Feb 85

TSX—PLUS USER'S GROUP NEWSLETTER

Newsletter

TSXPUG
c/o Jack Peterson
Horizon Data Systems
1901 Wildflower Terr.
Richmond, VA 23233

Administration

TSXPUG
c/o Glenn Burnett
Zia Corporation
P.O. Box 351
Morris Plains, NJ 07950

BOILERPLATE!

Welcome to this, the fifth issue of our newsletter published by the TSX-Plus Users Group (TSXPUG), a volunteer organization dedicated to the exchange of information among users of TSX-Plus. TSXPUG is independent of and receives no financial support from S & H; expenses are paid from membership dues. The newsletter is prepared four times a year (February, May, August, and November) and mailed the following month.

Jack and Glenn, whose addresses are above, run TSXPUG. Jack writes and edits the newsletter; Glenn handles administration, collecting dues, maintaining the membership list, and mailing the newsletter. If you have a contribution for the newsletter, please write to Jack; direct other correspondence to Glenn.

Brief newsletter stuff may be submitted in hardcopy form, but we prefer lengthy articles on magnetic media. We can read RX01, RX02, and 1600 bpi magnetic tape; in a pinch, RX50 and RL02; in a real pinch, RL01. We will return media to you if you ask us to do so. Thanks for your cooperation and your contributions.

IMPORTANT NOTICE!

As you can see by the above, we have decided to copyright each issue of our newsletter. We have become aware of a few instances in the past where material published herein has been copied without acknowledgement and used for commercial advantage. This is unfair to the volunteers who provide material for the newsletter, and unfair to our members who pay good money to receive it. Feel free to use whatever information we provide to the greatest extent possible, but if you steal it and distribute it to others, you are breaking a law which imposes severe civil and/or criminal penalties.

MEMBERSHIP RENEWAL INFORMATION

All of you should have received a membership renewal notice. If you have [illegible], please notify Glenn immediately. Because this was the first time we went through the renewal process, we were a little late getting the notices out, so everyone is receiving a copy of this issue. The remaining three issues this year will go only to those who renew for 1985. We must receive your renewal before May 1 if you expect to receive the next issue.

USA and Canadian memberships are US$15.00 per year, US$20.00 per year in all other countries. We request foreign subscribers to remit in US$ with an instrument (such as an international money order) payable at a bank in the United States to avoid delays and service charges.

CONSULTANT HOTLINE

Two more firms have told us they provide TSX-Plus consulting services, bringing the total to five. The latest entries are:

Peter Heinicke
Precision Computer Methods, Inc.
No. 3 2S779 Winchester Circle
Warrenville, IL 60555
(312) 393-6478

Milton D. Campbell
Talisman Systems
Drawer CP-255
Manhattan Beach, CA 90266
(213) 318-2206

If there are more of you out there, please let us know. We will publish a summary of all known TSX-Plus consultants in the next issue.

DECUS NEWS

The Fall 1984 DECUS Symposium was held at the Anaheim Convention Center in December. Several items of interest to TSX-Plus users emerged from this meeting. With the help of Ned Rhodes, we were able to schedule a TSXPUG Birds of a Feather (BOF) session and a TSX-Plus Magic panel discussion. Both were well attended and well received; they will be featured again in New Orleans.

DEC announced RT-11 Version 5.2 (scheduled for April or May) and BASIC Version 3.0 (scheduled for June). The former appears to be of little interest to TSX-Plus users, but the new BASIC version, a superset of BASIC-Plus 2, has been long awaited. BASIC V3.0 will be distributed as a new release of BASIC-11 rather than as a new product, so if you have BASIC support, you will receive this new version automatically when it is released.

The Spring 1985 DECUS Symposium will be held in New Orleans, May 27 - 31. Several presentations related to TSX-Plus are planned. Two pre-Symposium seminars will be offered, one on RT-11 internals, one on IND. Seminars are all-day affairs held the Sunday before Symposia open (May 26 in this case), cost extra bucks beyond the Symposium tab, and are conducted only if enough people sign up. If you are interested in either of these, check your Symposium registration packet for the application.

There are many RT-11 presentations scheduled for New Orleans, at least 7 of which are of special interest to TSX-Plus users. Tom Shinal of General Scientific will present "The RT-11/TSX Office", a reprise of a good presentation given in Anaheim, on Wednesday from 1:30pm to 2:30pm.

Jack Peterson will chair two audience participation sessions. The first, Tuesday from 5:30pm to 7:00pm, is "TSX-Plus Magic", which explores problems, solutions, tricks, and workarounds for TSX-Plus. The second, Wednesday from 3:30pm to 4:45pm, is new, a "Third-Party RT-11 Application Software Forum" which permits users (not salesmen) to recount their experiences, good or bad, with third-party RT-11/TSX-Plus software. If you have such experiences you wish to share, or if you are considering purchasing a software package, come to this session and participate and/or learn from others.

Bob Manfredi of the University of Connecticut Health Center will present two very interesting papers. On Monday from 9:00pm to 10:30pm, Bob will discuss "TSX-Plus In The Laboratory", showing application of several TSX-Plus special features including shared run-time systems, PLAS support, job monitoring, record locking, and user interrupt processing. On Tuesday, from 4:30pm to 5:30pm, he will present a "User Written Keyboard Monitor (KMON) for TSX-Plus" which provides many interesting features including I/O redirection, pipes, User Command Linkage (UCL), and other goodies.

On Monday, from 8:30pm to 9:30pm, James Phelps and Harvey Cohen will discuss "TSX/FORTRAN Real-Time", a set of real-time routines they have developed for use under TSX. Last, and certainly not least, on Tuesday, from 3:30pm to 4:30pm, Phil Sherrod of S & H will present "TSX-Plus Scheduling Internals and System Tuning", a paper of great interest to system managers.

Concurrently with DECUS, New Orleans will host the DEXPO trade show which affords an excellent opportunity to see, hear, smell, and touch lots of DEC compatible hardware and software. DEXPO will be at the New Orleans Superdome Tuesday through Friday. The show's sponsors promise about 300 exhibitors who will collectively display 500 new hardware and software products. DEXPO is open to all DECUS attendees at no additional cost, but you must register separately for this event, which is not affiliated in any way with DEC or DECUS. We suggest you use the preregistration forms found in DEXPO flyers and ads to avoid long lines at the show.

All in all, New Orleans looks like a good bet in a great town! We hope to see you there.

RT-11 SYMPOSIUM TAPE TREE

Speaking of DECUS, Tom Shinal, who distributes RT-11 Symposia tapes, has created a new tape distribution tree. According to his scheme, Ralston Barnard creates the master tape, usually within two weeks after Symposium, and provides a copy to Tom, who enters it into his distribution tree. The tree has 43 nodes and a depth of 5, so within a few more weeks, all nodes should have a copy of the tape for distribution to end users.

This tree is run by the RT-11 SIG rather than the National LUG Organization, who have an independent tree. If you belong to a LUG, please get your copy from it. Otherwise, follow two simple steps. First, from the list below, select the node WHOSE ZIP CODE IS CLOSEST TO YOUR OWN. Then, forward a 2400-ft mag tape reel, appropriate postage and return addressing information, and a clear statement of what you want (Spring 1985 RT-11 Symposium Tape, for example). Sit back and wait for your copy to arrive. These people do this work as uncompensated volunteers, so please follow the rules to make their job as easy as possible.

The following is a list of all nodes in the tree, in zip code order. PLEASE SAVE THIS LIST. WE WILL NOT REPEAT IT, EXCEPT FOR CHANGES, THIS YEAR.

Robert Schor
Rockefeller University
1230 York Avenue
New York NY 10021
212-570-8600

Paul A. Kessler
Brookhaven National Lab
AMD Building 515
Upton NY 11973
516-282-4156

Dennis P. Costello
Cornell University
National Submicron Lab
602 Knight Lab
Ithaca NY 14853
607-256-2329

John F. Stitzinger
HRB Singer
Science Park Road
D/115
P.O. Box 60
State College PA 16801
814-238-4311 X2853

Thomas J. Shinal
General Scientific Corp.
1684 East Gude Drive
Rockville MD 20850
301-340-2772

Robert E. Langford
Medical College of Va.
Room B3-016
1101 E. Marshall Street
Richmond VA 23298
804-786-9843

R.H. Klausewitz
West Virginia Wesleyan
Computer Center
Buckhannon WV 26201
304-472-9520

Janet Anderson
Rockwell International
Suite 200
800 Eastowne Drive
Chapel Hill NC 27514
914-968-4836

Herbert Spies
AD/KRESS
Building 380
Eglin AFB FL 32542
904-882-5498

Michael Newell
Florida Inst of Technology
150 W. University Blvd.
Melbourne FL 32901
305-768-8101

Mike Minton
NOAA
Atl Ocean & Metro Lab
4301 Rickenbacker Causeway
Miami FL 33149
305-361-4356

Amanda F. Stokes
Teledyne Brown Engineering
Cummings Research Park
Huntsville AL 35807
205-532-1842

Dan Tipton
ORFMA
117 Flint Road
Oak Ridge TN 37830
615-482-5000

Jerry W. Ethington
Federal Land Bank
201 W. Main Street
P.O. Box 32390
Louisville KY 40232
502-566-7304

Daryl George
Ohio Wesleyan University
Dept. of Math Sciences
Delaware OH 43015
614-369-4431 X900

Laurence G. Owens
Owens Computer Services
2813 Indianola Avenue
Columbia OH 43202
614-262-4805

Rick Sharpe
Toledo Edison
M/S 3085
300 Madison
Toledo OH 43652
415-259-3000 X470

Tom Haase
SOHIO
4440 Wanersville Ctr
Cleveland OH 44128
216-581-5724

Marion M. Fauber
M/S 4-07
P.O. Box 1042
Dayton OH 45401
513-258-7707

Kreigh Tomaszewski
Amway Corporation
21A-2D
7575 East Fulton Road
Ada MI 49355
616-676-7402

Jim Flatten
Ames Laboratory
304 Metallurgy
Ames IA 50011
515-294-4823

Ed McGuire
Grinnell College
Computer Services
Grinnell IA 50112
515-236-2570

Mark Buda
12000 W. Park Place
Milwaukee WI 53224
414-359-9800

John Vilandre
University of Minn.
Div. of Epidemiology
Stadium Gate 27
Minneapolis MN 55455
612-376-4056

Jim Leedham
Mayo Foundation
200 1st Street SW
Rochester MN 55905
507-284-2620

Dale J. Travis
Argonne National Lab
D817
9700 S. Cass Avenue
Argonne IL 60439
312-972-6964

Frank Talaga
Univ of Illinois
College of Dentristy
M/C 621 - Room 504
801 S. Pailina
Chicago IL 60612
312-996-7520

Richard Wrenn
Washington University
Dept. of Chemistry
Campus Box 8094
St. Louis MO 63110
314-362-3354

Robert Fairchild
Nebraska Wesleyan Univ.
50th & St. Paul
Lincoln NE 68504
402-466-2371

Guy Dunbar
Wm. E. Davis & Sons, Inc.
37 NW 122nd Street
Oklahoma City OK 73114
405-751-4660

Bruce D. Sidlinger
ALCOR Inc.
10130 Jones Maltsberger Rd.
P.O. Box 32516
San Antonio TX 78284
512-349-3771

David Taft
USGS
Box 25046 M/S 978
Denver Federal Center
Denver CO 80225
303-236-1639

Ralston W. Barnard
Sandia National Labs
Nuclear Physics 7523
Albuquerque NM 87185
505-844-5115

Nick Bourgeois
NAB Software Services
3236 Candlelight NE
P.O. Box 20009
Albuquerque NM 87154
505-298-2346

James F. Harrison
Los Alamos National Lab
MP-1 M/S H810
Los Alamos, NM 87545
505-667-5688

Doug Gladden
White Sands Missile Range
NR-AD-A
WSMR NM 88002
505-678-3348

Clifford P. Lloyd
M/S C-81
P.O. Box 1912
Las Vegas NV 89125
702-295-0392

Dr. Jeffrey P. Lieb
REED NRC
UCLA
Los Angeles CA 90024
213-825-6713

Jim Hamm
Boschert Inc.
384 Santa Trinita
Sunnyvale CA 94088
408-732-2440

Kevin Douglas Walker
Unitek Electronics Inc.
Suite 222
2850 Paa Street
Honolulu HI 96819
808-836-0877

Robert Perry
Tektronix Inc.
M/S 19-333
Box 500
Beaverton OR 97077
503-627-5410

Ron Tenison
The Catlin Gabel School
8825 SW Barnes Road
Portland OR 97225
503-297-1894

Steve Loretzen
UNICOMP Corporation
500 Fourth & Battery Bldg
Seattle WA 98121
206-223-6453

FORKIN´ (The Bug Killer)

Stan Montz of Murry Bergtraum High School in New York had a problem using a TI-855 serial printer under the LS handler mentioned in the last issue. He has subsequently written to tell us he found the solution. He had inadvertently genned the serial printer port as a time-sharing line also, which causes conflicts over who owns the interrupt vectors.

If you want to use a serial port sometimes or a printer and sometimes for a time-sharing line, gen the port as a time sharing line and allocate at least one extra (unattached) CL unit by setting TSGEN parameter CLXTRA to the number of extra units desired. The port normally functions as a time-sharing line. To seize the line for use as a printer, use the SET CLx LINE=nn command (see TSX-Plus Reference Manual for details).

Also mentioned in the last issue was a problem Bill Leroy had using an 18-bit third-party tape drive (MT handler) in a 22-bit system. Bill specified MAPIO for the handler, but jobs running above 256K frequently crashed when accessing tape. Locking tape jobs into low memory solved the problem. Since this item was published, several readers have indicated similar problems with other foreign 18-bit peripherals.

This problem may be due to strange implementations of the QBUS protocol by some third-party devices. When an I/O page reference is made by a bus master device, it is supposed to assert signal BBS7 L. We think some third-party devices may assert this signal whenever the three highest bits of a (18-bit) DMA address are all ones, causing a reference to the I/O page rather than to memory.

I/O requests made to a MAPIO device when the user buffer is located totally within the first 256K bytes of memory are processed normally by TSX-Plus, i.e., internal mapping buffers are not used. If, however, a buffer extends beyond 248K, some third-party devices may assert BBS7 L during DMA operations. The technical support group at S & H is addressing this issue; perhaps we will have a fix in the next newsletter.

HINTS AND KINKS

by

Ned W. Rhodes
Software Systems Group
1684 East Gude Drive
Rockville, MD. 20850
(301) 340-2773

Welcome to what may become a regular column in the TSXPUG Newsletter. Both Tom Shinal (General Scientific) and I will be contributing items to the column. Two things motivated me to start this column. The first was that, as a consultant, I am called in to help with some unusual TSX problems. As a result, I see some things I cannot explain easily. So, in this column, I hope to share these unusual events with you; perhaps some of you have experienced the same problems and/or have developed fixes or workarounds. When I have a workaround, I will publish it here. If you have a workaround to an unusual problem, please write or call me, and I will pass it along in the next column.

The second reason for this column? I can´t stand to see Jack have all the fun doing this newsletter.

DY Problems

Our first problem concerns floppy disks. We have experienced frequent "Trap to 4" errors when copying large files to our floppies under heavy system load. We use third-party floppy disk systems with 18-bit controllers, and have tried both MAPIO and our own data forwarding handlers to enable operation in 22-bit systems, but the errors persist. By trial and error, we learned that using the /SLOWLY option on our COPY commands, or locking PIP, DUP, and DIR in low memory, eliminates the problem.

There are two ways to lock programs into low memory. You can modify the program itself to issue the "lock into low memory" TSX-Plus EMT, or you can edit TSGEN.MAC to add a program name to the list of programs requiring special attention when run by TSKMON. This list is located at label SRFPRG outside the section of TSGEN.MAC where user changes are normally made. To lock PIP, DUP, and DIR in low memory, add each to this list of programs, specifying the AF$MEM flag.

At this time, we do not know the cause of our problem, but as the FORKIN´ column above points out, this may be caused by improper device operation in the 248K to 256K address range. We, Jack, and S & H are actively pursuing this possibility, and will let you know what we discover in the next newsletter.

MT/MS Problems

We have two problems with our magnetic tape drives, also third-party 18-bit devices. The first problem, similar to that above, is copy failure under heavy system load. We have also noted the INITIALIZE command sometimes moves tape and appears to work properly, but the tape remains uninitialized. This problem also disappears by locking DUP in low memory. Again, we are not sure why this happens, but we suspect it is related to the problem above. I would be interested in hearing from you if you are experiencing similar problems.

CL Problems

Our system is configured with two dial-in lines and three printer ports, all driven by the CL handler. We can hang our system by printing on one CL port while running VTCOM on another. Running three printers simultanously without VTCOM works fine, as does any combination if VTCOM is not very busy (transfering a file, for example). Has anyone else experienced this problem?

[Editor´s Note: Apparently, some other people have. This problem has been recognized by S & H and is corrected in Version 5.1C, now in field test.]

The Lost System

The Lost System is our final problem for this issue. You encounter this problem when you run a program that normally works but it now traps, or you copy a file to another device but the copy is now garbage. We suspected a directory caching bug, so S & H supplied a feature patch to disable automatic directory caching of SY and to allow SY to be DISMOUNTed. With these patches installed, our systems don´t get lost anymore.

Rather than publish the patches here, we will continue to look at the problem. If anyone has seen this problem, please get in touch with me so we can compare notes.

And Finally

I hope there was something of interest for you in this column. Feel free to contact me with unusual TSX-Plus problems that may be of interest to others. For routine problems, contact your distributor or hire one of the consultants mentioned in the CONSULTANT HOTLINE (like me).

LOGICAL DISKS - A TUTORIAL

Readers have occasionally requested that we include brief tutorials on certain facets of TSX-Plus operation in this newsletter. Most recently, several people have requested information on logical disks and the LD handler. As an experiment, we have prepared an introduction to this subject. If you would like to see other such columns as a regular newsletter feature, let us know what topics you would like discussed. Better yet, offer to write one for us. We may consider paying a small honorarium ($25.00 to $50.00) for each one published.

Logical disks (also called virtual disks and disk subsets) are simply disk files which, through LD handler magic, take on most of the attributes of physical (real) disks. Under RT-11, the LD handler is an external file (LD.SYS or LDX.SYS), and mounted logical devices are common to all running jobs.

Under TSX-Plus, LD is implemented as an internal system virtual overlay enabled by setting TSGEN parameter LDSYS to 1; there is no LD.TSX handler file. Each TSX-Plus job (time-sharing line, virtual line, or detached job) has its own set of 8 logical disk units which may be mounted without affecting other running jobs.

Logical disks have many advantages and applications. They are frequently used to divide large disks, like RK07s, into smaller, more easily managed chunks. Because logical disks have independent directories, they permit a large device to hold more files than the the maximum of 2322 permitted in a single directory. Independent directories also eliminate file name conflicts. By granting each user (or group of users) their own logical disk(s), you can eliminate inadvertent deletion of other users´ files, and each user can assume responsiblity for managing his own disk space.

To create a logical disk, determine the number of blocks you wish the "disk" to have; create a file of that size on some convenient physical disk (logical disk files can reside within logical disk files, but this seldom used feature is often more confusing than helpful); use a MOUNT command to tell the operating system you wish to treat this file as if it were a disk; initialize the associated logical disk unit.

For example, to create a 2000 block logical disk called MYDISK.DSK (DSK is the default extension for LD) on DM2, use the following commands:

```
CREATE DM2:MYDISK.DSK/ALLOC:2000
MOUNT LDO DM2:MYDISK
INIT/NOQ LDO:
```

If you examine the directory of LDO, you will see that it has 1986 free blocks. Where did the missing 14 blocks go? A logical disk has the same characteristics as a physical disk. The first 6 blocks of all disks are assigned as 1 primary boot block, 1 home block, and 4 secondary boot blocks. Directory segments, each two blocks in length, begin in block 6, so our logical disk must have four directory segments (8 blocks), which accounts for the "missing" 14 blocks.

Why are there four directory segments? Because DUP, when initializing a disk in the absence of explicit instructions to the contrary, creates a reasonable number of directory segments based on the device size. If, in the example above, you created a 5000-block logical disk file, DUP would create 16 directory segments; for a 100-block disk, 1 segment. If you do not like DUP´s choice, you can assign from 1 to 31 segments during initialization by using the /SEG:nn option.

Suppose you want a logical disk with 2500 blocks of file space and 10 directory segments. You must create a file with 2526 blocks (2500 blocks of file space, 6 blocks of device overhead, and 20 directory blocks), so use the following commands:

```
CREATE DM2:MYDISK.DSK/ALLOC:2526
MOUNT LDO DM2:MYDISK
INIT/NOQ LDO:/SEG:10
```

Now that you have a logical disk ready to go, what can you do with it? Just about anything you can do with a physical disk except boot it. You can initialize it; squeeze it; open, create, and delete files on it; scan it for bad blocks; and assign logical names to it. There are two ways to do the last, in the MOUNT command or by using the ASSIGN command. For example, the commands:

```
MOUNT LDO DM2:MYDISK
ASSIGN LDO DK
```

are equivalent to the single command

```
   MOUNT LD0 DM2:MYDISK DK
```

If several logical names are to be assigned to a logical disk, you may assign the first with the MOUNT command and the others with ASSIGN commands, or you may do them all with ASSIGN commands.

You can have as many logical disk files scattered throughout your system as you like, but you can access no more that 8 of these at any given time, as LD0, LD1, LD3, LD4, LD5, LD6, and LD7. If you can't remember which LD unit is assigned to which file, use the SHOW SUBSET command to display each mounted unit, the file to which it is assigned, and the size of the file. You can release a logical disk file assignment using the DISMOUNT command, or you can change the file assigned to a given LD unit by issuing a replacement MOUNT command without an intervening DISMOUNT for that unit.

Logical disks can be used in many ways to help organize your files more effectively. Some sites, for example, assign each user a logical disk file for work in progress. Others assign logical disks on a functional basis. At Horizon, we use a combination of approaches. We maintain one logical disk for each development project or contract, one for bookkeeping and accounting, one for completed documentation, one for common libraries, one for electronic mail, and one for each user as a scratch area. Logical disks are MOUNTED and ASSIGNED as part of each user's start-up command file.

Universities and similar organizations with a large user body and a need to maintain fairly high levels of data security, can assign each user a small, private logical disk. By mounting this logical disk in the user's start-up command file and by using the ACCESS command, you can prevent that user from reading or writing files elsewhere in the system. You may also create another, common logical disk containing minimally needed utilities and compilers and assign it logical name SY in the user start-up command file. In this way, each user has access to sufficient resources to perform his assigned tasks, but is prevented from poking around in the affairs of others.

Some users have expressed interest in having more than 8 logical disks mounted at any given time. This cannot be done with LD, but some fudging with older XD, SD, and AR disk subsetting handlers can achieve this effect. I personally think it is very difficult to keep track of more than 8 simultaneous mounts. The only reason anybody really wants this feature is because it is equally difficult to keep track of where tens or perhaps hundreds of logical disk files are located, and to constantly mount and remount these files. The next column describes an IND command file that automatically finds and mounts logical disk files wherever they are located.

FINDING A LOGICAL DISK FILE

As mentioned above, Horizon assigns logical disks largely on a project basis. We keep two logical disks always assigned: LD0 is our electronic mailbox disk and LD1 is our library/utility disk. LD2 is dynamically mounted in accordance with the project any given user is working on and is assigned as default device DK. We presently have about 50 logical disk files stored on three emulated RK07s. We frequently move logical disk files around to optimize storage and disk backups, so we generally do not know from day to day where a given logical disk file is located. To counter this situation, we developed, using IND and UCL, a simple procedure that allows TSX-Plus to find a logical disk file by project name (which corresponds to the file name), mount it as LD2, and assign DK to LD2. We also implemented a general MNT command which functions like MOUNT, but also finds the logical disk file for us.

The first component of this procedure is an IND command file called FIND.COM, stored on our system disk. In the listing of FIND.COM that follows, a couple of command lines were too long to fit our two-column format, so we broke them at convenient spots. Where a single line has been divided, the symbol && appears. If you are keying this file into your system, when you encounter this symbol, ignore it and continue the next line, ignoring its leading blanks, on the same line.

```
   .; FIND AND MOUNT A SUBDEVICE FILE
   .; COPYRIGHT (C) 1984
   .; HORIZON DATA SYSTEMS, RICHMOND VA
   .;
   .; THIS FILE MAY BE FREELY COPIED,
   .; BUT NOT FOR COMMERCIAL ADVANTAGE.
   .;
   .; PUT DEVICE NAMES TO SEARCH (WITH
   .; TERMINATING COLONS BUT WITHOUT
   .; INTERVENING BLANKS) IN LIST
   .;
   .SETS LIST "DM1:DM2:DM3:"
   .ENABLE QUIET
   .DISABLE PREFIX
 .AGAIN:
   .TEST LIST ":"
   .IF <STRLEN> EQ 0 .GOTO NOFIND
   .SETN LENG <STRLEN>
   .TEST LIST
   .SETS DEV LIST[1:LENG]
   .IF <STRLEN> EQ LENG .SETS LIST " "
   .IF <STRLEN> NE LENG .SETS LIST &&
       LIST[LENG+1:<STRLEN>]
   .TESTFILE ´DEV´´P2´.DSK
   .IF <FILERR> NE <SUCCES> .GOTO AGAIN
   MOUNT ´P1´ ´<FILSPC>´ ´P3´
   .IF <EXSTAT> NE <SUCCES> .GOTO NOMNT
   .EXIT
 .NOFIND:
   ;?FIND-F-File ´P2´.DSK Not Found
   .EXIT <SEVERE>
 .NOMNT:
   ;?FIND-F-Mount Command Failed - &&
       MOUNT ´P1´ ´<FILSPC>´ ´P3´
   .EXIT <SEVERE>
```

File processing is quite simple. Three parameters are passed to the file, corresponding to the three arguments of a MOUNT command. The first is the logical disk device name, i.e., LD0, LD1, etc. The second is the name of the logical disk file to be found, sans device name or extension (.DSK is assumed). The last (optional) is the logical name to be assigned as part of the MOUNT command. FIND extracts device names from a candidate list and tests each in turn to determine whether the logical disk file exists in its directory. If so, a MOUNT command is executed; otherwise, the next device is tried. If all devices are tried without success, or if a MOUNT command fails, FIND displays an appropriate error message and terminates with a SEVERE error condition.

To customize this file to your system, change the quoted string in the first command line (.SETS) to a list of all devices you wish to search for the logical disk file; include the terminating colon for each device, but do not include other characters including blanks. For example, to search DL0, DL1, DM1, and DM2, change the first command to read:

```
.SETS LIST "DL0:DL1:DM1:DM2:"
```

You may include any number of devices in this list. Devices are examined in the order in which they appear.

Next, we have defined two UCL commands, MNT and TO, as follows:

```
MNT := R IND\SY:FIND ^
TO  := R IND\SY:FIND LD2 ^ DK
```

MNT replaces the MOUNT command and follows its format. For example,

```
MNT LD2 TSXPUG DK
```

finds a file called TSXPUG.DSK, mounts it as LD2, and assigns it logical name DK. The logical name (third parameter) may be omitted as in the MOUNT command.

TO finds a logical disk file, mounts it as LD2, and assigns it logical name DK. It requires only the name of the logical disk file. For example,

```
TO TSXPUG
```

performs the same function as the MNT command above.

CHRISTMAS WISH LIST REVISITED

Many of you responded to our Christmas Wish List for TSX-Plus features published in the last newsletter. We thank all of you for your input. We received a couple of dozen letters. Interestingly enough, EVERYBODY voted for Item 1, improved system security, some with great emphasis. We have forwarded this information to S & H and I think we will see some improvement in this area in the near future. Other big vote getters were Item 6 (global logical device assignments) and Item 15 (send messages to inactive terminals).

An unlisted item was mentioned by Ron Tenny, namely drop case distinction in form names for the spooler, i.e., std and STD should be equivalent. This change will appear in Version 5.1C. How´s that for service?

If you did not vote, please forward your choices to us. We will continue to compile data and make the results available to S & H. This is your chance to influence the design of your favorite operating system.

JOHN STEWART´S DBL/DIBOL COLUMN

Due to circumstances beyond his or our control, John Stewart´s DBL/DIBOL column arrived too late for inclusion in this issue. We promise his column will resume in the next issue.

IMPROVED MODEM SECURITY FOR TSX-PLUS

Milton Campbell of Talisman Systems, who also writes the "RT-11 Perspective" column for Hardcopy, has submitted the following article deaing with improved security on TSX-Plus dial-in lines. We have reproduced the program to which he refers at the end of the newsletter.

The access security provided by TSX-PLUS is adequate for many installations; however, additional security for dial-in lines may be desirable. Many systems, including TSX-Plus, display such things as the installation name or operating system name and version number when the attention character is typed. This is fine for a valid user, but it gives too much data to an intruder. The long version of the TSX-Plus message is even worse, since it provides a partial history of the system. The following procedure and program can improve the security of a system with few dial-in lines.

The procedure is to run a program (as a detached job) that monitors the modem line until it sees a legal password. The program then allows the caller to log into TSX-Plus in the normal fashion. After the caller logs off, the program begins monitoring the line again for the password.

The program uses a CL line to monitor each modem line. Since a separate CL unit number is needed for each modem line, the number of lines that can be handled is limited to the number of available CL units. The program supplied monitors only one line, but it can be extended to handle more at the cost of increased complexity.

The program first attempts to associate a CL unit with a TSX-Plus dial-in line. If this attempt fails because the line is already logged on, the program goes to sleep for a few seconds then tries again. If the attempt fails for any other reason, the program gives up completely.

Once CL association occurs, the program assigns a channel and begins reading characters from the line. At this point, it is no longer possible for the caller to interact directly with TSX-Plus. Characters read are compared to the internal password (=HI-THERE). When a match occurs, the CL assignment is released and the user has 10 seconds to logon.

The program has been tested only on DL lines. I don´t know what it will do on DZ or DH lines, or on DL lines with programmable baud rates. The program included here has been simplified to reduce its size, but there are several additions that should be considered. It should not be too much trouble to add support for additional lines, although the cost in CL units is a problem. The program should probably lock itself into memory while reading characters from the CL line.

Calling a modem controlled by this program is a little mysterious because there is no indication that the computer has responded to the call until the internal password is entered. This is intentional, but some kind of innocuous prompt could be added to give legitimate callers a warm feeling. The program could also be extended to add other user interactions such as a call-back feature. After validating the internal password, the program could request a location identifier, then hang up the line and call the originating terminal back.

This method is not a complete solution to security problems, even for one modem line. Among other things, there is a window after a user logs off but before the program recycles during which another caller bypasses the program. Beyond the cost in CL units, this program consumes a detached line and imposes some overhead while reading the password and while waiting for the user to log off.

CURRENT VERSIONS

The current distributed version of TSX-Plus is 5.1B, generally identical to the Version 5.1 some of you may have. Version 5.1C, scheduled for release in March, is in field test. Version 5.1C adds a few new features, but largely corrects several errors in 5.1B.

Some 5.1B errors corrected in 5.1C involve field-width terminal input activation, stack overflows caused by the parallel printer (LP) handler, case discrimination on spooler form names, extended channel (above 16 decimal) purging, low memory fragmentation, and multiple simultaneous CL usage (see HINTS AND KINKS column). As usual, the technical support group at S & H, headed by Jan Bramlet, has been very responsive to user problem reports in creating this new release. If you don't think we are lucky in this regard, talk to a few RSTS users.

READER CONTRIBUTIONS

We repeat our oft-stated request that readers contribute to this effort. You needn't write a lengthy article, but there are several things you can do to help us help you. We would like to publish brief (four paragraph or so) reviews of software packages you have purchased and installed. Tell us what the package is, what you wanted to do with it, what you ended up doing with it, and your general reactions to its quality, documentation, etc. If you have developed a few clever command files or UCL commands that make using TSX-Plus easier, share them with us. If there is something about TSX-Plus that you just don't understand, let us know; it may be a topic for a tutorial.

We will return in three months. The next issue should reach you right after the New Orleans DECUS. Goodbye until then!

```
    .TITLE  WATCH   TSX+ MODEM SECURITY PROGRAM
;
; BASIC MODEM SECURITY PROGRAM
;   BY:     MILTON D. CAMPBELL
;           TALISMAN SYSTEMS
                    ;
; SET THE FOLLOWING TO THE "RE-CAPTURE" CYCLE INTERVAL, IN TICKS
                    ;
RECYCL=120.         ;120. = 2 SECOND INTERVALS
                    ;
; SET THE FOLLOWING TO THE TSX LINE NUMBER THAT IS THE MODEM
```

```
                                        ;
MODLIN=2
                                        ;
; SET THE FOLLOWING TO THE CL: LINE NUMBER TO BE USED
                                        ;
CLLINE=1
                                        ;
        .MCALL  .TWAIT,.PRINT,.SPFUN,.EXIT,.LOOKUP,.PURGE
                                        ;
ERRBYT=52
                                        ;
START::
        JMP     TRY
;
; WE COME HERE WHEN AN ATTEMPT TO CAPTURE THE TSX LINE
;       FAILS. GO TO SLEEP FOR A WHILE, THEN TRY AGAIN
RETRY:
        .TWAIT  #TWBLCK,#TIME
        BCS     RETRY           ;FAILED? TRY AGAIN
TRY:
        MOV     #CAPTUR,R0      ;POINT R0 TO THE EMT BLOCK
        EMT     375             ;TRY TO GET THE LINE
        BCC     SUCCES          ;TO BEGIN THE P/W SEQUENCE
        CMPB    #5,ERRBYT       ;CODE 5 IS "LOGGED ON"
        BEQ     RETRY           ;IF SO, LOOP AGAIN
        JMP     FATAL           ;ANYTHING ELSE IS FATAL
SUCCES::
        MOV     #^RCL0,CLBLCK
        ADD     #CLLINE,CLBLCK
        .LOOKUP #LKBLCK,#0,#CLBLCK
BADPW:
        MOV     #PSSWRD,R4      ;POINT R4 AT PASSWORD STRING
        CLR     STATE           ;HAVE NOT STARTED PASSWORD
        CLR     SIZE
COLLCT:
        .SPFUN  #SPBLCK,#0,#203,#INCHAR,#1,#0
        TST     STATE
        BNE     30$             ;BR IF INSIDE PASSWORD
        CMPB    (R4),INCHAR     ;SEE IF STARTS PASSWORD
        BNE     COLLCT
        INC     STATE           ;SHOW COLLECTING PASSWORD
30$:
        CMPB    (R4),INCHAR     ;SEE IF CHARACTER MATCHES
        BNE     BADPW           ;IF NOT
        INC     R4              ;POINT TO NEXT CHAR
        INC     SIZE
        CMP     SIZE,#PWLEN     ;SEE IF SIZE OK
        BLT     COLLCT          ;IF NOT AT END
        .PURGE  #0
        MOV     #FREE,R0        ;POINT R0 AT DISCONNECT BLOCK
        EMT     375             ;DISCONNECT DL
        .TWAIT  #TWBLCK,#LNTIME ;GIVE USER TIME TO LOGON
        JMP     TRY
                                        ;
FATAL:
        MOV     #EBLCK,R0               ;POINT TO MESSAGE BLOCK
        EMT     375
        .EXIT
                                        ;
STATE:  .WORD   0
SIZE:   .WORD   0
                                        ;
; THE PASSWORD IS DEFINED IN THE FOLLOWING
                                        ;
PSSWRD::
        .ASCII  /=HI-THERE/
PWLEN=.-PSSWRD                          ;COMPUTE LENGTH
                                        ;
INCHAR: .BYTE   0
        .EVEN
                                        ;
; EMT BLOCKS
                                        ;
SPBLCK: .BLKW   6
LKBLCK: .BLKW   4
EBLCK:
        .BYTE   0,127
        .WORD   1
        .WORD   FMESSG
TWBLCK: .BLKW   2
                                        ;
CAPTUR: .BYTE   0,155                   ;"CAPTURE" EMT BLOCK
        .WORD   CLLINE
        .WORD   MODLIN
                                        ;
FREE:   .BYTE   0,155                   ;"FREE" EMT BLOCK
        .WORD   CLLINE
        .WORD   0
                                        ;
CLBLCK: .RAD50  /CL0/
        .RAD50  /WRKFILXXX/             ;FILE NAME (NOT USED)
                                        ;
TIME:   .WORD   0,RECYCLE               ;RE-CAPTURE CYCLE TIME
                                        ;
LNTIME: .WORD   0,600.                  ;USER HAS 10 SECONDS
                                        ;
FMESSG:
        .ASCII  /? WATCH-Fatal Error:/
        .ASCIZ  / Unable to Associate CL: line/
        .END    START
```