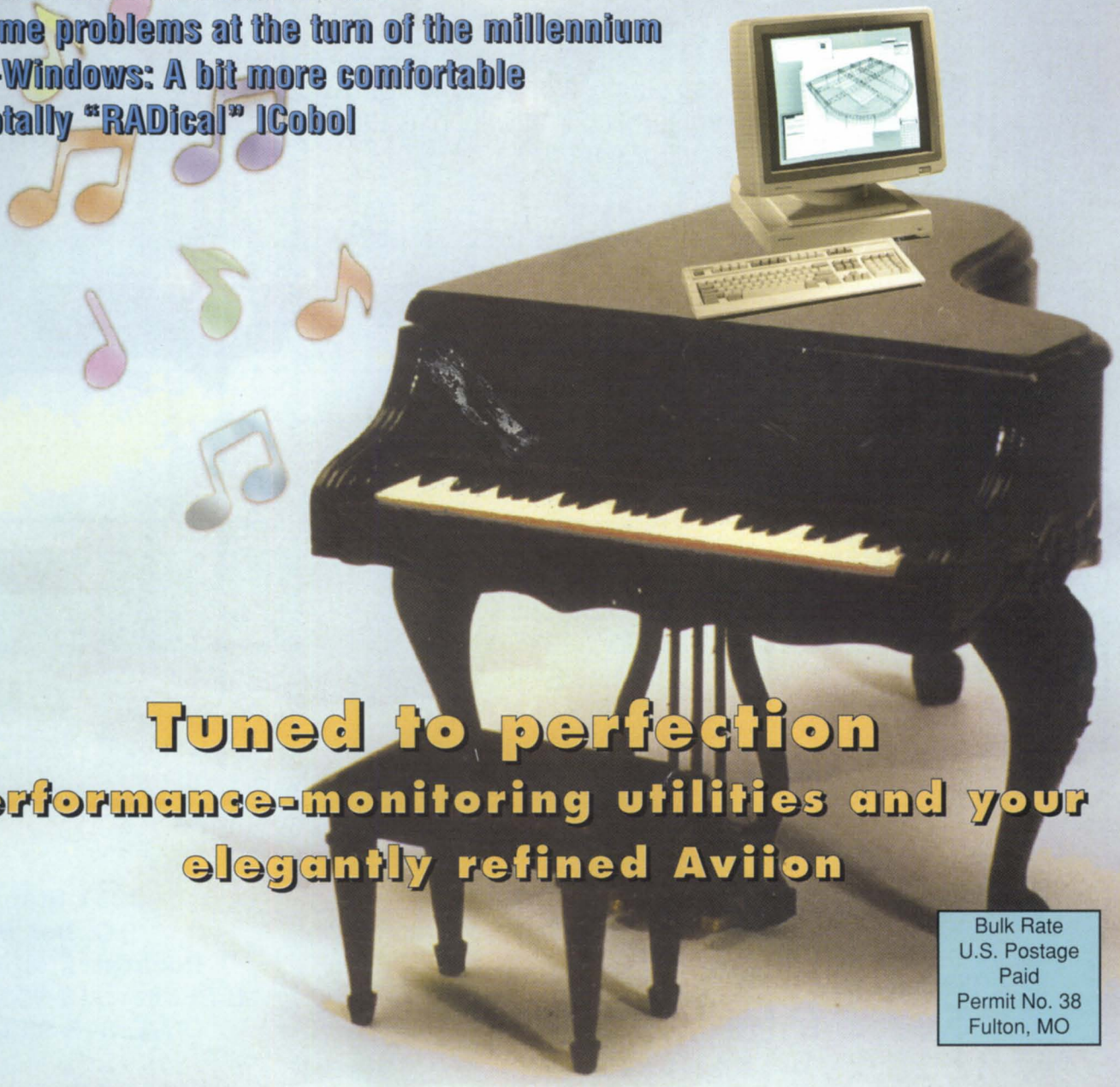
August 1993

FOCUS

The Magazine of the North American Data General Users Group

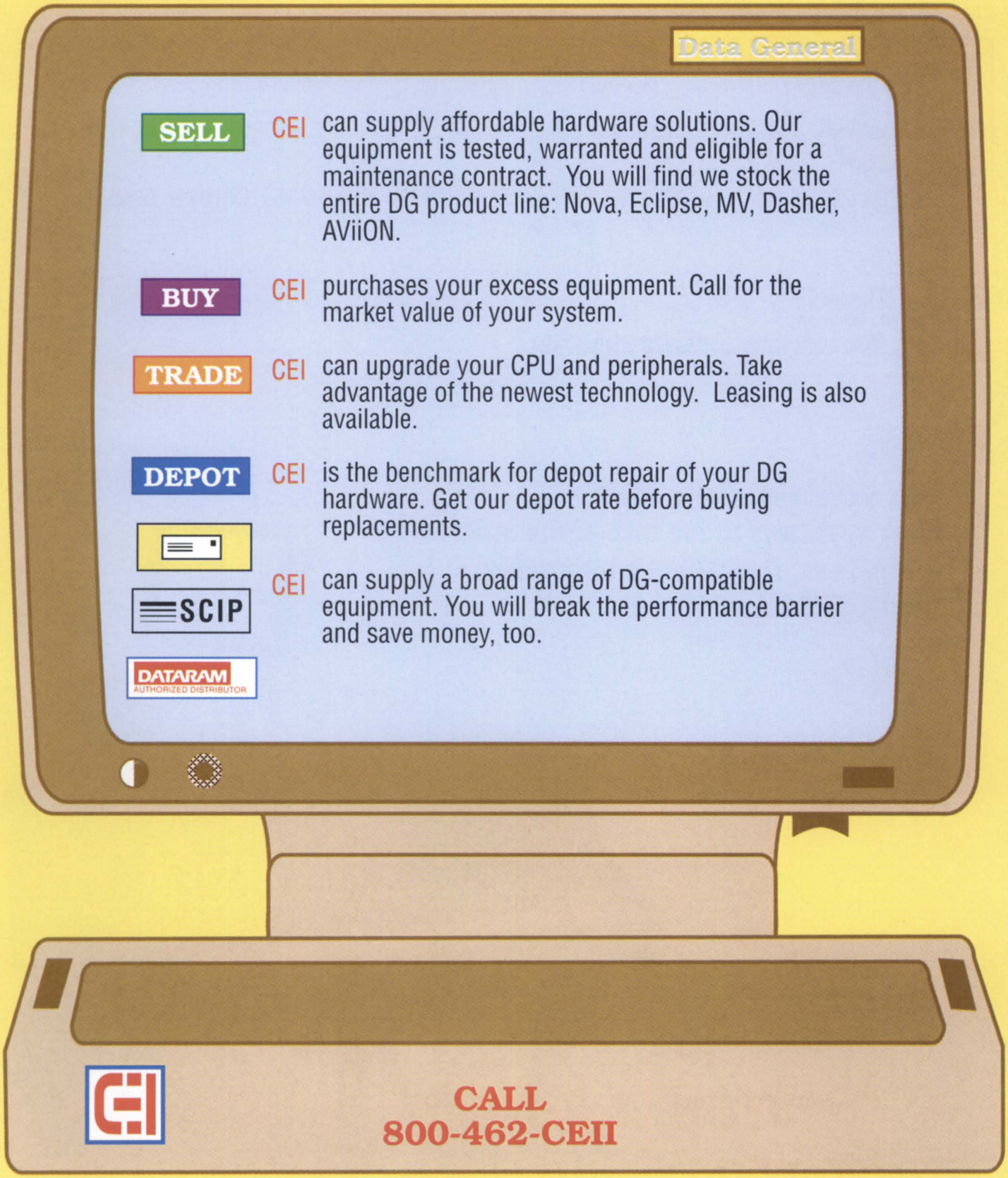
In Focus
Sar, the system activity reporter

Plus
Unix Notebook: Gone fishin'
Time problems at the turn of the millennium
X-Windows: A bit more comfortable
Totally "RADical" ICobol



Tuned to perfection
Performance-monitoring utilities and your elegantly refined Aviion

Bulk Rate
U.S. Postage
Paid
Permit No. 38
Fulton, MO



Computer Engineering International

2231 Star Court
P.O. Box 81755
Rochester, MI 48308
MI Phone: 313-853-0770
FAX: 313-853-0775

Circle 7 on reader service card.

Kill ICOBOL Bugs DEAD!

With *RAID**, ICOBOL bugs won't bite you anymore! *RAID* is the only ICOBOL debugger using pop-up windows, pull-down menus, mouse support and simple, speedy command operation. These features combine to give you the most potent, easy-to-use tool for finding and exterminating ICOBOL bugs.

RAID simultaneously monitors break points, program variables and source code in separate windows, allowing you to immediately zoom in on the problem. The result is faster debugging—and increased productivity!

And like all other Wild Hare products, *RAID* is completely portable, to zap bugs no matter where they hide.

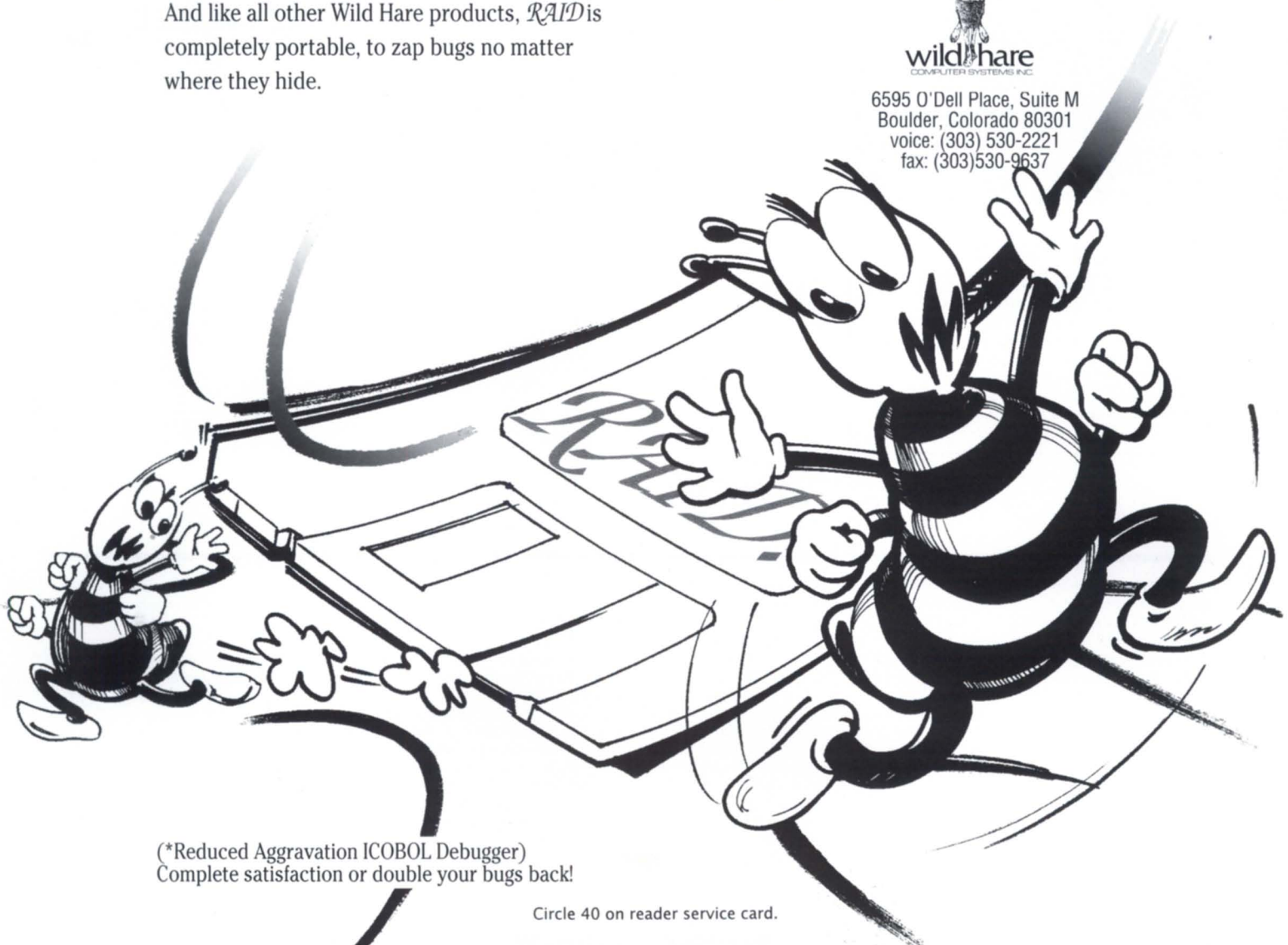
RAID is an integrated part of Wild Hare's *Choice!*™ ICOBOL run-time system and *Axis* ICOBOL compiler. We combined the most portable runtime, the fastest compiler and most powerful debugger to give you unparalleled programmer productivity!

Itching to try it? Order *Axis* with *RAID* by September 30, 1993, and we'll give you *Choice!* FREE! It's even covered by our 30 day sting-proof money back guarantee.

Get rid of ICOBOL bugs now. With *RAID*!

wild hare
COMPUTER SYSTEMS INC

6595 O'Dell Place, Suite M
Boulder, Colorado 80301
voice: (303) 530-2221
fax: (303)530-9637



(*Reduced Aggravation ICOBOL Debugger)
Complete satisfaction or double your bugs back!

Circle 40 on reader service card.

FOCUS

The Magazine of the North American Data General Users Group

Cover design by Ann Soto

FOCUS, the Magazine of the North American Data General Users Group (ISSN 0883-8194) is the official publication of the North American Data General Users Group (NADGUG) in cooperation with Turnkey Publishing.

Editorial and business offices are at P.O. Box 200549, Austin, TX 78720, phone 512/335-2286. NADGUG headquarters are located at NADGUG, c/o Danieli & O'Keefe Associates, Inc., Chiswick Park, 490 Boston Post Rd., Sudbury, MA 01776, phone 508/443-3330.

FOCUS, the Magazine of the North American Data General Users Group is distributed to members of the North American Data General Users Group. Membership fees are $60 per person ($110 outside North America). A one-year (12 issues) subscription to *FOCUS, the Magazine of the North American Data General Users Group*, costs $48. For memberships and subscriptions outside the U.S., add $50 to defray the cost of mailing.

The cost of single copies is $4. Requests to replace missing issues free of charge are honored only up to six months after date of issue. Send request to *FOCUS, the Magazine of the North American Data General Users Group*, c/o Turnkey Publishing.

NADGUG is an independent association of computer users; it is not affiliated with Data General Corporation, nor does it represent the policies or opinions of Data General Corporation. The views expressed herein are the opinions of the authors, and do not necessarily represent the policies or opinions of NADGUG or of Turnkey Publishing, Inc.

Advertisements in *FOCUS, the Magazine of the North American Data General Users Group* do not constitute an endorsement of the advertised products by NADGUG or Turnkey Publishing, Inc.

Volume 9, Issue 8.

NORTH AMERICAN DATA GENERAL USERS GROUP

DATA GENERAL IS OPENING A LOT OF EYES WITH THE BEST OPEN SYSTEMS SERVER IN THE WORLD!

The AViiON 9500– for performance range, high availability, data integrity, storage management and price/performance. It's simply the best open systems server in the world!

AViiON®–The complete open enterprise solution:
Flexibility–Choose the size server you require today.
Scalability–As your needs expand, the AViiON 9500 grows to 1600 MIPS of power.
High Availability & Data Integrity & Storage Management– The combination of AViiON servers, CLARiiON™ storage products and the DG/UX™ operating system is unmatched.
Price/Performance–AViiON is a recognized industry leader.
AViiON–Impeccable credentials:
A $1 Billion customer base in less than four years.
Voted #1 for Customer Satisfaction in Computerworld's "CW Guide to Servers."*
AViiON–The world's best server!
From a company that understands your enterprise computing needs. Call **1-800-DATA GEN**

Data General
Where the World is going!™

Circle 12 on reader service card.

So where is it?

Because some member listings and other information were inadvertently left out of its first printing, NADGUG's awaited 1993 Member Directory is in the process of being reprinted. The corrected directory edition will make its appearance later this year, so watch for it in your mailbox. NADGUG members receive a copy at no additional charge as part of their annual membership fees. The directory indexes NADGUG members by individual last name and also by company name. This year's directory includes Products and Services listings of Data General-related vendors.

In the interest of compiling as complete a profile as possible of NADGUG's membership for this year's directory *and* next year's edition, NADGUG requests you, the individual members, to volunteer information about what kinds of systems you're using—hardware, software, and programming languages. You can also make address changes (see form at the bottom of page 4; fax to NADGUG member services at 508/443-4715 or mail to NADGUG, c/o Danieli & O'Keefe Associates, Inc., Chiswick Park, 490 Boston Post Road, Sudbury, MA 01776). Deadline is September 1.

NADGUG wants to know more about its members

Please fill out the form below and fax to NADGUG member services at 508/443-4715 or mail to NADGUG, c/o Danieli & O'Keefe Associates, Inc., Chiswick Park, 490 Boston Post Road, Sudbury, MA 01776. Deadline is September 1.

Member Name: _____

Occupation Title: _____

Company Name: _____

Address: _____

City: _____

State: _____

Zip: _____

Phone: _____

Fax: _____

Please indicate the operating systems and languages your company currently uses (check all that apply):

- ☐ AOS/VS
- ☐ AOS/VS II
- ☐ Unix (DG/UX)
- ☐ RDOS
- ☐ MUMPS
- ☐ Pick
- ☐ Other: _____

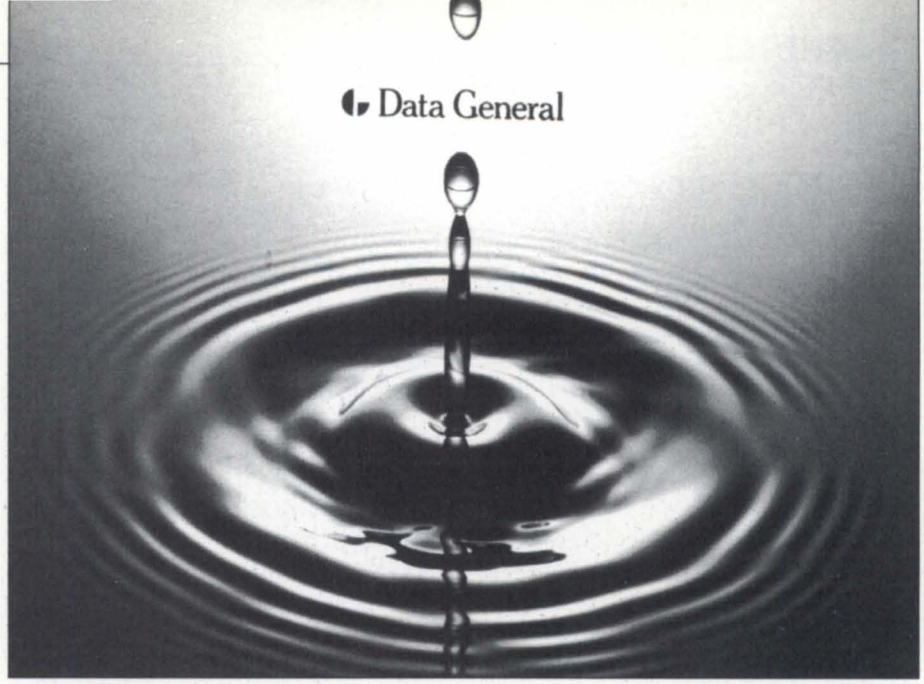
- ☐ Business BASIC
- ☐ ICobol
- ☐ Cobol
- ☐ C
- ☐ 4GLs
- ☐ Fortran
- ☐ Other: _____

Please indicate the Data General systems your company currently uses (check all that apply):

- ☐ Aviion (up to 5000)
- ☐ Aviion (6000 and up)
- ☐ MV (up to 5500)
- ☐ MV (midrange)
- ☐ MV (40000 and up)
- ☐ DG PC
- ☐ 16-bit Eclipse
- ☐ Nova
- ☐ Desktop
- ☐ Clariion

FOCUS
The Magazine of the North American Data General Users Group

New Aviions break previous speed record



Cover art from Data General's "Open Enterprise Servers" announcement

The day the August issue of **Focus** *magazine went to press, Data General was holding a press conference in Boston to announce its new line of Aviion servers. We'll bring you more details in future issues, but here's what we learned on June 29.*

Data General strengthened its position in the open enterprise computing marketplace with the introduction of a new family of Aviion enterprise computers and the addition of advanced features to its DG/UX operating system.

An open enterprise server manages the distribution of all corporate data, providing the filing, the printing, and the communications support for all the personal computers, workstations, and other desktop systems throughout the organization, DG says. It also manages the network.

The 16-processor Aviion 9500 server will be able to process up to 1,600 million instructions per second (MIPS) and up to 1,000 transactions per second (TPS), four times the performance of DG's previous top-of-the-line Aviion system. The eight-processor AV 9500 will provide up to 800 MIPS and 600 TPS performance.

Features like board-upgradeable systems; memory expansion to more than 2 GB; high-performance I/O; con-

nection to a terabyte of storage with Data General's Clariion disk arrays; fault tolerant backup with Clariion tape arrays; and high availability design make the AV 9500 systems suitable for mission-critical applications.

In addition to the high-end servers, DG also introduced two-, four-, six-, and 12-processor versions of the AV 9500, the AV 8500 mid-range server models for the office, and the AV 500 workstation.

AV 9500 prices will range from $84,000 for the two-processor version up to $280,000 for the eight-processor model. AV 8500 pricing begins at $36,000 and ranges up to $56,000. The AV 500 workstation will be priced at $15,000.

The company said that the two- and four-processor versions of the AV 9500, as well as the mid-range AV 8500 servers and AV 500 workstations, will be available immediately. The six- and eight-processor AV 9500 models will be available in the late fall, and the 12- and 16-processor AV 9500 systems will be available next spring.

DG/UX enhancements

The new release of DG's commercial Unix operating system, DG/UX 5.4, adds functions that directly support the needs of an enterprise server.

New high-availability features include a Distributed Lock manager that allows Aviion systems to be configured in clusters to improve applications availability; on-line controller restart; applications-transparent use of back-up LAN controllers; and dynamic bad block disk remapping.

Storage management enhancements include Virtual Disk Management (VDM), allowing customers to dynamically reconfigure disks on-line to enhance performance and availability, transparent to applications execution; and the bundling of Legato's Networker for unattended networkwide back up and recovery.

Support for Posix Threads allows applications with multiple threads of control, such as data base systems, to take further advantage of the Aviion multiprocessor environment. DG/UX 5.4, release 3.0, will ship in the fourth quarter of this year.

Industry support

Several of the industry's leading open enterprise-class software providers were scheduled to join DG at the announcement including Oracle Corporation, Computer Associates, Information Builders, Inc., and Dun & Bradstreet Software. According to Data General, more than 3,000 applications through partnerships with software developers are available on Aviion systems. Δ

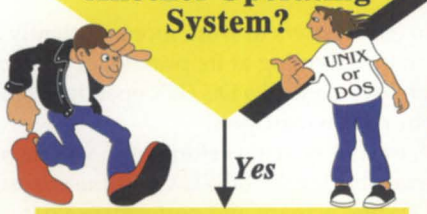
In General is compiled by Robin Perry. If you have an item for In General, please send it to Robin, c/o Focus magazine, P.O. Box 200549, Austin, TX 78720; 512/335-2286; Fax: 512/335-3083.

ATTENTION CEO USERS!

EAGLE Software Has Got You Covered!

Yes

Moving to Another Operating System?



No

Yes

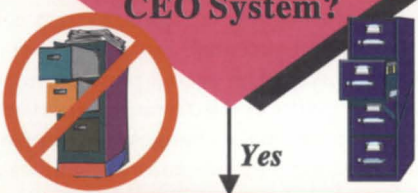
CABINET_CONVERTER™

Migrate CEO Files to UNIX® or DOS®

New!

- Allows mass conversion of CEO files to any word processor or spreadsheet supported by CEO_DOC_CONVERT or KEYpak.
- Moves information stored in the CEO document summary to the summary fields in WordPerfect.™
- Perform one mass migration or migrate in stages to meet users' needs.
- Flexibility allows customization to meet your specific migration needs.

Need to Optimize Your CEO System?



No

Yes

MAID™ for CEO®

CEO Optimization Utilities

- Recovers wasted disk space by the tens of thousands of blocks.
- Provides full-time notification of new CEO mail and reminders even if you're not in CEO.
- Helps avoid document loss and keeps CEO_INDEX errors from corrupting your files.
- Allows operations on the CEO filing system not normally available through CEO.
- Supercharges the performance of the CEO mail system.

Tight On Disk Space?

No

Yes

New!

DOCtAR™

DOCument ARchiver and Compressor for CEO

- Compresses and archives CEO documents up to 70% to any directory.
- Allows immediate restores by CEO users as a CEO user application or CLI program.
- Runs in conjunction with, or in place of, the CEO_ARCHIVER.
- Includes all the compression software you need at one low price.

Also from EAGLE...

FILE_MONITOR™
A superior file usage monitor and disk balancer.

DISK_PAK™ OnLine!™
Optimize your disks without shutting down your MV.

VS_TOOLBOX™
Powerful tools for MV management, optimizing INFOS®, and system security.

RUSH™
Now you can FAX documents directly from your MV terminals!

DISK_PAK™
The premier disk optimization product for MVs.

Call Today!

1-800-477-5432

EAGLE Software, Inc.
123 Indiana Avenue
Salina, KS 67401
Phone: 913-823-7257
FAX: 913-823-6185

Circle 16 on reader service card.

The System Activity Reporter

SYNOPSIS

The author begins a series of articles explaining the performance monitor utilities available for solving most common bottlenecks affecting DG/UX performance in a multi-user server environment.

by Thomas E. Soukup
Special to Focus

The goal of using any computer system is to enable users to work more efficiently and achieve their fullest potential. But if a system isn't performing at its peak level, corrective action must be taken. The overall performance of Data General's DG/UX operating system and the Aviion platform is extremely important to system managers.

To monitor and make changes that improve overall system performance, you must be familiar with performance-monitoring utilities available for the DG/UX multi-user system. Moreover, you must not only be able to collect system activity and performance data, but you should also be able to analyze the data and implement corrective action. Data General has included the following performance-monitoring and tuning utilities with DG/UX: System activity reporter (*sar*) utility; user application profilers; kernel profiler; highly tunable fast-file system.

To begin monitoring and tuning a system or applications, you have to know what resources users and applications are consuming. You must understand the availability and physical limitations of these resources, and how best to allocate them among users and applications. Excessive or even new demands on an already heavily loaded resource can significantly affect overall system performance. Often the solution is to modify a DG/UX kernel parameter, distribute data over more disks and controllers, rebuild data files to eliminate fragmentation, or redesign applications to consume fewer resources or use more efficient calls. However, sometimes more CPU power, more memory, or more disks and controllers must be added to the system to accomplish acceptable performance improvement.

Sar: the system activity reporter

The Unix tool *sar* (system activity reporter) is the front-line tool in any performance investigation. It provides enough information to direct you to the system resources causing performance bottlenecks. Hardware bottlenecks occur within the CPU, the memory subsystem, disk I/O subsystem, terminal I/O subsystem, network I/O subsystem, or a combination of two or more. Software bottlenecks occur within the operating system, relational data base, multi-user and batch applications, or a combination of two or more.

The Systems Evaluation and Performance Analysis Center (SEPAC) monitoring tool, the "Real-Time Performance Monitor" (UX/RPM) displays *sar* data in screen format. Screens logically group key *sar* statistics by functionality. UX/RPM provides users with current and historical data to track performance bottlenecks. UX/RPM simplifies the user's job. A single program collects critical performance data, thus eliminating the need to use

numerous DG/UX commands and utilities.

The AV SysScope performance monitor is a set of performance measurement tools for Aviions running DG/UX. It allows you to monitor, collect, log, and display the performance of one or more computers. The tool set consists of two graphical monitors: *sscope* for analyzing system performance (*sar* data), and *sscope-ps* for analyzing user-process performance (the Unix command "Process Status", *ps*). For each *sar* option, the following columns and descriptions are included: The significant data item that *sar* reports; the significant data item that the real-time performance monitor (UX/RPM) displays on its screen; the significant data item that the AV SysScope performance monitor displays on its screen; a brief explanation of each data item; some general performance guidelines; additional features included in UX/RPM and AV SysScope not reported by *sar*.

Many of the following items can also be obtained by using the *dg_sys_info()* system call. For further information, refer to the on-line manual pages for *sar* and the DG/UX system header file, *dg_sys_info.h*.

CPU activity (*sar -u*). The *-u* option lists CPU utilization. At any given time the processor is either busy or idle. When busy, a processor is either in user or system mode. When idle, the processor is either waiting for the completion of input/output or some other resources wait operation (such as a lock), or it has no work to do.

Figure 1: The *sar -u* option

***sar* item**	**UX/RPM item**	**SysScope item**
%user	User	Percent User Time

The percentage of time spent executing instructions in user space.

***sar* item**	**UX/RPM item**	**SysScope item**
%sys	Sys	Percent System Time

The percentage of time spent executing instructions in the DG/UX kernel, excluding time spent handling interrupts in user processes.

***sar* item**	**UX/RPM item**	**SysScope item**
%idle	Idle	Percent Idle Time

The percentage of time spent in the idle state. Idle time is time accumulated by a special kernel idle process that runs when there is nothing else for the system to do. Idle time represents unused CPU time.

Guidelines for CPU activity. The percentages of user, system, and idle time change based on what percentage of system mode calls versus user mode calls are being used by applications. Values for system, user, and idle time change based on the applications run. For example, applications using semaphores and messages (kernel space routines) will show a higher system time than applications not using semaphores and messages. For most system that don't use excessive kernel calls, the average of user mode calls will range from 60 to 75 percent. The average system mode calls will range from 15 to 25 percent. Idle time is normally 10 to 20 percent, unless it's a heavily loaded system. DG/UX will attempt to keep its processors completely busy when there are at least as many users as processors working on the system.

BACKUP 50 GB IN HALF THE TIME.

Two 8mm tape drives, each capable of storing up to 25 GB, can be combined in a single desktop cabinet. This dual configuration lets you take full advantage of our Advanced SCSI Processor.

Now, two drives can work together simultaneously to maximize backup speed and capacity with a variety of recording modes.

STRIPING MODE. In striping mode, you can write data to two tapes at once, in alternate blocks–doubling speed and capacity. In fact, you can back up 50 GB overnight this way.

MIRRORING MODE. In mirroring mode, you write the same data to two tapes at the same time. This is ideal for users who want to keep a backup off-site, for extra security, or for users who need to exchange data with other sites.

OFF-LINE COPY. An off-line copy function lets you copy the contents of one tape to another–without tying up the host. Off-line verify ensures that the tapes are identical.

CASCADE MODE. In cascade mode, data spills over to the second tape when the first tape is full.

FLEXIBILITY IS ONLY HALF THE STORY. Our data compression option is the fastest available. And it's switch-selectable, so you can still read and write standard 8mm tapes. Bright backlit status displays give you complete drive status information: recording mode, transfer rate, compression ratio,tape remaining and more.

Best of all, you get the advantages of 8mm helical scan technology–reliable, unattended operation and the best price-performance on the market.

Plug compatible with virtually every computer system, the drives are fully backed by our 12-month warranty that includes support from our in-house engineering staff.

For information, call today at

(804) 873-9000

CONTEMPORARY CYBERNETICS

TRUE PLUG COMPATIBILITY

Alliant	Convergent	DEC Unibus	ICL	Novell	PICK	Stratus
Alpha Micro	DataGeneral	Gould/Encore	Intergraph	OS/2	Plexus	Sun
Altos	DEC SCSI	HP	Macintosh	PS/2	Prime	Texas
Apollo	DEC BI-Bus	IBM AS/400	McDonnell	Parallel Port	Pyramid	Instruments
Arix	DEC DSSI	IBM Mainframe	Douglas	PC 386/ix	Sequent	Unisys
AT&T	DEC HSC	IBM RISC/6000	Motorola	PC MS-DOS	Silicon	Ultimate
Basic-4	DEC Q-Bus	IBM RT	NCR	PC Xenix/Unix	Graphics	Wang
Concurrent	DEC TU/TA81	IBM S/38	NeXT	Pertec	STC	*and more*

Rock Landing Corporate Center • 11846 Rock Landing • Newport News, Virginia 23606 • Fax (804)873-8836

Circle 9 on reader service card.

Features of UX/RPM and AV SysScope for CPU activity. "Load averaging" is a method of summarizing system performance. It is defined as the average number of processes in the kernel's eligible queue over a given time. Load averages are shown over the previous 1-, 5-, and 15-minute intervals. By comparing averages over the three intervals, it is possible to note whether the load is climbing or falling. If the 5-minute average is greater than the 15-minute average, and the 1-minute average is greater still, the load would generally be considered to be increasing. A falling average would usually mean that the load is decreasing. UX/RPM and AV SysScope both report the number of processors active as well as the ability to display *%usr*, *%sys*, and *%idle* by processor.

Kernel table management (*sar -v*). The *-v* option reports kernel table usage by processes, inodes, files, and locks.

Figure 3: The *sar -v* option

sar item	UX/RPM item	SysScope item
proc-sz	Processes	Process Table Size

The number of processes existing in the system divided by the maximum number of user processes that are allowed to exist at one time in the system. This number is a static kernel-configuration variable, NPROC.

sar item	UX/RPM item	SysScope item
inode-sz	Inode Table Size	Inode Table Size

The number of inodes opened/cached, closed/cached table entries currently being used or allocated in the kernel. This number is dynamically allocated.

sar item	UX/RPM item	SysScope item
file-sz	File Table Size	File Table Size

The number of object pointers in use. This number is roughly equivalent to the number of files in use, but also includes object pointers to such things as sockets, pipes, and local and *nfs*-mounted files.

sar item	UX/RPM item	SysScope item
lock-sz	Lock Table Size	None

The number of shared memory data segments in use. DG/UX does not report this value.

Processor Queue Activity (*sar -q*). The *-q* option reports the average queue length while the queue is occupied, and the percentage of time the queue is occupied.

Figure 4: The *sar -q* option

sar item	UX/RPM item	SysScope item
runq-sz	Bound Runnable Processes	Bound Runnable Processes

The number of bound runnable processes that are ready to run. This may include processes that are not eligible through no action of their own, but because the medium term scheduler (MTS) has made the process ineligible to reduce system load.

sar item	UX/RPM item	SysScope item
%runocc	none	none

The percentage of time the run queue is occupied. DG/UX always reports 100.

sar item	UX/RPM item	SysScope item
swpq-sz	Unbound Runnable Processes	Unbound Runnable Processes

Processes that could run, but have no virtual processors to run on.

sar item	UX/RPM item	SysScope item
%swpocc	none	none

The percentage of time the swap queue is occupied. DG/UX always reports 0 or 100 depending on whether the system is swapping or not.

Guidelines for processor queue activity. When "Unbound Runnable Processes" is non-zero, the system is severely loaded and NPROC is set too low. Processes must be bound before they can run; therefore, bound processes are able to run at lower cost than unbound processes, which must first be bound to a virtual processor. The number of bound processes is limited by the number of virtual processors, determined by the static DG/UX configuration variable NVPS, or if it is not set, a combination of the configuration variable NPROC and the amount of system memory.

Features of UX/RPM and AV SysScope for processor queue activity. Lists the number of processes bound to a virtual processor at the time of the sample.

Memory activity (*sar -r*). The *-r* option reports the number of memory pages (in 4 K pages) and swap file disk blocks (in 512-byte blocks) that are currently unused.

Figure 5: The *sar -r* option

sar item	UX/RPM item	SysScope item
freemem	Free Mem (frms)	Free Memory Frames

The number of free memory 4 K pages in the free memory pool, including pages in memory objects that are not mapped by bound processes, and pages of non-open files. Freemem is dynamically determined by an algorithm in the DG/UX kernel.

sar item	UX/RPM item	SysScope item
freeswap	Free Swap	Reserved Anonymous Pages

The number of free-disk, 512-byte disk blocks that have been reserved, but not necessarily accessed. Applications reserve swap pages for their uninitialized or bss data sections, for head space when they call *malloc* or *brk*, and for stack frames.

Guidelines for memory activity. Due to the current implementation of DG/UX (5.4), it is nearly impossible to determine how much free memory is truly available. In DG/UX (5.40), all the system's physical memory can be used

as an I/O buffer for user applications. Any system that has been running for a long time may have "used" all the physical memory for something. The *freemem* statistic is an attempt by DG/UX to estimate how much of the current memory can be used for something else without paying too high of a cost (such as swapping out some code or data). So, even if *freemem* drops very low (below 300, roughly), much of the memory could be easily be reused. For example, I/O buffers that were used, that have been flushed to disk, but have not been returned to the free memory pool are a very common source of memory. However, if one is seeing *freeswap* change often (especially downward), then you are definitely running out of memory. The solution is either to add more physical memory to the system, or reduce the number of users allowed on the system at one time. In the case of some relational data base products, you should reduce the amount of shared memory buffers used by that product.

Features of UX/RPM and AV SysScope for memory activity. AV SysScope reports the number of pages of physical memory and on the swap device that may be used for virtual address space. This item is listed as Total Anonymous Pages in AV SysScope. UX/RPM reports the total amount of physical memory configured in the system. This item is listed as "Physical Mem." in UX/RPM.

Memory swapping and switching activity (*sar -p*). The *-p* option reports swapping and switching activity.

Figure 6: The *sar -p* option

sar item	UX/RPM item	SysScope item
vflt/s	Hard Faults /Sec	Hard Page Faults /Sec

The average number of page faults that resulted in physical I/O (valid page not in memory) per second. To satisfy a *vflt*, a physical disk read is necessary.

sar item	UX/RPM item	SysScope item
pflts/s	POW Faults /Sec	Copy of Write Page Faults/Sec

The average number of protection faults per second. Generally, this value indicated the rate of "Copy On Write" (COW) faults when a forked process touches an unshared page from the parent process. Protection fault rates normally increase as fork rates increase.

sar item	UX/RPM item	SysScope item
pgfil/s	Fill Faults /Sec	Fill From Page Faults/Sec

The average number of hard page faults satisfied by a page in from the file system per second. This value includes those program text pages pulled from the program file on disk, and all mapped files, whether by *exec* or *mmap*. Generally, page fill fault rates above 10 per second should be investigated as to their cause. If the number of page-fill faults is equal to or just less than the total number of virtual faults, excessive disk faulting of program page is likely.

sar item	UX/RPM item	SysScope item
rclm/s	Reclaims/Sec	Frames Purged/Sec

The average number of frames freed for use by the frame purger daemon per second.

Guidelines for *vflt/s* and *rclm/s*. A high *vflt/s* value is an indication that DG/UX may be paging out process pages (program code and data). Interactive users may then notice long pauses in response time. If applications are accessing large files, the solution may be to decrease PERCENTBUF. This kernel parameter controls how much data are allowed to be cached up in memory.

A high *rclm/s* value that is consis-

tently in the low hundreds indicates the system is experiencing memory contention. When the frame-purger daemon has to start reclaiming hundreds of frames a second for an extended period, system performance will be poor. The system will start reducing the number of virtual processors it will allow to be bound.

The solution is to either add more physical memory or reclaim memory from either shared memory application buffers or use less memory per application. By redesigning the applications or reducing the number of users allowed on the system at one time. Possible kernel parameters that may assist in decreasing these values are MAXBUFAGE and PERCENTBUF (5.4.2). Decrease MAXBUFAGE from 60 to 30; this will cause more memory flushing of data but may increase overall performance. Decrease PERCENTBUF (5.4.2) from 100 percent to 50 percent or less. This will ensure executables remain in memory while flushing out more cached data files. Other kernel parameters that may need fine-tuning are MAXPAGEOUTS, MAXSLICE, and HOGFILESIZE.

MAXPAGEOUTS is the maximum concurrent pageout I/O operations that the system can have pending at one time. For higher throughput, set MAXPAGEOUTS to 3 times the number of disks your system has. MAXSLICE is the maximum time (in milliseconds) a user process can run before being suspended. My default MAXSLICE is set to 500; however, most processes are suspended long before the 500 milliseconds have passed, due to a resource wait. In some multi-user environments, reducing MAXSLICE to 250 can increase overall processing.

However, in most cases the default is fine. HOGFILESIZE is the maximum number of bytes of physical memory that can be used by a given file before that file will be treated unfavorably for physical memory resource allocation. By default, HOGFILESIZE is set to 1/4 MB. In some multi-user environments, reducing HOGFILESIZE can help to free memory to keep executables in memory. **Swapping and Ppro switching activity (*sar -w*).** The *-w* option reports swapping and switching activity.

Figure 7: The *sar -w* option

sar item	UX/RPM item	SysScope item
swpin/s	Process Binds/Sec	Process Binds/Sec

Rate at which the processes are being bound to virtual processors, usually caused by fork calls. Binding and unbinding incur overhead and, if they persist, result in impaired system efficiency. A process that has been unbound will be bound only when it is runnable. See "Fork System Calls/Sec and Unbinds/Sec."

sar item	UX/RPM item	SysScope item
bswin/s	Frames Purged	User Page Faults

The average number of user page faults per second.

sar item	UX/RPM item	SysScope item
swpot/s	Process Unbinds/Sec	Unbound Processes/Sec

Number of processes that are not bound per second.

sar item	UX/RPM item	SysScope item
sbswot/s	Bound Frames Purged	Bound Frames Purged/Sec

Number of purged pages that were mapped by bound processes.

sar item	UX/RPM item	SysScope item
pswch/s	Process Switches/Sec	Process Switches/Sec

Number of times processes are switches onto a physical processor to run.

Guidelines for *swpins/s* and *swpot/s*. High values for these *sar* items can indicate that user processes are issuing a high amount of system requests with little user code in between. In other words, they are not staying resident on a virtual processor due to their own behavior. The solution is to profile suspect applications, and modify and correct their design or reduce the kernel parameter MAXSLICE. Normally, the *swpin/s* rate and the *fork/s* rate should be very close. If the *swpin/s* rate exceeds the fork rate, the number of virtual processors is being limited in some way. The solution may be to increase the kernel parameters NPROC or NVPS.

Guidelines for *bswots/s* and *swpot/s*. Purging bound pages is expensive because bound processes will likely run again soon and require access to purged pages, resulting in hard page faults. A persistent non-zero value of this item may indicate insufficient memory or that PERCENTBUF is set too high. The *swpot/s sar* item indicates the rate at which processes are being unbound from virtual processors, caused by process exits. Binding and unbinding incur system overhead and, if they persist, will result in lowered system efficiency.

Guidelines for *pswch/s*. Unneces-

Continued on page 35

David Novy

Gone fishin'

SYNOPSIS

It was busy weekend for our columnist—catching a new disk drive, decommissioning some venerable MVs, and wrestling with adapters and an uncooperative RAID 7 box.

This weekend I decided to do some fishing. My home state of Minnesota is famous for its walleye, northern pike, and bass. For the more adventurous types there is the muskie. To catch a muskie it is said that one must be willing to make on the average more than 10,000 casts. But I was not on the trail of the usual famed fish of Minnesota this weekend. I was after a type of fish that had only rarely been seen in Minnesota before, and then only recently. I spent the weekend on the trail of the elusive Barracuda II. I'm happy to report that by the end of the weekend I had caught my limit of five.

Okay, so it wasn't a fish I was after, but instead a new disk drive. The Barracuda II is a new Seagate disk drive. It was designed and is manufactured in Minnesota. It is 2.1 GB in a 3.5-inch form factor, with an average seek time of less than 9 milliseconds. It runs at 7200 RPM and can read and write data at 2 MB per second. All I remember was hearing the 1.0 GB 3.5-inch form factor drive was on the way and suddenly, the 2.0 GB, 3.5-inch form factor drive appeared.

The arrival of this drive portends the end of the 5.25-inch disk drive era. The 5.25-inch drive is simply too heavy and hogs too much space to remain competitive.

There are two caveats when using the new Barracuda II. First, check the packaging before throwing it away. The Barracuda package with and without a drive weighs about the same. Second, although the Barracuda runs cooler than most 5.25-inch drives (~25 watts), it runs hotter than most 3.5-inch drives—10 watts idle and 15 watts in normal use.

Be sure to have adequate ventilation when using this drive. The drive is currently on allocation, but it should be

Further adventures with DOS

by David Novy

Recently I detailed my pioneering adventures with the new Microsoft DOS 6.0. Anyone who has read the trade literature lately will readily notice that I was not the only pioneer who suffered injury while trying to switch to the new operating system. I am running MS-DOS 5.0 once again.

The memory manager from DOS 6.0 was okay. However, I have since learned that for my normal work, DOS 5.0's "HIMEM.SYS" works just fine. And as far as disk compression goes, I think it isn't worth the potential grief. In addition, I have read that the first versions of Microsoft Windows NT will not support the DOS 6.0 disk-compression algorithm.

Therefore, anyone wanting to upgrade to Windows NT from a DOS 6.0 system with compressed disks will have to *un*compress the DOS 6.0-compressed disks before Windows NT can be installed.

Uncompressing a DOS 6.0-compressed disk is a venture only for the patient and the strong of heart and mind. It is *not* a simple process. I have continued using the Norton Desktop for Windows 2.2. It was written for MS-DOS 6.0, but on my system it works with DOS 5.0.

As long as you don't use the DOS 6.0 disk-compression or memory management algorithms, you shouldn't have any problems with it. I prefer using the Norton Desktop instead of the Windows "PROGMAN" because the Norton program is more user-friendly. Δ

available in quantity very soon. Cost of the new Barracuda II drives should be less than $2,600 each.

Good-bye, old friend

I retired my last MV machines this weekend. The decommissioning of the two remaining MV/10000s marks the end of an era. Both machines had a service life in excess of 9 years. I doubt that few machines in use today will be able to make that claim.

The MV/10000 was one of the finest machines that Data General ever produced. In more than 9 years of service I never had a CPU board failure. Power glitches never seemed to bother it. But their continued existence could no longer be cost-justified, so now they belong to the ages.

Disk controller troubles

I tried connecting a Data General Model 7430 VME SCSI-2 adapter (VSA) to a Storage Computer RAID 7 box over the weekend. I was really looking forward to having high-speed disk access from the RAID 7 box to an Aviion 6225. Unfortunately, the Model 7430 controller and the RAID 7 box appeared to have handshaking problems. (The ATTN line was not dropping when it should have.)

This is unfortunate since the StorComp RAID 7 box appears to run on just about anyone else's SCSI controller including the Data General SCSI-1 controller. It was also quite puzzling since the DG VSA controller and the RAID 7 box both use the NCR 53C710 I/O processor.

If anyone has any ideas why this problem is occurring, please get in touch with me. In the meantime, if you want to connect the StorComp RAID 7 box to a DG Aviion, you will either have to use the DG SCSI-1 controller or purchase a third-party controller, such as the new Ciprico new dual-channel fast SCSI-2 controller. Δ

David Novy is a technical computer specialist at 3M in St. Paul, Minnesota. He is past chairman of the AOS/VS special interest group, and current chairman of NADGUG's SIG/UX.

Brian Johnson

Time problems

SYNOPSIS

BJ ponders time and how, despite the direst of warnings, the turn of the the millenium just a few years hence will break most of the computer systems in the world. But you can do something about it.

I've been thinking about time a lot in the past few months, so I'm gonna let you in the results of my work. I first got interested in the subject of time about two years ago when I attempted to port the ANSI C functions in P.J. Plauger's book, *The Standard C Library*, to DG C. It culminated about six months ago when I was hired to design a message-passing protocol for a local account's Emergency Services System.

The EMS system involves a bunch of disparate systems connected by an existing TCP/IP network. The plan is to replace most of the voice-radio traffic associated with ambulance dispatching with direct computer-to-computer communication. So what's all this got to do with time and with the non-emergency systems that most of you are involved in? Well, if your systems are a) networked with other systems (whose aren't these days?), or b) involve time-stamped data, or c) experience problems twice each year when local time changes to and from Daylight Savings Time, then there's something of interest to you here.

I'm gonna make a prediction: on December 31st, 1999, most of the computer systems on this planet are gonna break. And they'll break again on December 31st, 2000, because you'll just apply chewing gum and paperclips to get through the first round of problems temporarily. After all, you'll have 12 months to get the real fixes in, right?

Here's a test: Go check your VSGEN dialogue file and see what the GMT offset is. 0:00? You and most other systems. With a little thought right now you could have six or seven years to get it right and party without interruption on New Year's Eve, 1999.

:TIME:BASICS

Volumes have been written on the quirky way that time has been handled in the past (see the references at the end of this column for a sample), but I'm going to simplify life considerably by restricting this discussion to the period after 1987 in North America, with some appropriate asides to our cousins in Europe and the Southern Hemisphere about differences that they should keep in mind.

First of all, we need to define some jargon. "Local time" refers to the time at your current location on the planet. "Calendar time" refers to the time on Planet Earth. "Time difference" is the absolute (i.e., positive) difference between two local or two calendar times. "Time-of-day" is the time difference between a local or calendar time and the previous local or calendar Midnight (e.g., 15:30:14, by itself). The word "time" used in any other context refers to a local time, a calendar time, or a time difference. Local time must include the date, the time-of-day, the location involved, and whether or not Daylight Savings Time is in effect.

For example, as I write this it is 11:15 PDT, June 10th, 1993. Luckily for us, instead of specifying my actual latitude and longitude in San Francisco, CA, it's sufficient for me to just name the time zone that San Francisco is located in. That's because someone agreed a long time ago to break the planet up into time zones and treat all sites within a zone as the same location (except for the Soviets, who declared the entire country to be one time zone). And yes, Virginia, there are more than 24 time zones, and some of them involve fractions of an hour. Calendar time also includes the date, time-of-day, and location. In this case the location is an exact location on the planet: a marker at the cute little observatory in Greenwich, England, situated on a picturesque hill near London with a spectacular view of The City, weather permitting. The time at Greenwich used to be called Greenwich Mean Time (GMT), but has since been renamed to Universal Time—Coordinated (UTC).

Calendar time has no notion of Daylight Savings Time, and therein lies its chief benefit; it inexorably increases with only occasional small bumps (more about that later). In North America since 1987, the agreed-upon rules for local Daylight Savings Time have been that it starts at 02:00 on the last Sunday in March and ends at 02:00 on the last Sunday in October. The memory aid for which way the time jumps on both Sundays is, "Spring forward, Fall back."

Because I've limited the discussion here to future times, we don't have to worry about how it worked prior to 1987. Here's what the transition to and from DST would should look like if you were watching the AOS/VS time-of-day clock on my system in San Francisco, and the transition occurred as it should:

Starting	Stopping	UTC
01:59:58	01:59:58	09:59:58
01:59:59	01:59:59	09:59:59
03:00:00	01:00:00	10:00:00
03:00:01	01:00:01	10:00:01

There are two interesting things to note here: local times-of-day between 02:00 and 02:59:59 on the last Sunday in March are impossible, and local times-of-day between 01:00:00 and 01:59:59 on the last Sunday in October are ambiguous.

:TIME:PROBLEMS

Because time and time arithmetic are so complicated, most computer manufacturers and software suppliers have messed them up badly. DG is no exception, but they're in good company; Microsoft blew it on MS-DOS and their C libraries, and even the renowned P.J. Plauger got it wrong in the source code provided with his book, *The Standard C Library*. Only the design-

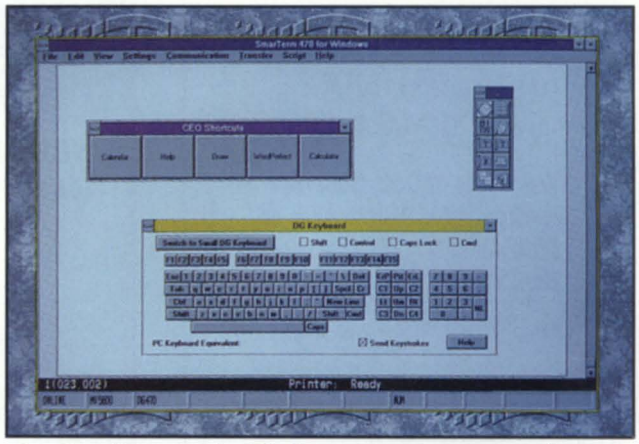
Discover the basic principle of connectivity ...

Defy the gravity of your connectivity problems.

It's a basic principle worthy of Newton:
When the complexities of multiple platforms weigh you down, discover superior PC-to-host connectivity with new SmarTerm®470 for Windows.

Take emulation to new highs.
SmarTerm 470 for Windows provides PC users with the only full-featured, precise Dasher 470C terminal emulation software for Windows with connectivity to MV and AViiON hosts. Whether you need emulation for text or graphics, SmarTerm connects on the highest levels.

Break free under your own power.
Escape from high internal support costs with the ease of Windows and SmarTerm's pioneering corporate support tools and utilities, including simplified graphical keyboard remapping, a pop-up virtual DG keyboard, an on-screen toolbox, user-definable button palettes, powerful scripting, scrollback, and a customizable help system. You'll reduce calls to the help desk and standardize your site's connectivity.



New SmarTerm 470 for Windows corporate support tools.

Discover SmarTerm.

SmarTerm 470 for Windows is the only emulation software that includes SmarTerm TCP/IP FREE. Get Windows Sockets TCP/IP and more, at no extra cost! Plus get drag-and-drop FTP, XMODEM, YMODEM, ZMODEM, and Kermit file transfer protocols. Discover the basic principle of connectivity today. Call **1-800-EMULATE.**

Persoft U.S.A. 465 Science Dr. • Box 44953 Madison, WI 53744-4953 • Phone (608) 273-6000 • Fax (608) 273-8227

Persoft Europe World Trade Center • Beursplein 37 • Box 30237 • 3001 DE Rotterdam, The Netherlands
Phone +31 10 405.3560 • Fax +31 10 405.5073

persoft®
CONNECTIVITY SOLUTIONS
DOS • Windows • Ethernet • Token Ring

Circle 27 on reader service card.

Figure 1: DSTMON.CLI

```
COMMENT Need to re-assemble and link DSTMON?
[!EQUAL,[!PATHNAME DSTMON.PR],]
 COMMENT Yep.
 XEQ MASM DSTMON
 XEQ LINK DSTMON
[!END]

[!EQUAL,[!PATHNAME DSTMON.PR],]
 WRITE ERROR: DSTMON could not be built.
[!ELSE]
 COMMENT Fire that mutha up.
 PROCESS/NAME=DSTMON/PRIORITY=2/RESIDENT DSTMON
 PAUSE 3.000
[!END]
```

Figure 2 - DSTMON.C

```
/*****************************************\
!* DSTMON - Daylight Savings Time MONitor *!
\*****************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "sys_dadid.h"
#include "sys_send.h"
#include "sys_time.h"
#include "sys_wdelay.h"

P_GTIME gtime_pkt; /* ?GTIME packet */
int dads_pid;/* Just what it says */
int dst_on; /* DST on time as MMDD020000 */
int dst_off;/* DST off time as MMDD020000 */
int local;    /* Local time as MMDDHHMMSS */

/* Send a message to Daddy. */
void send(char *msgp, int t) {
     char buf[256];
     int      bufl,ier;

     sprintf(buf,
          "(DSTMON) %s on %d/%02d at %02d:%02d:%02d\n",
          msgp,
          t / 100000000,
          (t / 1000000) % 100,
          (t / 10000) % 100,
          (t / 100) % 100,
          t % 100);
     bufl = strlen(buf);
     if (ier = sys_send(dads_pid,buf,bufl)) exit(ier);
}

/* Wait a while. */
void wait(double secs) {
     int ier,ms;

     ms = secs * 1000;
     if (ier = sys_wdelay(ms)) exit(ier);
}

/* Get local time as MMDDHHMMSS. */
void get_local(void) {
     int ier;

     gtime_pkt.time_pkt_pkt_id = $TIME_PKT_PKTID;
     gtime_pkt.time_pkt_func = $TIME_PKT_LOCAL;
     if (ier = sys_gtime(&gtime_pkt)) exit(ier);
     local = gtime_pkt.time_pkt_month;
     local = local * 100 + gtime_pkt.time_pkt_day;
     local = local * 100 + gtime_pkt.time_pkt_hour;
     local = local * 100 + gtime_pkt.time_pkt_minute;
     local = local * 100 + gtime_pkt.time_pkt_second;
}
```

Continued on page 20.

ers of Unix seem to have gotten it mostly right. What else would you expect from a bunch of PhDs at a facility that was essentially a physics laboratory? These guys were already accustomed to using precision clocks based on GMT for timing experiments. Our needs as programmers and users are much simpler. We need to know what the current local time is, if time *t1* is earlier or later than time *t2*, and by how much. Even with these modest requirements, we've got a couple of problems. Let's take a look at them and some possible solutions.

:TIME:PROBLEMS:1

What do you do to change your AOS/VS time-of-day clock on the last Sundays in March and October? If you're like most people you change it on Monday morning when you arrive at work. I used to do that too. And until just a few years ago the last Sunday in October presented a sticky problem; you had to reboot in order to change the local time backward, otherwise EXEC would crash with an arithmetic fault when computing an elapsed time difference.

DG finally fixed that a few revs ago, but now instead of a crash you just get an absurd elapsed time logged to :SYSLOG for all processes whose lifetime spanned the change and was less than the time difference involved in the change. Duh. A fancier solution is to queue up a batch job with /AFTER to handle the chore at 02:00, or as close to it as possible. Tacky, but it pretty much works. The ideal solution would be for DG to handle the chore inside the operating system with a little assistance from the VSGEN'er who would supply the rules for when DST starts and stops.

Since that last solution is out of our hands, we'll have to go with the tacky batch job, or at least something reasonably similar. It sounds pretty simple, but as usual, just as with coding a binary search program from memory, doing it right the first time is filled with pitfalls. Speaking of coding a binary search from memory, that's a great bar bet in any bar in Silicon Valley. Just keep a copy of Knuth in the trunk of your car to settle the inevitable disputes. The most obvious problem with the batch job scheme is that a batch job queued up to run at 01:59:59 will not actually run until sometime after 01:59:59, either because the batch streams were already running other jobs and this job had to wait its turn, or because it takes a few seconds for the batch job to "log on" before it gets around to executing the first step of the job. Clearly, we've gotta use a program that starts a bit earlier and then keeps a sharp eye on the clock so it can perform the switch precisely during the one-second interval between 01:59:59 and 02:00:00. The solution I use is to PROC up a program under PID 2 that does exactly that and then it terminates.

A less obvious problem is which of the myriad system calls should be used. You've got five to pick from: ?GDAY/?GTOD, ?ITIME, and ?GTIME/?NTIME.

?GDAY/?GTOD are calls left over from the days of RDOS, and have the problem that by the time you do the ?GTOD call the date you got from the ?GDAY call may have changed. ?ITIME gets around that problem, but it's a read-only call. ?GTIME/?NTIME get around both of these problems by returning or setting the date, time-of-day, and UTC offset all in a single atomic call. Now for the really obscure problem: AOS/VS actually keeps track of time-of-day using local time, so changing the local time-of-day will screw up any programs that use UTC instead of local time. As a result, when changing the time-of-day clock we also need to adjust the UTC offset in the opposite direction at the same time. For example, here in San Francisco I've VSGEN'ed my system with a UTC offset of -8:00 per the DG documentation. But when DST is in effect, my actual UTC offset is -7:00.

Because this program has to catch the operating system during that critical second between 01:59:59 and 02:00:00, it needs to be running continuously. The design I use involves using a program that you PROC up under PID 2 in the system UP macro to do the job. It just checks each second to see if it's time to start or stop DST. After DST ends, it recomputes the dates for the following year and patiently waits for them. This means that theoretically you don't even have to reboot your system at least once a year.

Figure 2 continued from page 18.

```
/* Return the date of the last Sunday for a given */
/* month and year as MMDD020000. */
int last_sun(int d, int m, int y) {
     int dgsun,ier;

     if (ier = sys_fday(d,m,y-1900,&dgsun)) exit(ier);
     dgsun -= dgsun % 7;
     if (ier = sys_cday(dgsun,&d,&m,&y)) exit(ier);
     return (m * 100000000 + d * 1000000 + 20000);
}

/* Determine DST start/stop. */
void init_dst(int year) {
     dst_on    = last_sun(31, 3,year);
     send("DST starts",dst_on);
     dst_off = last_sun(31,10,year);
     send("DST ends",dst_off);
}

/* The business end of this thing. */
main(int argc, char *argv[]) {
     int dt,ier;

     /* Get Daddy's PID. */
     if (ier = sys_dadid(-1,&dads_pid)) exit(ier);

     /* Get the program start time. */
     get_local();

     /* Determine DST on/off times. */
     init_dst(gtime_pkt.time_pkt_year);

     /* Check for impossible time. */
     dt = local - dst_on;
     if (dt >= 0 && dt < 10000) {
          send("Impossible time",local);
          exit(EXIT_FAILURE);
     }

     /* Check for ambiguous time. */
     dt = dst_off - local;
     if (dt >= 0 && dt < 10000) {
          send("Ambiguous time",local);
          exit(EXIT_FAILURE);
     }

     Need to adjust UTC offset for DST?
     if (local >= dst_on && local < dst_off) {
          gtime_pkt.time_pkt_pkt_id = $TIME_PKT_PKTID;
          gtime_pkt.time_pkt_func = $TIME_PKT_LOCAL;
          gtime_pkt.time_pkt_zone_hour++;
/*        if (ier = sys_ntime(&gtime_pkt)) exit(ier); */
          send("UTC offset adjusted",local);
     }

     /* Endless loop. */
     for (;;) {
          get_local();

          if (local == 101000000) {
               send("New Year detected",local);
               init_dst(gtime_pkt.time_pkt_year);
          }
          else if (dst_on - local == 4041) {
               gtime_pkt.time_pkt_pkt_id = $TIME_PKT_PKTID;
               gtime_pkt.time_pkt_func = $TIME_PKT_LOCAL;
               gtime_pkt.time_pkt_hour = 3;
               gtime_pkt.time_pkt_minute = 0;
               gtime_pkt.time_pkt_second = 0;
               gtime_pkt.time_pkt_zone_hour++;
               if (ier = sys_ntime(&gtime_pkt)) exit(ier);
               send("DST started",local);
          }
          else if (dst_off - local == 4041) {
               gtime_pkt.time_pkt_pkt_id = $TIME_PKT_PKTID;
               gtime_pkt.time_pkt_func = $TIME_PKT_LOCAL;
               gtime_pkt.time_pkt_hour = 1;
               gtime_pkt.time_pkt_minute = 0;
               gtime_pkt.time_pkt_second = 0;
               gtime_pkt.time_pkt_zone_hour-;
               if (ier = sys_ntime(&gtime_pkt)) exit(ier);
               send("DST ended",local);
               /* Recompute DST on/off for next year. */
               init_dst(gtime_pkt.time_pkt_year+1);
          }
          wait(0.5);
     }
}
```

Restrictions? Only one: you can't run your system UP macro if it will end up executing DSTMON during the two hours each year when the AOS/VS time-of-day clock has been set to a time that is either impossible or ambiguous. Gee, I think you can probably live with that, eh?

:TIME:PROBLEMS:2

The second problem we have to deal with has to do with rebooting the system. There are two kinds of MVs: with and without battery-operated hardware clocks. Those without ask the OPerator for the date and time-of-day during the boot sequence. Those with simply set the date and time-of-day without human intervention.

Unfortunately, the AOS/VS guys chose to store local time in the battery-powered clock instead of UTC, so after a reboot when DST is in effect the local time will be correct, but UTC will be off by an hour (except for you guys in Asia with the half-hour time zones). What to do? Clearly another little program is called for.

This program needs to check to see if DST is in effect, and if it is, then it needs to adjust the UTC offset accordingly. Pretty simple, eh? And because this program has a lot in common with DSTMON (including the restrictions), and it also needs to be in the UP macro, it makes sense just to combine the two programs into a single program.

The source for DSTMON.CLI and DSTMON.C is shown in Figures 1 and 2. I'm using C here because it's smaller and easier to read, but the real version of the program is in assembly language. The entire package, including both the C and the Assembly sources and the ready-to-run .PR file for the Assembly version, is available on the :SYSMGR BBS as item SMLOGS:SML9308X.

Note that DSTMON.CLI macro PROCs the program up as a resident program at a high priority. This ensures that DSTMON won't be preempted from seeing the critical second during which the change has to be made. By design, DSTMON's overhead is kept to a minimum so that the impact is negligible; a tiny working set size and less than 0.1 percent of the CPU time on my MV/4000, and correspondingly less on faster MVs.

:TIME:PROBLEMS:3

For those of you who depend on computing rates (e.g., transactions/second) or elapsed times (e.g., ambulance response time) based on logs of time-stamped data, then using local time is out of the question because of the problems I mentioned earlier with discontinuities and ambiguities. One possible solution is to use UTC any time you need a time stamp. UTC is practically continuous and always unambiguous. I had to say "practically" because UTC does jump occasionally. Luckily the jumps are always only one second) and involve adding or subtracting a second at midnight on June 30th and/or December 31st. The adjustments are made in order to ensure that UTC is never off from Mean Solar Time

by more than 0.9 seconds.

One problem with doing everything using UTC is that our humans could care less what time it is on the hilltop in England, so we need to have a way to convert UTC to local time for the benefit of the humans. I'm gonna talk about that next month, and show you the program that I use on our MVs to dial the U.S. National Institute of Standards each night at midnight to (re)set our clocks to UTC in such a way that the AOS/VS copy of UTC always advances, even when the MV time-of-day clock gains time and has to be adjusted backward, or when NIST subtracts a second from UTC. You've got a month to guess how I accomplish that feat.

While you're waiting for that, when's the last time you checked the battery on your MV clock? Δ

:REFERENCES

The 1989 Information Please Almanac, pp. 549-550.

Astronomical Formulae for Calculators, by Jean Meeus, 1982, Willmann-Bell, Inc.

Calendrical Calculations, by Edward M. Reingold and Nachum Dershowitz, 1989, University of Illinois at Champaign-Urbana.

The Discoverers, by Daniel J. Boorstin, 1983; Chapter 1, for an excellent historical discussion on calendars.

Practical Astronomy With Your Calculator, by Peter Duffett-Smith, 3rd ed., 1988, Cambridge University Press.

Scientific American, May 1982, pp. 150-151, "The Gregorian Calendar," by Gordon Meyer.

Scientific American, July 1993, p. 56ff, "Accurate Measurement of Time," by Wayne M. Itano and Norman F. Ramsey. See the box on page 65 for a brief synopsis of the relationship between International Atomic Time (TAI) and UTC.

The Standard C Library, by P.J. Plauger, 1992, Prentice-Hall, p. 111 and Chapter 15.

BJ is president of B.J. Inc., a San Francisco-based consultancy specializing in system auditing, system management, and performance analysis. :SYSMGR is a division of B.J. Inc. BJ can be reached at 109 Minna St., Suite 215, San Francisco, CA 94105, 415/550-1444 (voice) or 415/550-1072 (fax). The :SYSMGR Bulletin Board Service number is 415/391-6531 (300/1200/2400 with optional MNP class 5; 605X, 8/N/1) or 415/550-1454 (voice).

A bit more comfortable

SYNOPSIS
In the wonderful world of X-Windows you have the power to customize your operating environment.

by Joe Cannata
Special to Focus

New users to the X-Windows world often do not understand the flexibility of the Motif Window Manager *(mwm)*. The OSF/Motif product is made up of four parts:

1) a user-interface guideline;
2) a "C" API to a standard library of routines for building applications that conform to the guidelines;
3) the window manager *mwm*; and
4) a language, UIL, designed to ease user-interface development.

Therefore, *mwm* will run an application built with any X-Windows API, and in general an application built using the Motif toolkit will run under any window manager, providing the client is ICCCM-compliant. The *mwm* program (Motif Window Manager) is found in the directory "/usr/opt/X11/bin". It is one of many different window managers available. The *mwm* provides features in the frame of the window that make it attractive. It allows resizing and moving of the window, and also generates the three-dimensional beveled look.

One of the tools available to the *mwm* user is the root menu. This is a way of controlling what happens on your display, by simply clicking on the appropriate pop-up choice feature of the root menu. To get the root menu to appear, all that needs doing is to left-click anywhere outside of a client window, on the background area called the root

Figure 1: Part of the corresponding .mwmrc file that created the root menu shown in Figure 2 (page 27)

```
Manu RootMenu
{
@/usr/include/X11/bitmaps/xlogo32          f.title
"New Xterm"                                f.exec "xterm &"
no-label                                   f.separator
"Desktop Tools"                            f.menu Tools
"XClients"                                 f.menu XClients
"User Preferences"                         f.menu Preferences
no-label                                   f.separator
"Remote Logins"                            f.menu Logins
"Remote XAccess"                           f.menu XAccess
no-label                                   f.separator
"Refresh"                                  f.refresh
"Restart Mwm"                              f.restart
"Stop Mwm"                                 f.quit_mwm
}

# Menu to start Desktop tools
Menu Tools
{
"FrameMaker3.1"                            !"/usr/opt/frame/bin/maker -red -fg white *"
"Oval Clock"                               !"oclock -fg bisque -bg peru &"
"X Clock"                                  !"xclock -bg green -fg yellow -chime&"
"Xcalc"                                    !"xcalc&"
}
                                           .
                                           .
                                           .

# Menu to allow remote host access to X Display
Menu XAccess
{
"Access to X Display"                      f.title
"Allow all hosts"                          !"xhost +"
"Allow no hosts"                           !"xhost -"
}
```

New for 1993

Executive Seminars including:

- Migrating to an Open Systems Environment
- Downsizing and your Enterprise

Training Seminars including:

- Network Strategies for the 90's
- How to Select the Right AViiON Database Management System
- Recent Developments in AOS/VS II

An Extensive Exhibit of the Latest Products & Services

Unlimited Networking Opportunities

NADGUG 20th Anniversary Celebrations

OCTOBER 25-28, 1993
THE INFORUM
ATLANTA, GEORGIA

NADGUG 93

will blow you away...

Don't miss this opportunity to be part of the largest Data General gathering in the world!
Watch your mailbox for further details or for more information call 1-800-932-6663 ext. 1234

Circle 24 on reader service card.

window. Holding down the left button is required, otherwise the menu disappears. The standard root menu is controlled or configured by a file called "/usr/lib/X11/system.mwmrc". Making a copy of this file and moving it into your home directory, calling it ".mwmrc", allows you to customize the root menu. Most new users, however, will receive their ".mwmrc" file at the time of account creation. The "/etc/skel/.mwmrc" file will be placed in their home directory automatically when the system administrator builds their account. This file contains comments that are more helpful for customizing.

Figure 2 (page 27) shows a sample root menu. Figure 1 shows part of the corresponding *.mwmrc* file that created the root menu shown in Figure 2. The format of the *.mwmrc* file is simple. You define first the look of the root menu, then proceed to declare any other menus or operations. As long as you follow the rules of syntax, your creativity can take over.

The first line of the *.mwmrc* file contains the keywords, "Menu RootMenu", followed by a left "{" on the second line. This is where the menu you see in Figure 2 takes shape. The ensuing lines define the choices displayed. The line that begins with the "@" followed by the bitmap pathname, provides you the opportunity to select a bitmap pattern to appear at the top of your root menu. The pattern, "xlogo32" is a 32-point (72 pt = 1 inch, approximately) bitmap of the X-Windows logo, as seen in Figure 2. You could draw your own bitmap with the bitmap client and use that instead, or choose one from the standard X bitmaps found in the directory listed after the "@" sign. The second part of that line contains an entry "f.title". That is an *mwm* function, one of many.

These functions make up the *mwm* behavior in the root menu. "f.title" is used to define the menu title, and then creates a double divider line (in OSF/Motif 1.2 currently, in later releases a triple line) to separate it from the rest of the menu items. The remaining portion of root menu description is a list of all of the other choices, such as "Desktop Tools," and "Stop *Mwm*". Looking at Figure 2, any choice with an arrow on the right means that there will be a secondary or cascading menu to follow. Looking back at Figure 1, you see the function "f.menu" on the corresponding lines, followed by a submenu name. This means that elsewhere in the *.mwmrc* file, a separate menu entry must be made with the specific operations defined for that submenu. The line, "Menu Tools," is an example of that. The line above "Menu Tools," that starts with a "#" is a comment. These comment lines can appear anywhere and should be used for readability.

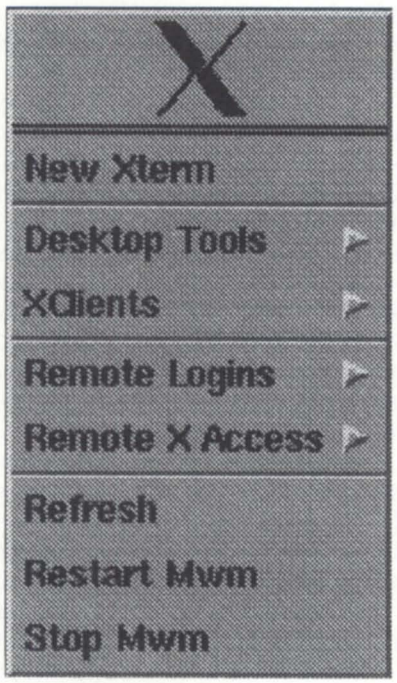
Any of the entries that contain the function *"f.exec"*, perform an operation directly. The first one for instance, fires up an *xterm*. If you wanted the 12x24 font in that *xterm*, "xterm -fn 12x24&"

would be the contents after the *f.exec*. In other words, after the *f.exec* is a standard command line to invoke an X client. The only requirement is that the command is surrounded by double quotes. The "no-label" line has a function called "f.separator". That causes a single line to be drawn across the root menu for emphasis or separation.

When you are in a submenu, such as "Tools" in this example, the only function you need to invoke a client is the "!" character. This is just another name for the "f.exec" function. As you see in the code in Figure 1, to start the oclock client, the "!" function is followed by the "oclock . . ." options.

The entry, "Stop *Mwm*", is a quick way via the root menu to end your session. If you logged on via *xdm*, more than likely you were exec'd into the *mwm* program. Therefore, stopping *mwm* will end all windows processes and return you back to the *xdm login* screen. The *mwm* function "f.quit_mwm" will accomplish this.

Figure 2: A sample root menu



When you click on this entry in the root menu, usually the last one on the list, a confirmation box appears, asking you if you want to "QUIT Mwm". You either click on "OK" or "Cancel". The remaining pieces of the *.mwmrc* file are self-explanatory.

If you want a menu to appear in the root window, other than the root menu, it is possible. To perform this task, it is necessary to identify the menu you want displayed by the name you declared for it in the *.mwmrc* file. Then, add this line:

```
<Btn2Down>    root    f.menu    Tools
```

under the section, "Buttons DefaultButtonBinding", in your *.mwmrc* file. Now restart *mwm* and you will have the "Tools" menu available to you each time you middle-click in the root window.

The default colors of the root menu in OSF/Motif 1.2 are black letters on light steel-blue. If you want to change

those colors, place these two resources into the *"Mwm"* file in your app-defaults directory:

```
Mwm*menu*foreground: color1
Mwm*menu*background: color2
```

This will allow you to choose any two colors you want to see. If you want any cascading menus to appear in different colors than the default or custom scheme you just set, it is possible by setting some other resources in your *"Mwm"* file. The format is almost the same:

```
Mwm*menu*Tools*foreground: color1
Mwm*menu*Tools*foreground: color2
```

In this case, you are adding a class of the name of the menu you declared, "Tools". All you need to do at this point is restart *mwm*.

It is important that you should create a variable called XUSERFILESEARCHPATH and set it to: *$HOME/app-defaults/%N*, where %N is translated by the system into the class (*Mwm*) in this case. This variable will indicate where your individual application resource files are found, the recommended way of setting defaults.

With a little work and creativity, you can set your root menu to make many of your normal operations available with the click of the mouse. For further in-depth information, consult the O'Reilly series book, *X-Window System User's Guide OSF/Motif 1.2 Edition*. DG part number 069-100229-03 for the latest on X11R5 and OSF/Motif 1.2. This newest revision of the book has been enhanced significantly, and is well worth ordering through TIPS.

Another good source is the *Unix Desktop Guide to X/Motif*, from Prentice-Hall. Of course, don't forget the obvious, the man pages on *mwm*, *X*, *xrdb*, and the clients.

Here is one piece of miscellaneous information that may benefit some lefties. It is possible to reverse the mouse buttons, so that the right button would replace what the left button did, and so on. Use the *xmodmap* program, and issue this command line:

```
xmodmap -e 'pointer = 3 2 1'
```

To reset it, either reverse the "3 2 1" or use "pointer = default" after the "-e" option.

With all of this information, you should now be able to make your operating environment in X-Windows a bit more comfortable and user-friendly. If you'd like to learn more about X and Motif, you might also try one of the Data General Ed Services CBT courses on X-Windows programming, or the video course on OSF/Motif. Δ

Joe Cannata is a systems training specialist, Educational Services, with Data General Corporation. He may be reached at the Atlanta Education Center, 4170 Ashford Dunwoody Rd., Suite 300, Atlanta, GA 30319; 404/705-2562.

Totally RADical ICobol

SYNOPSIS
Object-based technology is the simplest method of adding PC-like features to ICobol applications.

by Vibeke H. Arentz
Special to Focus

If Batman lived in Westboro, the Riddler might ask:

*What runs swiftly but has no feet; fixes problems, but uses no tools; has windows, but cannot see outside; leaps from one application to another, but cannot jump; has more keys than a piano, but plays no chord?**

Software applications have come a long way in the past 20 years, in appearance *and* functionality. But many minicomputer applications still look as if they were developed in the 1970s and haven't changed since.

With the growing popularity of PCs and success of Microsoft Windows, users expect software packages to have that PC "look and feel." They also expect software applications to be extremely user-friendly, and that PC-like tools will make applications easy to utilize. As a result, minicomputer software developers must find ways to build features such as windows, menus, and hot keys into their traditional applications to remain competitive. Value-added resellers (VARs) find that even if their software provides more functionality than that of the competition, the competition enjoys a sales advantage if its software offers PC-like features.

But adding these new features in minicomputer applications is no small task. It can take hundreds of man-hours to add the fewest, simplest windows or menus to Cobol applications, because every aspect of new features must be programmed directly into the application's code. This adds significant costs to applications and delays their marketability. An alternative might be to leave the applications alone. But with increasing competition in the software market, the status quo leaves a door open for users to defect from stodgy minicomputer systems to glitzy PC applications. PC hardware often costs less, and applications usually look better on PCs than on minicomputers, making PCs appear to be a good alternative even when PC network platforms and applications might offer less genuine business functionality than that of minicomputer applications. And not only can functionality be lost by mov-

ing to PC networks, but direct and indirect investment may be lost in the minicomputer application.

Now ICobol programmers have a third alternative. New technology has been introduced that allows programmers to add pop-up windows, pull-down menus, and hot keys to their applications in a matter of hours, not days or weeks. Founded on object-based technology, these facilities allow PC-like features to be added to minicomputer and mainframe applications without changing the existing code. Adding fancy features to applications without additional costs allows ICobol programmers to preserve and actually leverage their investment in those ICobol applications.

Traditional Cobol programming was never designed to easily create interactive user interfaces. The introduction of a non-ANSI-compliant "SCREEN SECTION" in ICobol improved the situation, but still requires additional source-level Cobol coding. Any changes to the information displayed requires source modification and subsequent recompilation.

Some Cobol systems now claim window support, but still require source code changes and recompilation of the program to modify the user interface. Even newer "interactive dialog editors" are not typically much more than code generators that still require program code recompilation for the slightest modification.

These situations make it extremely time-consuming, not only to program these features into applications, but also to maintain ICobol applications. ICobol programmers complain of spending weeks to program these features and additional hours recompiling applications after making simple changes to a single application window.

New technology

A new ICobol tool utilizes object-based technology to add PC-like features to ICobol applications in a matter of hours, without any changes to the existing ICobol code. Wild Hare Computer Systems developed the "Harestylist" RAD (rapid application development) tool to offer programmers a simpler, object-based path to the PC-like features users demand.

Object-based techniques utilized by this software tool eliminate all the "modify/compile" hassles required by the conventional method of adding interactive user interfaces to ICobol applications. Object-based technology can remove most of the user interface design from the Cobol programs and place it in completely separate window and menu "object" files. Object files are created and maintained using a nonprocedural, menu-driven, "point and click" interactive screen designer. These object files are treated like named objects and maintained independently from the Cobol programs themselves. Thus the user interface may be created and modified without touching a line of Cobol source code, and without even recompiling a single program.

The benefits of using object-based technology are many. Object-based technology saves programming time and effort, not only in the initial stages of programming windows and menus in ICobol applications, but in maintaining and customizing ICobol applications. Adding these features to existing applications without code-level changes or recompilations keeps development costs as low as possible.

Object-based technology's simplicity also makes it easy for programmers to customize applications' user interfaces. With procedural coding it could take weeks to move windows, adjust hot keys, and further customize and update the application for a single user's specific needs.

For example, a value-added reseller's (VAR's) applications menu system may be customized for a specific user by placing the customer's name in window objects. The customer is happy because the system then looks like a customized system, even though the VAR spent only about 30 seconds to do the customization. And programmers are happy because not one line of code was changed, and no recompiles were required.

Portability

Because applications developed with Harestylist run with Wild Hare's "Choice!" ICobol runtime system, which is fully portable and supports more than 60 different platforms and 18 different operating systems, programmers can offer applications with Harestylist features on virtually any platform or operating system without altering source code. Programmers who previously tried to add window simulation code to their ICobol applications for one platform do not have the flexibility of running that exact application on a different platform or operating system. It would require weeks or months to modify the application for a different platform, adding significant costs to the application.

Object-based technology is the simplest method of adding PC-like features to ICobol applications, saving programmers time and effort, and keeping the cost of applications low while maintaining their investment in existing ICobol applications. Object-based technology also simplifies application maintenance, and gives programmers a simple means of customizing applications for individual users.

Object-based technology is the totally RAD answer to bringing ICobol applications up to date, and to keeping programmers and users RADically happy. Δ

* Why, an ICobol program with Harestylist features, of course!

Vibeke H. Arentz is Marketing Specialist for Wild Hare Computer Systems, Inc. She may be reached at 6595 O'Dell Place, Suite M, Boulder, CO, 80307; 303/530-2221 phone or 303/530-9637 fax.

Using Harestylist

Harestylist operates on virtually any popular platform and operating system. When used with PCs, it can be controlled using a mouse or keyboard method. On other systems, functions are carried out with the keyboard.

When starting up Harestylist to design a window, the user sees a default window template. Once a new window form has been created, text and other enhancements may be added. The keyboard cursor keys may be used to position the display cursor to any desired point on the screen, as can holding down the left mouse button and moving the mouse, i.e., "click and drag."

The window can be moved using the "Move" function under the "Window" menu. The cursor control keys and/or mouse let you move the window to any location on the screen. When you move the window, an outline of the current window fill "follow" your cursor-mouse commands to indicate the proposed new window position. When you hit the enter key or click the left mouse button, the full window will be redrawn in the new position.

The window can be resized by using the "Resize" function under the "Window" menu option. The cursor control keys and/or mouse allow you to resize the window. When you resize the windows, an outline of the new window size will "follow" your cursor/mouse commands to indicate the proposed new window size. When you hit the enter key or click the mouse button, the full window will be redrawn in the new size.

Once the window is positioned, you can assign attributes to the window. All Harestylist functions are accessed by menus originating at the initial help/function menu. For example, you might want to add a shadow to the window to create the illusion of depth, and provide a sophisticated look to your applications. Select the "Shadow" menu from the initial help/function menu. From the "Shadow" menu, using either the mouse point-and-click method or the keyboard, select "Foreground color," which allows you to select the foreground color of the shadow. Then select "Background color," which allows you to select the background color of the shadow character. Then select "Attributes," which include blink, compressed, dim, reverse, and underline. Then select "Draw," which redraws the shadow with the current style, attributes, and color in effect. If you want to erase the shadow, select "Erase."

Any window-design process can be accomplished using this interactive technique. The "Border drawing" menu lets you select the window border's "Foreground color," "Background color," "Attributes," and "Style."

Use the "line drawing" menu to draw lines in your window. With this menu, use the cursor keys or mouse to determine the starting and stopping points of each line in the window. The "Line drawing" menu lets you select the line's foreground color, background color, and style.

The "Title handling" window allows a title to be associated with a window. It is controlled by the color, attributes, content, and location specified by the options in this section. The functions in this menu allow you to select the title's foreground color, background color attributes, and location. Location options are Left, Center, and Right.

The "Box drawing" menu allows the simple creation of rectangular box character sets in a manner similar to drawing lines. Boxes add a cosmetic enhancement to your windows. You create a new box by setting up the colors, attributes, and style you want, then specifying the top-left and bottom-right location of the box.

Harestylist offers many options in creating a window, allowing you to create a simple or complex window with just a few keystrokes or mouse clicks, and no recompilation. Once you have specified a window, you assign a name to it and save it to an object library.

Programming windows

Windows may be controlled by an application program in two fashions: implicitly through runtime system setup (no programming control required), and under program control with an Axis-compiled program (Axis is Wild Hare's ICobol compiler).

Automatic control. You can use the "Choice!" hot key and default window subsystem to define windows that are automatically displayed for any given program by the runtime system. This means you don't need to make program modifications in order for "Choice!" to call your windows into your Cobol program.

Program control. The new "windows" verb supported by the Axis ICobol compiler gives you utmost control of an applications-windowing environment. Simple commands include opening, closing, selecting, and modifying application-controlled windows at any time. The syntax is:

```
ADD
CLEAR
CLOSE
WINDOW        HIDE    lit-id [FROM filename]
OPEN
REDRAW
SELECT
SHOW
[[NOT] ON EXCEPTION imperative-statement]
[END WINDOW]
```

Each high-level function refers to a name the corresponding window was given when created or modified by Harestylist. Whenever a window name is required, an optional object library name may also be specified. This allows different window object files to contain objects with the same name. Δ

The latest products for DG systems

Terminal emulation

Atlanta, GA—Smarterm 470 for Windows is Persoft's full-featured Data General terminal-emulation software that allows PCs to precisely emulate the DG Dasher series of graphics and text terminals to connect to MV and Aviion hosts. Smarterm provides all the support needed for PC LAN-to-host connectivity, including Smarterm TCP/IP and corporate-support tools for standardizing PC-to-host interaction.

Smarterm 470 for Windows emulates the D470C, D461, D460, D450 color and monochrome graphics terminals, as well as the D411, D410, D400, D220, D215, D214, D211, D210, D200, and D100 text terminals. The software provides Telnet management for third-party TCP/IP applications, giving users flexibility in choosing the appropriate connectivity option for their environment.

Virtually all ethernet cards and all popular LANs are supported, including NDIS, ODI, and token ring drivers. The Windows version allows up to eight concurrent host sessions in separate windows.

Smarterm 470 includes sophisticated FTP capabilities: drag-and-drop FTP allows users to make convenient point-and-click file transfers; the standalone FTP client program allows direct access to FTP without running Smarterm; additional file transfer protocols supported include Kermit, Xmodem, Ymodem, and Zmodem.

A standalone version is priced at $395 retail. The file-server price is $1,795 retail for five concurrencies. Δ

Persoft, Inc., 465 Science Drive, P.O. Box 44953, Madison, WI 53744-4953; 608/273-6000.

AD INDEX

Company	PG#	RS#
Ames Sciences, Inc.	31	1
Asset Remarketing	5	2
Async Systems, Inc.	15	3
CDS	18	4
Claflin & Clayton, Inc.	27	5
Claflin & Clayton, Inc.	29	6
Computer Engineering International, Inc.	C2	7
Computer Wholesalers, Inc.	20	8
Contemporary Cybernetics Group	10	9
Data Bank Associates, Inc.	15	10
Data Bank Associates, Inc.	31	11
Data General Corporation	3	12
Data General Professional Services	33	13
Digital Computer Consulting, Inc.	18	14
Digital Dynamics, Inc.	28	15
Eagle Software, Inc.	7	16
Gerry Manning & Associates	22	17
Hanson Data Systems, Inc.	13	18
Intelligent Information Systems	19	19
International Computing Systems, Inc.	6	20
Mac Data, Inc.	32	21
Minitab Statistical Software	35	-
Monarch Software, Inc.	C4	22
NADGUG	23	23
NADGUG	25	24
National Computer Dynamics	22	25
Park Place International	21	26
Persoft, Inc.	17	27
R&D Computers	28	28
RAVE Computer Association, Inc.	12	29
Rhintek, Inc.	30	30
SCIP	11	32
Scott McIntyre & Associates	C3	33
Security Computer Sales, Inc.	9	34
Sysgen Data Ltd.	33	35
:SYSMGR, a division of B.J. Inc.	32	36
:SYSMGR, a division of B.J. Inc.	28	37
:SYSMGR, a division of B.J. Inc.	35	-
Threshold, Inc.	22	38
Universal Data Corporation	9	39
Wild Hare Computer Systems, Inc.	1	40
Wild Hare Computer Systems, Inc.	26	41

PRODUCTS AND SERVICES

Company	PG#	RS#
Persoft, Inc.	33	-

ON-LINE HELP

***Who to call for answers about* NADGUG *and* FOCUS**

NADGUG address:
c/o Danieli & O'Keefe Associates, Inc.
Chiswick Park, 490 Boston Post Rd.
Sudbury, MA 01776
FAX: 508/443-4715

FOCUS Magazine address:
c/o Turnkey Publishing, Inc.
P.O. Box 200549
Austin, TX 78720
FAX: 512/335-3083

NADGUG

Membership, RIGs, SIGs
NADGUG staff **800/253-3902**
(Outside the U.S and Canada.) **508/443-3330**

Electronic bulletin boards
(300,1200, 2400, or 9600 baud modem)
Rational Data Systems **415/499-7628**
DASH bulletin board (2400 bps modem) **800/DASH-CSC**
Data General Customer Support Center **800/344-3577**

FOCUS Magazine

512/335-2286

Editorial comments, article suggestions.....Doug Johnson
(please send product announcements to the address listed above)

Information about advertising.....Susan de Blanc

FOCUS back issues .....Turnkey Publishing staff

Continued from page 13

sary process switching is a waste of system time. The cache stalls and it may appear as CPU time. When this value gets into the thousands (depending on the type of system and number of processors), then too much time is being spent flushing cache. The solution is to modify applications to eliminate context switches. To accomplish this, the application may need to be modified to buffer up more data and do more processing at once. Applications should process more code with fewer calls to subprograms. Subprograms that use interprocess communication calls, pipes, reads, writes, excessive locking, and other system resources may force a process switch on a wait for a particular resource. Applications should be modified to more efficiently use semaphores, message queues, or signal-processing calls. The number of message queue calls and semaphores is listed by the *sar -m* option.

Features of UX/RPM and AV SysScope for swapping and process-switching activity. Reports the number of bound, unbound, and eligible processes. An "eligible" process is a one that is running or is chosen by the medium term scheduler (MTS) to run when a physical processor is available.

Summary

Performances monitoring and tuning the DG/UX system were once thought of as difficult tasks, and even now, performance monitoring and tuning are not exact sciences. There are many utilities available to help you analyze performance data and to decide on corrective action. The three areas where the system could be spending its time are in system mode (%sys), user mode (%usr), or idle time (%idle). The Unix system activity reporter (sar) is the first-choice utility in any performance investigation. Δ

Thomas E. Soukup, currently serving as the worldwide Benchmark Coordinator, is a member of the Technical Services Group at Data General Corporation in Schaumburg, Illinois. *Edited by Edward A. Sepich III, Total Environmental Systems Services Corporation.*

DASH Items...

Category: DG/UX

Author: Brian Paige
Subject: **Timeout on ports**

I need a way to check to see if a port has had any activity, and if not log it off. Not being extremely familiar with Unix, I was wondering if someone could tell me if there's a Unix command that will do this for me, or if I have to create something.

I played with the timeout option in *sysadm* for port services, but this seems to apply only to the login prompt/process.

Reply by: Doug Morgan

You can use the *who -u* command. The field just before the pid number is a time field.

The time is how long that port has to be idle. If a period (.) shows up in this field, the port has had activity in the last minute. You can write a script to kill any port that has been idle for so many minutes.

Category: Networking

Author: Ned Sheppard
Subject: **Printing w/portable Netware**

We are running Portable Netware on our Aviion and everything works well except for some printing problems. Text files sent to a queue that routes the job to *lp* services and then out to a printer defined on the Aviion print out very well.

Any graphics or Windows (3.1) word processing documents get trashed and are not legible when they get to the printer. The symptoms range from only the left side of the page printing to random characters all over the page. The Aviion printers are HPII and are defined as *hplaserjet II*.

Since I can print to a workstation printer (HP Laserjet) at a workstation running *rprinter* through the Portable Netware from any PC on the LAN and get good results with graphics files, I am led to believe that the problem lies with the printer definitions in *sysadm*.

Reply by: RTP Support

In general, when printing graphics or binary data to a printer connected and managed by the System V print scheduler, you should use an *stty* setting of "-opost".

This prevents output processing. The *w/o -opost* CRs are converted to CRNL pairs, tabs are expanded to spaces. You may also define two logical print queues to the same port.

For example, a queue named *laser_text* would not contain the *stty* setting of *-opost*. You would print all simple ASCII text files to this queue.

Next, you could define a second queue named, *laser_binary*. This queue would include the *stty* setting of *-opost*. All graphics or binary data should be printed to this queue.

If you do not wish to define a second binary queue, from the shell, you may issue the command: *lp -dprinter -ostt-opost file*. The *lp* command line option, "-ostty=-opost", will be passed to the interface script and no output processing will take place.

For additional information see the manual, *Managing the DG/UX System* - 093 70188-02 Chapter 11 page 11-93, "Adjusting the Printer Port Characteristics." Δ

DASH runs on an Aviion 5200 server located at the Customer Support Center in Norcross, GA. The bulletin board is available 24 hours per day, 7 days per week, free of charge. Call 800/DASH-CSC (800/327-4272) for the modem rotary.

Bits and bytes...

Asynchronous communications

From: Brian Johnson

There seems to be a lot of confusion about when ^S and ^Q are honored on IAC lines. The real scoop is this: they are always honored whenever the port is open and the current or most recent read or write was a nonbinary operation (?IBIN=0 in the I/O packet). The /IFC and /OFC characteristics specify that ^S/^Q should also be honored during binary reads and writes, respectively. All DG and third-party programs that I know of that require flow control to operate during binary reads and writes take care of turning on /IFC or /OFC themselves (e.g., CEO and Wordperfect), so these characteristics will only cause trouble if turned on by default (as I often see). If your program requires binary delimited strings (e.g., records terminated by an EOT or even CR/LF) then it should set both "?IBIN" and "?RTDS" and either default the delimiter table pointer (offset ?IDEL) or supply the address of a bitmap that specifies what terminators you want. The danger in using binary reads and writes with modems whose serial port speed is higher than the connection speed, or when on noisy phone lines, is that you really do need for the flow control to work, so you probably ought to be turning on /IFC and/or /OFC unless ^S/^Q are valid data characters. If so, then your only recourse is to run the serial port at the same speed as the modem port and take your chances on noisy phone lines. Δ

The NADGUG/Rational Data Systems electronic bulletin board is available to NADGUG members. Call 415/499-7628. No fees other than phone charges.

SCOTT McINTYRE & ASSOCIATES, INC.

"YOUR GLOBAL SOLUTION FOR D.G. & MOTOROLA"

313-299-0040

*** BUY * SELL * TRADE * DATA GENERAL & MOTOROLA ***

PRICE!

"It is with great pleasure that we have chose to work with Scott McIntyre & Associates, Inc. as our second source Data General supplier. The prices, service, delivery time and quality of the equipment we've purchased has never been equalled!"

Amaizo Yves
Managing Director
Sysinfo
Jauldes, France

PERFORMANCE!

"Scott McIntyre & Associates, Inc. has earned the respect of all the executives at our company for their honesty and diligence. They have always been flexible and accomodating!"

Andy Michael
Sales Director
Quiss Tech., Plc
London, England

QUALITY!

"Many other resellers in the USA would like to deal with us in Australia, and on some occasions we have brought from them. However, the quality of the goods and service has never matched what we get from Scott McIntyre & Associates, Inc."

Alan Frazer
Managing Director
Frazer & Associates PTY. LTD.

DELIVERY!

"It is good to know that there are still people out there who conduct their business with honesty and integrity."

William E. Herr
Director of Computer Operations
The Bon-Ton Store

VISA MasterCard AMERICAN EXPRESS

"August Specials"

"NEW" D413 CRT W/KB ONLY $435

6799-G 520MB HH DIFF. DISK ONLY $1,450

4370 IAC-16 RS232 ONLY $185

D410 CRT W/KB ONLY $125

4626 LAC-32 RS232 ONLY $2,450

MV2000 II W/4MB, 21MB CTD & 70MB DISK ONLY $950

"OVER $1,000,000 IN STOCK!"
"CALL FOR OUR SUMMER PRICE CATALOG!"

"Call Today"

***Scott McIntyre * Terry Smith ***
*** Robert McIntyre * Michael Nowacki ***
"Welcoming Lorraine LaCroix"

313-299-0040

FAX: 313-299-0041

P.O. BOX 82115 • 1856 STAR-BATT DR.
ROCHESTER, MI 48308-2115

Circle 33 on reader service card.

Plug And Play And Rock All Day!

***NOW PLAYING from* Monarch Software** ♪♪
Here's what you've been waiting for! The first indisputably easy migration path to DG/UX® for INFOS® and CEO® users.

OPEN ELECTRONIC OFFICE ™ (OEO ™)

Open Electronic Office is the only easy migration path for **CEO** users moving to DG/UX. **OEO** is keystroke-for-keystroke compatible with **CEO** and provides a familiar setting that ensures users will maintain, and even improve, their productivity. **OEO** also provides multiple choices in PC integration and MV interoperability.

OPEN FILE MANAGER ™ (OFM ™)

Open File Manager is the only high performance, high reliability migration path to DG/UX for **INFOS** applications. **OFM** uses the **INFOS** API so your programs won't have to change to take advantage of the improved performance and greater recoverability features designed into **OFM**. We also offer a VS COBOL-compatible run-time system and a sort/merge, all in one cost-effective bundle.

For More Information about easy migration to open systems call: 1-800-MIGRATE

Monarch™ Software

4904 Waters Edge Dr. • Raleigh, NC 27606
Phone: (919) 851-5408
Fax: (919) 851-9508
E-Mail: info @ monarch.com

Circle 22 on reader service card.