

N87-27236

## EXPERIENCES RUNNING NASTRAN ON THE MICROVAX II COMPUTER

Thomas G. Butler and Reginald S. Mitchell  
BUTLER ANALYSES GODDARD SPACE FLIGHT CENTER

## INTRODUCTION

The engineering analyst operates in an entirely new environment today compared to the days when NASTRAN went public. No individual could afford to own a computer capable of operating NASTRAN in 1972. And rarely did an engineering group have a computer dedicated solely to its purposes. Any NASTRAN-sized computer generally had a staff of people associated with its various requirements. It would have a systems staff, a utilities programming staff, a maintenance section and an operating crew.

The engineering analyst would generally interface with the operating staff (dispatcher) and submit his job to be processed by the operations group. This same operations group would collect his output, input, plots, and tapes and assemble them for him by job in his "output-box" or his courier station.

Later on when remote batch terminals (card read and print stations) became available, the user would by-pass part of this interface by inserting his job directly into the input queue. Operations people still manipulated and processed most of the output as before. With the improvement in remote terminals and with the extension of job control language, the analyst could take charge of this input and monitor the progress of his job and even change its status with respect to other traffic (delete, postpone, wait, control sequence). Generally, even though he knew that he had that access, he sought the tutelage of a systems programmer to set up a series of commands for him so that a job would proceed according to his wishes. Any demands on resources

were certainly routed through a systems programmer; i.e. extra memory beyond user quota, or extra disk space at a certain stage of his job, or routing storage to tape, transferring files between jobs, etc.

In essence, then, even though the analyst had access to commands to control the routing of his job, he generally opted for the easy way out by placing his dependency on the systems programmer. A fair number of analysts struggled along without help and lapsed into unenlightened ruts which were a combination of hints from others who operated similar jobs, reaction to scoldings from dispatch personnel, and occasional discussions with systems programmers. But in any such exchanges there must be good communication. The analyst must project his objectives in sufficiently jargon-free phrases that the systems programmer can grasp the requirements to provide a scheme of commands to allow resources in a timely fashion and obtain approvals for the temporary hoarding of storage or priorities or post-processing. One aspect of systems programming that was almost opaque to the analyst was (and still is) the tailoring of main memory, secondary storage, limits on system management facilities, allocation of utilities, assigning of queues, and assigning of priorities to a given software package. A program such as NASTRAN has needs such as few other programs. NASTRAN jobs can run sluggishly or efficiently depending on how well the systems programmer matches the allocations of computer resources to the program's resident memory requirements, scratch disk space requirements, and I/O requirements.

Interpretations of diagnostics to find out what mistakes the analyst made in his data were the tasks attended to by a computer's support staff. And in case an error was too elusive, the problem could be rerun with a core dump and any anomalous step could be pinpointed. Such is the help obtainable from the support staff.

Thanks to competition amongst computer manufacturers, two trends worked to the advantage of the analyst---system commands

became friendlier and editing languages improved greatly. With a little digging an analyst could prepare problem data more easily and could also have more confidence in simpler job control tasks. But he tended to reach a plateau of only simple skills and still depended on the computer staff for considerable support.

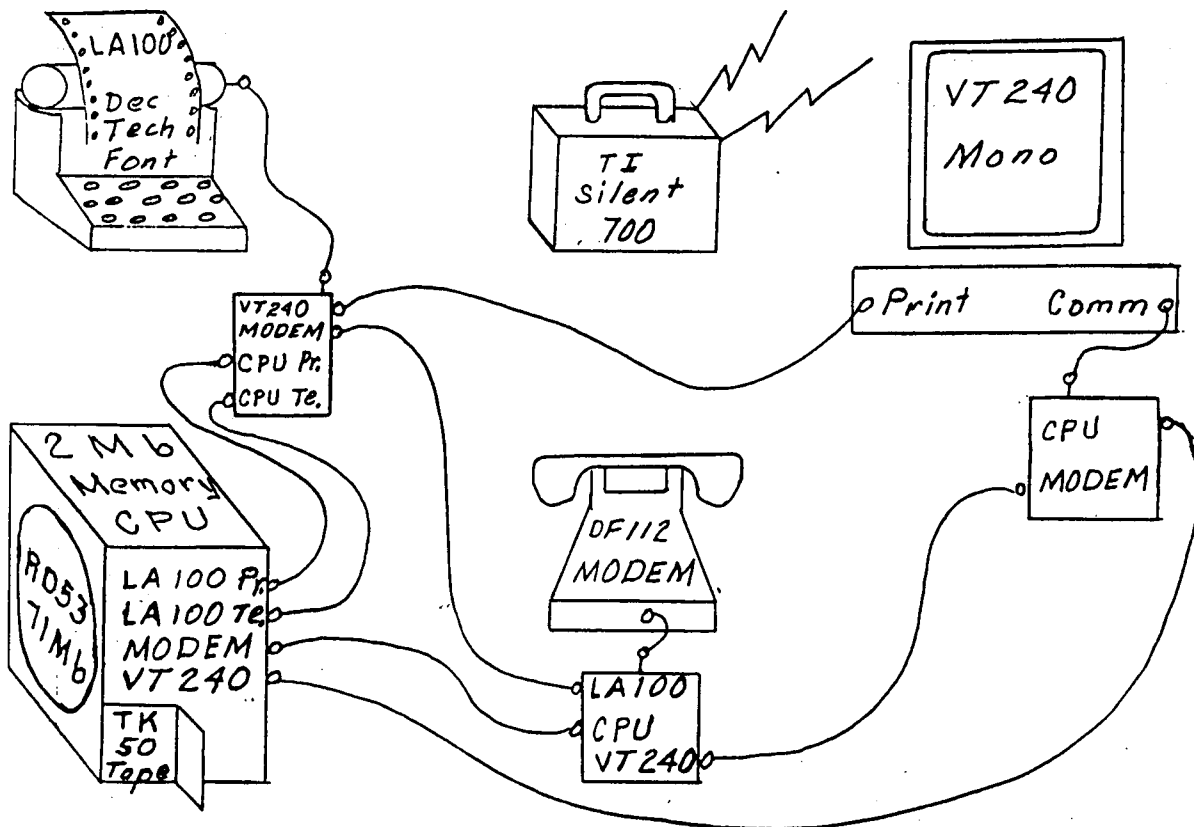
The next change of his computer environment came with the popularity of mini-computers. But the word mini implies less, so the budgets that usually went with the purchase of mini-computers were less than those for main-frames. This meant less staff--and sometimes no staff at all. Systems programmers who were assigned to a mini-computer would be available on prime-time shifts only and had so many other duties that they could give only limited help to analysts. This is a kind of good news--bad news story in which the engineers had their own computer, but were too ignorant in computerese to take advantage of it. A small cadre' buckled down and became computer wizards, but most engineers remained awestruck by the computer.

Then the unbelievable happened. Micro computers were marketed with such power that they could operate NASTRAN. Their cost was under \$50,000! Now it became economical and logical for an individual engineer to have his own unshared computer that could be linked to any other installation for which he had traffic. Good News and Bad News. He has his own computer BUT he has to be his own staff. When he gets a message for instance that max pages has been exceeded, he instinctively seeks help from a systems programmer. But then is me. Change hats and answer your own questions. Whoops! That's where we are today. Some of us got ourselves into this fix voluntarily and others became involuntary victims of this computer tyranny. The rest of the paper will be devoted to a description of the computer staff burdens that an individual analyst/owner of a MicroVAX II has to assume when he wants to use it to run NASTRAN. The topics will be divided according to the logic of getting NASTRAN to operate on a new MicroVAX installation.

## DECIDING ON THE HARDWARE

Many of the system configuration decisions depend on the hardware that the system is to manage and the applications programs which operate with it. The MicroVAX II hardware that I installed is as follows: 2 Megabytes of main memory, RD53 71-Megabyte Winchester Disk Drive, TK50 Drive for magnetic tape cartridges, VT240 Monochrome Terminal with Bit Mapping and port to printer, LA100 Typewriter Terminal/Printer with DEC Tech Math Font, DF112 Modem for 300 and 1200 baud transmissions, TI Silent 700 Typewriter Terminal for telephone access, Black Boxes for switching channels between components, Q-BUS backplane for expandable main memory and disk space, VMS Level 4.2 Operating System (No FORTRAN).

The configuration is shown in a diagrammatic sketch showing how the black boxes provide for various interconnections.



The modus that I intended for NASTRAN was: (a) to prepare the bulk data for analytical jobs and debug them well so that they can be reliable for submission to a remote main-frame; (b) to use NASCAD as a pre and post processor to help prepare input data and to massage the output from remote main-frames for final reports; (c) to make pilot runs to prove out solution strategies before enabling them on big jobs; and (d) to compose DMAP code for special applications and debug them thoroughly before incorporating them into an application run.

I chose not to subscribe to either a software or a hardware maintenance license, because the cost was prohibitive for a single user. Unfortunately, it would have paid off if I had, because my office had an air-conditioning failure which caused the computer to cook. Without a maintenance contract, the MicroVAX owner is treated by DEC like a second class citizen. It takes a long time to thread through their bureaucracy to pin point the right group to engage for service. With trouble one first has to establish whether the problem is software or hardware. Even after service is engaged, the non-maintenance-contract customer is shifted to lowest priority with days of waiting. It was a rude shock to be treated so shabbily. Once service was rendered, it was excellent. But to someone who is not systems oriented, it becomes a harrowing experience to have an emergency without a maintenance contract.

The systems approach to getting NASTRAN ready to operate on a MicroVAX-II will be discussed in this sequence: sizing the system; preparing the system disk; defining user quotas; installing NASTRAN; monitoring hardware errors; establishing day to day operations.

#### SIZING THE SYSTEM

The VMS operating system allows the computer manager to parcel all available memory (both main and disk) into quantities of storage that can be assigned to each process as it is submitted.

to the system. The name given to this parcel is "page" when referring to main memory and "block" when referring to disk space. A page on the VAX is a measure equal to 512 bytes. A disk block turns out to be the same size - 512 bytes. The system doles out memory to processes in units of pages. It seems logical to provide for the case in which the only traffic on the computer would be the largest program in the library. If the largest is NASTRAN, which needs 16,000 pages of memory for it to reside all at once in the computer, it would be most desirable to be able to give it a parcel of 16,000 pages. Since the MicroVAX does not come with that much real memory, we must take advantage of VAX's virtual (make believe) memory capability. Virtual memory means that only a portion of a program needs to reside in the real hardware memory at any given time (this portion is called the "working set"). The rest of the program and its data at any given moment reside on disk in the image (EXE) file and the system PAGEFILE, respectively.

The first task facing the new owner is deciding how much virtual (make believe) memory to have. This number must be chosen with some care because if it is too small, large program images will not run at all. On the other hand, making the virtual size too big wastes system resources. One can find out how much virtual space NASTRAN needs by consulting the LINK map that is part of the delivery package from COSMIC. To determine how many virtual pages are needed by NASTRAN, examine each LINK map (there are 15 for COSMIC NASTRAN) and note the largest value of virtual space (the figure is near the end of each map). The current value is around 8 megabytes. Since each page is one-half kilo-byte, this converts to about 16,000 virtual pages. NASTRAN and NASCAD need about the same amount of virtual pages to operate, so I chose a value of 16,000 for my Virtual Page Count (abbreviated VIRTUALPAGECNT). VMS also allows the computer manager to regulate how main memory is divided up amongst the traffic of processes it serves. A user-supplied system parameter defines the maximum parcel of main memory to be assigned to a given process at any time. This parameter is given the name Working Set Maximum (abbreviated WSMAX). Since the subject

machine has a 2 megabyte memory (4,096 pages of 512 bytes each), the decision then is to pick a fraction of 4096 pages as the maximum that any process will be allowed. First of all it would be a good idea to see if DEC supplied everything as you ordered it, memory-wise. There is a Digital Command Language (DCL) command that will report on this fact. Key in the command SHOW MEMORY. The response will be a display of a table of values. It confirms that there are 4096 total pages of physical memory available. The last line reports the number of physical pages occupied by the system. The table shows that my VMS 4.2 is occupying 1350 pages. Subtracting 1350 from 4096 gives 2746 pages that are available to processes. An example display of the SHOW MEMORY report is shown below.

System Memory Resources on _____				
Physical Memory Usage (pages):	Total	Free	In Use	Modified
Main Memory (2.00Mb)	4096	1945	2102	49
Slot Usage (slots):	Total	Free	Resident	Swapped
Process Entry Slots	12	6	6	0
Balance Set Slots	10	6	4	0
Fixed-Size Pool Areas (packets):	Total	Free	In Use	Size
Small Packet (SRP) List	117	24	93	96
I/O Request Packet (IRP) List	73	17	56	208
Large Packet (LRP) List	10	2	8	656
Dynamic Memory Usage (bytes):	Total	Free	In Use	Largest
Nonpaged Dynamic Memory	299520	197312	102208	193248
Paged Dynamic Memory	81920	24368	57552	23552
Paging File Usage (pages):		Free	In Use	Total
DISK\$MICROVMS:[SYS0.SYSEXE]SWAPFILE.SYS		3040	1056	4096
DISK\$MICROVMS:[SYS0.SYSEXE]SWAPFILE1.SYS;1		3448	0	3448
DISK\$MICROVMS:[SYS0.SYSEXE]PAGEFILE.SYS		13788	212	14000
DISK\$MICROVMS:[SYS0.SYSEXE]PAGEFILE1.SYS;1		2192	0	2192

Of the physical pages in use, 1350 pages are permanently allocated to VMS.

Even though I am the sole user I may have more than one process running at once on my VAX. I can have a NASTRAN job running in batch. I can be preparing another job with NASCAD interactively. Another job could be printing. Then I could telephone the modem with the Silent 700 terminal to interrogate the progress of the queues. If the maximum parcel of main memory is fixed at 2,000 pages of main memory, the system will be able to handle the traffic with this upper limit for any one. If the machine is busy, the actual working set for a given process may be smaller than WSMAX, but it may never be larger. So now the plan is to set VIRTUALPAGECNT to 16000 and to set WSMAX to 2000. How does one go about setting these values on the system? Log on as SYSTEM. Set the Default Directory to SYS\$SYSTEM. At the \$ prompt send the DCL command RUN SYSGEN. The screen will now display a new prompt SYSGEN>. Issue the command SHOW/MAJOR and the response will be a new table giving values to parameters. If they need to be changed, issue the command SET VIRTUALPAGECNT 16000 and SET WSMAX 2000. Issue another SHOW/MAJOR to confirm that the parameters are correct. Then type EXIT. When the \$ prompt is shown it will be necessary to issue the command @SHUTDOWN after which you should reboot the system by pressing the "Restart" button on the face of the CPU cabinet. During the rebooting, the system resets the values of parameters including the ones just prescribed. Check SYSGEN again to see that the parameters do have the new values. Now it is time for the next step.

#### PREPARING THE SYSTEM DISK

Parameters to be set in this section will govern the disk space to support the machine's memory. Three quantities will be set. The character of their names is that of files, because the system views these logically as files. Their names are PAGEFILE.SYS, SWAPFILE.SYS, and DUMPFIL.SYS. The reason that these files need to be created is that VMS is a virtual system. This means that only a fraction of a given IMAGE (program) is in main memory at any given time. What fraction is operating depends on the traffic in main memory, subject to the limit imposed



by WSMAX. If several programs are running and all are small, then they might all fit into main memory simultaneously, but if several large programs try to gain their allowable maximum, the system limits each to less main memory and completes their complement of virtual memory by calling for disk space from the file PAGEFILE.SYS. Ultimately the system will vacate low priority jobs from main memory and roll them entirely out to disk into a space called SWAPFILE.SYS. One more disk file is set aside for use by the system called DUMPFIL.SYS. Its purpose is to take over when an abort threatens and save everything that is in random access memory (main memory) and transfer it to the less volatile magnetic memory (disk). Here is the strategy for setting the size of these 3 files. Take a census of all the things that will be in permanent storage on disk and subtract that total from the rating of the disk. The quantity used to measure magnetic storage (disk, tape, and floppies) is the block. One block of disk space is set to 512 bytes (the same as a page), so we will use the block and page interchangeably in computing our disk space needs. The way to inquire of the system as to the sizes of these various disk files is to log onto the account with the directory in question and issue the DCL command DIR/SIZE=ALL. This will report on each file and give a total for all. NASTRAN occupies 16,000 blocks. The system files in the directory SYS\$SYSTEM occupy 14,674 blocks before any assignment is given to these three job management files. The system files in the directory SYS\$LIBRARY is 10,160 blocks. The system files for doing housekeeping of files on disk (directory DUA0:[0,0]) occupy 2,639 blocks. Other major programs like NASCAD and the WordMARC word processor occupy 8,900 blocks and 8,600 blocks respectively. This amounts to a burden of 60,672 blocks. Subtracting this from 138,000 blocks leaves 77,268 blocks. A rule of thumb for a MicroVAX sizing of PAGEFILE.SYS is to set it slightly larger than VIRTUALPAGECNT; e.g. 16,200 blocks. The rule of thumb for sizing SWAPFILE.SYS is to set it equal to half of PAGEFILE.SYS; e.g. 7,500 blocks. The rule for sizing DUMPFIL.SYS is to set it equal to the number of real pages in main memory plus 4; i.e.  $4096 + 4 = 4100$  blocks. Check to see that the total assignment

is within the remaining capacity of the disk: i.e. 27,800 vs. 77,268. There is a 49,460 block margin.

Now that we have figured out what size to make these files, the next step is to actually invoke this plan. Log onto the system account and set the default directory (SET DEF) to SYS\$UPDATE. Execute the system utility by issuing the command @SWAPFILES. Immediately the display gives a directory/size report on the subject files and reports on the available disk space. The utility prompts for the assignments first for the paging file, then for the dump file, and then for the swap file. After finishing, reboot the system as was explained in SIZING MAIN MEMORY above, in order to enforce these values into the system. All of this should take place before NASTRAN or other programs are loaded. Once the disk becomes cluttered with files, it becomes harder to find the solid block of room required for the PAGEFILE.SYS.

#### DEFINING USER QUOTAS

User resource quotas now need to be set in order to dictate how much of these computer resources will be allowed to a user at any one time. The resources that become important for a NASTRAN user are as follows. A limit is set on the maximum number of files that may be open at any one time (abbreviation FILLM). NASTRAN associates a large number of files with an execution so a satisfactory number would be 60. A limit is set on the amount of the PAGEFILE.SYS file that a user may have at any one time (abbreviation PGFL quota). To accomodate NASTRAN the user should get 10,000 pages. A limit is set on the maximum number of bytes the user can have involved in buffered I/O operations (abbreviation BYTLM). This should fall in the range from 8,000 to 12,000.

The way to implement these decisions is to log onto the SYSTEM account and set the default directory to SYS\$SYSTEM. Next issue the command RUN AUTHORIZE. A new prompt, AUTH>, will appear on the screen. If the user account already exists, use the commands:

MODIFY/FILLM=60/PGFLQUOTA=10000/BYTLM=12000 useraccount

and

EXIT.

In order to put these parameters into effect the user logs off then back on again.

There is a related task that must be attended to. There must be a Job Queue set up for running NASTRAN as a batch background job. This is done in two steps. First the queue is defined, then it is started. In order to define the queue, certain decisions have to be made with respect to how the manager wants it to operate for the NASTRAN users. These considerations are: How many jobs may be operating simultaneously? (abbreviation JOB\_LIMIT). For a MicroVAX it is wise to limit this to one at a time. What type of queue shall it be? It has already been decided this will be a batch queue (abbreviation BATCH). What priority will the background job be given? (abbreviation BASE\_PRIORITY). In order to give good turn-around to interactive users, the priority for batch jobs should be a low value like 2. What is the limit on the amount of main memory that can be assigned to a job in this queue? (abbreviation WSEXTENT). The system cannot give more than the value of WSMAX, but because consideration for NASTRAN was the prime factor in setting the value of WSMAX, WSEXTENT should be made equal to WSMAX. What protection should be assigned to jobs running this queue? (abbreviation PROTECTION). Other users should be able to find out what traffic there is on the computer so that they can adjust their activities accordingly, so set the protection to allow the World to Read. There is no need to set a time limit on the queue unless a host of users descends on your private world. To enable the queue definition, log into the SYSTEM account and set the default to SYS\$MANAGER. Send the DCL command

INITIALIZE/QUEUE/BATCH/BASE\_PRIORITY=2/JOB\_LIMIT=1/WSEXTENT=2000-  
/PROT=(W:R) SYS\$BATCH.

The other part of this task is to arrange for this queue to be available every time the user submits a job, which is tantamount to saying that it should be started every time that the system is booted. Incorporate the startup arrangement in the command procedure called SYSTARTUP.COM, by editing the file with the EDT editor. There is a system manager utility that has to be enabled before any queue starting command will be honored. Therefore this preceding command must be entered into the .COM file before doing the editing; \$START/QUEUE/MANAGER. Next insert the command \$START/QUEUE SYS\$BATCH. Save the file and reboot the system so that this queue will go into effect. NASTRAN's new computer home is now ready for it to move into.

#### INSTALLING NASTRAN

The computer environment for NASTRAN is now set. NASTRAN can be loaded into any account and any directory you may choose. It is advisable to give such a big program its own directory. The protection of this directory should be prescribed so as to allow access to the NASTRAN executable from other user accounts in other directories. In order to establish a NASTRAN directory, log into the SYSTEM and set the default to SYS\$MANAGER. Issue the command

```
CREATE/DIRECTORY DUA0:[NASTRAN]-  
/OWNER_UIC=[XXX,YYY]/PROTECTION=(S:RWED,O:RWED,G:RE,W:RE)
```

Now log onto the new NASTRAN account and load the cartridge containing NASTRAN into the TK50 tape drive and read all files from the tape into the NASTRAN directory. Included in this delivery are two prototype command procedures named NASTRAN.COM and NASTRANF.COM. There is a line in both of these procedures that needs to have variable parameters specified. It starts out \$DVC\_DIR:=. Whatever follows the equal sign should be deleted and the directory as defined above in the CREATE statement should be entered, but enclosed in double quotes; i.e. "DUA0:[NASTRAN]".

At last, NASTRAN is ready to run!!

## MONITORING HARDWARE ERRORS

The spectre of having a computer disaster which would disable everything is something that can be minimized. All that is necessary is to use the maintenance tools that DEC has provided with your MicroVAX VMS system. There is a utility procedure called STABACKIT.COM (Standalone BACKUP) which is nicely organized with prompts to generate a stand-alone BACKUP program tape (stand-alone means that the VMS system is bypassed while using the program). The program tape must be made before the need for it arises - sort of like filling the fire barrels before the fire starts. If your computer crashes and you can't communicate with the VMS system, this tape can be mounted and booted and a crutch set of system BACKUP utility commands can be issued from the keyboard. This Stand-alone BACKUP program can process the set of BACKUP save tape files to re-establish the system on disk. This works if there weren't any physical damage to either the tape drive or the disk drive.

The Stand-alone BACKUP program tape is generated by logging into the system and setting the default directory to SYS\$UPDATE. The MicroVAX is more limited than other VAX machines, so temporary space assignments must first be arranged to ensure that the utility will have room to operate. Reset the default directory to SYS\$MANAGER and issue the command RUN SYSGEN. The following parameter values need to be set.

```
NPAGEDYN 60000
NPAGEVIR 400000
PAGEDYN 190000.
```

Reboot as before.

Start the procedure by typing @STABACKIT.COM. From there on it is a matter of obeying or answering the prompts. After the stand alone backup tape has been generated, go back to SYSGEN and revert the parameters to their initial values.

Files on the disk should be backed up periodically. They can be full backups of everything on the disk, or they can be

incremental backups of just the new items since the last catalogue. There are certain parameters that are recommended for inclusion in the BACKUP command. They are:

- /RECORD -- this writes an entry in the directory header of every file to say when it was last backed up. When incremental backups are made this date is consulted along with the dates logged for last modification in deciding whether a new backup is needed.
- /LOG -- this displays the activity of writing the backup tape; diagnostics of anomalies are displayed on line.
- /OUTPUT=filename -- this creates a file on the disk of the tape contents with any diagnostics; this is useful for storing with the tape for reference.
- /BUFFER=5 -- this is peculiar to the MicroVAX; it sets the I/O buffers for efficient running of backup.
- /REWIND -- this ensures that the tape is at the beginning.
- /IMAGE -- this is used if a complete backup is desired.
- /SINCE=BACKUP -- this is used if an incremental backup is being run.

The VAX has an internal sensing of hardware misbehavior as part of its design. The VMS operating system provides utilities to record errors as they happen and later display them upon command. A record is kept of what device, what operation, and when the anomaly occurred. The utility that manages this service is called ERRFMT. It is well to ensure that this utility is enabled. If the command SHOW SYSTEM is issued from any account, it produces a display of system activities. If the utility ERRFMT is shown to be active, then it is enabled. If not, then steps should be taken to put it into operation. The first place to check if it is not operational is in SYSTARTUP.COM. It might be that the prototype COM file had the RUN ERRFMT statement commented out with an exclamation mark.

The user can get a gross report on the total number of errors that have been found for each device by issuing the command SHOW ERROR. More detail can be obtained by logging into SYSTEM and setting the default directory to SYS\$ERRORLOG where a number

of files pertaining to errors are kept. The most relevant file is called ERRLOG.SYS. This is a binary file and is difficult for the uninitiated to read. However, this file can get crowded, making it a chore to filter out the most recent information from the chaff. A routine maintenance step should be to periodically rename the existing file from ERRORLOG.SYS to ERRORLOG.OLD. The error files can be read with another VMS utility called ANALYZE. Issue the DCL command ANALYZE/ERROR\_LOG and it gives a complete report. DEC repair men are trained to read these reports and they are extremely valuable to them in case of an emergency. With a little trouble the computer manager can get the jist of what is going wrong at least to the extent that he might sense when it is time to call in a repair man.

#### OPERATION POLICIES

A certain number of does and don'ts are good ideas to observe. So long as the ambient temperature is acceptable, it is best to leave the MicroVAX running even when there is no traffic on it. The reason to apply this policy is to avoid the most damaging stress that the computer chips see--thermal cycling stress. Shut down when the computer is to be vacated for protracted periods or when ambient conditions are critical.

The Winchester disc is a delicate piece of machinery. Avoid jarring it or moving the CPU case where it is installed.

Install a thermal power cut-off switch which senses a temperature threshold and interrupts power to the computer when that threshold is exceeded.

Install a recording max-min room air temperature thermometer which can provide you with legal data of how well the air conditioning system was maintained prior to a catastrophe, so that compensation can be claimed.

Install a surge and spike protector in the power supply line to catch power anomalies. This is especially useful if your area

power lines are above ground. A lightning strike or car knocking down a power pole could damage your VAX in milli-seconds.

Read the error log periodically. Read the ACCOUNTING.DAT file periodically to check on foreign usage.

Keep a set of BACKUP tapes distant from the computer site, so that a magnetic record is available in case of fire damage or other major disaster. If several sets of BACKUP tapes are kept, it is a simple matter to cycle old tapes back to the computer and new sets to the off-site storage location. Remember, new VAXes are available from DEC in a matter of days, but your data base might take months or even years to replace.

Join DECUS (DEC User Society) for a useful exchange of information with other VAX and MicroVAX users. Every VAX owner may join at no charge. There is a large amount of exchange software available at little or no charge through DECUS.

## CONCLUSION

The MicroVAX operates NASTRAN so well that the only detectable difference in its operation compared to an 11/780 VAX is in the execution time. Execution immediately upon submission considerably offsets the slower running time. On the modest installation that was described here an individual engineer has all of the tools that he needs to do an excellent job of analysis. There is room to install NASTRAN, NASCAD, WordMARC and many handy utilities and still leave space for files of results. All of these tools, both hardware and software, have great capability at affordable prices. It is possible to expand main memory to about 10 Megabytes and to expand connected disk space to over 500 Megabytes. Running large NASTRAN jobs is possible. The biggest difficulty for most analysts of having a "private NASTRAN computer" is having to wear so many hats. This paper has tried to reduce the uneasiness one feels with the unfamiliar by suggesting guidelines for some of the essentials that the engineer/analyst/systems type must deal with.