UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Newsletter

26 September 1978 Vol. 4, No. 18

Send all comments, criticisms and contributions to the editor: T. W. Lanzatella
University Computer Center, 2520 Broadway Drive, Lauderdale, MN 55113
The University of Minnesota is an equal opportunity educator and employer.

TABLE OF CONTENTS

NOTICE OF CHANGES TO THE SYSTEM

The following changes will occur on Thursday, 28 September.

Tom Lanzatella repaired an error in RESEX processing of multiple equivalenced VSN's wherein the last VSN of an equivalenced VSN list appears in the preview display after the first event expires. At this site we only display the first VSN of an equivalenced VSN list. Tom also altered the account closed message to instruct users to call Judy Krieg instead of Jim Foster.

John Larsen installed a yet to be proposed new option for MSADR which provides the ability to convert from a physical to logical disk address.

Jim Mundstock installed a change to COMCMVE which improves the speed of the routine when a CMU unit is present.

Kevin Matthews altered several routines by installing another iteration toward common queues. Kevin also repaired an error in the PF utilities (PFU, PFDUMP and PF LOAD) which caused disk equipment to be occasionally turned off.

Don Mears again repaired 1CD so that a 7600 code will not print. Don also installed an important facet of the shared queue facility, that is the ability for 1MR to generate an Inactive Queue File Track (IQFT). Shared queues can now be recovered across deadstarts. Additionally, Don added code to EXPORT (actually 1HS) which will cause the routine to abort gracefully rather than to hang the system.

Brian Hanson added a correction to LINK which repairs the notorious RANDOM ADDRESS NOT ON FILE SYSLIB bug.

Marisa Riviere corrected an error in CALLPRG processing of MID option on the CALLPRG index entry.

Tim Hoffmann contributed an initial version of the CPU portion of the AUTODIVERT facility - now called SITEVAL. Program SITEVAL performs site validation.

Jeff Drummond contributed the following changes.

1) New versions of the console games LUNAR, LUN, STARWAR, WAR, KAL and MIC were added to MPLNOS.

2) A new console game DPR was added to MPLNOS. This game translates the CPU P-register into display coordinates and generates a display from this data.

3) Program ACCFAM was repaired so that when the ILLEGAL USER ACCESS message is output, a monitor call error does not result. This occurred because the message was within five words of FL.

4) The common deck COMCRDC was altered so that a READC request will skip to end of line if the working buffer fills up. (This mod was not proposed and cannot be fully tested. - ed.)

5) All "FET too short" situations in RESEQ were eliminated.

6) Program CATLIST was altered so that EP is set in the FET before calling PFM. This change repairs the situation where if a CATLIST request is issued for a removable pack when the pack is not mounted, the job bombs with RESEX FAILURE. A CATLIST request will now cause the job to roll just like any other PFM request. See Jeff's illuminating article later in this newsletter.

7) Jeff altered SYSCOM by adding a LINQ symbol for 8-lines per inch printing.

8) Jeff added a new (unproposed) option to CATLIST (PD) which causes the report to be printed at 8-lines per inch.

9) Jeff repaired an error in PFILES wherein error messages were not being reported if a PW option had been specified.

System procedure files U, K, DF and A were all changed in preparation for the demise of pack STF.

No KRONOS change will be installed this week.

PROPOSED CHANGES TO THE SYSTEM

MSADR REVERSED - by J. L. LARSEN

A need has been indicated for the control card MSADR to produce the logical address given the physical parameters.

This proposal, therefore, suggests that the MSADR control card be changed in the following manner.

The present control card reads:

MSADR,EQ=,T=,S=.
And produces the dayfile message:
Converted address UX CXXXX SXXXX.
Where the X's represent digits.
The new control card (which is the second option and is exclusive from the above form) reads:
MSADR,EQ=,PU=,PC=,PT=,PS=.
And produces the dayfile message:
Logical address TXXXX SXXXX.
The default values of the parameters are:
Equipment EQ=<DI>
Physical Unit PU=<0>
Physical Cylinder PC=<0>
Physical Track PT=<0>
Physical Sector PS=<0>

Note: For the new control card to be recognized, one of the "P" parameters must be present.

SYSTEM MAINTENANCE: People and Procedures

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

The following proposals were accepted or rejected.

1) Marisa Riviere's proposal to install a fast attach file manipulation utility was approved after some discussion of the potential interlock problems (see DSN 4, 17 p. 137).

2) Steve Collins proposal to add a parameter to relevant PFILE commands allowing users to manipulate the UCW was approved (see DSN 4, 17 p. 138). The new parameter will be UC=.

3) Andy Bremanis' proposed new account file message was approved in principle but rejected because the message was greater than 40 characters. Andy will work with Tim Salo in devising a shorter message.

4) Earl Schleske's proposal to add an assembly option to COMCRDW was accepted (see DSN 4, 17 p. 139). Earl agreed to do the work.

5) Steve Collins proposal to add CSET: NORMAL or CSET: ASCII to the STATUS report was approved (see DSN 4, 17 p. 139).

Larry Liddiard announced that the employee appraisal forms are due by 1 October.

//////////

Callprg and Library Tape News - by M. Riviere

On September 21, the 6400 Library Tape was modified to include a new version of MNFCLIB. This new version has some changes on the MANTRAP and FORSYS routines.

On September 21, the Cyber's Library Tape was restructured. Now this tape is composed of three different files. A Cyber 74 only file, a Cyber 172 only file, and a common file.

Current versions of SYSIO and COBOL5 were placed on the Cyber 74 only section. New versions of SYSIO and COBOL5 with DMS-170 interface are used on the Cyber 172. These products were submitted by Steve Reisman. In addition, Steve added QUERY, UPDATE, REPORT and DFRCV, a data base utility, on the Cyber 172 section of the tape.

Also on September 21, I largely modified the Callprg index in the Cybers. This modification consisted of dropping the old KRONOS index section, rearranging packages according to the programmers responsible for them, and removing the ones no longer in use. I needed a lot of help from whomever was willing to contribute, a lot of phone calls, and a lot of luck. The new version of Callprg ended being almost right. Not perfect. Thanks for your help and patience, all of you.

In addition to the index reconfiguration, the following modifications took place:

Steve Reisman introduced DDL, a data base description language. DDL is a package available only on the 172. Steve also rotated the versions of TRIAL making future the current version and the current, past.

S. Yen inserted a new set of BMDP packages, BMDP77, and SIR a scientific information retrieval system.

I inserted an entry to make accessible the 6400 Callprg index program library. The entry is CPOPL64, fetch type. I also introduced an entry for PERT78. PERT78 is a CDC product that I installed at M. Frisch's request.

//////////

Removable Packs at UCC - by J. J. Drummond

This article summarizes the changes made to PFM, PFILES and CATLIST with regard to removable pack requests. On a stock NOS 460 system, a permanent file request (such as CATLIST) for a file on a removable drive that wasn't mounted generated a *RESEX FAILURE.* message unless the *NA* parameter was present. The *NA* parameter caused the job to rollout and generated an operator request to mount the specified pack. Since our tape management system traps out requests for non-existant packs, it was felt that jobs should always rollout for a removable pack that wasn't mounted. This change was effected by setting the *NA* parameter (error processing bit in the FET) in PFM if the file being accessed was on a removable pack. However, this caused some unfortunate side effects since other errors could occur and not be reported because the error processing bit was set. Since PFILES already sets the error processing bit (to generate better looking error messages and process the *WB* (wait busy) parameter), CATLIST was also altered to set the error processing bit. The code to do this automatically was then removed from PFM. This means that the CP programs doing permanent file requests (PFILES and CATLIST) both set the error processing bit and do their own error processing (as opposed to PFM setting the error processing bit and no error processing being done). Staff who have programs doing internal permanent file requests from removable devices should adjust their programs accordingly.

//////////

A Multi-Primer - by Bill Elliott, Minnesota Educational Computing Consortium

MULTI is a set of operating system routines and modifications which provide a way for one copy of a program (called a task) to interact with several time-sharing terminals simultaneously.

Since one job is executing for several terminals, file interlock and availability problems disappear. Two-second response time is nearly always possible, even on a loaded system. Programs which use the permanent file system for inter-terminal communications would find the MULTI environment faster and more efficient. The possibility of running some CAI programs in Multi-mode would be worth investigation.

A primary roadblock to faster response time is the speed at which jobs can be moved to and from control points for execution. While jobs are waiting to be executed, they are placed in a rollout queue. Jobs are scheduled for execution from this queue, on a priority basis, as memory becomes available. Rather than speeding up this rolling process, MULTI reduces the number of jobs which need to be rolled, and thus reduces system overhead.

Because one program executes for several terminals, the percentage of time spent doing useful work is increased. This results in better response time for all users, not only those in Multi-user mode.

"Task" is the name given to a program designed and compiled to be executed in Multi-mode. Two classes of tasks exist: system tasks and user tasks. Task execution begins by saving the task binary as a permanent file which is public or explicitly permitted to UN=MULTI. To initiate Multi-mode execution, one types MULTI,TASKNAME for system tasks, or MULTI,TASKNAM,USERNUM for user mode tasks. If the desired task is not present, it is initiated by submitting a job to the MU (Multi-user) job origin.

System tasks execute subject to validations for UN=MULTI. They may reside on the DS Tape (if SSJ or other special permissions are necessary) or on a permanent file (VIA CALLPRG, or special UN=SYSMU).

User tasks are executed in a sub-control point to prevent use of exotic RA+1 requests and excessive memory and PFM usage. Because of this, extensive error diagnosis can be performed and the exchange package plus memory dump saved for inspection.

Each task is a separate job functioning as any other job in the system except that its queue priorities rival those of the MT origin and that CIO processes read/write of its INPUT/OUTPUT file with a different overlay. The heart of MULTI is the CIO overlay. All the rest of it merely gets things into position. This means minimal TELEX overhead (600 words), minimal CPUMTR overhead (no SIC functions used).

MULTI was designed to co-exist with TRANEX. However, since TRANEX has not been available, we have been unable to check this out.

Tasks communicate with terminals through their INPUT and OUTPUT files. Text and status information is placed into the task's input buffer by MULTI. Each text line and status entry is separated by a control word. This control word identifies the entry type (text or status), the terminal of origin and one or two arguments. In some cases, there is no text to associate with the control word. In this case, a null text line is always supplied for consistency. Possible status notifications which may appear in the input buffer are:

New User Wants to Log Into Task
Existing User Hung Up the Phone
Existing User Interrupted His Terminal (S or I Key)
Terminal Ready for More Output

Based on the content of its input buffer, the task responds by writing to its output buffer. As with the input file, each text line or command must be prefaced by a control word. This control word designates the nature of the text line following it (Text or Command), the destination terminal and one or two arguments. Possible commands which may appear in the output buffer are:

Delink User from Task and Return to Telex Mode
Acknowledge Terminal Interrupt
Request Input from Terminal
Request Notification When Output Nearly Complete

Tasks do not execute more frequently than once in two seconds. As input becomes available from terminals, it is transferred to the input buffer. When an endfile status (030) is returned, no more text is available from the terminals assigned to the task. The task should complete processing those input directives and text which were provided, flush its output buffer and perform other housekeeping functions before reading its input file again. A second read with no data available will return another endfile status and roll the task out until input text is again available and at least two seconds have elapsed.

On a machine language level, tasks differ widely from their program counterparts. The manner in which data are received from the users (text prefaced by control words) varies from the normal method (text separated by end of line marker only) where the program is dedicated to the individual terminal. This is to be expected.

Fortunately, it is possible to minimize this difference by providing common decks and versions of existing compilers to handle most of the bookkeeping. A set of COMPASS common decks and a version of the BASIC and MNF compilers have been developed which will aid in the conversion of existing programs into tasks which may then be executed in a Multi-user environment.

All control bytes except the 0004 byte (log off user) are processed when operating in Multi-user mode. User logoff can still be directed when the task issues a BYE or HELLO as the logout text when it returns the terminal to the normal TELEX mode of operation.

Even with the aid of common decks and compilers, not every program can be converted into a task. One must keep in mind the increased memory which is required to maintain the status of each individual user on the task.

It is the responsibility of each task to internally GET or ATTACH those permanent files required during the execution of the task. Tasks are permitted only to GET, ATTACH and REPLACE permanent files. A limit of 10 permanent file operations is imposed.

Up to 64 terminals may execute a user task simultaneously. There is no upper limit for system tasks. The task may enforce a limit of fewer terminals by simply issuing its own apologetic message and returning the terminal to TELEX mode. When a terminal is returned to TELEX mode, it is in the same state as before entering Multi-user mode. The subsystem and input mode (ASCII or NORMAL) remains unchanged.

MULTI was developed at MECC to meet the demand placed on the system by a few heavily used interactive programs. Current tasks include several demonstration routines with a system MAIL and editor routine in the wings. A few users have written their own tasks with more under development. MULTI has been well received and is contributing to better response time this year for MECC users.

A special thanks to Bill Wells for his suggestions and guidance with the current design. MULTI is brought to you courtesy of those (in)famous MECC systems programmers Rick Nordin, Mark Rustad, and yours truly.

//////////

Computer-Related Terms Used in Documentation - by Andy Mickel

One of the by-products of the User's Manual project was to establish a unified documentation scheme which would make sense to the ordinary user. It is heartening to see this unified documentation scheme firmly in place, functioning and being carried forth by Mary Boyd and Sara Graffunder. A complete description appears in Chapter 1 of the User's Manual.

Another by-product was a partial attempt to choose terms carefully in the User's Manual. Now this should be linked to other documentation. It is important to reduce the total number of terms and to discard those which can be potentially confusing. Actually, the list below was collected mostly by talking informally with various people at UCC, particularly systems programmers at Lauderdale.

Not So Cool (Out)	Cool (Like, wow-man! It's IN)
account number	user number
control card,	control statement,
job card,	job statement,
data cards,	data record,
job deck.	job file.
(To be general enough to cover submit batch jobs.)	
7-8-9 6-7-9 6-7-8-9 cards	end of record, file, information
(All of the above correspond to current Control Data usage. They prevent users from asking, "Where do I put the cards into the teletype?")	
timesharing	interactive
TELEX command	interactive command
(A much more accurate description of the mode of processing. Batch jobs are "time-shared" too, thanks to swapping.)	
unit record,	line
card image,	line
line-image.	line
(I know, I know, the Record Manager people will protest. IBM/CODASYL terminology has pervaded their thinking so much, they don't know a simple concept when they see one. Control Data's solution at level 472 is to introduce a whole new set of different, redundant, and confusing terms for Record Manager.)	
coded file	text file
(text files are stored in "binary" form too)	
real-time programming	time-dependent programming
legal	valid, correct
illegal	invalid, incorrect
(Bucking a trend to allow lawyers from talking their way into the computing profession; apologies to Pat Jarvis!)	

//////////

Why We Shouldn't Necessarily and Blindly Go To New Versions of NOS - or - What Do We Really Want From Our Operating System? - by Andy Mickel

During the first weeks of August while working furiously out at Lauderdale helping with the NOS conversion effort, I saw on a blackboard, a half-erased description of NOS (Network Operating System):

1. Complexity
2. Obscurity
3. Unreliability
4. Dozens of character sets to choose from
5. New and interesting design flaws
6. Cyber Record Manager
7. Common Memory Manager
8. NAM (Network Access Method)
9. Etc.

I would agree this is amusing because of the all-too-familiar ring of truth it carries. Since the demise of KRONOS 2.0 in 1973, we have seen the operating system, as produced by CDC, get a little more unwieldy with each release.

In switching to NOS, we have just completed the largest operating system change since the MOMS-KRONOS conversion in Fall, 1974. Unlike the change to KRONOS, which was well worth the effort (and was a lot of work), I feel like a lot of other systems programmers that the recent conversion was only marginally worth the effort. I found myself running very hard all year just to stay in the same place. I'm tired, and what's worse, I still hadn't recovered fully from that MOMS-KRONOS conversion four years ago! I certainly don't want to do it again, and so I'm moved to invest some more time and effort into writing this essay.

I'm tired of the operating system changing out from under me when the expected benefits are outweighed by the magnitude of the disruption; I'm a user too.

Maybe not everyone at UCC realizes the extent of the effort mostly put forth by systems programmers (unsung heroes). Over 40,000 lines of locally-developed code were converted. This is the code that tailors the bare-bones CDC system to our site's needs (the features added are far too numerous to mention here; the surest way to appreciate the difference is to log-in on a stock CDC system!).

Because of the modification scheme (the Minnesota Modification Method) we use (which preserves a stock version of the operating system), we were able to adapt these mods reasonably quickly and reliably. Contrast this to other sites such as the University of Colorado, which has heavily modified its KRONOS 2.1 level 7 system (but not by using our modification scheme which makes anOPL out of the mods themselves). This year they estimated it would take approximately the same time to move to level 10 KRONOS (12 months) as it would to retrofit the newer product set into the level 7 system.

Or contrast our situation with say, the University of Nevada or Stanford Research Institute, for example. They are locked into KRONOS but have also bought NOS (very expensive), not to run it, but must to steal code to put into KRONOS in various places.

The point is that it should not be taken for granted that these conversion efforts are accomplished without a lot of sweat and tears, and that we at the University of Minnesota Computer Center are privileged to have an operating system so flexibly maintained.

Why did we go to NOS?

Last December when we more or less decided to go to NOS, it was argued that we needed to have: 1) Multi-Mainframe (MMF), 2) Full-Tracking Disks, 3) Transaction Facility (TAF), 4) NAM 2550 support, and 5) the ability to run the new product sets of the future. NOS 1.2 (level 3 or PSR level 460) only provided a couple of these; most of the goodies come with later versions of NOS and so NOS 1.2 was deemed our first step. Several of us argued against going to NOS but were convinced by the conventional wisdom, and at least the NOS Conversion Task Forces helped air out the issues. But we succeeded in converting to NOS 1.2 as smoothly as can be expected plus having thrown in a few other "minor" changes to boot (SRU accounting, new character set, etc.) by working very hard all this year (from late December to present). Many of us are fatigued to say the least.

And there is still talk of moving the whole system to NOS 1.3 (level 4, PSR level 472 or correction level 477) within a year.

We no longer are compelled to go to NOS 1.3 and beyond.

Let's reassess the situation. We have full-tracking now thanks to Kevin Matthews who retrofitted it into NOS 1.2. We nearly have a MMF that fits our needs (CDC sure doesn't make one) thanks to Kevin Matthess, Jeff Drummond, Don Mears, and others. We decided not to buy TAF.

NAM has been a can of worms from the beginning. We don't need the new product set. (More on these two items in a moment.)

Two considerations should be made from now on for every across-the-board operating system change. 1) It is a struggle to write good user documentation and then see it wiped out in a conversion. The Tape Guide is a case in point: it must now be updated to reflect a few NOS differences at a time when we need to spend our efforts writing documentation for other areas which have never been documented. "Part of the normal course of events," you say. Well it just so happens we wouldn't have a Tape Guide if it weren't for a dedicated programmer named Bill Elliott who could and did write user documentation. And we don't have him now to fix it. Similar documentation examples can be cited as well. 2) University users should be polled at least once a year (or before a proposed change) to obtain their judgments on what our software and hardware improvements should be. As it is, they aren't even consulted, and if it weren't for them, we wouldn't be here.

Back to NAM (How can CDC sell a thing with a name like 'nam?). NAM has two parts: Remote Batch Facility (RBF) and InterActive Facility (IAF) to replace EXPORT (and SUPIO) and TELEX. Its reputation at VIM last year in Minneapolis caused reasonable people to froth at the mouth. NAM's liabilities when compared with TELEX and SUPIO are that it consumes a large portion of central memory, and performs badly in CP time. Releases from CDC have been consistently late in delivery.

Besides the CM portion of NAM, there's a front-end processor running in the 2550 which has a better reputation, and it is (*wow, gee*) written in Pascal. But when Arnie Nelson showed me the Pascal code, the Pascal cross-compiler, and the performance of same, I agree with him that it's a mess: Pascal transformed into a bizarre perversion!

Let's look at the product-set argument. CDC is producing FTN 5 for the FORTRAN-77 standard. But FTN accounts for less than 10% of Fortran usage on our machines, and this share is even shrinking. The new FORTRAN standard has few new useful features if you listen to the FORTRAN experts, and the vast momentum of existing FORTRAN IV programs will cause the new FORTRAN to be accepted slowly. To counter FTN 5 we will have M77 to implement FORTRAN-77. Besides CDC has insured that the FORTRAN common library will be unusable from MNF at level 472.

So what about COBOL 5 (implementing ANSI 74 COBOL)? We already have it working under NOS 1.2 (level 460). We won't need to change it.

What about maintenance, you ask? This is a case where it will be less disruptive to the vast majority of users to retrofit the new product set into the operating system (if there really is a need to acquire it) than to change the entire operating system to fit a few products. The numbers are dramatic: Just looking at languages, (there were also over 100,000 runs each of SPSS and S2000) MNF and TSF had a combined use of 910,000 and Pascal 272,000 versus 81,000 for FTN and 154,000 for COBOL. It's even a good bet that there aren't that many FTN and COBOL users that use the new features anyway.

Another upcoming horror attached to the new product set is Common Memory Manager which inhibits the very feature it is supposed to enable: dynamic memory. Further, it's another slower and bigger software system which would cause us to lose another chunk of our machine.

So if CDC is leading us down the path to ruin, then why do we need to follow them? We have shown for four years that we can ably maintain (document, etc.) a heavily-modified operating system and even perfect it to the point of 100% up-time in a month (something we never can do when we stay current with CDC). We are planning to replace the Cyber 74 (with a good possibility of a non-CDC machine) in a few more years and that means that the 172 or its replacement would merely front-end the new, major system, so why would we need to run the current CDC operating system then and afterwards on the lesser machine?

CDC seems to be imitating IBM more and more. First of all, they're making money now and have discovered that they like this better than delivering cost-effective computer systems. Prices of manuals, maintenance, and software have soared. CDC has learned from IBM that if you produce operating systems which consume more computer resources (mainly Central Memory and CPU), then the customer will buy more central memory and CPU. CDC is incidentally naming its software components with obscure three-letter abbreviations (as illustrated by NAM, CMM, CRM, IAF, BAM, RBF, TAF, etc.) and CDC sales are using unfortunate pressure tactics (they won't sell any other operating system besides NOS or NOS/BE to run on a 170). In short, they have degenerated unfortunately to IBM's level.

What are our goals for our computer system?

I think we'd all agree that we want to provide the most reliable, cost-effective computing possible to the University community and then to outside users. And we should strive to make the computer system simple to understand and thus to use so that we open it up to more people, not less.

Why then do we feel compelled to follow CDC's software mistakes which are now running contrary to cost-effective computing? I refer to mistakes which are software systems that: 1) most importantly make the computer system more complex and obscure and thus less simple to maintain, document, and to use, and 2) consume computer system resources (run slower and larger). Example: Record Manager. If it weren't for MNF and MSUIO, record manager would have dealt a serious blow to the 6400's timesharing performance. Little things add up. When you have to swap 10K extra for a dozen interactive jobs all the time, you've lost 10K of your machine. Example: Cyber Loader - it's easily 50-100 times slower in loading absolute overlays and takes far more memory for any kind of load. Make Cyber Loader default on MERITSS and you'll be looking up at six fathoms of water. What has it done to the Cybers? We just lost another chunk of our machine to overhead.

The conventional reply to all this is that Record Manager and Cyber Loader provide features which were desirable and requested by CDC sites. My reply to this is: "Fine, but I doubt that those features necessitated performance that poor." If one can sum up recent advances in programming, it has been clearly shown that there are much simpler ways of getting computing done. Software need not be this complicated or inefficient.

Quite often contradictory arguments are given in system group meetings to provide the rationale behind proposed changes to the system. Example: "We want to add this feature to the system as a separate control statement so as not to invalidate existing CDC documentation and to avoid altering a similar program (e.g., CATLSYS vs. CATLIST)." In contrast: "ROUTE must be modified to fit our needs; the X parameter will be made non-functional and the COPYAP parameter will be added."

My opinion after watching us vacillate without a clear systems strategy for four years is that it certainly depends on the individual case. But I feel we should be less afraid to alter existing CDC programs (control statements), because the plethora of redundant programs is proving to be harder to explain in general documentation. We have demonstrated that with important programs like DISPOSE and now ROUTE our local documentation is sufficient to describe the feature to the user community in spite of the erroneous description remaining in the NOS manual. We should take this course more often.

//////////

Cyber 74/172 Deadstart Dump Analysis from Monday, 11 September 1978 to Sunday, 24 September 1978 - by K. C. Matthews

Tuesday, 12 September

16:40 (DD-17) Cyber 74

Export hung with no apparent activity. A level 3 deadstart was required.

21:10 (DD-20) Cyber 74

Telex was down with 1TA, 1TD hung at the Telex control point.

Wednesday, 13 September

08:42 Cyber 74

Disk channel 27 hung up preventing a complete deadstart. The engineers fixed the problem temporarily.

12:00 (DD-21) Cyber 74

PP program 1SB hung at the Export control point. A level 3 deadstart failed, and a level 0 was required. We have had several cases where it looks like a level 3 recovery should have worked but didn't. This will have to be investigated.

12:19 (DD-22) Cyber 74

CIO hung at a control point. The problem was analyzed and was corrected on 22 September.

Thursday, 14 September

13:49 (DD-24) Cyber 74

CIO hung again. Same problem as the day before.

Friday, 15 September

The Cyber 74 hung several times in the morning with problems on channel 27. This problem was left over from Wednesday and was corrected.

15:05 (DD-50) Cyber 74

The system hung. This dump has not yet been analyzed. There were many EXPORT aborts around this time, however. The EXPORT problem was corrected by Don Mears at 17:00.

Sunday, 17 September

22:55 (DD-15) Cyber 74

1CJ hung. There was an old 1CJ problem for which KCM's fix was bad. The real fix is in on the current deadstart tape.

Monday, 18 September

18:48 (DD-2) Cyber 74

A disk error on device DN12 caused a job to hang up. The disk error was found to be a solid flaw by L. Ozga and has been flawed out. We will have to investigate why the disk error caused such a problem.

09:02 (DD-31) Cyber 172

PP program 1CJ hung. This was the same problem as on the Cyber 74 on the previous day.

09:19

Same problem as above.

Wednesday, 20 September

12:12 (DD-5) Cyber 74

The same disk error as on Monday hung the Cyber 74.

13:11 (DD-6) Cyber 74

The system hung. This has not yet been analyzed.

Thursday, 21 September

15:08 (DD-7) Cyber 74

CIO hung. The old problem that was fixed the next day.

17:55 (DD-10) Cyber 74

Same problem as above.

17:04 (DD-32) Cyber 172

Monitor was hung. This dump has not yet been analyzed.

Friday, 22 September

14:45

The power failed; all machines were down. Up at 15:10.

//////////

6400 Deadstart Dump Analysis (9/-11- 9/24) - by R. A. Williams

Date	Description	Tape
780922	A power failure caused the system to go down.	N.A.