library tape should be like is the following:

FILE 1 (CYBER ONLY)

FORTRAN - FTN 4 Library

FTN - FTN 4

MNF - CYBER version

MNFOLD -

APLSYS - CYBER version

Overlays for CALLPRG products not available in the 6400 (FTN 3, ALGOL 2, SORT/MERG 3, COMPI\$, COMP2\$ SYSTX20)

RMERTXT -

TXT6RM -

COVERTXT -

COBOLX - Set up to use Time Share COBOL, CYBER version. (Time Share COBOL is a disk file).

COBLIB - CYBER COBLIB or TSCLIB

FILE 2 (CYBER and 6400)

SYSLIB - CYBER 74 SYSLIB

FT3LIB - CYBER 74 FT3LIB

SYSIO - CYBER 74 SYSIO

MNFLIB - ?

SYSMISC - CYBER 74 SYSMISC (see BASIC)

PSCLIB - CYBER 74 PSCLIB

BASIC - CYBER 74 BASIC, if possible. Otherwise 6400 BASIC and 6400 BASLIB instead of SYSMISC.

PASCAL - 6400 PASCAL

SNOBOLC - 6400 SNOBOLC

SORTMRG - CYBER SORTMRG from CALLPRG

FILE - CYBER FILE

Overlays for RUN and RUN23

CB5TEXT - To be used by COBOL5 (COBOL5 is a CYBER CALLPRG product).

FILE 3 (6400 only)

MNF - 6400 version

MNFOLD

APLSYS - 6400 APLSYS

APLUM - 6400 APLUM

Overlays for 6400 CALLPRG products (ALGOL 2)

COBOL - 6400 Time Share COBOL

COBLIB - 6400 COBLIB

With the contents of the common Library Tape as described above some special arrangements will have to be made with the CYBER and 6400 products that are not included in the future ones. Also some CYBER CALLPRG products should become library tape products to be easily accomodated in both machines.

RUN and RUN23 - should be in the 6400 section of the CALLPRG index to be associated with their corresponding version SYSLIB.

SYSLIB - (current 6400 SYSLIB) should be in the 6400 section of the CALLPRG index as PAST, SYSLIB. (Users should be notified in advance about this).

LISP - Should be in the 6400 section of the CALLPRG index

SMTEXT and COPYCL - should be FETCH type in the 6400 section of the CALLPRG index. (Users should be notified in advance about this) SMTEXT and COPYCL should be also replaced by the CYBER updated versions.

PAST, FTN - PAST, SORTMRG, SORTMRG and PAST, ALCOL - should have entries in the 6400 section of the CALLPRG index stating that they are not available.

PRECOB - Should be in the 6400 section of the CALLPRG index.

SORTMR3 and PAST SORTMRG - should be in the CYBER CALLPRG index.

SORTMRG - Current should be removed from the CYBER CALLPRG index and placed in File 2 of the Library Tape.

PASCAL - Current should be removed from the CYBER CALLPRG index and placed in File 2 of the Library Tape.

STRATEN - Should be placed in the 6400 section of the CALLPRG index.

CPCTEXT, CPUTEXT, IOTEXT, IPTEXT, LDRTEXT, PFMTTEXT and SCPTTEXT - Should be placed as FETCH types in the 6400 section of the CALLPRG index or, if their usage is very small they could be retrieved by the CALLPRG program from the CYBER. If they are kept as 6400 files the newer version (level 12) should be used. (Users should be notified about it.)

A question arises on the documentation of the Library Tape modifications. Both machines may not be releasing a new tape at the same time. I understand that although it should be ideal to change systems in both machines at once this may not be possible due to the different allocation of System, Maintenance and Dead Start times. I understand that a delay may be introduced for changes to take place in the 6400 after they have been in production in the CYBER for a while, in order to insure 6400 System stability.

One possibility could be to have two versions of SYSMODS and update each machine version as the changes take place.

Other possibility should be to keep updating SYSMODS available only in the CYBER 74 and transferring a copy to the 6400 at each 6400 Dead Start Time.

If only one version of SYSMODS is kept it may be necessary to state to which machine belongs each described modification.

I will greatly appreciate comments and suggestions from the system's group about this proposed arrangements.

//////////

CALLPRG AND WRITEUP INDEXES ORGANIZATION FOR CYBER 74 - CDC 6400 - by M. Riviere

The arrangement of the CALLPRG and WRITEUP indexes products will not require many changes thanks to the modification that J. Mundstock is introducing in the CALLPRG program to add the capability of retrieving CALLPRG products that reside on the CYBER 74 disk pack.

The 6400 will have a CALLPRG index that will contain a 6400 section as a first section and the CYBER index as a second section. Products without pointers in the 6400 section of the index will be retrieved from the CYBER using the CYBER section pointers. A similar setup will be made for the WRITEUP index.

The source for both CALLPRG indexes will be the same one (CPOPL). The documentation of the index products changes will be extended for the 6400 with the 6400 section changes.

The 6400 CALLPRG and WRITEUP indexes will have to be modified each time that the CYBER indexes are or at every 6400 Dead Start. This setup will be very unlikely to effect 6400 stability since changes in the 6400 sections will be independent of the ones in the CYBER.

The 6400 section of the CALLPRG index will contain initially a few entries to satisfy the arrangements necessary due to the contents of the future Library Tapes. I will be inserting new cards following CALLPRG packages owners request.

For the 6400 section of the WRITEUP index I will be inserting entries to documentation files also upon request.

//////////

MORE CONSOLIDATION INFORMATION - by Bob Williams

- A) Testing dates for the consolidated 6400/CYBER 74 system have been set for July 25, August 1, and August 8 leading up to the August 15 final conversion. On these Sunday afternoons from noon to 4 P.M. the system will be up for testing. All elements should be available to users as they would be on the consolidated system so problems for the users and for us can be identified.
- B) The number of total files up at a control point at any time in a job validation limit formula is inadequate for time sharing use. For this reason we intend to modify the formula from the CURRENT:

INDEX * 20B + 30B
TO: INDEX * 20B + 10B

This will be done by just changing the KNFI base value from 30B to 10B. In addition, the 0 value will be treated specially. A value of 0 in VALIDUZ will mean 12D local files may be used (not 8D as the new formula would give). This means that instead of steps (30B, 50B, 70B, ..., 170B, unlimited) the number of local files steps will be (14B, 30B, 50B, ..., 150B, unlimited).

- C) The total indirect access file storage space limits are inadequate also. The lower value, therefore, will be changed from 1000B PRU's to 400B PRU's (CSRNG1 = 4 in place of 10).