UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Newsletter

20 May 1980 Vol. 6, No. 10

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

Kevin Matthews corrected an error in 1SB, SUPIO's helper PP-routine. The problem was that 1SB never checked the response from COMPRSS (read system sector) when the system sector for a print file was read. If the system sector happened to have been bad then eventually the system would hang. Because 1SB is short of space, a new zero-level overlay (OSS) was invented which does nothing except check the response from COMPRSS. Kevin also corrected a minor problem in UQM wherein the wrong byte in the EFNT (ECS part of the FNT) was checked in order to determine whether a file bound for another mainframe had been picked up. Additionally, Kevin introduced two new common decks COMCCV8 and COMCC16 which can be used to convert to eight or 16 word catalog entry (respectively).

Bill Sackett repaired a mod clashing problem in LFM which caused a breach of execute-only file security. The error, an old one actually repaired several months ago, was that execute-only files could be UNLOCK'ed and subsequently read.

Brad Blasing installed the following changes.

1) A non-upward compatible aspect of a R5 PSR modset was corrected. The problem was that the KCL IF(FILE(lfn,MS)) function failed if lfn had been VSN'ed.

2) Brad installed some PSR code into CATALOG which guarantees that the output buffer will be flushed if CATALOG aborts. A bug discovered and corrected by Marisa years ago.

3) Brad installed some PSR code into OVJ which causes the search for a 26 or 29 on job cards to quit at end-of-card. A bug fixed long ago by D. W. Mears.

4) Brad installed a PSR modset into TCOPY which corrects end-of-line conversion under the 63-character set. A bug corrected by Brad shortly after our upgrade to R4.

5) RETAIN function processing in PFILES was changed to use UNLOAD rather than RETURN to avoid decrementing the users resource count when RETAINing on a removeable pack.

6) Brad changed DSDI, the deadstart dump interpreter to reflect the changes he made to the control point area in order to install the schedule-around-hung-control-point feature.

Jeff Drummond changed TRN to improve threaded jumps.

Arnie Nelson added three new site ID's to SUPIO.

PROPOSED CHANGES TO THE SYSTEM

Random Access to Procedure Files - by P. Kachelmyer

I wish to propose a change to BEGIN processing in CCL. The suggestion is that if the procedure file has a directory (OPLD) then CCL should recognize this and use it to locate the specified procedure. If the procedure file does not have a directory then normal sequential access would be used.

//////////

Guess Which Version of COMPASS We Are Going to Link With the Next Compiler? - by M. Riviere

The first release of FTN5 that we have received is level 509. This version does not link properly with our current version of COMPASS which is level 485. Well, this is not news, the COMPASS compiler incompatibility across levels shows up quite often.

The way the new COMPASS is installed this time is by installing renamed versions of the COMPASS (1,0) and (1,1) overlays in the System and modifying the compilers to use the renamed products. This set-up implies the maintenance of several similar versions of COMPASS overlays in the System or the modification of the compilers according with the name changes. I made a modification to FTN5 and to COMPASS level 509 in order to make FTN5 load the (1,0) COMPASS overlay from a local file call "COMPASS" whenever FTN5 itself was loaded from a local file.

Additionally, I made a modification to COMPASS in order for the assembler to load its higher overlay (1,1) from the local file "COMPASS" if COMPASS itself was loaded by a compiler and the compiler makes the specific request for the local file load. I used bit 59 of location CP.USER of COMPCOM for the compiler-assembler communication. CP.USER is labeled as "space reserved to user" by CDC.

FTN5 is finally linking properly with "its own" COMPASS. Now they are both offered as Callprg fetch type products. Both products, assembler and compiler, do not need other modifications to be installed as System products (COMPASS still needs to have its local modifications added, but that is an independent problem).

Now, after all this information I can present my two past proposals:

1) To install the COMPASS modifications as a permanent modification;

2) To propose and document how the two words (or the bits of the two words) reserved by CDC for users in COMPCOM are going to be used.

//////////

About OPT=2 on FTN5 and FTN - by M. Riviere

The optimization level on FTN5 and FTN is currently set as:

	Default (No OPT)	Second Default (OPT Only)
FTN5	0	2
FTN	1	2

If we are going to make the optimization default value 2, what are we going to use as the second default: 0 for FTN5 and 1 for FTN?

Additionally, since FTN5 is a new product in our system, should we install FTN5 initially with the new default value? The change may be less confusing if we start with the new default values.

//////////

LIBEDIT U Parameter to Call LIBGEN - by Mike Frisch

I propose that a U parameter be added to LIBEDIT which will automatically strip off any ULIB record at the beginning of its operation and automatically call LIBGEN after the editing phase. This will greatly simplify the user library editing process.

//////////

User's CALLPRG Permanent File (Second Try) - by Mike Frisch

After the last Systems Group meeting, I was informed that MECC has implemented my proposal for searching a user's permanent file named CALLPRG. Their method solves several of the problems brought up at our meeting. I'm therefore submitting a revised proposal that uses the MECC ideas.

They have a validation bit that says whether you want to have a permanent file named CALLPRG searched. (I presume it can be set or cleared by SETVAL.) The bit is copied to the job's control point area. If it is set at execution, a permanent file named CALLPRG is searched. If none is found, CALLPRG clears the bit in the control point area. Thus, if no permanent file exists, only one permanent file search is made per job or timesharing session.

At our meeting, I was told that I had the current order of the search written incorrectly. It is:

search the user's local file CALLPRG if there is one,
search the SYSTEM file,
and then search the UCC file CALLPRG,

so that a user program of the same name would be found on the user's CALLPRG file first. I'm changing my earlier proposal to use the MECC scheme of permanent file access so I am also changing the point where it is searched to be: if there is no local file named CALLPRG, search for a permanent file if the appropriate bit is set in the control point area and GET it.

If a user has the appropriate bit set in the validation file and has a CALLPRG permanent file but doesn't want to use it during a given job or session, he or she can create an empty local file named CALLPRG (for example with WRITER,CALLPRG.).

I now feel that a defective CALLPRG file should cause a job abort with an appropriate dayfile message. This will help in finding errors in the CALLPRG file.

The value of this proposal is important to course instructors who want to make it easier for their students to automatically access special programs without needing GET(CALLPRG).

SYSTEM MAINTENANCE: People and Procedures

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

The meeting began with a dandy presentation by CDC on the LCN (Loosely Coupled Network). Copies of the folios used during the presentation are being circulated at Lauderdale and Exp. Eng.

The following proposals were presented.

1) Larry Liddiard's proposal to change M77 and MNF for the benefit of student time-sharing users was divided into two parts (see DSN 6,9 p. 64).
   a) We all agreed that OPT=2 should be the default optimization level for FTN and FTN5. The change will be made at a quarter break.

b) We all agreed that T=E was an acceptable default option in MNF and M77 for time-sharing users. However, we did not like the parameter -T on the RUN command used to disable tracing. Don felt that cracking this parameter would be difficult. We left open the question of how, exactly, the option would be disabled but Don seemed to favor T=0.

2) Mike Frisch's proposal to make CALLPRG search for a permanent file named CALLPRG and to use this file (if found) as the Callprg index was rejected (see DSN 6,9 p. 65).

3) Mike Frisch's proposal to imbue CATALOG with the ability to print out the 7700 table fields preceeding the comment was accepted assuming the ITEMIZE couldn't already perform this task (it can't) (see DSN 6,9 p. 64).

////////

Callprg and Library Tape News - by M. Riviere

There are no Callprg index or Library Tape modifications scheduled for May 27.

The next set of Callprg and Library Tape Modifications will be taking place on June 14, in order to fit within the end of the quarter break.

On June 14, COPYCH in the Library Tape will be modified by a new version provided by A. Hastings.

Other modifications for June 14 should be submitted by June 5 at noon.

//////////

Cyber 74/172 Deadstart Dump Analysis from Friday, 2 May through Thursday, 15 May - by K. C. Matthews

Tuesday, 6 May

11:34 (DD2025) Cyber 74
PP program 1SB (a SUPIO auxillary) was hung requesting an illegal mass storage driver. Analysis showed that 1SB was not handling system sector errors properly, and this led it astray. We have no idea how the bad output FNT entry (and hence system sector error) were generated.

Friday, 9 May

20:00 Both Machines
ECS errors appeared on both Cybers, bringing the Multi-mainframe system down until it was repaired at 21:25.

Sunday, 11 May

21:05 (DD2001) Cyber 172
The System failed with the CP #1 exchange package in low core. CP #2 was doing a TIM request for MAGNET at the time. This is the third dump like this in the past year or so, Brad Blasing noticed. We still aren't sure what to do to try to duplicate the problem.

Thursday, 15 May

15:15 Cyber 74
The maintenance procedure detected a CPU failure on the Cyber 74. We were down for a total of only 32 minutes while the problem was analyzed and repaired.

Polishing Our System - Andy Mickel

At the Valentine's Day System's Group meeting, my contribution to the round-robin discussion of ideas for future systems changes was that we should "polish" our system to make it more "user-friendly." I gave the specific example of the need to repair the inconvenience of routing a full-ASCII text file to the printer without specifying EC=A9. On any day one can find some listing in the site bins which is full of up-arrow excape characters.

Both Larry and Dr. Patton have emphasized the need for us to do our best in the face of "competition" caused by other departments getting their own computer systems. I would like to add that personal computer systems (micros) which offer full-ASCII capability, superior software-writing tools (such as screen-oriented text editors), and simple, user-friendly command languages also pose a challenge. Fortunately the Network Interface Box (NIB) proposal of Tom Jacobson and Pete Zechmeister may provide the means for our centralized systems to continue to contribute a necessary service (rich software culture, number crunching, and large data-base abilities) for our users by being easily accessible by personal computers acting as intelligent terminals, etc.

I'm not proposing we rewrite the operating system from the top down, or make any drastic changes. I stress that the bywords are "fine-tune" and "polish." There are definite constraints on what we can accomplish with the current system, both in hardware and software. We can only strive to improve what already exists according to specific software-engineering goals.

Since Valentine's Day I've been working on this article. Below is a list of criteria, gained from experience in language design, which I claim we should consider as we make modifications to software. For more such nuggets I think that we should comb Larry's collection of SOFTWARE Practice and Experience! I know I have complained in the past (in the DSN and at systems-group meetings) that we should agree on the "set of correct ways of doing things," probably because I didn't recognize that these things are open questions for everyone in our field (just look at the literature), and partly because I was just plain frustrated.

My point is to provoke thought. I want to stimulate us into thinking in whole-view terms instead of perpetuating ad-hoc solutions, suggestions, etc. I've observed that we've given contradictory rationale for specific software changes in a cyclical fashion since are switched to KRONOS (1974), and I, for one, don't like it.

Human-Oriented Designs

Designs should be "human-oriented." This doesn't imply that they be goof-proof or hand-holding, rather a conscious effort should be undertaken to make them straightforward.

For example: We have modified the stock CDC system to accept decimal instead of octal seconds for time limits. In other words SETTL(20) is human-oriented when 20 = 20 decimal and not (a machine-oriented) 20 octal (= 16 decimal) seconds.

Human Engineering

We should think in terms of "human engineering." This means: "not setting up an

accident to happen" (as my Dad used to tell me after I placed something near the edge of a table and later knocked it off and broke it). An example is the pitfall described earlier in routing full-ASCII files to the line printer. I suggest that at least a flag be reserved in the system sector to indicate a "full-ASCII" attribute. The attribute would be set by at least WRITEUP, the "COPY"-type utilities, and the text-mode mechanism, and can then be used automatically by various operating-system programs, which are output-device drivers, to do the correct things. This would not completely solve the problem in all cases, but would minimize the some pain and suffering. [Don Mears argues that the result of a "partial solution" might be even more confusing.]

Another example is the "feature" (having historical origins and perpetuated by the familiar rationale of "upward compatibility" with the past) of the MNF control statement which automatically loads and executes the binary program by default without requiring an explicit parameter to do so. This represents a pitfall when people assume it is similar to all other compilers which require a separate LGO statement for simple jobs.

Stated in another way (and by another person, namely Bill Price of Tektronix--they strive for "excellence"), human engineering means striving for designs simple enough to use correctly.

Ease of Use

We all pay lip service to "ease of use." That's why we're all programmers--we're lazy, right? But "ease of use" for a narrow aspect of the system (e.g. a single control statement) can be offset by the enormity and complexity of having too many control statements.

When viewed as a language, the set of operating-system control statements should:

a) be small in number. (This reduces the number of things we have to remember.),

b) be non-overlapping in function where possible. (This helps (a) above, and avoids confusion in our minds.),

c) incorporate some structure to reduce complexity. (An example is the TAPES control statement which possesses different "modules." Another good example is a discernable group such as permanent-file control statements which represent a more-or-less unified set of commands roughly at the same level of abstraction. This is not true for resource-allocation control statements such as SETJSL, RFL, RESOURC, etc. Another example of non-uniformity is the PASCAL control statement which does not have equivalenced parameters, like every other compiler.),

d) avoid unwanted generality. (While power can be provided by generality., it hampers ease of use!

(Examples: UNPAGE and COPYU each has a plethora of frills and options which are hard to remember when non-default usage is required. The "powerful" (general, flexible, etc.) control statement COPYCH is principally used for reading and writing tapes with possibly non-standard blocking formats while fully converting EBCDIC or ASCII characters. In addition, COPYCH is general enough to convert character sets from one disk file to another. This caused unclear descriptions in the COPYCH documentation, because the same descriptions had to be "correct" to handle these different uses.),

e) incorporate efficiency of implementation. (In other words, this goal balances ease of use. It would be nice if ease of use directly (not inversely) reflected the amount of work the computer system must perform to carry out a task. That would be an honest message to all of us who use the computer system. I think the best we can hope for, though, is to try not to hide inefficiencies where possible.), and

f) avoid dangerous side effects. (The biggest pain of all is the woeful lack of warning and lack of documentation telling a user that the OLD and NEW commands clear all local files (true even on some non-CDC systems!). Maybe this side effect shouldn't exist, so that we wouldn't have to worry about warning/documenting! However, pleasant side effects are desirable for a "user-oriented" system.)

Old Standbys
---

a) Names of control statements for utility purposes should be active verbs.

b) Defaults for control-statement parameters should be tied to common usage.

c) We should establish conventions for parameter names.

Experimenting

We as systems programmers write programs for all the users, not just ourselves. I think we should make an attempt to experiment with small groups of real users before we implement software changes or provide new and permanent software features. [Mary Boyd points out that this is especially true for documentation--if we only had the lead time!] The book: Psychology of Computer Programming by Gerald Weinberg, is a modern classic which describes the benefits derived from performing controlled experiments on computer users.

An Experiment

I conducted a informal, simple, little experiment of my own early this school year. I gained the cooperation of Tom Rindflesch (a linguist working in the Ancient Studies Group) and Dick Rubenstein (a project assistant) to write down what they thought were the deficiencies of our computer system. This was before both Tom and Dick became "institutionalized" to accept the faults our system has. Dick made his list after having had only 1-1/2 weeks exposure to our system.

What I hoped we would gain from their points of view was a fresh perspective. We veterans are sometimes "hardened" in spite of ourselves. Appendices follow which present Tom's and Dick's lists just as they gave them to me. I'm reluctant to annotate their lists; they are presented here just as they were to me. The lists contain incorrect, correct, good, and irrelevant points. Some are simply the result of lack of information, but then that also reflects somewhat on the state of our system's documentation....

[Don Mears suggests that an analysis of Help-Line questions would be better and more complete than my ad-hoc experiment. I agree.]

An Example

I would like to pick just one item as an example to illustrate how thinking about any one of their points would contribute to our own learning. Tom's first point "eliminate the slash" really struck me the wrong way! Without thinking I would have argued back: "but you don't understand the function of the slash, Tom, it provides the structure to the control-statement syntax." After thinking though, it occurred to me that I was only considering the slash in the context of Permanent-file control statements (except CATLIST!) and not PASCAL, GTR, COPYU, etc. In other words, because the slash is not implemented uniformly it is confusing to a beginner (as in: "When do you use a slash?").

I felt humbled.

Dick's points: 1., 6., 7., 9., and 13. struck me as being especially worthy of speedy consideration.

Specific Items

Below are some suggested software and documentation changes resulting from the application of the general criteria outlined above for human engineering / ease of use.

Sara Graffunder gave me a list of four things:

1) Eliminate PACK after text mode - see DSN article many moons ago with same suggestion by Bill Sackett. [Jeff Drummond, Brad Blasing, and Steve Reisman say that even CDC will make the change.]

2) Make ASCII the default CSET for TELEX jobs (less difficult to reverse effects of error). [This would require of course a change of habit for most users who are accustomed to the unimportance of different case for letters (which is now default) for entering programs.]

3) Minimize differences between direct and indirect access files from user's point of view (ACQUIRE and RETAIN help - but

```
GET,FNAME
FNAME NOT FOUND
```

when FNAME is DA is not helpful)

4) Change "ILLEGAL COMMAND" to "NOT AN INTERACTIVE COMMAND" or "COMMAND NOT VALID IN THIS SUBSYSTEM." [Don Mears adds that "ILLEGAL COMMAND" is inappropriate when "Job Active."]

I'd like to add several items:

5) Make the default ROUTE disposition code ("DC" parameter) "PR" (to reflect most frequent usage) instead of "IN" (equivalent to SUBMIT--how unnecessarily redundant!). I assert that using non-CDC defaults (which necessitates additional UCC documentation) is a poor counter-argument (and Bill Sackett agrees).

6) It's an incongruity of our system to press the escape key which produces a "DELETED" message when there is a delete key on the keyboard. [This has confused none other than than short-course wizard Rich Franta. Larry suggests the message "LINE DELETED."]

7) There is no UCC solution yet for providing special-request instructions for SUBMIT (batch) jobs.

8) Our character set conversion 1-1/2 years ago essentially traded in six bad and different character sets for one bad, uniform character set. Therefore we reduced the complexity of the problem. But now a rough edge of this character set really sticks out: there are two translations for @ and ^.

Rather than implement full-ASCII wastefully as CDC proposes (which clobbers the goal of efficiency and which would require a rewrite of nearly all the software) maybe we should try to think about this single rough edge. Our own Steve Collins and Mark Rustad at MECC have ideas.

9) The terminology of "normal" and "extended" is unfortunate. How about "subset-ASCII" and "full-ASCII?"

10) Empty files should be treated uniformly by the system. RETURN, COPY, etc. should print messages informing us that a file is empty. (Be meaningful, friendly...)

11) I would like to eliminate extraneous and oddball control statements. If a control statement has very low usage and a function which is duplicated by another processor, then we should remove it from the list. Example: Just in the area of text editors there is dead wood! AMEND is just a crude text editor, invented before the days of XEDIT. CDC EDIT is hardly used at all.

12) Three of the "aggravating little changes" which occurred when we changed operating systems last September do not represent different ways of doing things, but rather changes when constitute a step backwards. They are the three interactive commands: NORMAL, BRIEF, and PRINT.

a) NORMAL which was used by most people to switch back to subset-ASCII mode has the side effect of resetting several (TAPE, BRIEF, PARITY, and AUTO) states. This is a real problem with TAPE mode, because NORMAL is the only available means to get out. Fix this and NORMAL is nearly useless because of all its side effects. (I realize that Don Mears warned us at least 2 years ago to get used to the CSET command to switch character sets.)

b) BRIEF now works less well than it did when we had it only as a local feature (it used to toggle brief mode). [Bill Sackett says we should expand its generality--other processors should check it.]

c) PRINT was replaced by a more complicated (general, powerful, flexible, harder-to-use) NOTE control statement which has the side effect of rewinding the OUTPUT file (of all things!). This has been PSR'ed to CDC by another site, I understand.

13) I claim that after a successful attempt to RECOVER when logging-in, the message: "TYPE *CR* TO CONTINUE" is technically incorrect and misleading. This quarter I personally witnessed a user type the characters: *CR* followed by carriage return. I would suggest and improved message (which corrects the error): "Press the carriage-return key to continue." [Sara Graffunder and Mary Boyd and others prefer "press" to my and others' "hit." Note the proper punctuation, and the use of

upper-and-lower case which meets Steve Collins' approval. Many others said: "Better yet, why to we have to press the return key anyway?]

14) WRITEUP(anything=*) should work if "anything" is not an indexed writeup. I would interpret a user's intentions from this example as wanting the whole writeup just in case it's indexed. Instead, the message "WRITEUP FILE NOT IN CORRECT FORMAT" (starting in column 1 (not 2) and not terminated by a period) is displayed.

15) There should be an attempt to improve all acknowledging dayfile messages such as "EOI ENCOUNTERED." I suggest: "END OF INFORMATION ENCOUNTERED."

Jeff Woolsey suggests:

16) Purging a DA file which is attached to your job should change its type to LO so it can be ROUTEd (or whatever).

17) Time should be bi-directional (when it's not too late to stop an event). Hence you should be able to UNROUTE a file and PULL a submitted job if the requests are still in the queue.

18) Full-ASCII card processing is not a system feature, either. One has to read/punch binary cards and translate in the CPU. [CRAY provides this support.]

Bill Sackett suggests:

19) "Efficiency of implementation" is too often used as an excuse for poor user interface. For example: the 40-character dayfile message limit, and also the use of "X," to force batch command processing interactively.

20) The documentation should specify parameter mnemonics and how they came to stand for what they do. [Thanks, Bill--a good example is my description of the "RS" parameter in the CONVERT writeup.]

21) The operating system should prompt you when you are in doubt or error.

Vicky Walsh suggests:

22) Users should never see an octal dump--especially when the problem is not enough field length (MNF).

Mary Boyd suggests:

23) The message "ILLEGAL TERMINAL" (appearing after 4 unsuccessful log-in attempts) is not very informative.

Conclusion

I think some other people in the systems group besides me should use and build on the ideas here to present individual DSN proposals or "white papers" to polish our system as systems projects. The trick seems to be to find the right balance for change by introducing specific changes which move us definitely forward (such as most of our DSN proposed changes are). Contrast this to installing whole releases of CDC systems which change some small number of things for the better and add some small number of useful features, while they change some other small number of features for the worse and add some large number of often redundant, and for our

site, unnecessary and wasteful features which make the whole computer system more complicated to use.

I think some other people in the systems group should publish lists in the DSN of established conventions (see Old Standbys section above).

A final word: Computer users (including myself) yearn for simplicity and have tasted it in various forms so that we know it's achievable. [The best example I can think of is the simple elegance of Pascal when compared to the baroque complexity of PL/1--two languages which give the user virtually the same power. Once it became apparent that there were simpler ways of doing the same things, there was no going back! I predict that Pascal will be cited as the major reason for PL/1's death.] Remember, we are faced with competition from simple, user-friendly, personal-computer command languages.

P.S. Upon reading an earlier draft of this article, Larry added a list of "On-Going Questions." They are presented in Appendix C.

Appendix A - Tom's List

Andy,

Here are a few matters that have come to mind. I apologize for the "preliminary" aspects of this. I have written prelims for the Ph.D. next week and am harried. After prelims, if there's still the opportunity, I would be interested in spending some more time on this: possibly coming up with more suggestions for improvements and possibly coming up with more comprehensive and articulate remarks on the overall syntax of the control language.

-Tom Rindflesch
2-13-80

Suggestions for Improvements to the Operating System - Tom Rindflesch 2-13-80

MORE GENERAL MATTERS

Syntax of Control Statements.

- Eliminate the slash.
- It would be nice if all parameters were mnemonic. In many instances this is the case: for example "EC" (in ROUTE) stands for "External Characteristics." However, many other parameters are not so instructive. The queue to which the file is to be routed is indicated by the parameter "DC." I realize the difficulty involved in making all parameters mnemonic given the available resources. However, if the entire control statement language was overhauled (with perhaps input from linguists) such an objective seems achievable. An extensive review of the language would also allow all parameters which serve the same purpose to be given the same name (from one control statement to the next). In most instances this is the case, but there is room for improvement.

In interactive processing, the user should not have to distinguish between control statements and commands. That is, the use of "X" should be eliminated. The system should be able to distinguish between commands and control statements.

The user should not have to be aware of the difference between direct-access and

indirect-access files. I realize that changing this would mean changing many programs which distinguish between the two; nevertheless I think it would be worth it.

MORE SPECIFIC MATTERS

The system should automatically pack a file after exiting from TEXT mode.

LIST & RUN (LNH, RNH) should not have equivalenced parameters for non- primary files.

RETURN should be renamed. It is misleading; most new users assume that it does "return" the file (for later use). RELEASE would be an improvement.

OLD & NEW should not clear other local files. Especially, since CLEAR is available.

Appendix B - Dick's List

file=GRIPES Gripes About NOS Operating System
Richard D. Rubenstein - 10-13-79

1. NOS ought to flag a file as to its type (i.e. normal or ascii), and then set the mode automatically when processing the file. This feature could, of course, have a defeat option. This idea could extend to such statements as ROUTE, processors like XEDIT and all the language processors.

2. Seven ("7") character file names are restrictive and do NOT lend themselves to proper documentation concerning files.

3. There is no hierarchy of file organization, except as to the pack that the file is on, but this is hardly workable.

4. I don't know of read and/or write keys that can be placed on a file for read and/or write protection. I understand files can be read and write inhibited, but this does not provide the flexibility of keys.

5. The NOS system ought to be able to get permanent files regardless of the physical pack they're on. The PACKNAM command should not HAVE to be used. This is nearly as bad as IBM DOS in which the physical device address must be specified in order to reference a file.

6. NOS ought to be able to flag a file as to its logical type (i.e. binary, source, print-image, etc). Then, source and binary files for the same program could have the same name, instead of making the user provide a manual system to keep track of this.

7. The user batch accounting system is far too easy to break! Individual users within a batch account should be identified to the system by student number or some way to identify a user INDIVIDUALLY. This would help decrease fraudulent use of computing resources.

8. There ought to be a way to save and/or replace (the system could determine which one is applicable to which local file) ALL local files, and then clear the local file pool.

9. XEDIT ought to be able to move lines of text around within a file without having to do COPYD and then READ. Modern text editors have this feature.

10. NOS ought to have a provision for specialized "transparent" commands, transparent to the program that is being executed. In this way, interactive users could get system status, file status, user status, etc. without disturbing the state of the program they're in.

11. The batch RFL command ought to be handled automatically by NOS, again, with a defeat option.

12. NOS ought to be able to search the permanent-file directory to find a file if that file is NOT a local file. Again, a suitable defeat option could be made available for users not wanting the feature.

13. COPY, COPYCF, COPYBF processors print only "EOI ENCOUNTERED" when they cannot find the file to be copied as a local file. A more appropriate message might be "INPUT FILE CANNOT BE FOUND." The user would then know what is going on.

14. When a permanent file is obtained through the use of "OLD,fname", the subsystem which created the file is automatically entered. Instead of printing "READY", the systems should respond "ENTERING XXX SUBSYSTEM."

Appendix C - Larry's On-Going Questions

Do we want to try to work with CDC on cleanup?

1) Advantages - If we stay with CDC systems, then we obtain the future benefits of not having to re-fit our mods each time.

2) Disadvantages - Long-term approach: > 1 or 2 years to get this in.
- May have to make compromises for acceptability.

How do we see the future of our current CDC software and systems?

1) 1 year, 2 years, 5 years? Currently the 74 is paid for, the 172 is one year paid for and thus we could have a new system in summer '81. The 720 3-year lease is up in 1982.

2) Some scenarios:

a) Summer 1981 - change to other than CDC large mainframes; Move 172 ~~load~~ to MERITSS system and run residual CDC requirements there also. Drop 3rd lease year on 720.

b) Summer 1981 - change to new CDC 180 series with virtual memory and run both current system and new one gradually changing to the new (native) one. Move 172 load to MERITSS.

c) Summer 1981 - as stopgap for 2 years, replace the 74 & 172 with a 262K, 20PPU Cyber 750.

d) Do nothing until summer 1982.

How much time should be spent on this project (of polishing our systems) and who should lead?

1) 3 summer months?

2) 1 year?

How many documentation changes can we afford?

What are the top priority items to create a more user-friendly system if we cannot do them all?