

The VM/370 Resource Limiter (RESLIM), a facility available on the computer systems of the IBM Thomas J. Watson Research Center, enables users, user management, and the site's Computing Center to monitor and control usage of various computing resources. If a user's consumption of a particular resource exceeds a previously established limit, RESLIM takes actions designed to improve system performance and resource availability. Possible actions include degrading the user's priority, forcing the user off the system, or simply sending a warning message to the user and/or other VM users.

The VM/370 Resource Limiter

by D. M. Chess and G. Waldbaum

The interactive computer services of the IBM Thomas J. Watson Research Center at Yorktown Heights, New York, available through its Computing Center, depend on VM/370 (Virtual Machine Facility/370). Every day, about two-thirds of the Research Center's more than 1500 employees use the Computing Center's VM/370 systems: an IBM 3033 Multiprocessor and a System/370 Model 168 Attached Processor. Whenever one of these systems is unavailable or overloaded, as many as 500 users may be affected, with the result that the Research Center's productivity is severely impacted.

The extent of users' dependency on the VM/370 systems requires the Computing Center to provide a high level of service to its users. With the aid of annual user surveys, monthly goals for key measurements of performance, availability, and reliability have been established. The Center manages these goals and informs users and their management whether or not they are met.¹ (For background on VM/370 management, see References 2 and 3.)

Copyright 1981 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

The Computing Center's performance management strategy is a combination of capacity management, system tuning, and workload management. The Center attempts to acquire the capacity needed to satisfy users' forecasted requirements and its own performance goals. It also attempts to achieve its performance goals with its current equipment. To do that, the Computing Center monitors and tunes the systems and monitors and controls large usage during prime shift (9 am to 5 pm). In an environment like the one at the Research Center where demand generally far exceeds capacity, demand has to be controlled in order to ensure that all users do not experience long queues and delays for their interactive work.

One strategy used for managing the Research Center's workload is budgeting prime-shift CPU time. The Research Director's Technical Staff obtains the CPU requirements of the departments and allocates a prime-shift CPU time budget to each department that reflects the priorities of the Director and his Technical Staff. The sum of these allocations is less than the amount of CPU time that would jeopardize the Computing Center's monthly performance goals.

The Research Center's VM/370 Priority Changer controls each department's usage according to its CPU budget. This facility degrades prime-shift service for the largest CPU users in departments that exceed their allocations. The degree of degradation is determined by the extent to which the department was over its allocation for the past 20 days. It is also a function of system performance: the higher the aggregate use of system resources, the more the users in departments over allocation have their service degraded in relation to the users in departments who have not exceeded their CPU budgets. This strategy encourages users and their management to control large prime-shift usage and to move less critical heavy usage to lightly loaded periods. It also encourages user management to do a more thorough job of planning for their departments' future needs.

This paper describes the VM/370 Resource Limiter (RESLIM), a facility developed by the Computing Center. It is used in conjunction with the Priority Changer to control excessive system usage. RESLIM enhances the CPU budgeting process by providing users and their management with a tool for monitoring and controlling excessive CPU usage on a real-time basis. Such a facility is needed in VM/370 because users often mistakenly think that the system is "down" or overloaded when, in fact, their programs are looping (due to user or system errors), and they exacerbate the problem by powering off their terminals without stopping their programs. RESLIM also allows the Computing Center to manage excessive usage, especially that caused by runaway loops in disconnected virtual machines used to perform tasks for users.

RESLIM

What makes RESLIM especially noteworthy is its implementation. Running as an automatic virtual machine, it is on a level between the VM scheduler and the performance management staff. While much has been written on the measurement and tuning of VM systems at various levels (see References 4 through 7), RESLIM is one of a very few facilities that can tune the environment in real time and retain the flexibility of a virtual machine that is independent of the scheduler.

The primary functional component of RESLIM is a virtual machine that accepts user requests and monitors resource consumption. The RESLIM virtual machine periodically compares each user's consumption of various computer resources (currently, CPU time and virtual storage size) with limits established for that user. If no such limits exist, the user's consumption is compared instead with limits established for the user's department. If that limit does not exist, the user's consumption is compared with limits established for all users who do not have individual or departmental limits.

When RESLIM discovers that a user's resource consumption exceeds the applicable limit, it takes one or more actions designed to improve system performance and resource availability. Possible actions include warning the user, degrading the user's priority, and forcing the user off the system. For every action taken, RESLIM may also send a message to one or more other VM users (typically system programmers, performance management personnel, or concerned user department administrators), notifying them of the user involved and the action taken.

Lists of RESLIM limits, enabled actions, and users to be notified of actions taken are stored in files on RESLIM's disk. They may be modified or queried with the aid of the RESLIM machine and the RESLIM command. Users may set certain of their own limits. RESLIM "administrators" may set limits for any user or for any department. The format and syntax of the RESLIM commands are described in the Appendix.

The uses of RESLIM

The RESLIM virtual machine records on its disk a summary of the actions it has taken since it last logged on. Each record contains the identification of the user that exceeded a limit, the type and value of the limit exceeded, actual consumption, time, date, and whether the user was receiving messages when the action was taken. This file, combined with followup discussions with users and management, makes it possible to "tune" the RESLIM options and to evaluate the effectiveness and primary usage of RESLIM.

RESLIM has been monitoring VM/370 systems at the Research Center for two years.⁸ However, the ability to establish department limits, and thus to bring every user in the installation under RESLIM, has only been available since July 25, 1980. Since that date, RESLIM has been used to reduce the amount of resource usage that is faulty, inefficient, or unapproved by management. The concomitant reduction in CPU demand, paging, and I/O operations has resulted in more resource availability and better service.

The primary purposes of RESLIM are:

**purposes
of RESLIM**

1. To detect undesired program loops and minimize their negative impact on system performance. Unusually high CPU usage by a user or system task often indicates an undesired software loop. If the particular user or responsible system programmer is speedily informed of the anomaly, large amounts of computer resources can be saved and the impact on system performance minimized. In the absence of user action, a satisfactory improvement in usage and service can be achieved by having the priority of the offending task degraded automatically.
2. To detect large resource users and minimize their negative impact on system performance. When RESLIM finds a user consuming unusually large amounts of CPU time or virtual storage during periods of heavy system usage, it may be the result of a large or time-consuming task whose results are not needed immediately. In these cases, the task should be scheduled for deferred or after-hours execution.
3. To allow individual departments to monitor usage and stay within their CPU usage budgets. RESLIM permits the establishment of entries that apply to all users in a particular department who have no individual RESLIM limit. By this means, department administrators may be notified when RESLIM detects a user consuming a large amount of CPU time. A more effective approach for user management is to institute automatic priority degradation for users exceeding their departments' CPU limits. Within one month of the time this option was offered, it was adopted by all department directors.
4. To allow individual users to monitor and control their own disconnected virtual machines (virtual machines which operate without a connected physical terminal). Users in charge of machines that run disconnected may use RESLIM as an automatic supervisor for those machines, degrading their priority or logging them off if they consume more CPU time than expected.

The RESLIM virtual machine

The RESLIM virtual machine is automatically logged on at system initialization time. RESLIM then performs various housekeeping

functions largely through CMS (Conversational Monitor System) EXECs and starts the main program.

The RESLIM program is a collection of PL/I routines. Its basic action-sequence is simple. RESLIM sets an interrupt to occur at some future time, prepares itself for receiving user requests, and goes into a wait state. When the RESLIM virtual machine is aroused from the wait state, it determines what "awakened" it and takes the appropriate action. RESLIM can be awakened by a user request or a timer interrupt.

The RESLIM command enables users to communicate with the RESLIM virtual machine by means of the Research Center's XMSG (Extended MSG) facility (a local version of the SMSG of VM/370 System Product, also called VM/SP). The RESLIM command sends the RESLIM virtual machine an XMSG that awakens the machine with an external interrupt and fills a buffer in its memory with the text of the user's request. RESLIM examines the request for authorization and syntax. If the request is unintelligible or came from a user unauthorized to issue it, RESLIM sends an error reply (via Control Program MSG) and takes no further action. If the request was valid, RESLIM performs the required action and transmits a confirmation. A user's resource limits, the actions to be taken when the limits are exceeded, and the "Cclist" (the list of users to be notified of actions) may all be altered and queried in this way. All responses to queries and confirmations of alterations to RESLIM's files are performed by way of the Control Program (CP) MSG.

When the timer set by the RESLIM virtual machine expires, the RESLIM machine is awakened. At the Research Center, the machine is awakened in this way every 10 minutes. The length of this interval is adjustable. A longer interval will reduce RESLIM's CPU consumption but increase the impact of undesired program loops and other cases of excessive usage. RESLIM now uses an average of two to three minutes of System/370 Model 168 CPU time per day on each of the Computing Center's two systems.

Upon awakening in this manner, RESLIM reads the VM/370 control blocks which contain usage statistics for each user running on the system. RESLIM compares each user's usage with the user's limit, if such a limit exists, or if no such limit exists, with a limit for the user's department. RESLIM keeps two separate sets of limits, one for the period of primary usage (User Period 1, 9 am to 5 pm on working days) and one for off-shift (User Periods 2 and 3). Departments at the Research Center have computer usage budgets only for prime shift since demand for computer resources is very heavy on prime shift and substantially lighter off-shift. Thus, RESLIM's off-shift limits are very high (over an hour of CPU time on a Model 168) and are intended only to detect accidental and anomalous resource usage.

If RESLIM discovers that a user's resource consumption exceeds the applicable limit, it determines the next enabled action for the user. Possible actions, in the order they may be executed, are:

**possible
actions**

- Send a warning message to the user (by way of CP MSG), informing him that his usage has exceeded his limit.
- Severely degrade system response for the user by setting the user's scheduling priority to 99, and inform the user of this action.
- Cancel the user's session by using the CP FORCE command to log him off the system. This is done only with prior agreement of the user or the user's management.

Whenever RESLIM takes an action because a user exceeded a limit, RESLIM determines whether the user's Cclist flag (described in the Appendix) has been enabled, and if it has, sends a message describing the action to each user in the offending user's Cclist. If that user is not receiving messages because he is not logged on, is running disconnected, or has set "MSG OFF," RESLIM also sends the message to that user on the other VM/370 system.

The first time RESLIM discovers a user over a limit, it takes the first action (warning, priority degradation, or cancellation) that is enabled for the user. If this user is still over the limit the next time it checks consumption, it takes the next enabled action in the series. This procedure continues until the user logs off the system, RESLIM runs out of enabled actions, or the user's limit is increased to a level exceeding the user's current resource consumption.

Efficacy—examples of use

Even while monitoring the entire population of (over 400 simultaneous) VM users, RESLIM uses less than three minutes of Model 168 CPU time during the eight hours of prime shift. The time it saves and the resulting improvement in system performance are difficult to estimate accurately, but the following examples indicate RESLIM's value to the Research Center.

The first Monday that RESLIM enforced CPU time limits for every VM user at the Research Center, the default limit was 20 minutes of Model 168 CPU time. At 10 am (one hour into prime shift), RESLIM discovered a disconnected virtual machine that had consumed almost 23 minutes of CPU time in that hour. Upon investigation, it was discovered that the machine's owner had been on vacation since the previous week and that the machine had been a heavy CPU user all weekend. The user's manager was contacted and the machine forced off the system. If we assume

that without RESLIM's warning the runaway machine would have run until at least 11 am, RESLIM paid for itself for a week as a result of this action.

A week later, a user in one department was using a piece of faulty system software which contained a subtle program loop. When this problem occurred, the user mistakenly thought the system was down and turned off his terminal. However, his program continued to loop while he was disconnected. RESLIM sent a warning message to responsible performance personnel after the looping user's usage exceeded 20 minutes, but the loop used more than five more minutes of CPU time before the user was contacted and the program halted. The next day, the user's department director instituted automatic priority degradation as the RESLIM action for his department.

One user disabled his CPU time limit and failed to reenale it. Some weeks later, his program entered an undesired program loop that used 53 minutes of CPU time before it was noticed. A reasonable RESLIM limit could have prevented at least half of that waste.

To allow easy testing of new versions of RESLIM, a virtual machine called RESSIM was established to test modified RESLIM code without running the risk of taking action against undeserving users. When a new modification of RESLIM (running on RESSIM) went into a loop, RESLIM detected its "twin's" excess usage, and alerted the programmer in charge. Thus, RESLIM keeps down its own development costs.

If one of the VM systems fails and does not come up again until the morning, virtual machines that had been scheduled to perform extensive tasks overnight are likely to run instead during prime shift. RESLIM detected and degraded the priority of at least four such machines in August 1980.

An inexperienced user, whose CP directory entry had recently been changed, used 10 megabytes of virtual memory and a large number of drum pages during prime shift. RESLIM detected the excessive virtual machine size and informed a VM consultant. The consultant helped the user set his storage size to a reasonable level before the user had adversely affected system paging rates.

One of the Research Center's system tasks (a disconnected virtual machine that performs some work for users) used up its disk space and began to loop. After RESLIM notified the responsible system programmer about the excessive CPU usage, the programmer brought the machine down and cleared space before users were impacted by poor service.

Hardware failures may cause users' programs and system tasks to loop. In one case, a hardware problem that generated nonstop interrupts was identified only after RESLIM degraded the priority of the system task responsible for handling those interrupts.

Two users responsible for disconnected virtual machines, which should only run overnight, have set 0 second CPU time limits for prime shift, and arranged to have their machines forced if the limit is passed. This means that the machines are logged off if they erroneously run into prime shift, even if their owners are absent.

These examples illustrate some of the benefits RESLIM has provided our VM computing facilities.

All departments at the Research Center now have RESLIM limits of 20 minutes of equivalent Model 168 CPU time or less with priority degradation as the first enabled action. Most also have a Cclist that includes a department manager or a senior manager's secretary. Departments whose budgets have been closest to exhaustion have generally chosen the lowest limits.

All VM users, except those with a demonstrable need, are also subject to a two-megabyte limit on virtual storage size throughout prime shift except for the two hours around lunch time (11:30 am to 1:30 pm). This limit has the effect of shifting some load to the lunch period when the load is generally lighter.

RESLIM is a virtual-machine-size monitor that is sensitive to the amount of each user's virtual storage currently on system paging devices. No action is taken against a user whose virtual storage is above his limit *unless* such usage is also above a certain threshold (currently 256 pages). This feature helps ensure that RESLIM will take action only when a potential performance impact exists.

Possible enhancements to RESLIM

Several enhancements are being considered in order to extend RESLIM's utility:

1. Monitor the usage of other system resources, including temporary disk space, spool files, and tape drives.
2. Allow limits to be set for *rates* as well as absolute amounts of resource use (e.g., CPU time per hour).
3. Use historical usage data to calculate appropriate limits for most users (subject to individual overrides) by determining a "normal" value for each user monitored.
4. Allow a manager to specify that a user (or all users) in the manager's department should not be able to change his RESLIM entry.

5. Take actual system performance into account. When performance is satisfactory, RESLIM could be more lenient in its actions. When performance is below established objectives, RESLIM could reduce user and department limits.
6. Allow for variable wake-up intervals, depending on limit values. For example, a user who is very close to his limit at the last inspection might be checked again in five, rather than ten, minutes. This check would reduce the delay between the time the user exceeds a limit and the time RESLIM takes an action against that user.
7. Prepare a monthly report, by department, of actions taken, limits exceeded, and requests processed.
8. Introduce an interactive "front-end," preferably taking advantage of the full-screen capabilities of the IBM 3270 displays. It would be easier for users to determine and modify RESLIM entries because, as the Appendix illustrates, the syntax of the RESLIM command is rather cumbersome.

Conclusion

RESLIM has proved to be a useful workload management tool for users, user management, and computing center personnel. On a daily basis, RESLIM discovers about ten undesired program loops and automatically reduces the performance impact of these loops by degrading the offending virtual machine's priority. This saves hours of processing time yet preserves the faulty state of the users' virtual machines to facilitate on-line debugging.

The most notable feature of RESLIM is its unique position in the resource allocation hierarchy. It is more specific in function and more easily modified than the VM scheduler and yet makes possible a degree of automation that can free management from the responsibility of directly watching over the system and taking action manually. While it is likely that certain of RESLIM's functions could be performed by a sufficiently sophisticated scheduler, it is also clear that the ease with which management can alter its functional parameters stems directly from its implementation as a separate virtual machine. And in an environment which is heavily resource-constrained, the ability to make prompt, specific changes in the allocation and control of those resources can be a great advantage.

ACKNOWLEDGMENTS

We wish to extend our appreciation to Donald Sannella, a former summer employee in the Computing Systems Department, for writing the first version of RESLIM, and to Norman Brenner and Leroy Junker for their efforts in enhancing and maintaining RESLIM.

Appendix: The RESLIM command

Following is a representation of the RESLIM documentation as it is printed out and which is available to users on line. It consists of a CMS file (on the same disk as the RESLIM EXEC), which the user can edit by typing RESLIM ??

The RESLIM command communicates with the RESLIM virtual machine, a facility for monitoring and limiting the resource usage of individual VM users.

The basic format of the RESLIM command is

RESLIM subcommand subcommand-operands

where 'subcommand' is one of:

SETLim

SETAct

SETLIST

Query

Dellim (for the use of RESLIM administrators only)

CONS(" " " " " " " ")

STOP (" " " " " " " ")

and 'subcommand-operands' vary with the subcommand.

The syntax and function of each of the public subcommands follows. (For limitations and temporary conditions, see NOTES at the bottom of this file.)

1. SETLIM

syntax: RESLIM SETLim < CURRent | ALL | UP1 | UP23 >
type value < unit >

function: The SETLIM subcommand establishes a limit on the amount of a given resource that the issuing user may consume in one terminal session. RESLIM checks the resource consumption of each user every 10 minutes, and, if any user is over his limit, takes appropriate action (as specified by the SETACT subcommand).

operands: If CURRENT is specified in the SETLIM command, the given limit applies only for the current terminal session: when the user logs off, it will be forgotten.
If ALL or UP1 is specified, the limit will apply for the current session, and all future sessions, while User Period 1 is in effect (subject to temporary overriding by SETLIM CURRENT).
If UP23 is specified, the limit will apply for the current and all future sessions off-shift (User Periods 2 and 3), subject to the same overrides by SETLIM CURRENT.

'type' is the type of resource monitored. Current types monitored are CPUTIME and VMSIZE (VMSIZE limits may only be set by RESLIM administrators). CPUTIME is the amount of total CPU time consumed in the current terminal session, or since the last change of shift (as reflected in CP QUERY TIME).

'value' is the limit to be set. For CPUTIME, if no 'unit' is specified, 'value' is assumed to be in CPU seconds. Alternately, MIN may be entered for 'unit,' and 'value' will be taken in CPU minutes. A 'value' of * indicates that no limit should exist (i.e., monitoring ceases), and a value of +value or -value indicates that the existing limit should be changed by 'value.'

'unit' is blank (implying SEC) or SEC or MIN.

examples: RESLIM SETL CURR CPU 30

A limit of 30 CPU seconds is established for the current terminal session.

RESLIM SETLI UP1 CPU +2 MIN

The limit that applies to every terminal session in UP1 is increased by 2 CPU minutes.

RESLIM SETLIM UP2 CPU 1 MIN

The limit that applies to every session during UP2 and 3 is set to 60 CPU seconds.

RESLIM SETLIM CURR CPUTIME *

The CPU time limit is removed for the current session.

2. SETACT

syntax: RESLIM SETAct <CURRent|ALL|UP1|UP23>
type act-options

function: The SETACT subcommand determines what actions will be taken when the user is found to be over his limit. There are three levels of action: the issuing of warnings, degrading of priority, and finally, cancellation of the terminal session (i.e., forcing logoff of the VM). Each of these levels may be enabled or disabled via the SETACT command. The first time RESLIM finds a user over his limit, the first enabled action in the hierarchy (warning, priority, logging off) will be taken. If the user is still over his limit ten minutes later, the next action that is enabled will be taken, and so on until either all enabled actions for that type of limit are exhausted, or the user is again within his limit.

operands: <CURRent|ALL|UP1|UP23> and 'type' are as described above, under SETLIM. 'act-options' are one or more of:
<WNG|NOWNG> <PRiority|NOPR> <NOCANcel|CANcel>
<NOCCList|CC>

The first three control the enabling of the three levels of action described above. The fourth, CCLIST, controls whether or not a list of other users is notified of all action taken (the list is established by SETLIST).

The SETACT subcommand always sets all four of these options. If fewer than four are given on the command line, RESLIM acts as though the default values (WNG, PR, NOCAN, NOCC) had been specified for each that is omitted.

examples: RESLIM SETA CURR CPUTIME WNG PR CC CAN
Enables all the levels of action for CPU limits during this terminal session, and instructs RESLIM to notify the user's CCLIST of all action taken.
RESLIM SETAC UP1 CPUTIME NOPR
Because of the assumed defaults, this is the same as:
RESLIM SETACT UP1 CPUTIME WNG NOPR NOCAN NOCC

3. SETLIST

syntax: RESLIM SETLIST <CURRENT|ALL|UP1|UP23> type
<ADD|DROp> name

function: The SETLIST subcommand is used to create and modify the list of users (the "CCLIST") that will be notified of all action taken by RESLIM against the issuing user, if the CC option has been specified in a SETACT subcommand.

operands: <CURRENT|ALL|UP1|UP23> and 'type' are as described above under the SETLIM command. If ADD is specified, the userid 'name' is added to the user's CCLIST. If DROP is specified, the userid 'name' is removed from the list. RESLIM checks for the validity of the userid given, before processing the command.

note: The special forms:

RESLIM SETLIST <CURRENT|ALL|UP1|UP23>
type DROp name (uid

and

RESLIM SETLIST <CURRENT|ALL|UP1|UP23>
type DROp name (ALL

may be used to remove one's own userid from the CCLIST of another user (with userid 'uid'), or from the CCLISTS of ALL users.

examples: RESLIM SETLIS CURR CPUTIME ADD GIMLET
Adds GIMLET to the list of users to be notified of actions due to CPU overruns, for this terminal session only.
RESLIM SETLIST UP1 CPUTIME DR FRING (TORPID
If issued by user FRING, removes FRING from the permanent prime-shift CPU-time CCLIST of user TORPID.
RESLIM SETLIST UP23 CPUTIME DROP PLESK (ALL

If issued by PLESK, removes PLESK from the permanent off-shift CPU-time CCLISTS of all users.

4. QUERY

- syntax: RESLIM Query <CURRent|ALL|UP1|UP23>
<EVery|type> <(uid)>
- function: The QUERY command displays (by way of CP MSGs) the current status of the issuing user or other VM users, including resource limits, enabled acts, CCLISTS and pending actions (if any).
- operands: <CURRent|ALL|UP1|UP23> and 'type' are as described above. If EVERY is specified, the status of both types of limits is displayed. EVERY is the default. If 'uid' is given, limits displayed will be for the user specified. If it is omitted, limits will be for the user issuing the QUERY.
- examples: RESLIM QUERY ALL CPUTIME (GORMOLE
will display the UP1 and (if issued in UP1) CURRENT limits for CPU-time, the enabled actions, the CCLISTS, and any pending actions against user GORMOLE.
RESLIM Q
is equivalent to RESLIM QUERY CURRENT EVERY.

NOTES:

1. When the shift changes from UP2 and 3 to UP1 or vice-versa, all limits are reset to the default limit for the period being entered, exactly as if the user had logged off and back on again at precisely 9 am or 5 pm. Any previously issued RESLIM commands affecting only the user's CURRENT status are thus no longer in effect.
2. When the PRIORITY option of the SETACT command is enabled, (and, if WNG is enabled, a warning has already been sent) and the corresponding limit is exceeded, RESLIM will set the user's priority to 99. Raising or removing the limit will immediately restore the previous priority.
3. RESLIM keeps a list of userids which are able to change the status of others ("RESLIM administrators"). If your RESLIM status changes without your doing it, it was probably one of these users.
4. When RESLIM, in attempting to send a message to someone on a user's CCLIST, gets a non zero return code from the CP MSG (indicating that the message's target is either logged off, disconnected, or has MSG set OFF), it will send a message to that userid on the other Yorktown VM system, via the RSCS link.
5. If you have set no RESLIM limits for yourself, a CPU limit (as determined by your department director) may still apply to you. Also, the following CURRENT VMSIZE limits may apply, unless you have made other arrangements:
Off Prime Shift — No Limit
11:30 am to 1:30 pm weekdays — 4 M (4096 Kilobytes)
Other Prime Shift — 2 M (2048 K)

6. No actions are taken against users violating VMSIZE limits unless the user's system disk and drum residence exceeds 256 pages.
7. The only command by which a nonprivileged user may access his VMSIZE limits is QUERY. If you think your VMSIZE limit, actions, or CClst should be changed, contact a VM consultant.

CITED REFERENCES

1. *Interactive Computing at IBM's Thomas J. Watson Research Center*, GK20-1194, IBM Corporation (1979); available through IBM branch offices.
2. L. H. Seawright and R. A. Mackinnon, "VM/370—a study of multiplicity and usefulness," *IBM Systems Journal* **18**, No. 1, 4-17 (1979).
3. W. J. Doherty and R. A. Kelisky, "Managing VM/CMS systems for user effectiveness," *IBM Systems Journal* **18**, No. 1, 143-163 (1979).
4. P. H. Callaway, "Performance measurement tools for VM/370," *IBM Systems Journal* **14**, No. 2, 134-160 (1975).
5. H. A. Anderson, Jr., M. Reiser, and G. L. Galati, "Tuning a virtual storage system," *IBM Systems Journal* **14**, No. 3, 246-263 (1975).
6. D. Ferrari, *Computer Systems Performance Evaluation*, Prentice-Hall Inc., Englewood Cliffs, NJ (1978).
7. E. C. Hendricks and T. C. Hartmann, "Evolution of a virtual machine subsystem," *IBM Systems Journal* **18**, No. 1, 111-142 (1979).
8. D. Sannella and G. Waldbaum, *On-Line Monitoring and Control of VM/370 Usage*, Research Report RC 7586, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (1979).

The authors are located at the IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598.

Reprint Order No. G321-5157.