

UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Newsletter

6 February 1979

Vol. 5, No. 3

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 SYSTEM11

PROPOSED CHANGES TO THE SYSTEM.12

 THE EX PARAMETER ON CALLPRG - M. Riviere12

 NOS RELEASE 4 - T. W. Lanzatella13

 SRU LIMITS - THREE PROPOSALS - K. C. Matthews.13

SYSTEM MAINTENANCE.19

 LAST WEEK'S SYSTEMS GROUP MEETING - T. W. Lanzatella19

 CALLPRG AND LIBRARY TAPE NEWS - M. Riviere19

 THE ARITHMETIC OF TIME LIMITS - T. W. Lanzatella20

 ONCE MORE ABOUT - M. Riviere20

 CYBER DEADSTART DUMP ANALYSIS - K. C. Matthews21

 6400 DEADSTART DUMP ANALYSIS - R. A. Williams.21

NOTICE OF CHANGES TO THE SYSTEM

NOS Changes

The following changes become effective on Thursday, 8 February.

Kevin Matthews supplied the following changes.

- 1) Program 1CJ was altered to issue a dayfile message when it cannot process an output file because the file does not reside on a shared device. Currently the tracks are dropped with no message. This situation could occur if an output file was RESEQed or UNPAGED.
- 2) Some TWX code from CDC was installed into PFLOAD which corrects the spurious ARCHIVE FILE FORMAT ERROR messages.
- 3) Program REC was altered to hang at deadstart time if the disk used for system dayfiles is bad.
- 4) Kevin partially installed a new multi-mainframe feature called the MMF STATUS WORD (MMSL in low core). Eventually this word will tell which mainframes can receive files, run jobs, etc.
- 5) Program UQM was corrected so that the jobname for routed output is consistent. Currently the banner page jobname for files routed from TELEX is not correct.
- 6) Program ØEF was altered to prohibit files from being routed to the 6400. This is a kludge until the MMF status word (mentioned above) is complete.

- 7) Programs SETTL, SETASL and SETJSL were altered to assume a decimal rather than octal argument.

Tim Hoffmann altered CLEAR processing in PFILES so that if the primary file has been NODROPPed the file is not EVICTed.

Don Mears contributed the following changes.

- 1) Program LSB was repaired so that jobname and external characteristics are passed to IMPORT.
- 2) Program UNPAGE was corrected to assign its destination file to an output type device. This ensures that the file resulting from an UNPAGE command will reside on a shared disk.
- 3) Full ASCII support was added to the 8090 IMPORT package (a big change).
- 4) An addressing problem was repaired in LKT, the Link Test utility.

Jeff Drummond installed the following changes.

- 1) Program PFM was corrected so that when a user CATLISTs a recently created file which resides on a shared device, the file size will be correct. This code is from PSR485.
- 2) MAGNET abort processing was altered to ensure that a dayfile is appended to all dumps.
- 3) Program MSM was altered to output dayfile messages describing specific tracks involved when TRT linkage errors are discovered.
- 4) Programs TRANSIT, TRN, COMSTRN and SEND were reinstalled. The change installs the long-awaited new SEND command. See WRITEUP(SEND) for details.

Brad Blasing repaired a critical error in SFP which was not validating any of the CM addresses in its request. This situation has been PSRed.

KRONOS Changes

The following changes become effective on Thursday, 8 February.

Brad Blasing installed a similar change to SFP as described above.

Jeff Drummond installed new versions of program TRANSIT, TRN and COMSTRN to accommodate the new SEND command. Programs XMIT, XMT, XMTVAL, COMSXMT and ECSXFER were deleted from WPL.

PROPOSED CHANGES TO THE SYSTEM

The EX Parameter on Callprg and RA+64 - by M. Riviere

The EX parameter is used in a Callprg index entry to specify the name of the file that should be loaded and executed when, for a given control statement callable package, that file name happens to be different than the package's name.

The EX parameter is very seldom used and its main function is to force the intervention of Callprg for successive loadings of the same program since no file name associated with the user's statement is left available among the job's files. A new Callprg intervention may be forced in this case to recreate

a specific loading environment, such as adjustments to field length values.

When the EX parameter is used, a request for the loading and executing of the file specified by the parameter takes place through Callprg without any modifications being made to the contents of RA+64. That is, RA+64 remains showing the name of the package requested by the user's control statement.

I have been requested to change Callprg to replace the contents of RA+64 by the execution file name.

A reason in favor of the request is that a given program or overlay may be looking at RA+64 for the name of the file that it was loaded from.

A reason against the request is that should the execution of the program end in an error which produces a core dump of low core locations, the user will find in RA+64 a file name that will not have any meaning in relation to the originally requested package.

What is the most consistent name to use?

//////////

NOS Release 4 - by T. W. Lanzatella

As an effort to initiate general discussion, I propose the following schedule for the installation of NOS Release 4. The conversion effort has already begun for several staff members. By mid June, I suggest we offer Sunday test time to users with total conversion by the end of second summer session. I refer the reader to System Strategy Committee Minutes in DSN 4, 22 p. 185.

//////////

SRU Limits - Three Proposals - by K. C. Matthews

The job card time limit was used in MOMS and KRONOS to limit the maximum amount of CPU time a job could use. This usually was enough to limit the maximum system resources a job could use, since a job normally accumulates CP time as it runs. When a job is totally I/O bound, it may still run up an enormous bill for mass storage within a reasonable CPU time limit.

The SRU limit should now replace the time limit as the primary resource limitation device. This is one of the reasons why CDC invented it. And since job cost is mostly a function of the number of SRU's involved, users can easily limit the maximum cost for a job.

The time limit is inadequate for several reasons.

- (1) It is hard to understand quickly (see TWL's article in this issue).
- (2) A job can burn up a lot of resources even within a moderate limit.
- (3) The time limit now applies to a single job step (i.e., control statement) rather than to the CP time for the entire job.

- (4) The time parameter implies different amounts of computing power depending on which mainframe one is using. At least for the same job, T40 means a larger amount of processing (and cost) on the Cyber 74 than on the 172.

Users who are cost conscious are by now familiar with the SRU. The SRU limiting control statements, SETASL and SETJSL, are more unfamiliar. Rich Franta explained their functions in an article in the February newsletter. SETJSL sets the job step limit - the number of SRU's that can be used for a single job step. SETASL sets the account block limit - the number of SRU's that can be accumulated between CHARGE cards. Where CHARGE cards are not used (UCC), the account block limit is the maximum that can be used for an entire job.

The SRU maximums can limit the total cost of a job for batch job but not for Telex sessions. The situation is analogous to that for the time limit. When a Telex job exceeds the Job Step SRU limit or the Account Block SRU limit, the message

SRU LIMIT

appears at the terminal. The user can type S,N where N is a number less than or equal to his SRU validation. The job can then continue until the next SRU limit. This process can continue indefinitely, so there is no limit on the number of SRU's used per session. We don't think this presents any problem in practice.

Most user numbers are now validated for unlimited SRU's. We would like to begin assigning limited SRU validations, particularly to student user numbers. We would also like to encourage greater recognition of the SRU as the primary factor in job cost. Users should recognize that there is an SRU limit and that raising the SRU limit for a job will mean higher costs in case of an error. To adequately meet these needs, we present three distinct proposals:

- (1) Change the SRU validation Formula;
- (2) The SETSL control statement;
- (3) The job default SRU limit.

Proposal 1 - SRU Validation Formula Change.

This is not a proposal to change the formula by which SRU's are calculated; I don't understand that formula. This has to do with how the SRU validation index (a 6-bit quantity) in the validation file is converted to the SRU limit: the number of SRU's a job can use. There are several 3- and 6-bit fields in the validation file entry for a user number which are converted to actual limits by formulas or table look-ups. We first present the stock CDC formula, which we are using now. Remember that the index is in the range 0-77B. (77B is always interpreted to mean unlimited.) Since the internal accounting rate (non-delay time) is \$0.16/SRU, we will also show the internal cost for the SRU limit.

Formula A - Stock CDC.

$$\text{LIMIT} = 64 * \text{INDEX} + 8$$

This gives the following table:

INDEX	LIMIT	COST
00B	8	\$ 1.28
01B	72	\$11.52
02B	136	\$21.76
03B	200	\$32.00
.
.
75B	3912	\$625.92
76B	3976	\$636.16
77B	Unlimited	Unlimited

Note that multiplications by powers of two are necessary in these formulas because they must be calculated in the PPU. The problem with formula A is that it is not fine enough at the low end. One would like to be able to limit a student job cost to, say, around \$3 or \$5. With the CDC formula, this is impossible. If a limit of \$1.28 isn't enough, \$11.52 is the next available one. The most obvious correction is to reduce the step size. In formula B, we reduce the step from 64 SRU's (\$10.24) to 8 SRU's (\$1.28).

Formula B.

$$\text{LIMIT} = 8 * \text{INDEX} + 8$$

INDEX	LIMIT	COST
00B	8	\$ 1.28
01B	16	\$ 2.56
02B	24	\$ 3.84
03B	32	\$ 5.12
.
.
75B	496	\$79.36
76B	504	\$80.64
77B	Unlimited	Unlimited

This is almost good enough. The low end of the scale is fine enough for good regulation. And the high end of the scale is almost high enough. For if we are willing to let someone spend \$80 at a crack, we might almost be willing to abandon the limit and let the user decide. Jim Foster indicated that Formula B would be acceptable but that he would prefer a few high (but limited) validations.

There are two easy ways in which we can have small step sizes at the low end of the scale and higher step sizes at the high end. We can either approximate an exponential function or we can change the formula above a certain index value. We used the exponential method for time limits in KRONOS, and the formula was copied to NOS. Formula C is an acceptable version for SRU limits. The exponential methods have two unsettling attributes: the formulas are not easily understood and they are not monotonic functions of the index variable.

In Formula C, we divide the index into 3 bits of exponent (E) and 3 bits of mantissa (M). We then present the table in order of increasing functional value, not increasing index.

Formula C.

$$\text{INDEX} = 8 * E + M \text{ (or } 3/E, 3/M)$$

$$\text{LIMIT} = 32 * (M + 5) * 2 ** E$$

INDEX	M	E	LIMIT	COST
00B	0	0	5	\$.80
10B	1	0	10	\$1.60
20B	2	0	20	\$3.20
01B	0	1	37	\$5.92
30B	3	0	40	\$6.40
02B	0	2	69	\$11.04
11B	1	1	74	\$11.84
.
.
67B	6	7	14656	\$2344.96
74B	7	4	17024	\$2723.84
75B	7	5	21120	\$3379.20
76B	7	6	25216	\$4034.56
77B	7	7	Unlimited	Unlimited

In formula D, we change the step size at an index of 66B. Before 66B, the step is 8 SRU's (\$1.28); above 66B it is 2048 SRU's (\$327.68).

Formula D.

$$\text{LIMIT} = 8 + (\text{INDEX} + 1) \quad (\text{INDEX} \leq 66B)$$

$$\text{LIMIT} = 2048 (\text{INDEX} - 66B) \quad (\text{INDEX} > 66B)$$

INDEX	LIMIT	COST
00B	8	\$1.28
01B	16	\$2.56
02B	24	\$3.84
03B	32	\$5.12
.	.	.
.	.	.
66B	440	\$70.40
67B	2048	\$327.68
70B	4096	\$655.68
71B	6144	\$983.04
72B	8192	\$1310.72
73B	10240	\$1638.40
74B	12288	\$1966.08
75B	14336	\$2293.76
76B	16384	\$2621.44
77B	Unlimited	Unlimited

An informal poll showed opinion divided between formulas C and D. Jim Foster preferred formula D. His vote should count as more than 1, since he is the one that has to live with the formula. Hence, I propose that we change the SRU limit calculation to formula D.

Proposal 2 - The SETSL Control Statement.

If we succeed in encouraging SRU limit use for controlling job costs, users will be changing SRU limits in jobs and in time sharing sessions. This would involve using the SETJSL and SETASL control statements. Unfortunately, if both of these statements are to be used, they must usually be used in the proper order. Sometimes SETASL must be used first; other times SETJSL must come first.

Example A.

My user number is validated for unlimited SRU's. When I log in, the Job Step Limit (JSL) and Account Block Limit (ASL) are both set to 500B by Telex. (This is true for almost all user numbers now.) If I want to run an expensive compilation (4000B SRU's), I will have to increase my JSL or else sit by the terminal and say "S,N" when required. But I cannot simply enter:

```
SETJSL, 4000B.
```

This JSL is greater than the limit for my entire account block. When I enter the above statement, it aborts with the message:

```
JOB STEP EXCEEDS ACCOUNT BLOCK.
```

Hence, I must increase the ASL first. Thus, the correct sequence is:

```
SETASL,4000B.  
SETJSL,4000B.
```

Example B.

My user number again. This time I am running a batch job. I am afraid the program is going to get lost when I execute the binary, so I want to limit the total cost of the job to 100B SRU's. So you would think I'd simply insert the statement:

```
SETASL,100B.
```

after my user card. Wrong! When I try that, the job aborts with the same message as before:

```
JOB STEP EXCEEDS ACCOUNT BLOCK.
```

This is because I am validated for unlimited SRU's and so that my default JSL is unlimited. (In Proposal 3 I will want to change this, but the present problem will still exist when I am setting a low ASL.) Hence, the correct sequence is:

```
SETJSL,100B.  
SETASL,100B.
```

Thus, the general rule seems to be that when you want to change both JSL and ASL to some common value, use SETJSL first if you are decreasing their value; use SETASL first when you are increasing their value. If users are like me, there is a 50% chance of getting the order correct when typing in a hurry. Hence, I propose the SETSL control statement. Its form is:

SETSL,N.

where N is a number of SRU's. SETSL will set both the JSL and the ASL to N, in the proper order. N is assumed to be a decimal number unless a suffix B appears. This is consistent with our current (as of today) SETJSL and SETASL statements.

Proposal 3 - Job Default SRU Limit.

When I begin running a batch job, my JSL and ASL are both set to my validated SRU limit. (Remember that number we were calculating in proposal 1?) This is not so good if I have an unlimited SRU validation and make a terrible mistake. Suppose I wish to run a 1000 second job updating my big data base. But before I get into the part which uses a lot of CPU time, I get hung up in my preprocessing program which reads the disk all night. I can do this and accumulate enormous SRU totals from the mass storage transfers, all within 1000 CPU seconds. Hence, it would be nice to have a default limit on the number of SRU's to prevent my job from going forever.

Perhaps a user should have to execute a SETSL control statement to spend over a specified amount of SRU's. For example, choose N (Nachtsheim's Constant) = 6250. 6250 SRU's is \$1000 at 16¢/SRU.

We could change the system so that a batch job would have JSL and ASL initially set to the minimum of the user number's validated SRU limit and Nachtsheim's constant. If a user wishes to spend more than this amount, he would have to include a SETSL (or SETASL + SETJSL) statement. But jobs spending more than this certainly qualify as "big" jobs. The big jobs would have to ask for it to spend an exorbitant amount of resources in a dumb loop. But at least some other exorbitant refunds (some have already been given) might be avoided.

If this proposal is implemented, it would certainly have to be well advertised. Even then, we will be bound to terminate a few big jobs at the Nachtsheim limit at first, and their owners will be furious. This will have to be considered in debating the proposal. I doubt if this change has to coincide with a quarter break, since probably few classroom projects approach the big N.

Another problem is that somebody is going to ask the reasonable question: "You guys say I have to increase my SRU limit. All right, what should I set it to? My job reads from 1 to 32000 case cards and then...etc., etc., etc." In many cases we won't know what a reasonable limit is either. The user may just have to say

SETSL,77777B.

(77777B means unlimited). So some users might be stuck always inserting this card in all their decks. I think the change is still worth it because of the protection it gives to the normal class of jobs.

Another reasonable suggestion would be to have an "S," parameter on the job card which would specify the initial ASL and JSL. This would work like the "T," parameter which now specifies the time limit. In fact, an S makes more sense since the T, which historically applied to the whole job, now sets the time limit for a job step only. Unfortunately, there is not enough room in the job card translator (ØVJ) to do both S and T. Adding S and deleting T would be great if we didn't have to keep everyone's current decks working. We still might want to replace T with S someday, perhaps in conjunction with the release 4 or release 5 conversion.

What exactly am I proposing now? Simply that we decide on a value for Nachtsheim's constant, and implement it as the maximum default batch job SRU limit.

SYSTEM MAINTENANCE: People and Procedures

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

1. Larry Liddiard mentioned that all individuals who have received a copy of Arnie Nelson's questionnaire on the future RJE terminal architecture should respond promptly. Our intentions are to replace 5-8 1004's per year.
2. Mike Frisch reviewed his article on the status of packages on UN=LIBRARY (see DSN 4, 24 p. 197).
3. Larry presented a discussion of the differences between MNF and M77. Larry also announced that MJF, EJM and JPS or JJD would be attending the next VIM conference (5 May).
4. New Business - Larry distributed a copy of the mission statement for a new Micro Computer Group written by Mike Skow. Larry also distributed copies of a recently completed study by Jerry Larson on I/O security. A copy of this is currently being distributed around Lauderdale.

//////////

Callprg and Library Tape News - by M. Riviere

On February 6, Steve Reisman placed SORT/MERGE and SRTLIB on the CDC 6400 Library Tape. These versions of SORT/MERGE and SRTLIB are the Kronos Level 13 versions that were available on the CDC 6400 as Xmit type package until last autumn. Steve needed to install some modifications to the main SORT/MERGE overlay in order to force overlay loading from the System. SRTLIB was also modified in order to remove the CIO= routine that was causing duplicate entry point error diagnostics at load time.

Also on February 6, Michael Frisch and Bill Sackett installed modifications to FORTRAN, MNF and TSF on the 6400 Library Tape. This combined modification consisted of the removal of the University of Minnesota arithmetic routines section from FORTRAN and the addition of the NOS routine SPA. SPA, Substitute Parameter Address, allows NOS and Kronos routines to be compatible at execution time. The compilers were modified in order to set up the loader tables in a fashion such that the scanning of the FORTRAN libraries at loading time is now made in the following sequence: MNFCLIB, MSUIO, SYSIO, FORTRAN, MINNLIB. The FORTRAN library modification was implemented by M. Frisch and the compiler modifications by B. Sackett.

Also on February 6, I replaced SYSLIB on the Cyber Library Tape. This new version of SYSLIB contains most of its relocatable binaries obtained from the current WPLNOS common decks. This version of SYSLIB also has modifications and additions from the CDC level 477 loader routines. The addition of the loader routines was requested by Brad Blasing.

The Cyber Callprg index was modified on February 6 with the following additions.

- 1) J. Mundstock introduced M77, the new version of MNF, as a fetch type package.
- 2) S. Yen changed the future version of SIR.

The next Callprg and library tape modifications will take place on February 27. Modifications for that date should be requested before noon on February 15.

//////////

The Arithmetic of Time Limits - by T. W. Lanzatella

This article presents a brief examination of the arithmetic operations used to manipulate time limits under NOS. The article will reveal that the time limit is not a practical tool for limiting job cost when the limit is intended to be small or accurate.

The job step time limit can be specified in two ways. The first is via the T parameter on the job card. The second is with the SETTL command. In either case, the number specified is rounded up to the nearest multiple of 8 and placed into bits 30-47 of CPJW in the control point area. Programs CONTROL (DISPLAY) and ENQUIRE look at this word when the user requests a display of time limit. This explains how a user can enter T1 or SETTL(1) and then observe a time limit of 8.

Once the time limit is specified, it must be enforced and this is the root of the confusion. Every job step is initiated by CPUMTR with the RLMM monitor function. This is actually a function used to request SRU or time limit but one of its subfunctions is to begin a job step. At the start of every job step, the time limit in CPJW is converted to milliseconds by multiplying by 1000. This 36-bit quantity is then truncated to 24 bits and added to the top 24 bits of CPTW in the control point area. The bottom 36 bits of CPTW is the CP time accumulator for the job and is updated every few milliseconds. Each time the accumulator is updated, CPUMTR also checks whether or not the job has exceeded its job step time limit (subroutine CPT). This is done by restoring the 24-bit field in the top of CPTW to a 36-bit field and then subtracting the CP accumulator. If the result is negative, then the time limit has been exceeded.

Numerically speaking, the above operation can be represented with the following formula in which all division is without remainder, all numbers are decimal and TL is the job card time limit.

$$(((TL + 7)/8)*8*1000)/4096)*4096=L$$

where L is the actual time limit which is enforced in milliseconds.

The simplest way to explain this to users is to say that the enforced time limit is the job card time limit rounded up to the nearest multiple of 8 and then rounded down to the nearest multiple of 4.096.

//////////

Once More About Old Files on Callprg and Writeup - by M. Riviere

The SPL pack is nearly full by now. It would not be a bad idea to prevent it from being completely full instead of letting it be and then think about what to do.

Let us try to make a review of the CALLPRG and WRITEUP files. An RFCAT (UN=CALLPRG(or WRITEUP), PN=SPL statement will produce a list of files catalog information where unused files are flagged with one, two or three asterisks accordingly with the number of days that the files have not been used (25, 60, or 90 days).

Please take a look at your files and consider purging the unneeded ones and placing on magnetic tapes the large ones that are very infrequently used.

//////////

Cyber Deadstart Dump Analysis from Monday, 22 January 1979 through Sunday, 4 February 1979 - by K. C. Matthews

Tuesday, 23 January

12:43(DD-1)

Cyber 74

The display console simply went blank. Everything in the dump looks normal. No resolution of this appears likely.

Tuesday, 30 January

11:36

All Mainframes

A 3 1/2 second power failure brought all three mainframes down. One motor generator could not be restarted causing a delay in coming up.

13:14

Cyber 74

Low core looked hopelessly clobbered. Since the system behaved so erratically, maintenance routines were immediately run and no dump was available. Unfortunately, the maintenance diagnostics revealed nothing wrong. The system came up and ran the rest of the day without a problem.

//////////

6400 Deadstart Dump Analysis (1/22 - 2/4) - by R. A. Williams

There were no unscheduled deadstarts except for a power failure on Tuesday, January 30.