UNIVERSITY OF MINNESOTA COMPUTER CENTER

Deadstart Systems Newsletter

22 June 1976 Vol. 2, No. 12

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

TABLE OF CONTENTS

LEVEL 12 - STATUS

Presenting a brief report on the tribulations surrounding Level 12 and its installation:

Control Data added three new modifications to program LINK at level 12. The first modification was to enable LINK to recognize the new 5400 loader table (machine characteristics). The second modification (LINK39) was to repair a problem with tape resident load files while the third modification (LINK38) repairs a problem whereby LINK would accept only up to four characters as the load file name. Unfortunately, only the first of these three modifications worked correctly. Discovering and repairing both these errors took about three days.

N. L. Reddy discovered that a faulty CDC mod, CVFE1 (to COMCVFE), caused QREC to malfunction for the first three days of Level 12. This modification has been removed.

Program SEND stopped working properly when Level 12 was installed. This problem remained in the system for six days and arose because a local modification to COMPUSS had been inadvertently left out of the running system.

The final problem relates to the installation of level 12 FTN and occupied the majority of M. Riviere's time for over two weeks. With level 12, CDC introduced a new machine characteristics loader table, the 5400 table. Confusion began when Marisa discovered that LINK only recognizes 5400 tables while the CYBER loader actually allocates space. Assembling FTN with 5400 table capability and then loading with LINK caused a discrepancy between what FTN believed were correct entry addresses and what LINK believed were correct transfer addresses by a factor

of 10 words. This problem has been repaired by lengthening the FTN (0,0) overlay by 10 words in the area preceding the labeled common block COMPCOM.

NOTICE OF CHANGES TO THE OPERATING SYSTEM

Kevin Matthews supplied a modification to the local common deck, COMSPMS, and supplied a new version of his performance measurement statistics modification, PMS. The change frees a bit in low core to be used for the forthcoming delayed input queue.

Jim Mundstock installed a small fix to BATCHM which was not checking the account open bit in the Access Word on the VALIDUZ file before running student batch jobs.

Tim Salo installed several modifications from the 6400/MERITSS system.

1) CARGU1 - Provide conditional assembly in COMCARG to specify the maximum number of characters per parameter.

2) CUPC1 - Provides conditional assembly in COMCUPC to specify the maximum number of characters per parameter.

3) CATLU1 - Alphabetizes the file names in the CATLIST short list.

4) CATLU2 - Adds the BR (brief) option to CATLIST. Specifying the BR parameter suppresses the title and headers.

5) CONTU1 - Implements the PRINT control card. PRINT lists an optional expression and comments on file OUTPUT.

6) CONTU2 - Fixes a mode 1 error in CONTROL.

7) SUN - Implements the SUN (set user number) control card (System origin only). SUN functions like a USER/ACCOUNT card with no password necessary.

Bob Zalusky reinstalled his modification to PFILES allowing the PN=0 option on the permanent file commands. Bob discovered that PURGALL (a separate program from PFILES) accepts the PN=0 option but declines to act on it.

Bill Elliott submitted a new source for his common deck COMSEXF. The new version provides better security for the TAPES index file.

Allan Johnston contributed the following batch of corrections (his last).

1) Trim 160 words from TELEX's field length by moving the old 63 character set translation table from CM into the PP, 1TD.

2) Reinstall an old level 9 mod which streamlined outputting of TELEX warning messages (FASWRN).

3) Repair a problem in TELEX which caused a hung port if a user typed STOP and then hung up the phone.

4) Alan supplied a new source for his program USERS.

5) Install a new version of the TELEX rotary check. The new algorithm allows for off-hours user validation (after 5). Additionally, Alan supplied a new source for the PORT file source processor, CPORT. Program CPORT will become part of MPL and will also be placed on the deadstart tape.

Brian Hanson finished installing the level 12 version of the CYBER loader.

Bill Wells corrected a small problem in XEDIT Y command processing and supplied a new version of XEDIT for the deadstart tape.

PROPOSED CHANGES TO THE SYSTEM

Barry Fox suggests that programs AFD (Account File Dump) and DFD (Dayfile Dump) be modified so that a user with CSOJ may use the utilities.

Rich Franta has observed that when the delayed input queue is installed, the COST control card must be able to reflect correct job cost for delayed jobs. Rich also suggests that program CALLPRG be corrected so that WRITEUP, CONTROL does not print the record name INDEX at the start of the listing. Additionally, CALLPRG should be modified so that indexed writeups begin at the top of a page.

//////////

Busy File Utility - K. C. Matthews

Occasionally, when TELEX aborts, a direct access file is left "busy". This sometimes happens after a level - 3 recovery. (Busy files eventually become unbusy during a level zero deadstart.) Whenever this happens, the user often calls the Help-line or Operations because he wants to use the file. For each file, someone must bring up DDF at the console, search the correct catalog track for the file name to find the system sector, and then go clear the interlock in the system sector. This can be very time consuming, especially when a System 2000 data base (6 direct access files) is left busy. I propose that we install a new control card called UNBUSY which calls a new UFM function to unbusy such files. The UFM function will be legal for system origin jobs only.

The format of the control card is:

```
   UNBUSY(pfn,ui,pack)
where
   pfn=permanent file name
   ui=the file user index
   pack=the pack name (if needed)
```

Examples:

```
   UNBUSY(SAM2716)
```

will unbusy file SAM under user index 2716 on the default devices.

```
   UNBUSY(SAM,2716,STF)
```

will unbusy file SAM under user index 2716 on pack STF.

UNBUSY is still intended to be used only in exceptional circumstances.

The second parameter could be user number instead of user index with a little extra code in the CP routine.

//////////

Delayed Input/SUPPRESS - D. R. Lienke

A. Sub-topic: Ferocious attack on attitudes.

Damn. You guys did it again. In the last DSN, I complained about the practice/procedure/attitude of giving only a few hours time between the publication of a proposal and a vote on the topic (or what amounts to the same thing - an implied need to vote). I don't think that people have enough time to consider a subject in a few hours - certainly not enough time to be able to vote intelligently, let alone make constructive suggestions.

I have noticed a lack of discussion involving most 'fresh' proposals. Do you think that such a lack is due to the typical apathy of a large group, or rather an indication of 'no preference' due to lack of time to ponder and mull over a subject? Recall the spirited, animated discussion on DIVERT - 2 weeks after the proposal was published. Compare it with the lack of discussion at the first meeting. And yet, at the very meeting where I indicated the inappropriateness of hasty approval of proposals, the proposal for the DELAY queue was approved. I hardly think that I would have to mention that haste leads to embarrassment, lost time/money, bad public relations, ill-advised changes, imprecise definition, and inconsistent (philosophically) design and implmentation. I will have more to say on the DELAY proposal later in this communication.

B. Sub-topic: SUPPRESS.

I shall define 'SUPPRESS' as my recent, and by no means orisinal, proposal to put 'UNPAGE ability' in the output system (BATCHIO, SUPIO, EXPORT) where I consider that it rightfully belongs. I quote from the DSN Vol 2, No 11:

> " I think that all the I/O packages should recognize a user-controllable bit in the system sector for such an ability. That would save lotsa "UNPAGE" time ..."

Such a proposal is reasonably clear, and it has been brought up before, although perhaps not formally. I recall being told to formally describe such such a proposal, since (evidently) it had such sweeping implications for the queue exit system. Well, the fact of the matter is, I doubt if I could, since I'm not the I/O driver expert. I do know that there is sufficient space in the system sector, and that there were 8000+ UNPAGE executions in (during) MAY76. To cause such a (presumably) large amount of data to be passed through PP's twice (via UNPAGE) in addition to normal processing by the output system seems wasteful to me. I

have nothing against UNPAGE - it was (and is) excellent as a stop-gap measure, but gap-stopping should not be our goal, nor should the transmutation of stop-gaps (essentially KLUDGES) into permanent system features. I don't know how much work is involved in the implementation of SUPPRESS, but I would think that looking over all the packages (BATCHIO, EXPORT SUPIO) might lead to some re-design ideas so that such 'accross-the-board' changes (e.g. SUPPRESS) would not be so difficult in the future. Consider: the UCC is the authority over programs and rates of cost (in the user's view, anyway); we control the programs that waste paper and we (usually) increase the price of paper. Our solution to save paper/money is to utilize other resources, namely CPU time and mass storage transfers. Yet it is the user's time and transfers that we suggest (some would say 'force') he use. A single control card to set a bit, some undefined amount of work in routines that will save everyone time/money/paper - isn't that worth at least a look at? (In addition, don't EXPORT & BATCHIO already have an operator-initiated SUPPRESS mechanism?) In short, I think my proposal was definite enough to warrant a study of the matter (it may well be that, in terms of manpower, it won't be worth the effort - you won't know 'til you look at it).

C. Subtopic: DELAY.

Has enough consideration been given to the user? Has enough consideration been given to the effect on the system? Has enough consideration been given to the uniformity and consistency of the design? Will the user be able to ascertain that his job has actually been placed in the DELAY queue - i.e. are you going to give the user immediate output? Will a user be able to 'un-queue' a job that he wishes to run at either a different time or not at all? Will a user be able to judge how likely it is that his job will run at all (i.e., can he get a list of the DELAY queue with estimates of total CPU times of other jobs that his will be competing with?)?

To answer such questions surely requires some thought. Obviously NLR has considered the internal ramifications, but only based on what was presented as the external view.

I disagree with the decision to place the DELAY parameter on the Job card for several different reasons.

A) All the I/O packages would have to be changed. There have been two rejoinders to this comment. NLR claims that this is a fallacious statement since it is merely the queue interface routines that have to be modified. If this is true, bravo, but I suspect that at least a re-assembly of the I/O packages will have to be performed, and that some logic will have to be added to deal with the JOBCARD parameter. (Is SUPIO internally clean - i.e. does it call standard CPU common decks, does it use 2TJ, does it have B1=1, does it (really) still have

calls to MOMS program GFM, etc., etc. - as promised so long ago, so that anyone could easily modify it?) I also contend that the job entry routines are not the place to determine anything other than validation, and certainly not the place to shunt files off to queues other than INPUT. Otherwise, let us have options on the JOBCARD for immediate disposal to punch, print and film queues; obviously people wish to print, punch and film just as they wish to have jobs run cheaply (i.e., in the DELAY queue).

KCM has said that there is no 'a priori' reason for NOT putting a parameter (in general) on the Jobcard. My reply is that I feel the JOBCARD to be functionally obsolete. The operating system does not now restrict the user to any JOBCARD parameters - that is, the user may reset his FL and his TL, etc. For all practical purposes, except possibly an operations consideration, we have no real use for a Jobcard, and therefore, it seems like a bad design to place yet other options on it.

B) There is no provision for proving to a user that his job was actually placed in the DELAY queue. As I see it, this involves at least two methods for getting information to the user. One is that the processor that places jobs in the DELAY queue must return a dayfile message to the user. With the current proposal this would mean all the I/O packages would have to produce some output - obviously extravagent in terms of manpower and almost certainly the wrong place for such action. Secondly, some way must be given to users to find out the contents of the DELAY queue independent of of the original job dayfile. This implies a listing processor, probably similar to ENQUIRE.

C) The priviledge of DELAY queue access will probably need validation. Currently, the job entry processors do not perform full validation - although NLR says that this is coming.

D) The documentation for the JOBCARD will have to be changed. This would not be so for a completely separate processor.

E) The Chairman (LAL) has indicated to me that he is highly critical of ANSI 76 FORTRAN features because they lacked uniformity and do not go as far (are not completely implemented) as they could. Should the system group be less critical of a proposal that lacks uniformity and completeness?

F) As long as I have invoked the name of the leader, I would remiss if I did not say that it was presumptuous and overbearing of the original proposal to psychologically undermine the imagination of the systems group by using the phrase " a parameter must be allowed on the JOBCARD ...". (Note: MUST) The systems group is only nominally a democratic group in such cases - but of course, LAL never has said differently. Perhaps LAL could set aside the decrees from the proposals in some fashion so that dictator-style proposals could be more easily recognized and need not be voted on. (I add the note - LAL himself used the term dictator as a self-referent.)

G) There is no way to 'un-queue' a job. Unless, of course, you consider phone calls to the OD and the operator as acceptable methods; a method which I, as an OD, strenuously oppose.

I therefore propose that: a) the original proposal to place the DELAY parameter on the JOBCARD be dismissed, and b) the following processor be implemented :

QUEUE(ACTION, QUEUE, QUALIFIERS)

The purpose of this processor is threefold:

1) Allow the user to control the destiny of his own jobs/files.
2) Unify the scheme of enqueue, dequeue and requeue files.
3) Include PLOT & FILM output files for queues instead of using ad-hoc procedures (viz. PSTPRC, MF501).

I envision the parameters as follows:

ACTION	FUNCTION
ENTER	Place file in queue - if file is DA, copy it.
REMOVE	Remove file from queue and make it local.
TRANSFER	Move file from one queue to another.
SUSPEND	Let file remain in queue, but allow no action.
RESUME	Continue action on file.
SUPPRESS	Suppress action on carriage control.
RECIND	Suppress SUPPRESS.
DROP	Set 'user drop' (like op-drop) for file/job.
PURGE	Remove & destroy file in queue.
LIST	Print contents of queue.

QUEUE	DESCRIPTION
ACTIVE	Job currently running.
INPUT	-
PRINT	-
PUNCH	-
FILM	-
PLOT	-
ALL	
ALL-EXCEPT	

QUALIFIER - Any type of argument that makes sense;
perhaps file names, perhaps Jobnames, etc
.

I have tried to anticipate the criticisms of the QUEUE proposal and answer them as follows:

a) Full implementation is unlikely by 01JUL76.
This fault lies foresquare with the Executive Committee, as nearly as I can tell. There was simply too short a time to allow for a decent design. Due to the time constraint, I would suggest that the manual solution be adopted, that is, holding the decks for submission after midnight. On the other hand, it may not be difficult to introduce just a part of the QUEUE proposal right away, for instance:

QUEUE(ENTER, DELAY)

would be preferable to the JOBCARD solution and could be used as a base for further work. The user wouldn't have to relearn another card.

b) Some combinations of keywords are inappropriate.
So, in all probablility is: COPYBF,LGO,PUNCH.
We can't protect everyone from himself.

c) People will drop other people's jobs.
Of course, one should only be able to drop one's own jobs. However, the possibilty of malicious action must be considered. First of all, as in regular KRONOS, the user hash must be used for all jobs, and the hash must be guaranteed unique. Naturally, the use of job-dropping facilities should probably be validated / controlled.

d) The proposal goes too deeply into the system.
The implementation doesn't have to be completed at one fell swoop - do it in pieces, but do it.

e) The philosophy is too divorced from the current system.
So come up with 500K and buy NOS. We should either provide decent service ourselves or be content to follow CDC down the garden path to their bank.

f) People from Applications do a lot of talking, but don't contribute any effort toward actual coding.
Applications people aren't white sheep, that's for sure, but we are often in touch with users who are attempting to do reasonable things, but the things don't have any way of getting done. In short, what difference does it make where the ideas come from? (Was that an 'ad hominem' argument?)

g) QUEUE is too time-sharing oriented.
That's the way the world is headed. I feel that we have been deficient in not giving the general user access to, say, a TTY strategically located near batch Job submission sites so that he can monitor his job if, when and how he wishes, and act to affect its course through the system. After all, we are supposed to be a service group; we don't own the user's job, HE owns it, and we are merely the purveyers of a service. I think we have been entirely too proprietary about Jobs once they have been handed to us. The restrictions which we place on information about Jobs is surely too tight. As KCM has observed, simply because a user has changed physical locations doen't mean he's someone else, and he SHOULD have access to any and every bit of information about his job, regardless of what site he originally submitted his job, regardless of where it is going, and regardless of what site he may to use to interrogate the system.

B. Summary.

There exists an inversion in the perception of the placement of 2 system features: SUPPRESS belongs in the system I/O packages, not in a utility routine, and, generalized user-controllable queue management belongs in a CPU utility routine, not in the Job entry facility. Such and inversion should be corrected.

For a good discussion of extensions to this queue proposal, the is a memo about Job Administartion available from TWL describing a series of generalized interfaces between users and the system. (Note - I was elated to see that many of the ideas in this QUEUE proposal were covered in the previously mentioned paper.)

//////////

Master User Revisited - B. Williams

When we discussed the operating system consolidation, the master user concept was only briefly touched upon. It is worthwhile to redefine its implementation (at least it is for me since I don't know what we decided on some of the issues). MERITSS currently gives "master user numbers" which have *'s in them to each department or user group to maintain those user number "under them" which match the master number in all non-* characters (e.g., 2104901 is under 2104***).

This master may access all permanent files belonging to those "under" the master. The master may, if the Telex priviledge command bit is set, logoff or monitor the activities of those under the master number with the ACCESS/MONITOR/LOGOFF commands. The master also changes passwords and maintains permanent files for those subordinate numbers by ACCOUNTing to the numbers and PURGEing or using the PASSWOR command (access word permission is carried over ACCOUNT's but only those validated may use multiple ACCOUNT's).

The theory behind this setup was that individual departments should have control over the passwords, files, and activities of user numbers they control without having to contact UCC all the time, thus making it easier on them and us. The ability for each master group to control all user numbers in the system was not desired for security reasons. A security check can't be run on each master and often these master numbers can fall into the wrong hands. Also, incorrect entry of the terminal number on LOGOFF is less disasterous. The number "*******" was created to control all system numbers but its use was restricted and its password changed often.

The issues here are several. Is the addition of the LOGOFF command desirable? Should we restrict ACCESS/MONITOR/LOGOFF to only those under the master (thus, for non-master numbers to only others up on the same number) or can we use some other method to allow masters to control only some users (such as, ORG!, another bit)? Will the PASSWOR,X=Y/UN=___ capability eliminate need for PASSWOR permission to be carried across ACCOUNT's (is it, in fact, on the CYBER)? Is it desirable to allow any user to ACCOUNT as much as they wish (is that, in fact now possible on the CYBER)?

By answering these questions, we can define the master user implementation.

SYSTEM MAINTENANCE: People and Procedures

Last Week's Systems Group Meeting - T. W. Lanzatella

1. Kevin Fjelsted agreed to perform a study to determine the functional differences between WRITEUP and INFORM. This study will be used to help determine what kind of enhancements ought to be added to WRITEUP to fill the gap when INFORM is abandoned.

2. MERITSS/MIRJE Merge.

   a) Regarding the library tape and compilers, we passed over each of the major products and decided which would be common and which would be different.

(* means common)

LISP	*
BASIC	* (after testing for additional field length requirements)
MNF	The 6400 will run the short core version
APL	The 6400 will run ~~APL*CYBER~~. The 74 will run ~~APLUM~~
COBOL3	* needs repairs
SNOBOLC	*
PASCAL	*
ALGOL	the 6400 will run ALGOL2 + mods the 74 will run ALGOL 3
FTN	not on the 6400
RUN23	not maintained but remains on the 6400 only along with combined I/O
TSRUN	Phase out

b) The MERITSS conversion document, now available as a MERITSS sysnote, will be published in the next UCC Newsletter.

3. A long, heated discussion on Don Hamnes' DIVERT/DISPOSE proposal resulted in the following changes and additions.

a) Change the Divert keyword from DVRT to DIVERT.

b) Eliminate the NODVRT command and add an ON/OFF parameter to DIVERT.

c) Make DIVERT errors non-fatal.

d) Eliminate the utility DIVLIM and replace it with a utility which processes the entire VENUS file producing a report to be displayed at each remote terminal.

e) Change the conditions under which a user receives a message indicating that a file was diverted so that a user is always informed that a file was diverted regardless of equipment status.

f) Embellish the comment page resulting from a ~~secondary~~ output file being diverted.

g) An attempt to dispose to an illegal site should be a fatal error.

h) ~~DISPOSE~~ macros should be moved to COMCMAC (form CPCOM).

5. Delayed Input Queue

a) The general mechanism suggested in the last DSN (2, 11 p4,5) was accepted in principal.

b) A D on the jobcard (before the period) will determine a delayed input job.

6. Re: K. Fjelsted's SETVAL proposal (DSN 2, 11 p2)

   a) We agreed that since the MERITSS version of program LIMITS already possessed certain of the proposed features, we would continue to enhance LIMITS.

7. Re: E. J. Mundstock's CALLPRG proposal (DSN 2, 11 p2)

   a) We concluded that an ability to transfer CALLPRG bites between machines would be useful and therefore we accepted the proposal.

8. Re: B. Fox's ACCSTAT proposal (DSN 2, 11 p3)

   a) The proposal was accepted with the exception of the default values of the line width parameter which should be determined by the equipment type associated with the output file.

//////////

CALLPRG News - M. Riviere

COPYM, PLAP, SPRUCE AND DECODE, previously announced in the May 24 Newsletter were introduced as fetch type Callprg packages on June 12. DECODE was originally announced with the name PSCODE. Andy Mickel is maintaining COPYM and PLAP and SPRUCE. John Strait is in charge of PSCODE. Andy transferred the maintenance of PASCAL to John Strait and the maintenance of TYPESET to Michael Frisch. At the same time John made current PASCAL to become past, future to become current and introduced a new version of PASCAL as FUTURE. Also in June 12 Dennis Lienke changed the current and past versions of IMSL. This change is described in the UCC Newsletter of June.

During last week I changed the entry for Michael Frisch fetch type library RUNMIN, the entry for Howard Kurs fetch type library SYSMISC (COBOL 3 library) and my entry for COBOL in order to retrieve the past version of SYSLIB with those products.

In June 19 Michael Frisch moved several of his packages to magnetic tape (FUNPACK, fetch and past, PLOTTAC and UMPRG.)

Also on June 19 Andy Mickel made future LISP to become current and current to become past. Andy also made available documentation for LISP in a Callprg fetch type file, LISPDOC. LISPDOC is 120 pages long and reside on magnetic tape.

Brian Hanson added the CALLSYS routines to the future version of FORTRAN and to FT3LIB. The CALLSYS routines were announced during last quarter. The documentation of the CALLSYS routines is still under review but available on a public indirect access file, CALLSYS, on the STF pack, under Brian's account number (YZE6069). Michael Frisch is considering renaming CALLSYS to PROCPAC. The name is independent of usage and is used only for documentation purposes.

Also in June 19, I introduced level 12 FTN and level 12 FORTRAN as future products and I replaced the installation texts by their level 12 versions.

On the Library Tape, James Mundstock changed the routine RANSET$ in FT3LIB to add the entry points RANSET, RANGET and RUNGET$ for MNF.

//////////

Proposal on Text Processing - M. Frisch

Presently there are three text processing programs on the Cyber 74. TYPESET, written by Mike Huck, can be used for output on the time-sharing terminals or batch line-printers. PLAP, from the University of Colorado is similar in capabilities to TYPESET and is a FETCH file at Andy Mickel's request. TXTEDT, heavily revised by Marisa Riviere from a Purdue University program, can be used for output on the 3M Electron Beam Recorder (EBR) microfilmer or batch line-printers. I have recently received a program from the University of Calgary called FMT which has similar capabilities to those of TYPESET and PLAP but which is better designed, I believe.

My proposal is to install FMT in CALLPRG as a current program, keep TYPESET as a current program unless its usage drops significantly, keep PLAP as a FETCH program, and move TXTEDT to FETCH since we have never supported it wholeheartedly. This last is because the EBR doesn't produce readily reproducible output.

I have just ordered a special 30 cps time-sharing terminal for documentation. It has a replacable type-element allowing different character fonts. Its printing quality is very good and will be used for upper/lower case final copies of writeups and probably the newsletter. Other computer centers are using similar terminals for these purposes. We need good software for such a terminal and FMT has been used successfully for this purpose at Calgary. (TYPESET and PLAP could also be used since the terminal communicates in ASCII mode.) To fully use the terminal, some additional software features will be needed. In the long run, I plan a revision to FMT to accomplish this.

The writeup for FMT is presently available to staff via:

```
ATTACH,FMTDOC/PN=STF,UN=YZE6031.
```

//////////

B1=1 in Relocatable Versions of Common Decks - by K. C. Matthews

The question is, how should the relocatable versions of the common decks be assembled for SYSLIB?

Many COMPASS routines in the KRONOS system set register B1 to a 1 and leave it 1 throughout the program execution. This results in shorter object code in some cases where a 15 bit instruction may be used to add 1 or subtract 1 instead of a thirty bit instruction.

The SYSCOM macro anticipates this usage by allowing a B1 argument.

The macro call

```
        SYSCOM          B1
```

defines a symbol called B1=1

(the macro defines some other symbols, of course). The symbol B1=1 is used to determine whether the convention that B1 contains 1 exists for this program.

Many common decks want B1 to be 1. Some, like COMCWTW and CDMCWTO, check the symbol B1=1. If it is not defined, they generate an instruction to set B1. This is done with the two statements.

```
IF      -DEF,B1=1,1

SB1      1
```

Thus, B1 is always 1 for the common deck; but if the symbol is defined (indicating that B1 is 1 globally) there are two savings:

(1) The 30 bit instruction is not generated.

(2) The 30 bit instruction doesn't have to be executed each time the common deck is called.

Some common decks, like COMCARG and COMCSTF use the fact that B1 is 1, but don't have the assembly check. Instead, it is documented in the common deck listing that B1 must contain 1 as an entry condition.

Currently, all of the common decks are available as relocatable routines. Since they are only assembled once, some assumptions must be made about B1. It seems that there are three possibilities.

(1) Assemble all common decks without the

```
SYSCOM B1
```

statement. Then common decks like COMCWTO will set B1 to 1 upon entry, whereas decks like COMCARG still require B1 to be 1 as an entry condition. This is how common decks were assembled on the old SYSLIB.

(2) Assemble all common decks with

```
SYSCOM  B1
```

this is what is done in the latest version of SYSLIB. Then practically all common decks require B1 to be 1 on entry (as documented in WRITEUP, SYSLIB).

(3) Change the common decks like COMCARG to check if the symbol B1=1 is defined. If it is not, have them generate a

```
SB1          1
```

instruction. Then all common decks would not require B1 to be 1 on entry.

Option 2 produces the most efficient code, and is in effect now. Option 3 offers more protection for users, especially those who don't read the SYSLIB writeup. Option 1 seems too inconsistent. A few users were burned when SYSLIB went from option 1 to option 2, but they may all have changed their code by the time we can make any changes, so they should probably not be a factor in any decision we make.

//////////

System Consolidation Status - B. Williams

Everyone has been working diligently on the 6400-Cyber 74 system consolidation. To insure that two (or more) people are not working diligently on the same feature while no one is working on another, and to make sure we have covered all areas to be converted, I have prepared an outline, of sorts, which may be used to match people to various conversion projects. This outline follows:

Also, we have placed a sysnote on the 6400 system explaining the consolidation to users. This note will be updated as new conversion problems are discovered and decisions made. A copy of the note was sent to all on the Systems meeting list last week. Please comment on this note!!

We should tell the users when a trial of the consolidated system will be available as soon as possible. This will hopefully be sometime in July (18 or 25) during 6400 system time (Sunday 12 - 4 or something). Can we have the system looking externally as it will on August 15 by one of these dates?

I. Operations

Manual/Operator Instructions.
J. SELL/R.D. WILLIAMS

CMR, IPR, LIBDEC
WILLIAMS/LANE

ISF/SYSPROC.
WILL/LANE

U,A,UTILITY PROCEDURES.

DEADSTART/LIB. TAPES
MR, TNL, BU

II. DEADSTART

CEJ CHECK (SETU3).
WTS OR KCM (ACK TS)

DAYFILE RECOVERY.
AUTO

INITIALIZE WAITS FOR AUTO.
B.W.

DELETION OF UNUSED ROUTINES ON TAPE.
WTS (SUSPEND ALIGNOFF)

SYSTEM= IN CMRDECK (SETU10).
KCM/TS

III. PERMANENT FILES.

ULTRA - PRIVATE (PFMU6 + others).
TS

SS= CAPABILITY (PFILU2 + others).
TS

PERMIT ERROR CHECKS (PFDUMP,LOAD,ATC)
TS

CONTROL WORD DUMP MODS (PFDUU1).

PFCOPY "SAVE FILE" KLUDGE (PFCOPY).
B.W. WILL [illegible] ABOUT

PFM SPEED UP MODS (PFMU1).

EXTRA PFM PAUSES (PFMU2).

IV. MTR/CPUMTR.

CPUMTR RUNS IN MODE 7.

CIO TTY READS (SETU2,CMTRU7 + others)

ECS CLEAR CORE MODS (COMTRU3).

MTR TRAP BAD TELEX REQUESTS (MTRU6).

V. TELEX

NOTICE/NOTIFY CALL (TELXU78).

NEW NOTICE FEATURES.

SUBSYSTEM ON LIST, RUN (TELXU28).

MASTER USER, LOGOFF (TELXU64).

TRAPS (TELXU68,etc.).

ERROR ENHANCEMENTS (TELXU48, etc.).

WRITEUP COMMAND.

BARROW TERMINAL TYPE.

APLUM COMMAND, or APL CHANGE.

ROTARY ALGORITHM.

CPORT.

GAME COMMAND (PFILES).

ACCSTAT COMMAND.

1TO POTS TO DUMP FIX (1TOU2).

TELEX LOW CORE HAS 1TD IDLE ADDRESS (1TDU26).

VI. USER VALIDATIONS.

DEFAULT VALUES CORRECT IN MODVAL.

ACCOUNT CARD PERMISSION BIT.

DSD PERMISSION BIT.

PASSWORD HASHING.

MODUAL EFFICIENCIES, SECURITY (MOVLU10).

LIMITS CHANGE CAPABILITY, PARAMETERS (MDULU11).

VII. XMIT.

VIII. DAYFILE TIME PARAMETER.

IX. NOTICE/NOTIFY PROGRAMS.

X. ECS.

ROLLOUT SHORT FILES (IROU7).

TRACK SIZE OF 1010B SECTORS.

CORRECT TRACK CALCULATION IN SET.

RECOVER FROM PARITY, DATA ERRORS.

XI. MASS STORAGE DRIVERS.

DEFAULT CHANNEL NUMBERS.

6DI ENHANCED ERRORS (7SI, TSP).

CORRECT 808 CODE IN ERROR (6DBU1).

XII. TALK.

SOURCE AND PROGRAM.

FAFT IN ISF FOR IT (ISF*U2).

XIII. CATLIST

"BR" PARAMETER (CTLSU3).

ALPHABETIZE (CTLSU2).

XIV. COMPILERS/LIBRARIES.

MNF/RANIO,SET/SAVSCT, ETC.

COBOL.

BASIC.

ALGOL.

LISP.

SNOBOL.

PASCAL.

APL.

RUN23.

COMPASS.

-TEXT.

-COMMON DECKS.

XV. WRITEUP.

INSTALLATION.

CONTENT SELECTION.

CONVERSION OF INFORM DATA.

ADDITION OF DATE/INTERACTIVE CAPABILITY.

XVI. CALLPRG.

INDEX CREATION/CONTENT SELECTION.

ADDITION OF XMIT CAPABILITY.

XVII. MISCELLANEOUS JOBS.

CONVERT FILE SUBSYSTEMS.

CREATE NEEDED UN's AND UI's.

CONVERT VALIDUX FILE.

CONVERT ACCOUNT DATA BASE.

XVIII. GAME/DEMONSTRATION PROGRAMS.

TEN.

EYE.

STARWAR/WAR.

LEM.

POOL.

TWL/BW

XIX. ACCOUNTING.

PF STORAGE WARNING MESSAGE.

ACCOUNTING MONTHLY/QUARTERLY REPORTS.

BJF

MONTHLY USER PF REPORTS.

STATISTICS REPORTS.

BW/Kam

ACCSTAT.

BF

XX. CONVERSION. BW/TWL

DOCUMENTATION.

PROGRAMS.

BW

XXI. PRINT CONTROL CARD.

TS

XXII. 026/DIS.

TS

CORRECTIONS.

TS

ENHANCEMENTS.

TS