

# NASA TECH BRIEF



NASA Tech Briefs are issued to summarize specific innovations derived from the U.S. space program, to encourage their commercial application. Copies are available to the public at 15 cents each from the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151.

## Real-Time Operating System/360

The Real-Time Operating System (RTOS) is a real-time control program developed to support the Apollo Spaceflight missions. The program has a cost savings advantage for other real-time applications, such as those having random inputs which require a flexible, fast data routing facility; display systems which can be simplified by a device independent interface language; complex applications (having many asynchronous paths of logic) which need added storage protection and data queuing; and high response problems that require rapid access to a large number of programs and/or data. RTOS is an example of high reliability software that has demonstrated its value during many hours of Manned Spaceflight support. Elements of this control program are now considered standard for real-time supervisors.

Independent task management is the basic control facility in RTOS/360. Processing under OS/360 is conducted by means of tasks or "units of work". An OS task depends on its creator in order to exist. It has the storage protect key of its creator, must have at least one load module executing under it, and its dispatching priority is a function of its creator's priority. An independent task, however, does not require the existence of its creator in order to exist—in fact, independent tasks can be created by the system. An independent task has a unique protect key; it can be dormant with no load modules executing under it, and it assumes its priority from the request it is processing. These characteristics make it possible for the real-time system to receive and process varying data loads rapidly and efficiently.

In order to handle multiple requests for a given independent task, RTOS must build and maintain a queue of work requests for the task. Information about each request is held in elements of a real-time work queue. By varying the characteristics of the queue, it is possible to control the order of or limit

the number of requests to be processed. Thus, by thorough queue management, the system's work flow can be regulated and core lockouts can be prevented.

The heart of RTOS is the routing capability. In a real-time environment data must be processed immediately when they appear. Data routing, acting as the interface between the hardware interrupt facility and the independent task management function, identifies the criteria which link a particular data type to an independent task. If an input message matches the current data definitions, it is routed to the task that will process it. If no match is found, the message is discarded. Messages for a given task can be allowed to accumulate until a specified number is reached before a request for the task is generated. Certain real-time functions require work requests to be generated after a given amount of time has elapsed. Time routing, acting as the interface between the time management and task management modules, accomplishes the job.

In order to meet the demands of a real-time environment, special input/output facilities are needed to handle input/output requests rapidly and efficiently, and to support special real-time I/O devices in use at the Manned Spacecraft Center. These functions are achieved through the use of the special Real-Time Access Method. With this access method, it is necessary to OPEN each data set or device only once during the course of a real-time job step, and it allows all tasks access to the device. The Real-Time Access Method supports standard and special purpose graphic and communication devices. By creating device independence, RTOS allows substitution of sequential devices (printers and tape units) for the special real-time I/O devices during testing.

A special facility was developed to handle the large amounts of data processed by the Real Time Computing Complex (RTCC). A special set of utility pro-

(continued overleaf)

grams was written to support data tables—arrays of data maintained on direct access disk storage in OS/360 partitioned format. The standard OPEN/CLOSE logic was eliminated, increasing the speed at which data could be read or updated. Many different tasks can access a given table at the same time. Data table integrity and consistency are achieved by “locking” the table, which prevents other tasks from accessing the table until it is “unlocked”. Thus, various segments of a table can be read through different requests and the user is assured that no update has taken place between requests.

One of the major features of RTOS is the ability to support many different types of display devices each with different internal format requirements. The Display Formatting Language (DFL), which consists of a set of reentrant subroutines loaded at system initialization, insulates the user from the unique characteristics of a given display and the internal format changes resulting from modifications to these devices.

In the standard OS/360 environment system, library routines are included within high-level-language-load modules produced by the linkage editor, frequently causing multiple copies of some routines to be present in main core at the same time. Real-time linkage resolves the problem by allowing load modules to reference common resident reentrant library routines. In addition, certain numerical constants are frequently needed by many different programs. By using real-time linkage to maintain these constants in common tables, the system coordinators can insure that all programs are using identical constants. Real-time linkages also allow parameters such as task priorities to be held in a common table. Thus, it is possible to “tune” the system by simply modifying the desired priorities rather than reassembling a large number of programs.

In the RTOS environment, Large Core Storage (LCS) is used as secondary storage. Associated with the LCS units is the storage channel feature which allows a selector channel to access memory independent of the central processor. Programs and data are brought into LCS from direct access storage in anticipation of their use by a set of support programs called LCS allocation. When a load module is requested by a task, it is transferred from LCS into main memory for execution by a set of support routines called the Large Core Storage Access Method. LCS provides a significant reduction in the overhead related to the repeated loading of modules into main memory. As LCS becomes filled, the allocation routines examine the modules and purge those which have the least use.

RTOS has a complete spectrum reliability and testing features to qualify its use in complex real time environments. Both an input simulator and statistics gathering facility are available for extensive debug and analysis of the application. Catastrophic

hardware recovery facilities for CPU, channel, and devices are provided through extensive switching and restart features. Intermittent hardware and software failures are recovered efficiently through retry and refresh logic to ensure continuous support of the application.

### Notes:

1. It is possible to run RTOS on an IBM System/360 Model 50; 512K bytes of high speed memory are required.
2. Two updates to the program have recently been achieved. The first update added the following capabilities:
  1. Dynamic Large Core Storage (LCS) allocation
  2. System Task
  3. Alternate Device Support
  4. High Speed Restart
  5. Performance and Reliability improvements which represent an improvement in the total system.

The second update added the following capabilities:

1. OS/360 Release 14
  2. OS/360 Release 15 Input/Output Supervisor
  3. OS/360 Release 16 FORTRAN H Compiler
  4. Multi-jobbing Capabilities
  5. Job Step Timing
  6. Public/Private Device Allocation
  7. Job Scheduler Allocation Recovery
  8. Management Information System (MIS)
  9. Houston Automatic Spooling Priority System (HASP) Version II
  10. Capability to introduce jobs to HASP from another job
  11. Single 2314 System Residence Volume
  12. System Recovery Facility (SRF)
  13. General Performance improvements to RTOS facilities
  14. Main Core Storage 2K Block History Utility
  15. Stand Alone Restore program for 2314's.
3. Inquiries should be made to:  
COSMIC  
Computer Center  
University of Georgia  
Athens, Georgia 30601  
Reference: B69-10386

### Patent status:

No patent action is contemplated by NASA.

Source: R. L. Hoffman, H. H. Mueller, R. S. Kopp, W. D. Pollan, B. W. Van Sant, and P. W. Weiler of IBM Corporation under contract to Manned Spacecraft Center (MSC-12148)