UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Newsletter

9 December 1980 Vol. 6, No. 21

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 repaired several problems related to support of the MICR job card option.

1. The change was causing the scopes to blank occasionally due to a modification error.

2. Jobs using the MICR option which were submitted on a machine without an EFNT were being scheduled rather than held in the queue.

3. Jobs using the MICR option which are submitted on the C172 will now abort. This is mainly for logistical reasons.

Kevin also repaired the CPUMTR memory checksum function. The function was failing on machines with more than 400K words of CM - like the C172.

Marisa Riviere installed the following changes:

1. MFEBR was changed to not throw away lines with zero words of data and to ensure that the number of frames generated is stored in a R-register.

2. A small documentation change was made to CALLPRG.

3. RFMOD was changed so that long files can be changed to direct access type.

Bill Sackett added several PAUSE requests to the PFM CATLIST function. The reason for this change is that PFM was reading permit sectors without pausing. When a file has many permit sectors associated with it, this can tie-up a control point for quite a while. Private files are limited to two permit sectors but no limit is enforced for semi-private files.

Jeff Drummond changed mod PPSAT to adhere to coding standards.

Brad Blasing installed some PSR 501 code which corrects a problem in 1SJ where an SSJ= program (with no SSJ block) interrupted from a terminal would result in all subsequent job steps executing at MXPS.

Andy Hastings repaired an annoying error (unfortunately ours) in 1RI which caused 1RI to hang if a rollin error is encountered while rolling in a DMP= job.

PROPOSED CHANGES TO THE SYSTEM

The Cray Station - by T. W. Lanzatella

By now, most people are aware of the Cray service which UCC is about to offer to the user community. The current plan is to dump Cray jobs to tape and transport the tape to the Cray site via UCC courier. Turnaround time for this process is about 26 hours because of the courier schedule. This means that the same job with corrections or new data can be resubmitted only once every 48 hours. Although we lack proof, we suspect that this will be a deterrent for a certain segment of the potential Cray users. I would like to propose a major change in the way Cray service is rendered.

An active development area at Cray is to produce packages which allow various mainframes to communicate with the Cray via NSC adaptors. A package which runs under NOS, the Cray station, will be completed soon. Hardware to accomodate the Cyber/Cray link is currently installed at Lauderdale. The communications media is a 9600 baud line. We would like to run the Cray station on our production system and use this as the primary means of transmitting jobs to and from the Cray. We say primary because for jobs over a certain size, the effective baud rate of a courier would be better.

Running the Cray station is not without drawbacks. The package needs a dedicated PP and uses 30K of CM. Neither the C74 or the C172 can stand this kind of resource drain if the package is up all day. These deficiences have been discussed with Cray analysts and we don't think it will be very hard to make the CP program roll and the PP bounce. As an interim measure, I suggest we schedule the station manually for 15 minutes each hour.

//////////

Profile Changes - by G. Jenson

Profile is a system routine which offers various methods of control over certain user numbers. The structure of control is;

1. "Master user" number
2. Charge number
3. Project number
4. User number being controlled

The controls are in the form of "Master user" and installation-set limit registers. The registers which limit access include;

- TOTAL SRU's a project can use (SML - set by "master user")
- TOTAL SRU's a project can use per job/session (ISV - set by "master user")
- Expiration date for a project (PEX - set by "master user")
- Expiration date for charge (CEX - set by installation)
- Times of day between which a project can be used (TI/TO - set by "master user")

In addition to these registers which are dynamically updated by the system at end of job/session, or at the second and subsequent execution of the CHARGE control statement, 8 pairs of Registers at the project level, and 8 index registers at the charge level are set aside for use by the installation in any way if desired. Those registers, which are not updated by the system are;

- LR1-LR8 installation limit registers
- AR1-AR8 installation accumulator registers

The only interaction the system has with these registers is to check AR1 against LR1 etc., down the line to assure that the accumulator register does not exceed its respective limit register. If the accumulator exceeds the limit (eg,AR1 > LR1) than the project number in question can not be accessed. This means that if bit *CCNR* is clear in the access word (which means that a valid charge statement must be supplied to the system in any job directly following the USER or ACCOUNT statement) the user can not gain access to the system. The 8 index registers at the charge level allow a "master user" to place default values (using the index algorithum mentioned later) into their corresponding limit registers at the project level when a project number is created. (E.g., IR1 is the index for LR1 etc.) If any of these registers are to be used at a site, the site must determine for what they are to be used, and also provide programs to update the registers in the PROFILB file.

The Computer Science Department, among other users, has long desired some way of more closely controlling users of their accounts in such ways as those mentioned earlier, but more importantly in a dollar-value-resources-expended way.

One way to do this would be to assign a unique user number to each individual requiring control, allowing each account N number of dollars to perform his/her task, and then shutting off the account when these dollars have been used up. This method has some drawbacks, however, which include,

- overhead at the accounting end in managing the added user number etc.
- great amount of work at the Administrative end in creating user numbers, shutting off user numbers, billing irregularities, paperwork, etc.,

The next rational method to approach this problem is outlined in the following proposals.

1. I would like to use 3 of the pairs of limit/accumulator registers at the project level (as well as their corresponding indices at the charge level) designated for use by the installation to limit/control pseudo dollar amount limits for projects. Namely;
   - AR1/LR1 would become SRU related charge accumulator/limit registers.
   - AR2/LR2 would become communications related charge accumulator/limit registers.
   - AR3/LR3 would become supplies related charge accumulator/limit registers.
   - IR1-IR3 would be used as the index for the respective limit registers (LR1-LR3).

2. Since the mnemonics (AR1-AR3), (LR1-LRC) and (IR1-IR3) are in reality meaningless, I would like to change the mnemonics for directives and listings to;
   - AR1 becomes SRA (SRU dollar value accumulator)
   - LR1 becomes SRL (SRU dollar value limit)
   - AR2 becomes CMA (communications dollar value accum.)
   - LR2 becomes CML (communications dollar value limit)
   - AR3 becomes SPA (supplies dollar value accumulate)
   - LR3 becomes SPL (supplies dollar value limit)
   - IR1 becomes SRI (index for SRL)
   - IR2 becomes CMI (index for CML)
   - IR3 becomes SPI (index for SPL)

3. The current algorithm for index values applied to limit registers is; Limit = (index*100B)*100D. Decimal points are not allowed in PROFILE so the dollar value limit if index is equal to 1B would be 6400D or $64.00, the values going upward by a factor of 2 from there. This does not give good control at the low end of the dollar-value-world however, as $64.00 is a bit high for the bottom rung of the index ladder. I therefore propose to change this algorithum to; Limit = (index **2) * increment (increment being initially set at 2). In this case, if index was 1B the limit would be 200 or $2.00. The limit would progress nicely from there into the large dollar figures. In both instances, an index of 77B implies no limit (infinate dollars). This proposed algorithum gives good control steps at both ends of the limit spectrum.

4. Currently the registers which would become SRL, CML and SPL are only able to be updated by users with *CSAP*. Thus, where the pseudo dollar limit is reached, UCC Administration must raise the limits. I propose to give "master users" access to these registers (SRA, CMA, SPA, SRI, CMI and SPI would still require *CSAP*) so that they can add 'funds' to the limit registers without interface with UCC Administration.

5. I would like to implement the program to update the appropriate registers in the PROFILB file into the accounting system to be run daily. There by implementing these controlling mechanisms. Bear in mind that these registers in no way constitute an accounting record, but merely allow users some level of control over the individuals who use their accounts. Although these registers contain "dollar" amounts, these amounts are but a method to facilitate this control.

If there are any questions about PROFILE, I will attempt to answer them. A document entitled PROFILE is also available from the reference library.

//////////

Add MACRO11 and PDPSIM to Callprg - by Richard Rubenstein

I propose installing two new processors in the Callprg index for both MERITSS and the Cyber 74 and 172. These processors are to be inserted as control statement callable (i.e. NOT fetch-type).

The first entry is called PDPSIM and is the PDP/11 Simulator program. It's written to load and run the binary output from the PDP/11 Cross Assembler (ASM11).

The second entry is called MACRO11. This is a combination PDP/11 Assembler and simulator in one package. MACRO11 assembles PDP/11 source programs and, if no assembly errors are generated, loads and runs the assembled program.

MACRO11 will be used initially Winter and Spring Quarters by Dr. G. M. Schneider of Computer Science. PDPSIM will serve those who wish to assemble and run their MACRO11 programs separately.

I have written PDPSIM and MACRO11. MACRO11 creates a procedure file which, in turn, executes ASM11 and PDPSIM. PDPSIM is currently in the process of extensive testing. I wish to have these Callprg entries installed by the beginning of Winter Quarter, in time for use by Computer Science Classes.

//////////

Randomly Selected Procedures - by S. E. Collins

The MERITSS Group and User Services would like to conduct a survey of users. Past surveys have been plagued with a lack of response, as they have been mailed to users. To gain a higher user response, the questionnaire will be a program on the system. Initially, the questionnaire was to be on account number LIBRARY, and users could get it and run it at will. However, this idea has two major flaws:

1. The data gathering method is not statistically valid. The sample population is choosing itself. To gather valid data, one must choose the sample (at random) and then get a response from every person in the sample.

2. Users would still have to be told how to get and run the survey, thus forcing them to read and learn something of marginal usefulness.

The solution to point two is to automatically run the questionnaire program for the user, so nothing has to be done by the user. To solve point one, users must be selected at random, and "forced" to at least begin running the questionnaire program.

I propose adding a feature to NOTICE/NOTIFY which will allow "random message blocks" to be created. Already, NOTICE/NOTIFY can be used to force a user to run a procedure file. By simply adding a "chance factor" to the definition of a message block, the procedure call at login could be forced only at random intervals.

I will add the RANDOM command, which will specify the proportion (of 1000 logins) for which a message block will be processed:

RANDOM,nnn.

will cause the message block to be processed in only nnn of each 1000 logins (chosen at random).

This proposal presents one problem; users will be unable to recover if they are chosen to run the questionnaire program. Of course, they could log in again, and hope they are not chosen twice in a row.

//////////

Don't Change That Channel/Don't Touch That Dial - by D. W. Mears

I propose to change the PP programs LKT and PDP to allow the TELEX PDP11's to be dumped, tested and loaded while 1TD still has the PDP11 channel reserved if all the TT equipments on the PDP channel are "OFF" in the EST.

The basic problem is that 1TD cannot properly drop and reserve the channel when equipments are turned off and on in the EST because there is not enough time and room in 1TD to handle the bookkeeping needed to correctly process channel reservations when several equipments share the same channel. This means that when 1TD determines that the PDP11 has crashed it turns the equipment off in the EST, but does not drop the channel.

The resulting problem is that the operators must either stop TELEX or manually drop the PDP11 channel reservation with a DSD DCH command if they need to reload the TELEX PDP11 during the day. Neither of these solutions are workable. Stopping TELEX unnecessarily interrupts users coming in through the 6676's or the 2551. The DCH command is bad because it is dangerous and confusing. The DCH command is different on each system, If a DCH is done on the wrong

channel, the system will usually hang when owning PP attempts to drop a channel which is no longer reserved. Most people confuse the DCH command with the DCN command, which does completely different things.

With my proposed changes installed, the procedures the operators use to reload the TELEX PDP11 after a crash will be the same procedures they use to load it at beginning of operations.

This code is already written and in the system on the KLUDGES file.

SYSTEM MAINTENANCE: People and Procedures

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

Our meeting was short and delt with only two items.

1. Tom Lanzatella's proposal to add an account file message to keep track of Cray job charges was approved (see DSN 6,20 p. 174).

2. Adrian Swanson's discussion topic suggesting a new CCL feature to control parameter substitution was considered at length. Although we thought the feature was useful, it doesn't provide any new capabilities. Furthermore, no one with the capability to install the change has time to work on it. We decided finally to submit the suggestion to CDC as a RSM.

//////////

Callprg and Library Tape Modifications - by M. Riviere

On December 20, as an end of the quarter modification Jim Mundstock will be replacing M77 and its library, M77LIB with future versions. The current version of M77 will then become past. This will be the last System modification that Jim will be making at UCC. Perhaps, for some people, this may seem the end of confusing days when "future replaces current and current becomes past, but suddenly, current has a bug and past disappears." But others like myself, who have been tracing these transitions for many years while working close to Jim, know that all the confusion stemmed from Jim's creativity and was the result of his continuous effort for improvement in work that undoubtedly produced great results. The end of confusing days may also be the end of many positive experiences.

Also on December 20, Michael Frisch will be modifying several of the System libraries. Part of this modification is the addition of a new routine, MINNLIB, which, if called within a FORTRAN program, sets LDSET instructions for the Loader to scan the corresponding I/O libraries (M77IOL for M77, FT4IOL for FTN, etc.). This modification is made according to a proposal which complements the library rearrangement (see Bob Williams' article in the July 8, 1980 issue of this DSN). Other modifications to Michael's libraries are: 1) in M77IOL, the addition of the routine GOTOER, and the updating of the PLOTPAC SYMBOL routine; 2) in MINNLIB, the replacement of DMXLNER and DMXLNEF with corrected versions; the addition of routines Q92CLIP

Q92COPE and Q92PUSH; a correction to FACTOR in the PLOTPAC section and replacement of CPU.CPM, CPU.CFM, CUP.OVL and CPU.PFM with the versions now used on SYSLIB.

On the same date, Steve Reisman will add a Callprg index entry for the AD2000 graphics package. This package will only be available in Cyber 172, and permitted only to a few users for the time being.

I will be on vacation until February 16. Please send requests for modifications to Tom Lanzatella in my absence. The scheduled dates for the modifications during that time will be:

Modification Date:	Submit Modification by:
January 13	January 1
January 27	January 15
February 17	February 6

//////////

Toward 1985: A White Paper Discussing the Formulation of a Comprehensive Communications Plan with Emphasis on Objectives, Processes, and Personnel Involved in Identification of Viable Alternatives, Evaluation Criteria, and Decision Making Methodology

-or-

Ask Not from Whom Bell Extracts Tolls... - by R. A. Williams

Communications is one of the "hottest" fields around right now, it seems. Consultants are springing up all over the place offering seminars on the topic and advice on tailor making a plan for your organization. Like predecessor "fads" (e.g. database management, hula hoops), communications has a cadre of buzzwords that make experts in the area seem even more so. I'm sure you've heard such terms as network, contention, datagram, and uplink creep into everyday conversation, confirming that communications has arrived as a science.

Of course there are several good reasons why communications issues have taken the spotlight of late. Technology and economics stand out as the basic incentives for most of the growing interest in communications. In this respect, UCC is no exception.

Communications is an essential and growing part of our operations. With campuses around the state and most mainframe equipment separated from its users by some distance, we depend on a network of connections between these points to operate. Further, rising costs and increasingly varied and powerful terminal and computer equipment, lead to a dramatic rise in the impact communications decisions have on our organization's future.

Here we plan to examine a portion of the UCC communications problem; that associated with the MERITSS instructional labs. The current state of affairs, future options and steps being taken to make the transition successful will be discussed.

This paper is presented not only to keep UCC staff aware of the activities of the participants in this particular project but to stimulate thought and discussion about the issues involved and the possible spinoff solutions that may be spawned to tackle other UCC communications problems.

Earlier we indicated that the two foremost issues facing communications planners are economics and technology.

One of our biggest economic challenges lies in the fact that most of the equipment we use to move data is leased from Bell Telephone. This includes not only the lines but modems to convert data between digital (computer equipment) and analog (telephone equipment) signals. These charges are assessed monthly, can increase, and have been known to do so. Further, there are indications that the computation of charges in the future will be based on use rather than by line. Thus, heavily used phone numbers, like those that move data, will cost far more than lightly used circuits, such as those just for voice conversations.

The technology issue has many facets but probably the most important to us is the rapidly changing nature of mainframe and terminal equipment. The watchword here is growth, in speed, capability, type and multiplicity of locations. To keep pace with this explosion, more powerful communications equipment will be needed.

Taken at face value, the objectives of any planning with regard to communications should be reduction, or modest increases in costs, while improving flexibility and capability. That sounds good but when the various tradeoffs inherent in almost any cost reduction-service increase equation are coupled with the technical intricacies of this problem (e.g. maintenance, reliability, security), many of which lurk in the background until actual implementation, one realizes the scale of the task.

To attack these issues as they relate to the MERITSS instructional labs, a committee was formed by Mike Skow with Ed Edmundson, Don Mears, Pete Zechmeister, and Bob Williams as members. This paper presents a quick (depending on your reading speed) synopsis of their findings so far and future plans.

A short outline of the MERITSS communications environment is in order. While it comprises a variety of configurations, the bulk of the demand exists in about a dozen clusters of from a few to thirty terminals each (hereafter known by the title of "instructional labs"). The majority of these labs are located on the Twin Cities campuses with one at Morris and one at Duluth. In addition to the labs, a block of dial-up access to the system is provided. Currently, many lab terminals are serviced by a dedicated "private" telehone line between the computer and the terminal with a modem located on each end. The lines and modems are rented from Bell.

A logical step to reduce costs is to piggyback an entire lab's terminals on one set of modems and line. In fact this has been done for many years in the case of the Morris and Duluth labs and was recently installed at

four large labs on the Minneapolis campus. This is a good example of how the obvious solution may not be totally cost-effective, however. The equipment to multiplex (piggyback) several terminals on one line is costly and the modems to push this increased amount of data through the line are more expensive than the single terminal kind. Given the dynamic (i.e. changeable) nature of the local labs and increased impact of service interruptions caused by failure of the line or multiplex/modem equipment. The obvious solution may not be the prudent choice.

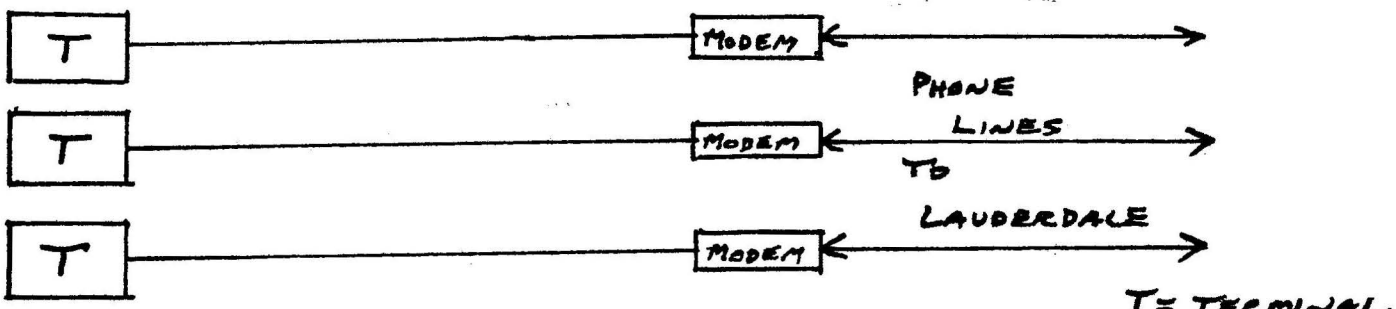

FIGURE 1

Time Division Multiplexing

The multiplexing scheme which we currently use is termed the "Timeplex" approach by those in the know. The trade name is derived from the time division method of multiplexing in which each terminal has a certain amount of space reserved on this single phone line for its data. Thus, if ten 300 baud terminals are multiplexed by the time division technique, the line must transfer data at 3000 baud (while not totally accurate, this explanation will suffice for the purposes of this discussion). I apologize to those of you I spoke with before learning of the derivation of Timeplex. I was under the impression that it was a multiple dwelling rental unit in an old Twilight Zone episode.

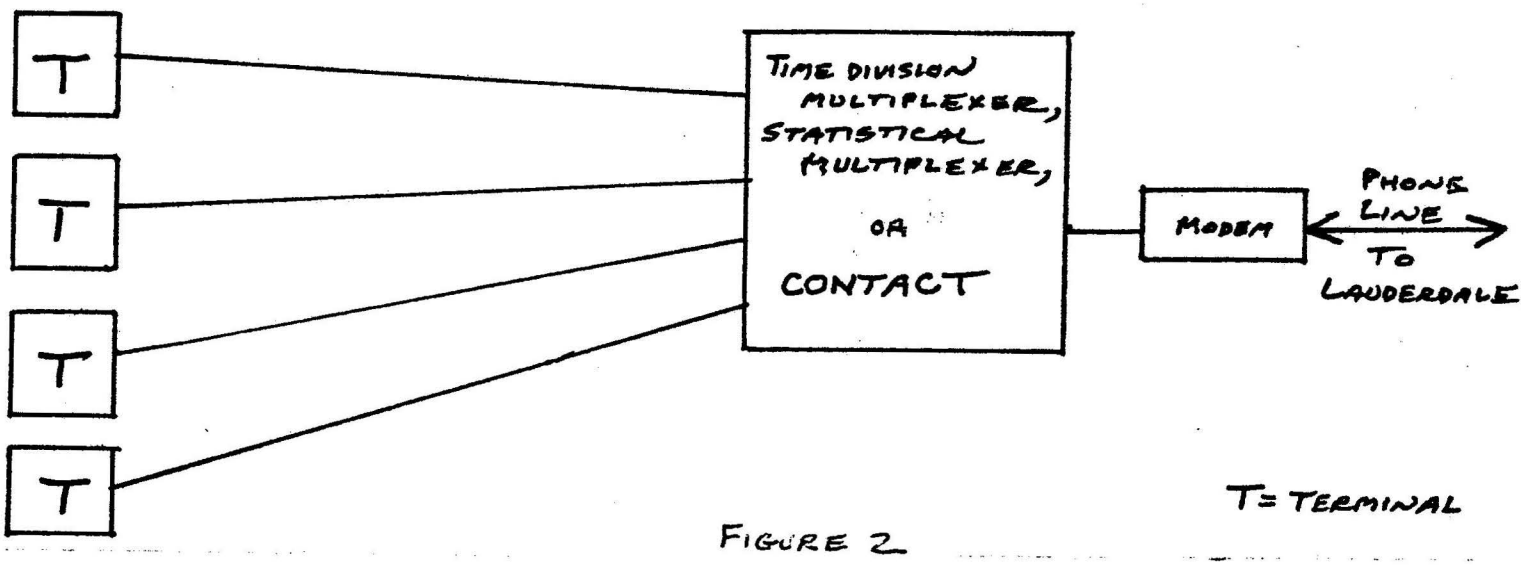

FIGURE 2

NIB

While on the subject of project names, I must indicate that NIB isn't a new snack cracker either, though it does involve chips. Tom Jacobson and Pete Zechmeister have devised a communications plan for the labs that is close to the leading edge of commercial communications technology. One Network Interface Box attaches to each terminal in a lab and a single cable connects all NIB's together. Note how this contrasts with other plans, which need one cable to each terminal from a central location in the lab. Data is passed from the computer system to the lab in the form of packets, identified by their terminal destination address. Each terminal NIB, with a unique address, grabs only that data which is destined for its terminal. The process works in reverse for data from the user.

The advantages of the NIB are almost all rooted in flexibility. A myraid of terminal devices may be attached to the NIB and network reconfiguration is easy. This is due to the simplicity of the scheme (e.g. only one cable is needed to wire all terminals together) and its generality (e.g. each NIB is programmed to "talk" to its host device, whatever that may be). Each NIB is also relatively inexpensive to build (around $250). Like the other proposals we will discuss, the NIB's, in contrast to the time division multiplexers, make efficient use of the baud rate provided by the modem and phone line (i.e. if no data is to be sent, "space" is not reserved for it) which is crucial, since the cost of modems is usually tied directly to their data rate.

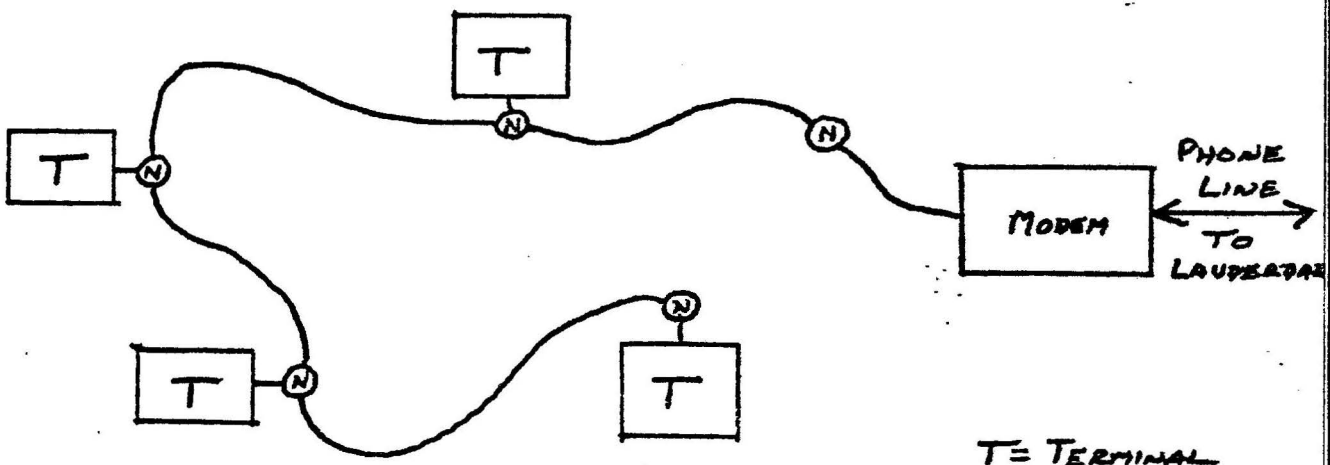

FIGURE 3

MG

The MG communications scenario advanced by Pete Zechmeister does not, contrary to popular belief have Pete motoring from lab to lab in a British sports car (purchased for him by UCC) with the top down and a stack of data at his side. It calls, instead, for construction of a single board micro-processor Modem Gateway device. Like the NIB, this stored program unit

is designed to be adaptable to different terminal characteristics. Each MG is able to handle up to 32 terminals and does not require a pair of modems, despite its ability to reach an astounding 880K baud. (If Pete can pull this off he probably doesn't need a car to get around!) It is also cheap to construct (around $400).

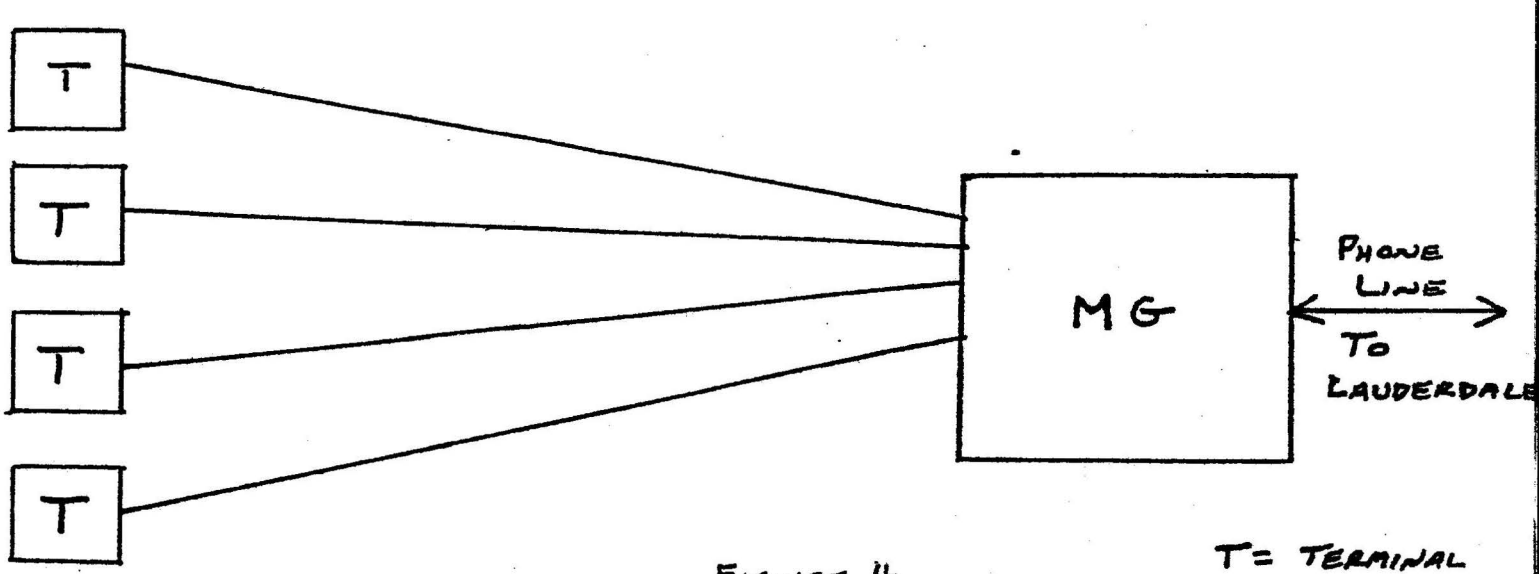

FIGURE 4

Contact

Another concentration proposal which, like the MG, sends the signals from many terminals over one telephone line, is CONTACT (CONcentrated Terminal And Communications Teleprocessing), advanced by Ed Edmundson and Pat Snyder of UCC's Engineering Services. CONTACT is built around a standard LSI-11 (the heart of a Terak microcomputer) microprocessor with standard Digital Equipment Corporation peripheral devices to interface terminals. It, too, has a stored program. The strongest points of this plan lie in its ease of maintenance, relying on proven, "off the shelf" components (i.e. units are constructed from a small number of commercially available devices rather than being designed at the chip level). This means that elements of the system are well documented and that service of the equipment may be performed by existing Engineering Services Technicians.

Statistical Multiplexing

Yet another option is statistical multiplexers. These devices operate like many of the concentration plans covered to date (and judging by the length of this paper the date has probably changed while you are reading it) in that, unlike time division multiplexing, they take into account the fact that all terminals won't operate at their rated speeds at all times (e.g. some will be inputting, others outputting, still others doing nothing at a given instant). Clearly, other non time division concentration plans depend on this fact as well (e.g. NIB, MG, CONTACT) but statistical

multiplexing, as used in this context, refers to units that are totally "off the shelf", requiring no assembly or programming (i.e. batteries are included). The State University System uses this approach to communicate with MERITTS. Their equipment is manufactured by Codex (sounds like a Star Wars character, doesn't it?)

Purchased Bell Compatible Modems

Finally, we could remain archaic (but happy?) by staying with a single line per terminal configuration. Costs could be held at bay by purchasing the modems used on the lines. Since Bell's monthly modem lease fee is the bulk of our costs in most cases, this move could prove fruitful in the long run.

All of these ideas have merit. Of course there are drawbacks. Statistical multiplexers and CONTACT have high initial price tags but, when maintenance headaches are factored in, may prove to be more viable in the long run than MG or NIB. What of Timeplex? Is it really over the hill or can it be helpful in solving UCC's communications problems?

To get the answers to these and other questions we have decided to study all of the options further (a typical committee action). Money has been committed to make prototypes for all of these plans. Pete Zechmeister is building on MG and writing software for it in the MODULA language. Several NIB's have been constructed and Bill Sackett is busy writing code to drive them. Bill Wells is engaged in the design and coding of software for CONTACT, to run in a prototype that Ed Edmundson and Pat Snyder will put together, using a Terak as the processor. Meanwhile, Brad Blasing is writing code so the MERITSS front-end can handle all of the systems. Timeplexers are being used for some lab ports, as we mentioned earlier, providing thousands of dollars a year in savings already and we use Penril-brand modems for 1200 baud service on all UCC systems, avoiding costly rental fees. Engineering Service's, Doug Parkes is researching other options in the purchased modem area as well.

Our plan is to spend about $1500 for materials to construct these prototypes. Then, by early next year (1981), we should be able to test the plans, first in UCC terminal rooms and then in actual lab environments. This will provide needed data on the issues of reliability, maintainability, and security to add to already gathered information on initial purchase cost and flexibility.

Hopefully, by the fall of 1982, CONTACT, MG, NIB, statistical multiplexers, Timeplexers, owned modems, or some combination plan will be providing increased communications service to MERITSS users at a substantial long term savings per terminal.

Disclaimer: I have tried to accurately present the essence of our group's work to date in this area but since it is such a complex topic and I don't profess to be an expert on all of these plans (in fact on any of them) some half truths or outright lies may have crept in. Nevertheless, if I have conveyed to you the flavor of this project and stimulated thought about it, the paper was a success.

//////////

REDACT, an Interactive Program for Writeup's Upkeeping - by M. Riviere

Since some time ago, there have been several demands for the creation of a utility that could accomplish and/or simplify all the tasks of writeup upkeeping, (e.g. edit text, create-modify-purge files in the Writeup account number, update the Writeup index, maintain statistics on writeup usage, handle file ownership validation, produce writeup's lists, etc.)

L. Fetcher, R. Franta and D. Larson tried to produce such a utility. The result was a memo issued on February 4, 1980 (Subject: The Revamp of Writeup) where they stated all the features and guide lines for the utopic routine, together with an opinion about the project being too complicated for them to accomplish and being more suitable as a System's group project than one of their own. REDACT is my partial answer for the requested utility and it includes several of the featues suggested in that memo.

REDACT replaces RFM and Upwrite for writeup files maintenance and index redacting (editing). By merging both functions, REDACT consolidates validation requirements on the Writeup validation file, straightens the file owner-index entry relationship and makes file modifications to be simultaneously reflected in the Writeup index. In addition, REDACT produces catalog list of WRITEUP files sorted by RFM's ownership criteria. When running in information mode REDACT gives explanations about the Writeup program, the Writeup index, the index parameters, and whatever else that it occurred to me could be useful to know for any one creating and maintaining machine retrivable documentation in our installation.

It is important to mention, however, that REDACT does not fill all the requirements of the Fletcher-Franta-Larson suggested utility. Some other of the answers to all the "super-utility" features are already in or could be added to the current Writeup program and all the UCC available text editors. Some of those requirements were almost impossible to implement in only one program. Contrary to what the memo suggests, REDACT does not create texts or edit them; this is a task to be accomplished by specific text editors. REDACT does not generate writeups' lists; the Writeup program is quite suitable for that (*). REDACT does not control the internal format of a document. It is up to the document's author to design the required format, although REDACT can give guidelines about how a document should be and show how to create and maintain it. (This includes descriptions about the handling of indexed writeups and a detailed explanation of the writeups' attributes that are represented by the index's parameters.)

REDACT is now available for test purposes. Type "REDACT" (on the Cybers 172/74 only, for the time being) and let it show you, through an interactive session, about its tasks. You can use REDACT now to modify files in the Writeup account number. The file modifications will be done as if they were done by RFM. The Writeup index that REDACT is modifying for the time being, however, is not the real Writeup index but file INDXNW in UN=WRITEUP, PN=SPL. That is, any modification to the test index file made though REDACT will not show up when the Writeup index is updated.

The information file of REDACT is still in the process of being cleaned up (mainly for linguistic errors for which I apologize) but possible to be understood.

Please give some trys to REDACT. I would like to have some tests done and some feed-back about its design before I consider installing it as the official Writeup maintainer utility.

(*) The WRITEUP program is now in the process of being split away from Callprg by Yvonne Murray. Once a program in itself, Writeup will be a much easier package to have modifications implemented on it.

//////////

Cyber 720 Deadstart Dump Analysis (11/16-12/8) - B. E. Blasing

Wed.	11/19	23:15	The scopes blanked. Analysis showed the recurring mode 40 problem struck again. The SCR also showed a CM double bit read error. The dump was given to the CE's. A level-3 recovery was successful.	DD13
Fri.	12/5	0740	The scopes blanked. Subsequent attempts at deadstart failed. The mainframe then disgorged a dazed and dumbfounded CE. A level-0 deadstart was then successful.	

//////////

Cyber 74/172 Deadstart Dump Analysis from Tuesday, 19 November through Thursday, 4 December - by J. J. Drummond

Monday, 24 November — Cyber 172

14:05 — DD2012

An enterprising member of the systems group was finally able to reproduce an elusive checkpoint/restart error. A level three recovery was successful.

Wednesday, 26 November — Cyber 74

22:29 — DD2013

The system hung-up with several disk errors. A level three recovery was successful. Problems with a disk controller were also noted the following day during system's time. The CE's eventually worked on the problem.

Thursday, 4 December — Cyber 74

21:49 — DD2014

LFM hung dropping a tape equipment during a tape assignment. A level three recovery was successful. It is not clear how LFM could have thought that it had the equipment assigned to the control point when the equipment was unallocated. More investigation is required.