Technical
Paper
2426
C2
January 1985

Operating System for a Real-Time Multiprocessor Propulsion System Simulator

Users Manual

Gary L. Cole

NASA

NASA Technical Paper 2426

1985

Operating System for a Real-Time Multiprocessor Propulsion System Simulator

Users Manual

Gary L. Cole

Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics and Space Administration

Scientific and Technical Information Branch

CONTENTS

SUMMARY

The NASA Lewis Research Center is developing and evaluating experimental hardware and software systems to help meet future needs for real-time, high-fidelity simulations of air-breathing propulsion systems. Specifically, the Real-Time Multiprocessor Simulator project focuses on the use of multiple microprocessors to achieve the required computing speed and accuracy at relatively low cost. Operating systems for such hardware configurations are generally not available. A real-time multiprocessor operating system (RTMPOS) that supports a variety of multiprocessor configurations has been developed at Lewis. With some modification, RTMPOS can also support various microprocessors. RTMPOS, by means of menus and prompts, provides the user with a versatile, user-friendly environment for interactively loading, running, and obtaining results from a multiprocessor-based simulator. This report is a users guide for RTMPOS. The menu functions are described and an example simulation session is included to demonstrate the steps required to go from the simulation loading phase to the execution phase.

INTRODUCTION

Simulations of complex dynamic systems require a versatile interface between the user and the simulation computer. To maximize the usefulness of the simulation as an engineering tool, the interface should provide a user-friendly means of programming, interacting with, and obtaining results from the simulation. In the past, different types of computers have satisfied some but not all of these requirements. For example, the use of analog computing equipment provides immediate results and an extremely versatile user/simulator interactive capability. But efficient programming and changing of the programs can be difficult. On the other hand, digital computers are more easily programmed but their operating systems are generally not designed to provide the interactive capabilities needed for many simulation tasks.

Simulations are becoming more sophisticated in terms of both the complexity of the systems being modeled and the level of model detail being sought. There is also a need for the simulations to execute in real time (e.g., testing of control algorithms and hardware-in-the-loop tests). More powerful computer hardware and operating systems software are needed to support these real-time simulation efforts. An example is the real-time multiprocessor simulator (RTMPS) being studied at Lewis. The objective of the RTMPS project is to develop and evaluate experimental hardware and software systems for the real-time, interactive simulation of air-breathing propulsion systems. The RTMPS project is focusing on the use of multiple microprocessors to achieve the required computing speed and accuracy at low cost relative to hybrid and mainframe digital computers. Although the required hardware is complex, it is

anticipated that the desired user-friendliness can be achieved through proper design of the operating system.

Grant and contract efforts related to the RTMPS project have been documented in a number of reports. Investigations of computer architecture and related hardware requirements are described in reference 1. References 2 and 3 discuss possible approaches to partitioning simulation models for solution on multiple microprocessors. Requirements for high-level programming languages for multiprocessor systems are discussed in reference 4.

The Lewis in-house RTMPS efforts, to date, are documented in references 5 to 9. Reference 5 describes, in general, the multiprocessor simulator concept. The designs of experimental hardware now in use are given in reference 6. An overview and users manual for a real-time multiprocessor programming language (RTMPL) is documented in references 7 and 8.

A real-time multiprocessor operating system (RTMPOS) has been developed at Lewis to support experimental versions of the RTMPS. An overview of the design philosophy and features of the RTMPOS is presented in reference 9. The RTMPOS was designed to allow interactive, engineering-level programming and operation of a variety of multiprocessor configurations. To provide this flexibility, RTMPOS makes use of information contained in data-base files of the type that are generated by the RTMPL utility (refs. 7 and 8). The RTMPOS provides special functions and tasks that operate in conjunction with a manufacturer-supplied disk operating system (DOS). Most RTMPOS tasks are written in Pascal and should require little or no modification to be compatible with other disk operating systems. The remaining tasks and routines, written in assembly language for the Motorola MC68000 microprocessor, provide interfaces between Pascal, the DOS, and the simulator. Although these assembly programs are generally quite simple, the extent of the modifications required to adapt them to other multiprocessor systems would of course depend on the particular DOS and type of microprocessor being used.

This report is intended as a guide to the use of the RTMPOS. The report begins with a description of the general multiprocessor architecture that is supported by the RTMPOS. Next, the RTMPL programming procedure is described to show how RTMPL output files relate to the RTMPOS. Then the general programming and operational functions that are provided by the RTMPOS are described. Next the RTMPOS menu selections are described. Finally an example of a simulation session is provided to illustrate the steps required to go from program loading to execution.

GENERAL SIMULATOR CONFIGURATION

The real-time multiprocessor programming language (refs. 7 to 8) and the real-time multiprocessor operating system (ref. 9) were developed to provide user-friendly programming and operation of the general simulator configuration shown in figure 1. The following paragraphs are intended to familiarize the reader with this configuration and its associated terminology. Additional details can be found in references 6 to 8.

The primary elements in this configuration are the multiple simulation channels (1 to n), the front-end processor (FEP), and the real-time interface.

The real-time interface provides for communications between the simulator and external devices (e.g., strip-chart recorders and controls). The FEP serves as the simulator controller and the interface between the user and the simulator. These functions are facilitated by the RTMPOS, which resides on the FEP. Data are transferred between the FEP and the simulation channels via the interactive information bus. The RTMPOS provides for simulator run-time operations such as program loading, parameter changes, simulator mode control, and data handling. The FEP also serves the simulator peripherals (terminals, disk drives, printers, etc.). The manufacturer-supplied DOS provides such typical utilities as an assembler, a linkage editor, a text editor, and file-handling services. The resident DOS has a multitasking capability (i.e., it permits many tasks to run concurrently in a time-slice mode). This capability is essential to the RTMPOS concept since the RTMPOS includes several tasks that permit the simulator to send advisories to the user via the FEP while the simulation is being run.

Each simulation channel (fig. 1) consists of two processors - a computation processor (COMP) and a preprocessor (PREP). Each COMP executes its assigned portion of the simulation (an RTMPL program) and interfaces its channel to the FEP. The PREP's also execute RTMPL programs and distribute information from the COMP's to the other channels via the real-time information bus. The COMP and PREP in a channel communicate through a shared memory. Each simulator processor contains specially designed firmware (software burned into EPROM's) that governs the communication between the FEP and the COMP's and between the COMP's and the PREP's. The firmware performs such functions as initializing and checking the processor memory, transferring data, setting the execution mode of the processors, sending interrupts out to the FEP, and, in some cases, timer control.

One of the simulation channels (1 to n) is assigned special tasks to perform. It serves as a real-time extension of the FEP and is known as the RTX channel. Its COMP is available to perform any real-time analysis required by the user to support the simulation. For example, it could be used to sample and process data from the other channels, which are then uploaded to the FEP by means of the RTMPOS. The RTX PREP is used to distribute data on the real-time information bus and serves as controller for that bus. It is also responsible for all timing and control of simulator operations. The remaining channels are designated as digital simulation computers (DSC's) and are available for solving the simulation equations.

The communication paths that are available in the general simulator configuration provide a high degree of programming flexibility. As a result, the RTMPS can emulate a variety of multiprocessor systems. This flexibility also means that the RTMPOS design can be viewed as generic to a variety of multiprocessor systems that are subsets of the general configuration shown in figure 1. The only hardware constraint is that the FEP must be able to communicate with the RTX COMP.

The current Lewis RTMPS experimental hardware, the FEP, and the peripheral equipment are shown in figure 2. The Lewis RTMPS uses a Motorola EXORmacs[1] development system as the FEP, and the resident DOS is Motorola's VERSAdos[1] (ref. 10). The FEP and the simulator processors are based on the Motorola MC68000 microprocessor with an 8-MHz clock.

---

[1]Trademark of Motorola Inc.

FORMULATING A SIMULATION

Before programming a simulation for the RTMPS, a user must partition the simulation equations into segments that can be solved in parallel. Those segments are assigned to the computing elements available in a particular RTMPS configuration. The simulation is then formulated by using the RTMPL utility (ref. 8). The RTMPL allows the user to program the various elements of the simulator in a high-level, engineering-oriented language. As shown in figure 3, the output of the RTMPL is a set of assembly language source program files (one for each processor that is being used in the simulator) and a set of simulation data-base files that relate the actual simulation implementation to the RTMPL source programs.

Each source program file consists of records that are given one of the following identifiers: VARIABLE, CONSTANT, ARGGROUP, EXEC, or TASK. The EXEC and TASK records for each source program define the executable parts of the corresponding simulation segment. The VARIABLE, CONSTANT, and ARGGROUP records define the local arguments for the EXEC and TASK records. A program file must contain at least one executive (EXEC record). Executives are like main programs and are used to govern the computational flow of the program. Executives have the ability to perform tasks. TASK records are used like subroutines to construct a more readable and versatile program. EXEC's and TASK's can have names up to eight characters in length.

VARIABLE records define the time-dependent variables that appear in the simulation equations. In RTMPL, variables are assigned a set of attributes: size, data type and precision (DTP), scale factor (SF), hold value, and initial-condition value (IC). The DTP and SF attributes are described in reference 8. Size refers to the number of current and past value terms that the variable has; that is, how many current and past values are to be saved in the simulation. The minimum variable size is 1 (i.e., only the current value is saved). The hold value is used to set the value of the variable during HOLD mode execution of the program. The IC value is the starting value of the variable that is loaded when the simulation is initialized. Hold and IC values can be changed at run time by means of RTMPOS commands.

In RTMPL, constants also have DTP, SF, and size attributes. Size for a constant specifies the number of elements in a multivalued constant array. The minimum size is 1. Only constants that have been specified as parameters are adjustable through the RTMPOS at run time.

An argument group (ARGGROUP) is a set of arguments that can be referred to by a single name. The arguments may include constants that are local to the program and variables that are local or external. An external variable is a variable that is used in a program but whose value was computed in and transferred from another program. Argument groups provide a convenient mechanism for transferring large volumes of data between the FEP and the simulator channels. Argument groups have DTP and size attributes. For ARGGROUP's, the size attribute refers to the maximum number of arguments that can be contained in the group. Arguments may be added as long as the specified size is not exceeded and the argument has the same DTP as the ARGGROUP. Arguments in the ARGGROUP may be added, deleted, or replaced at run time through the RTMPOS.

The data-base files that are generated by the RTMPL are read by the RTMPOS. These files include information that the RTMPOS needs to support truly interactive execution of the simulation on the multiprocessor-based simulator. For example, a CONSTANT record specifies its name, DTP, SF, size, value, and assigned memory location. Reference 8 contains a sample data-base listing that includes the names of the data-base files as well as their contents.

The data-base files include definitions of global constants, messages, operating system tasks, and general groups. Global constants have the same attributes as local constants but may be referenced in any program. Messages are advisories that are to be relayed from the simulator to the user via the RTMPOS under certain run-time conditions. The messages are defined by the user while the simulation is being programmed but may be modified at run time through the RTMPOS. As described in reference 8, the RTMPL utility provides for the definition of operating system tasks (OSTASK's) that the RTMPOS will perform upon initiation from the simulation. However, the current implementation of the RTMPOS does not support OSTASK's. That capability could be added as a future enhancement. A general group (GENGROUP) is very similar to an argument group. Two major differences are (1) that the general group cannot be specified through the RTMPL utility but can be specified by the user at run time and (2) that the general group is not tied to a specific program (i.e., it can include variables and constants that are computed on different processors). Also, there is considerably more flexibility in adding, deleting, or changing items in the GENGROUP. There is no restriction on item attributes (e.g., DTP's do not have to be the same). Like argument groups, general groups provide for large-volume data transfers between the FEP and simulator channels (e.g., displaying values of local variables in multiple channels).

OPERATING SYSTEM FUNCTIONS

All RTMPL data-base and program load-module files are saved on disk files that can be manipulated by the RTMPOS. Once the simulation has been formulated, the RTMPOS provides the following general programming and operational functions:

(1) Simulation loading/IC and program control
(2) Data-base management
(3) Simulator initialization and mode control
(4) Run-time monitoring
(5) Simulation results management
(6) Miscellaneous

Simulation Loading/IC and Program Control

The previously described source program files are assembled and linked by using the FEP-resident assembler and linkage editor to form program load modules. At run time this function allows for loading the modules into the simulator processors, setting the initial-condition values of simulation variables, activating the desired EXEC in each program, and enabling/disabling tasks (if any) to be performed under each EXEC.

Data-Base Management

Data-base management functions include loading the data-base files into the FEP memory, editing the data base, saving an edited data-base file on disk, and listing the data base on a terminal or printer. The editing function allows the user to change the simulation at run time. For example, IC and hold values of variables, values of local or global constant parameters, messages, and items contained in arguments and general groups may be changed. The simulation programs are automatically updated to reflect the data base currently loaded in the FEP regardless of whether the programs are loaded into the simulator before or after the editing is done. However, the load-module and data-base disk files themselves remain unchanged. The edited data base may be saved at any time for future use, either by overwriting the original data-base disk files or by saving a new set of files. Note that saving the data base does not affect the RTMPL source code files. Hence the changes will not appear in later RTMPL versions of the simulation unless changes are made in the original source code.

Simulator Initialization and Mode Control

Mode control refers to the user controlling the execution mode of the simulator. There are four possible modes - stop, run, hold, and cycle. When the simulator is in the stop mode, execution of all simulation programs ceases. The simulator must generally be in the stop mode to do data-base editing. For example, initial-condition or hold-condition values can be loaded only when the simulator is in the stop mode. The other simulator modes (i.e., run, hold, and cycle) can only be entered from the stop mode. In the run mode, all simulation programs are executing repeated update cycles. The simulator continues to execute until the user issues a stop command or the simulator or simulation itself issues a stop command. In the latter two cases, the user is issued an advisory, via the message device, that the simulation has stopped. In the hold mode, variables in each program are held at user-specified, fixed values. Once the user specifies which variables are to be held, all programs in the simulation execute a user-specified number of update cycles. The simulator then returns to the stop mode and an advisory is issued that the hold mode is complete. At present, the cycle mode is undefined and has not been implemented.

Simulator initialization is a hardware test that checks COMP-to-PREP communications and the shared memory within each channel. This must be done before the simulation program load modules can be loaded and is usually done at the beginning of a simulation session. The process may have to be repeated in the event of a processor failure or hangup. The simulator update-interval and analog-to-digital channel timer clocks are first set when the simulator is loaded. This function provides for making subsequent changes to these clocks.

Run-Time Monitoring

The run-time monitoring function allows a user to receive advisories via interrupts from the simulator and to record all actions that are taken during a simulation session. The advisories can be user or operating-system messages or they can be requests to read data from the simulator. The messages are displayed on the user-designated message device. Current (M) and halt (H)

advisory messages that are generated by the operating system are listed in table I. The read advisories would be built into the simulation by the user and would be transparent to the user at run time.

A powerful feature of run-time monitoring is the self-documenting session history - a disk text file that saves all user menu selections and resulting prompts and messages from the RTMPOS. Any advisory messages that occur are automatically entered into the session history, as are user-generated comments. The session history provides a means for correlating the session with the data file and can be listed and edited by using the manufacturer-supplied utilities that are resident on the FEP. Furthermore a session history can be executed by the RTMPOS. That is, menu selections can be input to the RTMPOS from a session history file rather than entered manually from the keyboard. This allows the user to quickly bring a simulation to a previously obtained condition by executing the session history from that session. The user may also create a file with the text editor that can be used to execute the routine parts of a simulation session (e.g., loading a data base and the simulation program load modules). After the execution of the session history is completed, control is returned to the user at the keyboard. An example session history is presented in a later section.

Simulation Results Management

The simulation results management function provides the capability of saving, listing, and plotting simulation results. The user can display simulator CPU time and simulation time, computation time for individual processor programs, and delay times, if any, for advisories and external variable transfers. The user can also display values of simulation variables as they are computed although this slows simulation execution time.

Data can be saved by either of two mechanisms. The READ advisory (ref. 8) is built into the simulation by the user through the RTMPL. The READ advisory causes an interrupt that initiates an RTMPOS interrupt service task. The service task reads the user-specified argument group directly from the designated processor to an RTMPOS auxiliary storage memory. The interrupt process is transparent to the user. When the simulator is put in the stop mode, the data are automatically saved in the user-specified data disk file. These data can then be listed or plotted. At the present time, plotting is done outside the RTMPOS.

The second and more versatile mechanism for saving data is by using the so-called analysis functions. These allow the user to specify general groups at run time that save sampled values of variables, save maximum and minimum values of variables, and specify the maximum allowable rate of change of the variables. The sampling process works in the same manner as the read advisory. Interrupts are issued at a sampling rate that is specified by the user. Once the sampling process is complete, an advisory message is issued on the message device. The user must then initiate saving of the data to the data file. The user may display the maximum or minimum values at any time. If a variable exceeds the maximum rate of change specified by the user, the simulation stops and an advisory is displayed on the message device. The particular variable having the rate-of-change violation is displayed on the user's terminal.

Once the RTMPOS has been invoked by the user, there are two ways to exit from it. Using the quit option terminates all of the RTMPOS tasks permanently. Alternatively, the user can temporarily suspend the RTMPOS, perform other tasks, such as editing and listing a text file, and then resume the RTMPOS at the point where she/he left off. In the latter case the simulator can continue to run with any advisory messages output to the message device as usual. The data read advisories are also processed.

As a diagnostic aid, the user has the option of reading or writing memory locations in the simulator processors, the shared memory blocks, and the RTMPOS auxiliary memory block.

INVOKING THE OPERATING SYSTEM

Normally, the first step in invoking the RTMPOS will be to log onto the FEP. In the case of the Lewis system, this is done by

(1) Turning on the system power (including the simulator)
(2) Pressing the keyboard return key <CR> after the cursor appears on the monitor
(3) Pushing the system boot button on the FEP (EXORmacs) chassis control and indicator panel
(4) Pressing the break key on the terminal keyboard

As shown in figure 4, the user will then be prompted for information. (Note that the user should turn on the all-caps key before making entries.) In the following example the user responses are underlined to distinguish them from system prompts:

ENTER DEFAULT SYSTEM VOLUME: USER No. = SYS:0 <CR>

ENTER DATE (MM/DD/YR) = 3/28/84 <CR>

ENTER TIME (HR:MIN) = 15:10 <CR>

A series of messages are then displayed on the screen. The messages are followed by a line displaying an equal sign followed by the cursor. The user then enters "MPOS.CF S <CR>." This entry invokes a chain file (a series of commands) that results in the RTMPOS tasks being loaded from disk files into the FEP. If the FEP has been logged on and is in use, the user should enter "USE SYS:0 & <CR>" followed by "MPOS.CF S <CR>."

The next user entry, "STAR <CR>," causes the RTMPOS to start executing. A message is displayed first showing what version of the RTMPOS has been accessed. (The functions provided by version 1.0, 090684, are described in this report. The addition or deletion of functions is indicated by a new version number.) The RTMPOS then goes through an initialization routine that attaches the simulator processor memory and the auxiliary memory to the RTMPOS tasks. A message is displayed to indicate whether or not the memory was successfully attached. The user is then prompted to designate the message device and a file for saving session history. The user may opt to use the default names. However, if the option to use the default session history file is

elected, it must be saved by copying it to another file at the end of the session or it will be lost the next time the RTMPOS is invoked.

The user is now ready to proceed with loading and execution of a simulation. The user is prompted to press the keyboard return key <CR> to proceed to the main menu of RTMPOS functions or to enter a desired simulator mode command. The following section describes the main menu and the submenus, which reflect the major functions described earlier. Note that the simulator mode can be set by entering "S" for stop, "R" for run, "H" for hold, or "C" for cycle in response to any main or submenu selection prompt.

USING THE OPERATING SYSTEM MENUS

The RTMPOS has a main menu and five submenus. A summary of all menu selections for RTMPOS version 1.0, 090684, and the page they are discussed on is given in table II. A "<CR>" in response to a submenu selection prompt returns the user to the main menu. After a menu selection is made, the RTMPOS will prompt the user for additional inputs if necessary. An illegal or improper entry will cause either appropriate error messages to be displayed or a repeat of the prompt. (The appendix lists abbreviations and symbols that are used in the RTMPOS prompts and error messages. Not all of them are referred to in this report.) A general rule to remember is that entry of "#" or any set of characters terminated by "#" will cause exit from the current level of execution. Entry of "#" in response to more than one prompt may be required to exit a functional level. This is illustrated in the section Example Simulation Session.

If the required response to a menu prompt is the name of an EXEC, TASK, or ARGGROUP record, the syntax of the response is as follows:

PROGRAM.TYPE.RECORD<CR>

Here, PROGRAM and RECORD denote the actual program (channel) and record names, which can be eight characters long. TYPE denotes which processor type the record is associated with and is entered as a "C" (COMP) or "P" (PREP). The periods are used as delimiters. The RTMPOS always provides and displays to the user a default program name and a default processor type. Therefore the program name or processor type must be entered only when it is different from the default. The following entries represent valid record names for EXEC, TASK, or ARGGROUP records:

PROGRAM.RECORD<CR>

.TYPE.RECORD<CR>

RECORD<CR>

Whenever a new program or processor type is entered, it becomes the new default value and will be displayed by the RTMPOS. Since a GENGROUP is not associated with a specific simulation program, only its name, up to eight characters long, is entered in response to a prompt.

If the required response to a menu prompt is the name of a CONSTANT or VARIABLE record it is entered as follows:

PROGRAM.TYPE.CONSTANT$ITEM<CR>

or

PROGRAM.TYPE.VARIABLE$ITEM<CR>

The syntax is similar to that described earlier except for the addition of another delimiter ($) and ITEM, which denotes the item number of interest in the particular CONSTANT or VARIABLE array. (Recall that constants and variables have a size attribute.) The following entries represent valid record names for CONSTANT or VARIABLE records.:

PROGRAM.CONSTANT$ITEM<CR>

.TYPE.CONSTANT$ITEM<CR>

CONSTANT$ITEM<CR>

CONSTANT$<CR>

CONSTANT<CR>

The use of "$ITEM" in an entry is interpreted as wanting the specified item and all succeeding items in the record. The use of only "$" is interpreted as wanting all items in the record. The omission of "$ITEM" is interpreted as wanting the first item or all items depending on the menu selection being used. Entry of a nonexistent item number will result in an error message. Although ARGGROUP's and run-time-generated GENGROUP's also have size attributes, they are treated as entities for most RTMPL operations. The use of "$" or "$ITEM" with their entry is ignored or treated as an error by the RTMPOS.

Responses that involve disk file names must be entered in accordance with the FEP DOS rules for naming files. In the case of the Lewis FEP, a VERSAdos file name consists of five parts delimited as follows:

VOLUME:USER.CATALOG.FILE.EXTENSION

Each part (except USER) is alphanumeric and must begin with an alphabetic character. "VOLUME" refers to the disk name and may be four characters long. "USER" refers to the user number and must be a number up to four characters long. "CATALOG" and "FILE" are names up to eight characters long and "EXTENSION" must be two characters long. Whenever a file name must be entered, the user is prompted for the individual parts except for the extension, which is always supplied by the RTMPOS. In some cases, the user is prompted only for "VOLUME" and "USER".

Recall that all user entries must be followed by pressing the return key "<CR>".

Main Menu

The main menu of RTMPOS functions is shown in figure 4. The simulator mode is set by entering the first letter of the desired mode (e.g., "R" for run). When displayed on the user's terminal, the current simulator mode is indicated by an asterisk. A cycle mode has not been implemented in the current version of RTMPOS. The allowable simulator modes are stop, run, and hold.

Stop mode. - "S" allows the user to put the simulation in the stop mode when it is in run or hold. In response to the "S" entry, the RTMPOS waits for notification from the simulator that it is in stop. The user is notified whether or not the simulator goes to stop. When a stop is executed, any user or operating system M and H advisories generated since the last stop are automatically saved in the session history, and read advisory data (but not sample data) are automatically saved in the data file. System M and H advisories are saved in the session history only by number. The system message numbers and corresponding messages are given in table I.

Run mode. - "R" allows the user to put the simulator into the run mode. In response to the "R" entry, the RTMPOS waits for notification from the simulator that it is running. The user is notified whether or not the simulator goes to run. The current run/hold number is displayed on the user's terminal and on the message device. The run/hold number is an identification number for transients during a simulation session. This number is used to retrieve data from the data file for listing and plotting. The run/hold number is displayed on the user's terminal and message device and entered in the session history each time the user puts the simulator in the run or hold mode.

Hold mode. - "H" allows the user to put the simulator into the hold mode. The hold mode must be entered from the stop mode. The user is first asked whether or not the same variables are to be held again. If no previous hold cycles were executed or if the user wishes to change the hold variables, the response is "N"; otherwise the response is "Y." The user may specify the program variables to be held either individually or by means of argument groups or general groups. Specification of the hold variables by means of a general group is recommended as the most convenient method. Variables to be held must be specified by their present values. After specifying the variables to be held, the user is prompted for the number of update cycles to be executed in the hold mode. The user is notified whether or not the simulator goes to hold. When the proper number of cycles has been executed, the simulation is automatically placed in stop and the user is notified, via the message device, that hold cycling is complete. The user may stop the simulator before completion of the hold cycling by using the "S" entry. In some circumstances (such as debugging) the user may wish to have the simulation execute a specific number of update cycles. The hold mode can be used for this purpose. When the user is prompted for the first variable to be held, entry of "#" will result in no variables being held and the simulation will execute normally. The current run/hold number is displayed on the user's terminal and on the message device when the simulator goes to hold.

Other functions, the general nature of which were described earlier, can be selected by entering the proper number. Entering a number between 3 and 7 will result in the display of a submenu.

Function 1, initialize simulator hardware. - The user is first prompted to reset the simulator processors (i.e., push the reset button on all simulator processors). A "N" entry causes return to the main menu. A "Y" entry indicates that the reset buttons were pushed and the hardware test then proceeds automatically. The user is notified which channels pass the hardware test and which channel is the RTX. If the RTX is not ready, the simulator cannot be used.

Function 2, display or set update interval and number of ADC channels. - The current values of the update interval and the number of analog-to-digital converter (ADC) channels are displayed first, followed by a prompt. A "Y" entry indicates that the values are acceptable and causes a return to the main menu. If a "N" entry is made, the update-interval and ADC-interval clocks will be reset. The user is first prompted for the update interval in microseconds. The value entered must be an integer multiple of 5 with a range of values from 100 to 81 915. The update interval is the length of time allotted to the processors to do the simulation calculations and required data transfers. If the update interval and the simulation step-size parameter are equal, the simulator will run in real time. The user is notified whether or not the simulator will run in real time when the clock is set following the specification of the step-size parameter.

The ADC-interval clock is set by specifying the number of analog-to-digital channels that are being used. The allowable number of channels will be governed by the length of the update interval. If too many channels are specified, the RTMPOS notifies the user. Both clocks may be set any time after the hardware has been initialized. The user is always prompted to set the clocks when a simulation is loaded.

Function 8, suspend RTMPOS temporarily. - The user can temporarily suspend operation of the RTMPOS without affecting operation of the simulator (i.e., the simulator will stay in its current mode). The user may then invoke any command recognized by the manufacturer-supplied operating system (e.g., VERSAdos). To resume operation of the RTMPOS, the user enters the command "MPOS.CF." (When using the Lewis RTMPS, if the simulator is running and no user entries are being made to the RTMPOS, it is recommended that the RTMPOS be suspended. This will prevent the RTMPOS task from being aborted by VERSAdos if RTMPOS is idle for about 15 min.)

Function 9, quit. - The simulation session is terminated if the current data base has not been edited or has been edited but previously saved. If the current data base has been edited but not previously saved, the user will be given an opportunity to save it before the session terminates. If the user elects to save the data base at that point, the RTMPOS will return to the main menu after the save process is complete, otherwise the session terminates automatically.

Selection of any of the following main menu functions results in the corresponding submenu display.

Function 3, Session History

```
SIMULATOR MODE (current mode displayed)
 (1) DISPLAY DATE AND TIME ON TERMINAL AND IN SESSION HISTORY FILE
 (2) ENTER A COMMENT IN SESSION HISTORY FILE
 (3) EXECUTE AN EXISTING SESSION HISTORY FILE
 (4) BE PROMPTED FOR FILE TO SAVE SESSION HISTORY
 (5) STOP SAVING SESSION HISTORY
<CR> OR SELECTION ? ..
```

The current simulator mode is displayed at the top. Even though the simulator mode selections are not displayed in the submenus, the mode can be set by entering the first letter of the desired mode in response to the selection prompt, as indicated earlier. A <CR> in response to all submenu selection prompts returns the user to the main menu:

Selection 1. - Selection 1 causes the current data and time to be displayed at the terminal and entered as a comment in the session history file.

Selection 2. - Selection 2 allows the user to enter a comment in the session history file (e.g., a description of the transient to be run). The user is prompted for the comment to be entered. A comment must be enclosed in asterisks, may contain any character (except an asterisk) and is limited to 64 characters. Two asterisks, spaced 64 characters apart, are displayed above the entry line as a guide to the user. An entry of "*#*" will return the RTMPOS to the menu. The current data and time are automatically entered in the session history file along with the comment.

Selection 3. - Selection 3 allows the user to execute the menu selections and responses to prompts contained in an existing session history file. The user is prompted for the name of the file to be executed. Session history is never saved during execution of the current session history file. Figure 5 shows a listing of a typical session history file that could be executed. Execution of the session history file ceases when the end of the file is reached. The RTMPOS then returns to the menu and a message is displayed if session history is not being saved. Warning! Care must be taken if the session history file to be executed contains selections to initialize the simulator, to load a data base, or to load a simulation. If it does, the following respective actions should be taken before executing the session history file: push the processor reset buttons, delete the current data base (submenu 4, selection 9), and unload the simulator (submenu 5, selection 6). Otherwise the session history will not execute properly. There is also a problem if the session history contains run (R) or hold (H) mode selections. User entries that are made while the simulator is in run or hold (e.g., display of simulation values or a stop (S)) are time dependent. This time dependence is not accounted for during execution of the session history. That is, each entry is executed in sequence as quickly as possible. In view of the aforementioned problems, the user must carefully edit a session history before executing it. Entries in the session history that will cause improper or unwanted actions by the RTMPOS can be deleted or made into unexecutable comments (to be explained later).

Selection 4. - Selection 4 allows the user to redefine the disk file for saving the session history. The user is prompted for the required information.

Selection 5. - Selection 5 allows the user to stop saving the session history file. After the command is entered, no further information is written to the session history file until item (4) of this submenu is selected.

Function 4, Data-Base Loading and Management

```
SIMULATOR MODE (current mode displayed)
 (1) LOAD SIMULATION DATA BASE INTO FEP
 (2) DISPLAY/EDIT ARGUMENT GROUP DEFINITIONS
 (3) DISPLAY/EDIT VALUES OF SIMULATION CONSTANTS OR VARIABLE IC/HOLD
     VALUES
 (4) DISPLAY/EDIT GENERAL GROUP DEFINITIONS
 (5) DISPLAY/EDIT USER-DEFINED ADVISORY MESSAGES
 (6) SAVE CURRENT EDITED DATA BASE ON DISK(ETTE)
 (7) GET LISTING OF CURRENT DATA BASE
 (8) DISPLAY DATA BASE STATUS
 (9) DELETE CURRENT DATA BASE FROM FEP MEMORY
<CR> OR SELECTION ? ..
```

Selection 1. - Selection 1 allows the user to load a simulation data base from the disk file into FEP memory. The user is first prompted for the name of the simulation definition file that is contained in the RTMPL data base (catalog name SIMDEF). The RTMPOS then displays the identifying contents of the file so that the user can be sure she/he has selected the right file. The user then has the option of loading that data base, selecting a new one, or quitting. Once data-base loading begins, the entire data base is loaded into FEP memory without any prompting of the user. After the data base has been loaded, the user may proceed with any of the data-base-management menu selections or go on to load the simulator. The simulator is automatically unloaded if it was previously loaded.

Selection 2. - Selection 2 allows items to be deleted, inserted, or substituted for other items in an argument group. The user is first prompted

ENTER ARGUMENT GROUP NAME . .

Entry of "#" will return the user to the menu. To modify an argument group named DATA, of type C, associated with program DATAPROC, the user response would be as follows:

ENTER ARGUMENT GROUP NAME . . DATAPROC.C.DATA

or simply "DATA" if the default program and processor type are DATAPROC and C, respectively. The resulting prompt is

ENTER ARGUMENT GROUP ITEM NUMBER OR NAME . .

Here, entry of "#" returns the user to the argument group name prompt. Otherwise, the user must enter the name or number of the item in the argument group that she/he wishes to modify. For example, if the second item in the argument group DATAPROC.C.DATA is to be changed and it happens to be the present value of DATAPROC.P.ANF (namely P.ANF$1), the user could enter either ".P.ANF$1" or "2" to modify that item in the argument group. The following display would then appear:

DATAPROC.C.DATA$2 = DATAPROC.P.ANF$1 (XV) . .

This prompt verifies that the second item ($2) in the argument group DATAPROC.C.DATA is the present value ($1) of ANF, from the PREP (P) processor for program DATAPROC, and that it is an external variable (XV). The user then has the following choice of entries:

Entry	Meaning
#	return to the argument group name prompt
<CR>	no change to this item - display the next item in the argument group (if there is one); otherwise, return to the argument group name prompt
/	delete this item from the argument group - all subsequent items in the group are renumbered accordingly
<	insert a new item in the argument group before the current one
>	insert a new item in the argument group after the current one
New item name	replace the current item with this new entry

An entry of "<" or ">" will result in the prompt

ENTER ARGUMENT TO BE INSERTED . .

The user may respond with "#" or the name of the item to be inserted. An item that is inserted or used to replace the current item can be a constant, a variable, or an external variable that is available to the argument group's program. But the new item must have the same DTP as the argument group. An argument group may be added to another argument group in the same program provided that enough space has been allocated and that the DTP's are the same. An argument group is automatically updated in the simulator if it is loaded and at the time of loading.

Selection 3. - Selection 3 allows values of local and global constants in the simulation to be displayed and values of parameters to be modified. IC and hold values of simulation variables may also be displayed/modified. The user is first prompted

ENTER CONSTANT OR VARIABLE NAME . .

Entry of "#" will return the user to the menu. An example of a valid entry is

ENTER CONSTANT OR VARIABLE NAME . . CORESIM.P.PRC.

Again, CORESIM or P may be omitted if they are default values. The display

CORESIM.P.PRC = 1.0000E+00 . .

would indicate that the present value of the constant is unity. The user then has the following choice of entries:

Entry	Meaning
#	return to the CONSTANT/VARIABLE name prompt
<CR>	do not change the constant's value - display the next item in the constant's array if there is one; otherwise, return to the CONSTANT/VARIABLE name prompt
D	replace the constant's value by the current default value
New value	replace the constant's value by the new value. The value may be entered with or without the engineering (E+00) notation

The default option allows the user to simply enter a "D" rather than the entire number. The default value is displayed whenever the user enters the CONSTANT/VARIABLE editing routine. Warning! Whenever a new value is entered and not rejected, it automatically becomes the default value. If the default value is used or a new value is entered, it must be consistent with the constant's DTP. An illegal value will be rejected with an error message and the user will be prompted again. In some instances the user may also receive a warning (e.g., when a decimal value is entered for a constant integer). In this case the value is not rejected unless it is out of range, in which case an error message would be given. In the event that the user modifies a global constant, it is automatically updated everywhere that the global constant is used in the simulation.

The procedure for displaying/modifying variable IC and hold values is essentially the same as that described for the constants. When a variable name is entered, the IC value will be displayed first regardless of selection number. The user will then have the same choice of entries as for a constant. Any entry other than "#" results in the hold value then being displayed with the same entry options. Warning! While the RTMPOS does allow present (ITEM = 1) and past (ITEM > 1) values of variable IC's to be different (i.e., simulation starting from a nonequilibrium condition), the firmware currently implemented results in all past values being set equal to the current value when the simulation IC's are set. Hence a simulation will start from a non-equilibrium condition only when a run (R), stop (S), run (R) sequence of mode changes is used.

Selection 4. - Selection 4 allows the user to build or edit a general group at run time. The initial prompt is

ENTER GENERAL GROUP NAME . .

Entry of "#" will return the user to the menu. Otherwise, the user must enter a name up to eight characters long - the first being alphabetic. No program name or processor type is attached to a general group. If the name entered by the user does not exist, the user is notified and a new general group by that name can be created. Processing of a general group and the choice of entries in response to prompts is the same as that for an argument group. Any constant, local variable, argument group, or general group may be added to a general group. When the user specifies an argument or general group to be added, the RTMPOS adds it item by item rather than as an entity. If an argument group contains an external variable, it is entered in the general group as coming from the source processor. There is no restriction on DTP or size. Once the general group is formed, it may be saved on a disk as part of the data base.

Selection 5. - Selection 5 allows the user to modify the user-defined M and H advisory messages that are sent to the user from the simulator via the FEP. The user is first prompted

ENTER MESSAGE NAME . .

Entry of "#" returns the user to the menu. Message names are alphanumeric, up to eight characters long, and are not specific to any program. For example, if the user wanted to modify a previously defined message named DUCTMESS, the entry would be

ENTER MESSAGE NAME . . DUCTMESS

The specified message would then be displayed followed by the prompt

OK (Y/N)? . .

A "Y" response would indicate that the message was all right - no change desired. The message name prompt would then be displayed again. A "N" response would mean the user wished to change the message. A message must be enclosed in asterisks, may contain any character (except an asterisk), and may be 64 characters long. Two asterisks spaced 64 characters apart are displayed above the entry line as a guide. An entry of "*#*" will return the RTMPOS to the menu.

Selection 6. - Selection 6 allows the user to save an edited simulation data base to disk files. If the data base has been loaded from a disk and has not been edited, the RTMPOS will not permit that data base to be resaved to a disk. However, the manufacturer-supplied copy utility may be used to save such files under a different name. If the RTMPOS determines that the data base has been edited and can be saved, the user is given the option of either overwriting the files from which the current data base was loaded or generating a completely new set of files. If the latter option is elected and there is sufficient room on the disk, it is recommended that different data bases for the same simulation be saved under different user numbers.

Selection 7. - Selection 7 allows the user to obtain a listing of the simulation data base currently residing in the FEP memory. Because the data base is not a text file, it cannot be obtained by using the manufacturer-supplied list utility. The user may elect to list all or part of the data base.

Selection 8. - Selection 8 allows the user to determine what data-base files have been edited. After the command is entered, either a message will be displayed indicating that the current data base has been saved or a prompt will be displayed asking if the user wants to display the names of the data-base files that have been edited.

Selection 9. - Selection 9 allows the user to free FEP memory by deleting the current data base. Only one data base at a time is allowed to reside in the FEP memory.

Function 5, Simulation Loading/IC and Program Control

```
SIMULATOR MODE (current mode displayed)
 (1) LOAD SIMULATION INTO SIMULATOR
 (2) SET SIMULATION IC'S
 (3) CHANGE BACKGROUND EXECUTIVES
 (4) ENABLE/DISABLE PROGRAM TASKS
 (5) DISPLAY PROGRAM STATUS (ACTIVE EXECUTIVES AND ENABLED TASKS)
 (6) UNLOAD SIMULATOR
<CR> OR SELECTION? . .
```

Selection 1. - Selection 1 initiates loading of simulation programs. A data base must reside in the FEP before this selection can be made. The user is prompted to select a channel for each simulation program that is specified in the simulation. If the configuration is DUAL, associated (same program name) COMP and PREP programs are automatically put into the same channel. The specified processor function (RTX or DSC) for each program must be compatible with the assigned channel or the assignment will not be made and an error message will be displayed. (When the simulator hardware is initialized, the RTX channel number is displayed.) Every program must have a channel assignment or an error will occur. Once the assignments have been made, the user is given the option to load or not to load. After the simulator is loaded, the RTMPOS automatically (1) updates each program's CONSTANT and ARGGROUP records to be consistent with the data base currently residing in FEP memory, (2) initializes variables, (3) enables/disables tasks, according to conditions originally set through the RTMPL utility, (4) activates the first zero priority background executive specified in each program, and (5) displays the simulation status showing the active executive and enabled tasks in each program. The user is prompted to enter the name of the integration step-size parameter, to set the update-interval and number of ADC channels, and to name the file for saving simulation results (data). The integration parameter should be specified as a global parameter when the simulation is being developed with the RTMPL utility. This makes it easier to modify at run time and to coordinate with the simulator update interval by means of the RTMPOS.

Selection 2. - Selection 2 allows the user to set the IC values of program variables in the simulator. The simulator must be in the stop mode. The user may specify the variables to be initialized either individually or by means of argument groups or general groups. There is also an option to set the IC values of all simulation variables without having to enter them individually or by means of a group. If not all variables are to be initialized, use of a general group is recommended as the most convenient method to specify IC's. Variables to be initialized must be specified by their present value. When variables are initialized, they automatically have their past values set equal to the present value by means of the simulator firmware. Whenever the simulation IC's are set, the run/hold number counter is incremented.

Selection 3. - Selection 3 allows the user to activate a specific zero priority executive in a selected simulation program file. Any other zero priority executives in the same program are automatically deactivated. The user is given the option of (1) entering the name of the executive to be activated or (2) reviewing the status of all executives with the opportunity to change the status of each. As stated earlier, eight characters may be entered for an executive name (plus program and processor type). However, only the first seven characters are recognized (ref. 8). Note that, when the simulation

programs are first loaded, the RTMPOS automatically activates the first zero priority executive specified in each simulation program.

Selection 4. - Selection 4 allows the user to enable or disable any simulation task - no other task is affected. The user is given the option of (1) entering the name of the task to be enabled/disabled or (2) reviewing the status of all tasks with the opportunity to change the status of each. As with executives, eight characters may be entered for a task name, but only the first seven characters are recognized. Note that, when the simulation programs are first loaded, each task is enabled or disabled according to the status initialized by the user through the RTMPL utility.

Selection 5. - Selection 5 causes the simulation status to be displayed, showing the active executive and enabled tasks in each simulation program. An example is shown in the session history file listing of figure 5 (lines 64 to 69).

Selection 6. - Selection 6 sets in the RTMPOS a Boolean parameter indicating that the simulator is not loaded. This selection must be used if simulation programs have previously been loaded and the user wishes to execute a session history file that contains a selection to load the simulator. Otherwise the session history file will not execute properly.

Function 6, Simulation Results Management

The selections in submenu 6 are

```
 (1) ACTIVATE/DEACTIVATE ANALYSIS FUNCTIONS
 (2) SAVE SAMPLE DATA, DISPLAY MAX/MIN VALUES
 (3) DISPLAY SIMULATOR CPU TIME AND SIMULATION TIME
 (4) DISPLAY COMP AND PREP COMPUTATION TIMES
 (5) BE PROMPTED FOR FILE TO SAVE SIMULATION RESULTS
 (6) LIST SIMULATION RESULTS DATA FILE
 (7) PLOT SIMULATION RESULTS DATA FILE
 (8) DISPLAY SIMULATION VALUES OF CONSTANTS, VARIABLES, AND ARGUMENT/
     GENERAL GROUPS
 (9) DISPLAY COMP AND PREP ADVISORY MAX-DELAY TIMES
(10) DISPLAY EXTERNAL VARIABLE DELAY TIMES
<CR> OR SELECTION? . .
```

Selection 1. - Selection 1 allows the user to activate or deactivate any of the four analysis functions: (1) sample values of variables, (2) save maximum values of variables, (3) save minimum values of variables, and (4) set maximum allowable rate of change of variables. The user is first prompted to select the desired analysis function or to quit. If the selected function is already activated, the user is so notified and given the option to abort it. Otherwise the user is prompted for the name of the general group containing the variables to be processed by the selected analysis function. The general group must already exist or an error message will occur. Each analysis function can process 16 variables. If the selected general group contains more than 16 items, only the first 16, whether constants or variables, will be used and a warning will be issued. All scaled fraction values are truncated to single precision and only single-precision integers are allowed. After setting up an analysis function, the user is prompted for another selection until the quit

option is selected. Except for the sampling process, analysis functions can only be aborted or deactivated by this submenu selection. The analysis functions can be set up with the simulator in any mode.

The sample process allows data to be read from any simulator processor and saved by using the same interrupt procedure that is used by the RTMPL read advisories (ref. 8). When the sample process is active, read advisories are ignored and tasks containing read advisories should be disabled so that unnecessary interrupts are not issued. The user can elect to have the sample process trigger as soon as the simulator is put in the run or hold mode or when the value of a variable in the general group reaches a specified level. In the latter case the user is prompted for the variable to be used, the trigger level, and an absolute-value bandwidth about the trigger level. The sample process then triggers if the value of the variable comes within the trigger level, plus or minus the bandwidth. If the run/hold trigger option is selected, the sample will begin after the user quits the analysis functions setup routine and the simulator is placed in run or hold. The user is also prompted for the number of update intervals between samples and the number of samples desired. The RTMPOS displays the maximum number of samples allowed based on the memory size allotted to saving data. When the sample process is complete, an advisory is issued on the message device. Processing of the data is discussed in item 2 of this submenu.

The max/min value functions keep track of the maximum and minimum values of variables in the general group selected for each function. The same general group could be used for both functions. The max/min values are initialized to the values of the variables at the time that the function is activated. Whenever the variable IC values are set (main menu function 5, submenu selection 2), the max/min values are also initialized to the IC values if the max/min functions are active.

The maximum rate-of-change function allows the user to specify the maximum allowable change of variables per update interval (or time step). If the allowable change is exceeded, the simulator goes to the stop mode and a message advisory is displayed on the message device. The next entry by the user will result in a display on the user's terminal indicating which variable caused the rate-of-change violation.

Selection 2. - Selection 2 allows the user to save sample data or to display maximum or minimum values resulting from the corresponding analysis functions. The user is prompted to select the desired function or quit. An asterisk indicates which of the functions are active. Unlike the read advisory data, saving the sample data to the data disk file must be initiated by the user. If the user attempts to save the sample data before the sample is completed, the user is notified and given the opportunity to terminate the sample. The user then has the option of saving the partial sample. The sample data are saved in the data file that was previously specified by the user. The user can also elect to display the maximum or minimum values if those functions are active. The results are displayed on the user's terminal and saved in the session history file.

Selection 3. - Selection 3 allows the user to display the number of update intervals, the simulator CPU time (update-interval clock times the number of update intervals) and simulation time (integration step-size parameter times

the number of update intervals) since the last simulation IC. These values can be displayed with the simulator in any mode.

Selection 4. - Selection 4 allows the user to display the time required to calculate the program residing on each simulator processor. The values can be displayed with the simulator in any mode.

Selection 5. - Selection 5 allows the user to redefine the disk file for saving simulation results. The user is prompted for the required information. The user is automatically prompted for this file when a simulation is first loaded.

Selection 6. - Selection 6 allows the user to obtain a listing of simulation results, in engineering units, that have previously been saved in a data disk file as a result of the read (R) advisory (ref. 8) or the analysis sample function. The user is prompted for the list device (e.g., a printer or the user's terminal) and the data file to be listed. The user can then elect to list results of a specific run/hold number or the entire file.

Selection 7. - Selection 7 allows the user to obtain plots of data that have previously been saved in a data disk file as a result of the read (R) advisory (ref. 8) or the analysis sample function. Warning! This function is not currently implemented in the RTMPOS. Therefore plotting must be done by means of a separate program. The user is prompted for the required information.

Selection 8. - Selection 8 allows the user to display current values of simulation constants and variables (local or external). All values are retrieved from the proper simulator processor except for nonparameter constants that are assigned as immediate data by the RTMPL. Immediate data values are retrieved from the data base and are preceded by the expression IMMED.DATA:. The user specifies an external variable by preceding the name with "@." For example, suppose it is desired to display the value of the external variable DATAPROC.P.ANF$1 that is used in program DATAPROC.C. First of all, DATAPROC.C must be the default program and processor type. The user then enters @DATAPROC.P.ANF$1 or simply .P.ANF$1 since DATAPROC is the default program. The value displayed is the most recent value of ANF$1 transferred from the PREP to the COMP and is retrieved from the COMP. An entry without "@" (i.e., DATAPROC.P.ANF$1) results in the value of ANF$1 being retrieved from the PREP. The user is prompted for the item to be displayed. The name of a constant, variable, argument group, or general group may be entered. All items of a constant or variable array will be displayed, beginning with the first one specified, until the user exits the array by entering "#." The user cannot change values by means of this function. The function can be used in any simulator mode. Values displayed on the user's terminal are also saved in the session history. Note that displaying values with the simulator in the run mode will slow down the simulator.

Selection 9. - Selection 9 allows the user to display the maximum delay times of message and read advisories from all simulator processors. The delays for all channels are displayed automatically at the user terminal and written into the session history.

Selection 10. - Selection 10 allows the user to display the transfer delay times of external variables. The user is given the opportunity to display delay times for each program that contains external variables.

Function 7, Read/Write Memory Location

```
SIMULATOR MODE (current mode displayed)
 (1) DISPLAY MEMORY LOCATION
 (2) WRITE MEMORY LOCATION
<CR> OR SELECTION? . .
```

Selection 1. - Selection 1 allows the user to display the contents of simulator and RTMPOS auxiliary memory locations. The simulator memory includes the COMP, the PREP, and the shared memory. The user first selects the type of memory to be read and then the channel number (except for auxiliary memory). The user is then prompted for the desired offset address where a zero offset represents the first byte of the memory being addressed. The offset can be entered in decimal or hexadecimal format. Hexadecimal values must be preceded by two apostrophes. Offsets that are not entered as an even number are automatically rounded down to the nearest even address. Eight bytes are displayed at a time and the values are given in hexadecimal. The user then has the option of entering a new offset, pressing the return key <CR> for the next eight bytes, or entering "#" to return to the menu to select another memory or quit.

Selection 2. - Selection 2 allows the user to write to simulator and RTMPOS auxiliary memory. In this case simulator memory includes only the COMP and PREP. Warning! The user should obviously be knowledgeable about the memory locations being written to, so as not to affect simulator operation or simulation execution. The user is prompted for the type of memory, the channel number, and the offset address. When writing to the auxiliary memory or COMP memory, only one byte is written at a time and the address can be even or odd. When the PREP is selected, two bytes must be written and the offset must be even.

DISK FILE MANAGEMENT

The RTMPOS can generate files that contain the following types of information:

(1) An edited data base
(2) Session history
(3) Simulation results

As explained earlier, the data-base files are an output of the RTMPL utility. The user can then edit those files by means of the RTMPOS and save the new files. Since these files may be large and there may be many of them, the user will probably want to copy these files to another disk and delete them from the operational disk. This procedure is recommended because the RTMPOS will crash if there is an attempt to overfill the operational disk.

File types 2 and 3 have default names defined by the RTMPOS. The user has the option of redefining those names. The session history and simulation results files will, in general, contain important information about a given

simulation session. It is recommended that they be redefined at the beginning of each session or that the default files be saved at the end of each session by copying them to another file or disk.

Files may be copied or deleted from a disk by using the appropriate manufacturer-supplied operating-system command on the FEP. For the NASA Lewis FEP the commands are COPY or DEL followed by the file name and desired options. The user should refer to the manufacturer-supplied FEP users manual for the exact form of the copy and delete commands.

EXAMPLE SIMULATION SESSION

The example presented here is intended only to illustrate the sequence required to get a simulation up and running. It does not illustrate the full capabilities of the RTMPOS. The example simulation is based on a typical small-turboshaft engine. A single-channel, dual-processor simulation was developed by using the RTMPL utility. The simulation equations were allocated to the COMP and PREP processors so as to minimize execution time for the two-processor configuration. Details of the partitioning are not presented herein.

The process of building the simulation by using the RTMPL utility is described in reference 8. The outputs of the RTMPL utility are simulation data-base files and assembly language source files for each simulation program. The program source files are listed as part of the data-base listing. For this simulation the source files are DEV1:0.OBJCOMP.T7ODUAL.SA and DEV1:0.OBJPREP.T7ODUAL.SA for the COMP and PREP, respectively. Before the simulation can be loaded and executed, the source files must be transformed into executable machine code for the target processors. In general, this is done by using a manufacturer-supplied assembler and linkage editor that are resident on the FEP. Once the user has logged on, as described earlier, the assembler and linkage editor are invoked. The following commands assemble and link the source file for the COMP processor on the NASA Lewis RTMPOS:

= ASM DEV1:0.OBJCOMP.T7ODUAL.SA

= LINK DEV1:0.OBJCOMP.T7ODUAL.RO

Both of the program source files must be assembled and linked in this way. The user should refer to the documentation for the manufacturer-supplied assembler and linkage editor for descriptions of the exact syntax to be used and possible options that are available. The example presented herein assumes that the simulation has been formulated by using the RTMPL utility and that the program source files have been assembled and linked.

Figure 4 shows the terminal display for beginning the example simulation session. An "=" generally indicates a prompt from the manufacturer-supplied DOS. Various displays followed by ". ." indicate a prompt from the RTMPOS. In all cases the user types her/his response and then enters it by pressing the keyboard return key <CR>. The following commentary explains the session as it progresses.

The first step is to invoke the RTMPOS as described earlier. Lines 1 to 4 show the log-on process; the blank lines (5 to 7) represent a series of log-on

messages from VERSAdos. The RTMPOS is then started by the chain file command "MPOS.CF S" (line 8). The chain file commands (lines 9 to 19) are then displayed. The RTMPOS is initiated by the STAR(t) command (line 21). The default message device is indicated to be the line printer designated as "#PR." The user accepts that by typing "Y" on line 30. The prompt on line 32 means that the user should be sure that the printer is turned on and properly configured to receive messages before pressing the return key.

Lines 34 to 49 define the session history file. The user elects not to use the default file name by typing "N" on line 34. The user is then prompted for the parts of the file definition. Session history file extension is always "SH." The user must be sure that the designated volume (disk) is loaded and the RTMPOS then tests if the file already exists. If it does, the user can overwrite the existing file or enter a new name. The session history name is displayed on the user's terminal and on the message device (fig. 6). The RTMPOS is now initialized and the user is ready to begin the simulation session. The message on lines 52 to 53 (fig. 4) indicates that the user can change the simulator mode in response to any main menu or submenu prompt. Pressing the return key (line 54) results in a main menu display and prompt (lines 60 to 79). The asterisk on line 63 indicates that the simulator is currently in the stop mode.

The remainder of the example session will be explained with the aid of the session history file that was generated during the example session. A listing of the session history is given in figure 5. All lines beginning with an exclamation point (e.g., line 1) are treated as unexecutable comments. These include all nonuser entries such as prompts and other terminal displays and messages. The exclamation point is a signal to prevent the RTMPOS from trying to execute that entry when a session history file is being executed. A user can therefore make an executable session-history entry into an unexecutable comment by inserting an exclamation point at the beginning of the proper line. This would be accomplished by means of the manufacturer-supplied text editor utility. Whenever a prompt is included in the file, the user's response is given on the next line (e.g., lines 5 and 6). A blank line indicates a carriage return <CR> response to a prompt (e.g., line 11). RTMPOS prompts that are repeated often or are not necessary for understanding the session file are omitted from the file to save space. Some user responses are entered in the session file only as a comment. These include main menu selection 8 and session-history-menu selections 3 to 5, which would cause problems if they were to be executed as part of the session history file. When an improper response is entered by the user, "!ERROR!" is written into the session file, but the entire error message is not written.

A typical user procedure after the RTMPOS is invoked would be the following:

(1) Initialize the simulator hardware.
(2) Load the data base for the desired simulation.
(3) Edit the data base if required.
(4) Load the simulation programs.
(5) Set up the desired analysis functions.
(6) Run the simulation and obtain results.

To initialize the simulator hardware the user enters main menu selection (i.e., function) 1 (fig. 5, line 6). (All subsequent line numbers refer to the session history.) The user is prompted to reset the simulator processors (i.e., push the reset buttons on the processor boards). A response of "Y" (line 7) indicates that this was done and the hardware test proceeds. (A response of "N" would have aborted the hardware initialization and returned the user to the main menu.) Lines 8 to 10 indicate successful completion of the hardware test and that the RTX is channel 1. The fact that the advisory link is operational between channel 1 processors and the FEP is displayed on the message device (fig. 6). During this session, only one channel was available. When multiple channels are available, the status of each channel and the RTX channel number is displayed. The <CR> entry (line 11) returns the user to the main menu, and selection 4 is made to do data-base-related functions. Data-base submenu selection 1 allows a data base to be loaded. The user is prompted for the parts of the simulation definition file, with responses on lines 16 to 19. The RTMPL catalog name for this file is always SIMDEF. (The user can change the name by means of VERSAdos utilities.) The file name T70ODUAL is the same as the RTMPL user-selected simulation identification name. The RTMPOS tests for the existence of the file, displays identifying information from the file on the user's terminal, and gives the user the option to load it or not.

Once the data base is loaded (lines 21 and 22), the user can make other submenu selections to edit it. In this session the user selects 4 to edit the general group, GROUP1. The group does not exist, so it is created (lines 27 and 28). Although it is not shown in the session history, the default program is the COMP. Therefore TIME and NP come from the COMP program. TIME is a running measure of simulation time in seconds, and NP is the power turbine speed in revolutions per minute. The item .P.NG is the compressor speed in revolutions per minute, and the .P. indicates that it is located in the PREP. Because the COMP and PREP programs in the same channel have the same name, T7002CH1 in this case, it is not required as part of the entry. The PREP is now the default processor until the user enters a prefix of .C. to change it. The user then escapes from the general group editing routine by entering "#" (lines 37 and 39) in response to two successive prompts.

The user next proceeds to load the simulator (main menu function 5, submenu selection 1). The user is prompted for the channel in which to load program T7002CH1, which is indicated to be an RTX program (line 47). Channel 1 must be selected (line 48) because it has been identified as the RTX (line 10). If there had been multiple DSC programs in the simulation, any ready DSC channel could have been chosen for any DSC program. Lines 52 and 53 indicate that the simulation programs have been loaded. The user is then prompted (lines 54 and 55) for the name of the simulation integration step-size parameter (assumed to have units of seconds) so that the RTMPOS can coordinate it with the simulator update interval prompted for on line 57. After the display of the current value of the step size (line 56), lines 57 and 58 display the default values of the update interval (5000 μs) and the number of ADC's (zero since analog-to-digital convertors are not currently implemented). The "Y" response on line 60 indicates that the default values are acceptable. An entry of "N" would have allowed the user to change those values. Lines 61 to 69 indicate that the simulator will run slower than real time and that the simulation is loaded and updated for the current data base and give the status of simulation executives and tasks. The user then selects the default data file by entering "Y" (line 70) to the name that is given on line 71. An entry of "N" would have allowed the user to rename the data file. The simulator automatically executes one

hold cycle to initialize simulation external variables and then returns to the stop mode. Execution of the stop mode causes previously generated advisory messages to be entered into the session history. Lines 75 and 76 are the advisory link operational messages previously seen on the message device (fig. 6).

Before running the simulation, the user may wish to set up an analysis function to obtain data. The set up is done on lines 81 to 93. This particular sequence is for setting up the sample process by using GROUP1, which was formed previously in the session. The user elects to trigger on run/hold mode (lines 87 and 88) rather than selecting a trigger variable. The user then enters the number of simulator update intervals per sample (line 90) and the number of samples desired (line 92). (The maximum number of samples allowed is displayed on the user's terminal but not in the session history.) The user exits the analysis function setup procedure by entering "Q" (line 93) and subsequently puts the simulator into the run mode (line 96). Note that the mode change takes place in response to a submenu prompt. When the sample process is complete, an advisory is displayed on the message device (fig. 6) and the user puts the simulator in stop (line 101). The sample process complete message is then entered in the session history (line 102). The user saves the sample data to the data file (lines 105 to 107) and a message is displayed (line 108) that 20 records of data were obtained as specified on line 92. The user then returns to the main menu and quits the simulation session (line 114).

Since the data base had been modified by creating a general group, the user is now advised that the edited data base has not been saved (line 115). The user selects the option to save the data base before quitting, to display the names of the data-base files that were edited, and to overwrite the existing data-base files. Only the simulation definition file and the modified files are overwritten, and in this case the modified files are actually new because there were no general groups previously. After the data base is saved, the RTMPOS returns to the main menu prompt (line 133). The user then terminates the session by entering "9" (line 134).

The user could create the same or any session history file by using the manufacturer-supplied text editor. For example, the user could create a standard text file that, with only minor editing, would allow loading of any simulation data base and the simulator without manual entry of all of the menu selections and prompt responses. However, care must be taken to include a response for all prompts that will occur. The prompts that occur will often depend on the responses to previous prompts. For example, in figure 5, responses (lines 16 to 19) to the prompts that occur after the load data-base selection (line 15) will depend on whether or not a session history file was previously loaded. The session history of figure 5 could, for example, be modified to load any single-channel, dual-processor simulation by editing the parts of the file definition (lines 16 to 19 and lines 50 and 51) and any other items (e.g., integration step-size parameter, line 55) that would be different. Lines 80 to 134 of the file, as well as any comment lines, could be deleted to simply bring the simulation to a loaded and ready-to-run condition. The resultant file would be like the one shown in figure 7.

CONCLUDING REMARKS

An operating system (RTMPOS) that provides versatile and user-friendly means for interactively programming, running, and obtaining results from a real-time multiprocessor simulator has been developed. The RTMPOS acts in concert with a manufacturer-supplied disk operating system that provides the usual operating system functions such as assembling, linking, text editing, and file handling services. The RTMPOS facilitates the running of simulations that have been formulated by using a real-time multiprocessor programming language (RTMPL) utility. The RTMPOS menus provide for run-time operations such as loading, modifying, and specifying computational flow of programs, simulator mode control, data handling, and run-time monitoring. Run-time monitoring is a powerful feature that provides the user with a self-documenting history of the simulation session. The user may then use that history at a later date to automatically recreate the session. The session history also coordinates the session with the data file.

The capabilities that were described represent the first edition of the RTMPOS. Future enhancements to the operating system are expected to include the following capabilities:

(1) Simulation static test
(2) Enhanced analysis and graphical display of simulation results
(3) More extensive simulator hardware status checks
(4) Handling of simulator advisories that will allow for such things as initiation of a user-generated operating-system task

Since the RTMPOS is an outgrowth of a research effort, the implementation of these improvements will depend on the results of further applications and the users' acceptances and responses. The present NASA Lewis application of the RTMPS is to the simulation of air-breathing engines to facilitate the study of engine dynamics and control systems. However, the RTMPS has potential for much wider applications and should be beneficial to any simulation or process that would benefit from parallel processing. For example, the RTMPOS itself could serve as the interface between a facility, such as a wind tunnel, and the operator. The RTMPOS would provide the means for the operator to make changes to and monitor facility operating conditions, while the simulator could act as the facility control system and be interfaced to the facility sensors, actuators, etc.

APPENDIX - SYMBOLS AND ABBREVIATIONS

ADC	analog-to-digital converter
A0,...,A7	processor address registers
C	COMP processor
CHAR	character(s)
CHG	change
CN	constant
D	refers to disable task or use default value for constant or variable IC or hold
DB	simulation data base
DEL	delete
D0,...,D7	processor data registers
E	refers to enable task
GC	global constant
HD	hold value
IC	initial-condition value
LOC	memory location
LV	program local variable
MAX	maximum
N	no
NUM	number(s)
P	PREP processor
PC	process program counter
Q	quit
RETURN	press RETURN key, <CR>
SIM	simulation, simulator
SP	processor stack pointer
SR	processor status register

XV	program external variable
Y	yes
#	return to higher prompt or command level
/	delete item from argument or general group
<	insert item before current item in argument or general group
>	insert item after current item in argument or general group

REFERENCES

1. O'Grady, E. Pearse; and Wang, Chung-Hsien: Multibus-Based Parallel Processor for Simulation. Proceedings of the 1983 Summer Computer Simulation Conference, Society for Computer Simulation, 1983, pp. 371-375.

2. Daniele, Carl J.; and McLaughlin, Peter W.: The Real-Time Performance of a Parallel, Nonlinear Simulation Technique Applied to a Turbofan Engine. *Modeling and Simulation on Microcomputers: 1984*, Ray Swartz, ed., Society for Computer Simulation, 1984, pp. 167-171.

3. Makoui, Ali; and Karplus, Walter J.: Data Flow Methods for Dynamic System Simulation: A CSSL-IV Microcomputer Network Interface. Proceedings of the 1983 Summer Computer Simulation Conference, Society for Computer Simulation, 1983, pp. 376-382.

4. Collins, W. Robert; and Feyock, Stefan: The Use of ADA in Distributed Simulations. Proceedings of the 1983 Summer Computer Simulation Conference, Society for Computer Simulation, 1983, pp. 364-370.

5. Blech, Richard A.; and Arpasi, Dale J.: An Approach to Real-Time Simulation Using Parallel Processing. NASA TM-81731, 1981.

6. Blech, Richard A.; and Arpasi, Dale J.: Hardware for a Real-Time Multiprocessor Simulator. NASA TM-83805, 1984.

7. Arpasi, Dale J.: RTMPL - A Structured Programming and Documentation Utility for Real-Time Multiprocessor Simulations. NASA TM-83606, 1984.

8. Arpasi, Dale J.: Real-Time Multiprocessor Programming Language (RTMPL): Users Manual. NASA TP-2422, 1984.

9. Cole, Gary L.: Operating System for a Real-Time Multiprocessor Propulsion System Simulator. NASA TM-83605, 1984.

10. Glaser, Jay G.: The VERSAdos Operating System. Microprocessor Operating Systems, John Zarrella, ed., Microcomputer Applications, 1981, pp. 9-1 to 9-20.

TABLE I. - OPERATING SYSTEM M AND H ADVISORY MESSAGES

Number	Message
201	WARNING! MESSAGE STORAGE MEMORY FULL - OVERWRITING OLDEST
202	WARNING! DATA STORAGE MEMORY FULL - NO MORE DATA SAVED
230	UNDEFINED MODE SELECTED
231	HOLD MODE COMPLETED
232	SIMULATOR HALT! WATCHDOG TIMEOUT
233	SPURIOUS PROGRAMMABLE TIMER INTERRUPT
234	SIMULATOR HALT! UPDATE INTERVAL FAILURE
235	SAMPLE PROCESS COMPLETE
236	ADVISORY LINK OPERATIONAL
237	SIMULATOR HALT! RATE OF CHANGE VIOLATION
238	SIMULATOR HALT! SPURIOUS INTERRUPT VECTOR
239	SIMULATOR HALT! UNINITIALIZED INTERRUPT VECTOR
240	SIMULATOR HALT! SYSTEM FAILURE
241	SIMULATOR HALT! BUS CLEAR LINE DRIVEN LOW BY BUS ARBITER
242	SIMULATOR HALT! SERIAL PORT INTERRUPT
243	SIMULATOR HALT! VERSABUS ACFAIL OR BUS RELEASE LINE LOW
244	SIMULATOR HALT! VM02 ABORT PUSHBUTTON PRESSED
245	SIMULATOR HALT! MISCELLANEOUS
246	SIMULATOR HALT! LINE 1111 EMULATOR
247	SIMULATOR HALT! LINE 1010 EMULATOR
248	SIMULATOR HALT! TRACE
249	SIMULATOR HALT! PRIVILEGE VIOLATION
250	SIMULATOR HALT! TRAPV INSTRUCTION
251	SIMULATOR HALT! CHK INSTRUCTION
252	SIMULATOR HALT! ZERO DIVIDE
253	SIMULATOR HALT! ILLEGAL INSTRUCTION
254	SIMULATOR HALT! ADDRESS ERROR
255	SIMULATOR HALT! BUS ERROR

TABLE II. - SUMMARY OF RIMPOS VERSION 1.0, 090684, MENU SELECTIONS

[Mode selections may be made in response to any menu prompt.]

Main menu selection:	Discussed on page
(S) Stop mode	11
(R) Run mode	11
(H) Hold mode	11
(C) Cycle mode	--
(1) Initialize simulator hardware	12
(2) Display/set update interval and number of ADC channels	12
(3) Session history (submenu)	13
(1) Display date/time on terminal and in session history	13
(2) Enter comment in session history file	13
(3) Execute an existing session history file	13
(4) Be prompted for file to save session history	13
(5) Stop saving session history	14
(4) Data-base loading and management (submenu)	14
(1) Load simulation data base into FEP	14
(2) Display/edit argument group definitions	14
(3) Display/edit values of simulation constants or variable IC/hold values	15
(4) Display/edit general group definitions	16
(5) Display/edit user defined advisory messages	17
(6) Save current edited data base on disk(ette)	17
(7) Get listing of current data base	17
(8) Display data base status	17
(9) Delete current data base from FEP memory	17
(5) Simulation loading/IC and program control (submenu)	18
(1) Load simulation into simulator	18
(2) Set simulation IC's	18
(3) Change background executives	18
(4) Enable/disable program tasks	19
(5) Display program status (active executives and enabled tasks)	19
(6) Unload simulator	19
(6) Simulation results management (submenu)	19
(1) Activate/deactivate analysis functions	19
(2) Save sample data, display maximum/minimum values	20
(3) Display simulator CPU time and simulation time	20
(4) Display COMP and PREP computation times	21
(5) Be prompted for file to save simulation results	21
(6) List simulation results data file	21
(7) Plot simulation results data file	21
(8) Display simulation values of constants, variables, and argument/general groups	21
(9) Display COMP and PREP advisory maximum delay times	21
(10) Display external variable delay times	22
(7) Read/write memory location (submenu)	22
(1) Display memory location	22
(2) Write memory location	22
(8) Suspend RIMPOS temporarily	12
(9) Quit	12

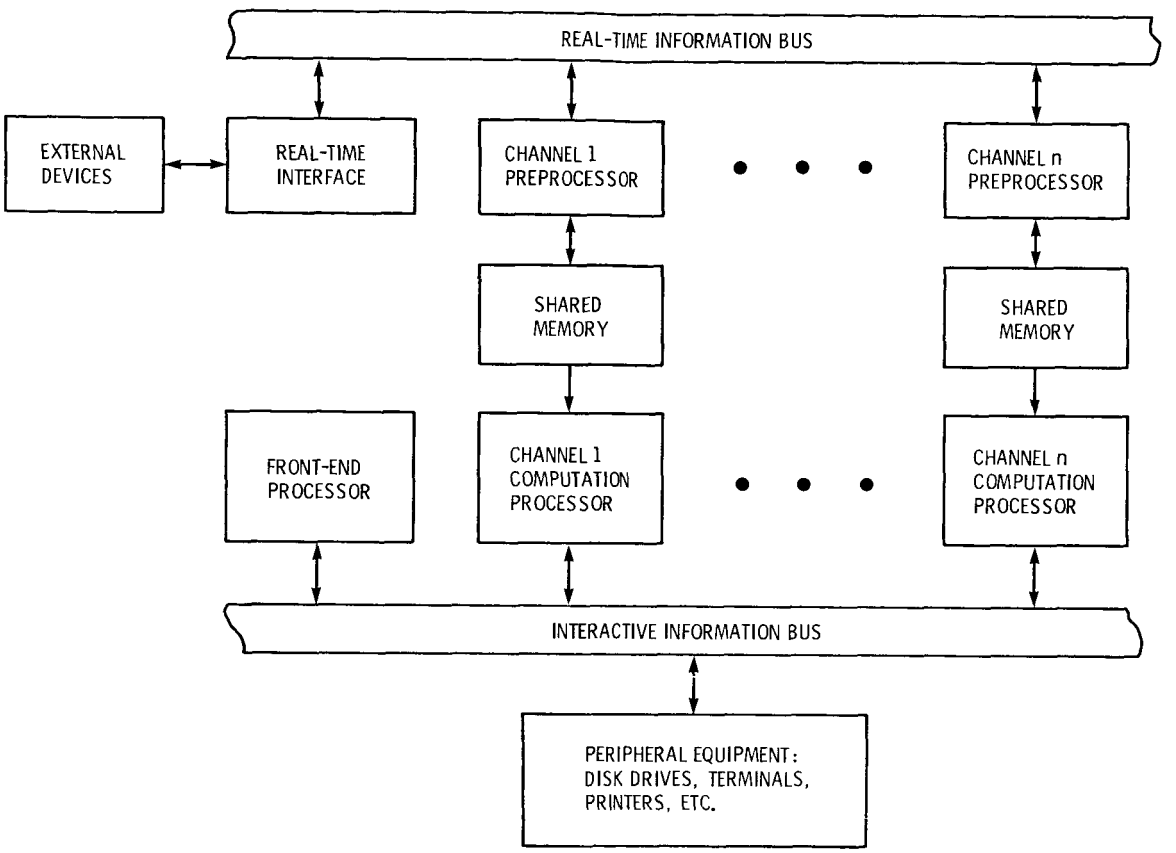

Figure 1. - General simulator configuration.

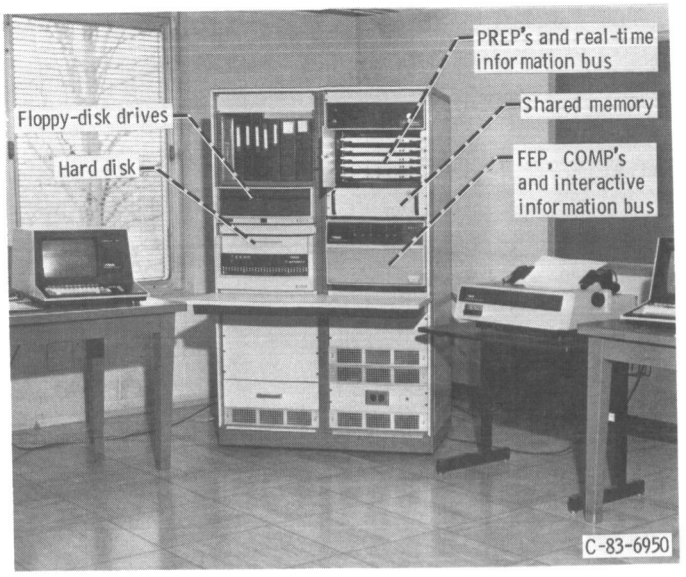

Figure 2. - NASA Lewis RTMPS experimental hardware.

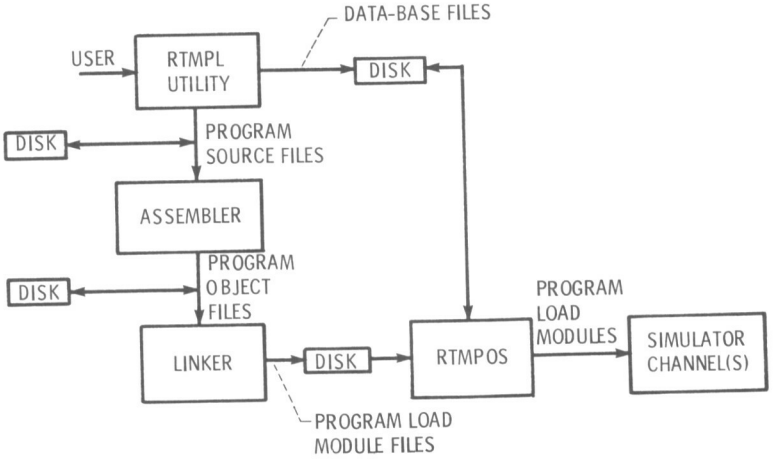

Figure 3. - RTMPL-generated simulation files.

```
 1  VERSADOS VERSION: 3.00, 8203221421
 2  ENTER DEFAULT SYSTEM VOLUME:USER NO.=SYS:0
 3  ENTER DATE (MM/DD/YR)=9/06/84
 4  ENTER TIME (HR:MIN)=13:10
 5      .
 6      .
 7      .
 8  =MPOS.CF S
 9  =@SYS:0.GLC.IC01
10  =@SYS:0.GLC.MC01
11  =@SYS:0.GLC.RC01
12  =@SYS:0.GLC.IC02
13  =@SYS:0.GLC.MC02
14  =@SYS:0.GLC.RC02
15  =@SYS:0.GLC.STPOPSYS
16  =LOAD SYS:0.GLC.RTMPOS
17   RTMP 0001: LOADED
18  =NOARG
19  =END
20  END CHAIN
21  =STAR
22
23
24  ****************  NASA LERC RTMPOS VERSION: 1.0, 090684  ****************
25
26
27  RTMPOS AUXILIARY & COMP PROCESSOR MEMORY BEING ATTACHED
28  MEMORY SUCCESSFULLY ATTACHED AND AUXILIARY MEMORY CLEARED
29
30  ADVISORY MESSAGE DEVICE IS: #PR                                    OK (Y/N) ? .. Y
31
32  <CR> WHEN MESSAGE DEVICE READY ..
33
34  SESSION HISTORY FILE IS: DEV1:0.SESSHST.SCRATCH.SH          OK (Y/N) ? .. N
35
36  DEFINE FILE TO SAVE SESSION HISTORY:
37
38  ENTER VOLUME NAME (4 CHAR MAX) .. DEV1
39
40  ENTER USER NUMBER (4 CHAR MAX) .. 0
41
42  ENTER CATALOG NAME (8 CHAR MAX) .. SESSHST
43
44  ENTER FILE NAME (8 CHAR MAX) .. T700DUAL
45
46  LOAD DISK(ETTE)  DEV1  TO SAVE SESSION HISTORY!   THEN HIT RETURN ..
47  *****  TESTING FOR EXISTENCE OF: DEV1:0.SESSHST.T700DUAL.SH
48
49  SESSION HISTORY FILE SAVED AS: DEV1:0.SESSHST.T700DUAL.SH
50
51  READY TO BEGIN SIMULATION SESSION:
52  ****  SET SIMULATOR MODE BY ENTERING 1ST LETTER OF MODE NAME
53  ****  IN RESPONSE TO ANY MENU SELECTION PROMPT
54  <CR> TO CONTINUE ..
55
56
57
58
59
60  **********************  MAIN MENU OF R*T*M*P*O*S FUNCTIONS  **********************
61
62  SIMULATOR MODE:
63  *(S) STOP MODE
64   (R) RUN MODE
65   (H) HOLD MODE
66   (C) CYCLE MODE
67
68  OTHER FUNCTIONS:
69   (1) INITIALIZE SIMULATOR HARDWARE
70   (2) DISPLAY/SET UPDATE-INTERVAL AND NUMBER OF ADC CHANNELS
71   (3) SESSION HISTORY
72   (4) DATA BASE LOADING & MANAGEMENT
73   (5) SIMULATION LOADING/IC & PROGRAM CONTROL
74   (6) SIMULATION RESULTS MANAGEMENT
75   (7) READ/WRITE MEMORY LOCATION
76   (8) SUSPEND RTMPOS TEMPORARILY
77   (9) QUIT
78
79  SELECTION ? ..
```

Figure 4. - Terminal display for example simulation session.

```
1  ! SESSION HISTORY FILE: DEV1:0.SESSHST.T700DUAL.SH
2  ! DATE: 09/06/84    TIME: 13:10:04    ** OF PREVIOUS ! COMMENT **
3
4
5  ! MAIN MENU SELECTION ..
6  1
7  Y
8  ! CHANNEL 1 PASSED HARDWARE TEST
9  ! CHANNEL 1 READY
10 ! CHANNEL 1 IS THE RTX
11
12 ! MAIN MENU SELECTION ..
13 4
14 ! SUBMENU SELECTION ..
15 1
16 DEV1
17 0
18 SIMDEF
19 T700DUAL
20 Y
21 ! DATA BASE HAS BEEN LOADED FOR SIMULATION FILE DEV1:0000.SIMDEF.T700DUAL.DB
22 ! DATE: 09/06/84    TIME: 13:12:24    ** OF PREVIOUS ! COMMENT **
23
24 ! SUBMENU SELECTION ..
25 4
26 GROUP1
27 ! GENERAL GROUP "GROUP1   " DOESN'T EXIST!    CREATE IT (Y/N) ? ..
28 Y
29 ! CREATE A NEW GENERAL GROUP: GROUP1
30 ! ENTER ITEM TO BE ADDED TO GROUP ..
31 TIME
32 ! ENTER ITEM TO BE ADDED TO GROUP ..
33 NP
34 ! ENTER ITEM TO BE ADDED TO GROUP ..
35 .P.NG
36 ! ENTER ITEM TO BE ADDED TO GROUP ..
37 #
38 ! GENERAL GROUP "GROUP1   " HAS 3 ITEMS
39 #
40
41 ! SUBMENU SELECTION ..
42
43 ! MAIN MENU SELECTION ..
44 5
45 ! SUBMENU SELECTION ..
46 1
47 ! T7002CH1 (RTX      ): CHANNEL NO. ..
48 1
49 Y
50 DEV1
51 0
52 !   CHANNEL 1 HAS BEEN LOADED WITH PROGRAM "T7002CH1.C"
53 !   CHANNEL 1 HAS BEEN LOADED WITH PROGRAM "T7002CH1.P"
54 ! NAME OF SIMULATION INTEGRATION-STEP-SIZE PARAM ..
55 TIMDEL
56 ! THE CURRENT INTEGRATION STEP SIZE IS  0.00020 SEC
57 ! UPDATE INTERVAL (MICROSEC): 5000
58 ! NUMBER OF ADC CHANNELS: 0
59 !    OK (Y/N) ? ..
60 Y
61 ! SIMULATION WILL RUN SLOWER THAN REAL TIME BY A FACTOR OF   25.00
62 ! SIMULATOR CHANNELS HAVE BEEN LOADED AND UPDATED FOR THE CURRENT DATA BASE
63 ! DATE: 09/06/84    TIME: 13:15:04    ** OF PREVIOUS ! COMMENT **
64 !                      *****  SIMULATION PROGRAM STATUS  *****
65 ! DATE: 09/06/84    TIME: 13:15:04    ** OF PREVIOUS ! COMMENT **
66 ! PROGRAM: T7002CH1.C;    CHANNEL: 1;    ACTIVE EXECUTIVE: T700EXE.
67 ! ENABLED TASKS: NONE AVAILABLE
68 ! PROGRAM: T7002CH1.P;    CHANNEL: 1;    ACTIVE EXECUTIVE: T700EXE.
69 ! ENABLED TASKS: NONE AVAILABLE
70 Y
71 ! SIMULATION DATA FILE: DEV1:0.DATA.SCRATCH.DF
72 ! DATE: 09/06/84    TIME: 13:15:20    ** OF PREVIOUS ! COMMENT **
73 ! SIMULATOR IN "HOLD" MODE TO INITIALIZE SIMULATION XV'S
74 ! DATE: 09/06/84    TIME: 13:15:20    ** OF PREVIOUS ! COMMENT **
75 !C1;PREP;0;SYSTEM ADVISORY MESSAGE NO. 236
76 !C1;COMP;0;SYSTEM ADVISORY MESSAGE NO. 236
77 ! SIMULATOR IN "STOP" MODE
78
79 ! SUBMENU SELECTION ..
80
81 ! MAIN MENU SELECTION ..
```

Figure 5. - Session history file listing for example session.

```
 82  6
 83  ! SUBMENU SELECTION ..
 84  1
 85  1
 86  GROUP1
 87  ! TRIGGER SAMPLE ON  RUN/HOLD MODE <CR> OR ENTER VARIABLE NAME ..
 88
 89  ! NUMBER OF UPDATES/SAMPLE ..
 90  500
 91  ! NUMBER OF SAMPLES (@   0.1000 SEC/SAMPLE) ..
 92  20
 93  Q
 94
 95  ! SUBMENU SELECTION ..
 96  R
 97  ! SIMULATOR IN "RUN" MODE    RUN/HOLD NO. 1
 98  ! DATE: 09/06/84    TIME: 13:17:25    ** OF PREVIOUS ! COMMENT **
 99
100  ! SUBMENU SELECTION ..
101  S
102  !C1;COMP;9500;SYSTEM ADVISORY MESSAGE NO. 235
103  ! SIMULATOR IN "STOP" MODE
104
105  ! SUBMENU SELECTION ..
106  2
107  1
108  ! 20 RECORDS OF SAMPLE DATA WERE SAVED IN DEV1:0.DATA.SCRATCH.DF
109  Q
110
111  ! SUBMENU SELECTION ..
112
113  ! MAIN MENU SELECTION ..
114  9
115  ! CURRENT DATA BASE HASN'T BEEN SAVED!    SAVE IT BEFORE QUITTING (Y/N) ? ..
116  Y
117  Y
118  !                              *****  DATA BASE STATUS  *****
119  ! DATE: 09/06/84    TIME: 13:21:19    ** OF PREVIOUS ! COMMENT **
120  ! THE FOLLOWING FILES OF "DEV1:0000.SIMDEF.T700DUAL.DB  " HAVE BEEN EDITED
121  ! DEV1:0.GGRPDEF.T700DUAL.DB        AND/OR    DEV1:0.GLSTDEF.T700DUAL.DB
122  Y
123  Y
124  Y
125  Y
126  Y
127  ! CURRENT DATA BASE FOR DEV1:0000.SIMDEF.T700DUAL.DB   BEING SAVED
128  ! DATE: 09/06/84    TIME: 13:22:01    ** OF PREVIOUS ! COMMENT **
129  ! GENERAL GROUP RECORD FILE SAVED AS: DEV1:0.GGRPDEF.T700DUAL.DB
130  ! GENERAL GROUP LIST FILE SAVED AS: DEV1:0.GLSTDEF.T700DUAL.DB
131  ! SIMULATION DEFINITION FILE SAVED AS: DEV1:0000.SIMDEF.T700DUAL.DB
132
133  ! MAIN MENU SELECTION ..
134  9
```

Figure 5. - Concluded.

```
DATE:  09/06/84   TIME:  13:10:04
SESSION HISTORY FILE SAVED AS: DEV1:0.SESSHST.T700DUAL.SH

!!!! FROM CHANNEL 01 PREP:
236  ADVISORY LINK OPERATIONAL
!!!! FROM CHANNEL 01 COMP:
236  ADVISORY LINK OPERATIONAL

DATE:  09/06/84   TIME:  13:15:02
SIMULATOR IN "RUN" MODE  RUN/HOLD NO. 1
!!!! FROM CHANNEL 01 COMP:
235  SAMPLE PROCESS COMPLETE
```

Figure 6. - Advisory message device display.

```
1  ! GENERAL SESSION HISTORY FILE FOR LOADING SINGLE-CHANNEL,
2  ! DUAL-PROCESSOR SIMULATION
3
4
5  ! MAIN MENU SELECTION ..   [INITIALIZE HARDWARE]
6  1
7  Y
8
9  ! MAIN MENU SELECTION ..   [LOAD DATA BASE]
10 4
11 ! SUBMENU SELECTION ..
12 1
13 DEV1
14 0
15 SIMDEF
16 T700DUAL
17 Y
18
19 ! SUBMENU SELECTION ..
20
21 ! MAIN MENU SELECTION ..   [LOAD SIMUALTION]
22 5
23 ! SUBMENU SELECTION ..
24 1
25 ! T7002CH1 (RTX     ): CHANNEL NO. ..   [CHANNEL 1 ASSUMED TO BE RTX]
26 1
27 Y
28 DEV1
29 0
30 ! NAME OF SIMULATION INTEGRATION-STEP-SIZE PARAM ..
31 TIMDEL
32 ! THE CURRENT INTEGRATION STEP SIZE IS  0.00020 SEC
33 ! UPDATE INTERVAL (MICROSEC): 5000
34 ! NUMBER OF ADC CHANNELS: 0
35 !    OK (Y/N) ? ..
36 Y
37 ! SIMULATION DATA FILE WILL BE DEFAULT FILE: DEV1:0.DATA.SCRATCH.DF
38 Y
39
40 ! SUBMENU SELECTION ..
```

Figure 7. - General session history file for initializing simulator hardware and loading a single-channel, dual-processor simulation.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2426		

4. Title and Subtitle	5. Report Date
Operating System for a Real-Time Multiprocessor Propulsion System Simulator - Users Manual	January 1985
	6. Performing Organization Code: 505-40-74

7. Author(s)	8. Performing Organization Report No.
Gary L. Cole	E-1998
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135	13. Type of Report and Period Covered: Technical Paper

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Washington, D.C. 20546	

15. Supplementary Notes

16. Abstract

The NASA Lewis Research Center is developing and evaluating experimental hardware and software systems to help meet future needs for real-time, high-fidelity simulations of air-breathing propulsion systems. Specifically, the Real-Time Multiprocessor Simulator project focuses on the use of multiple microprocessors to achieve the required computing speed and accuracy at relatively low cost. Operating systems for such hardware configurations are generally not available. A real-time multiprocessor operating system (RTMPOS) that supports a variety of multiprocessor configurations has been developed at Lewis. With some modification, RTMPOS can also support various microprocessors. RTMPOS, by means of menus and prompts, provides the user with a versatile, user-friendly environment for interactively loading, running, and obtaining results from a multiprocessor-based simulator. This report is a users guide for RTMPOS. The menu functions are described and an example simulation session is included to demonstrate the steps required to go from the simulation loading phase to the execution phase.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Operating systems; Multiprocessors; Real-time simulation; Interactive simulation; Simulators	Unclassified - unlimited STAR Category 62

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	39	A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1985

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

4 2 1U,G, 850115 S00161DS
DEPT OF THE AIR FORCE
ARNOLD ENG DEVELOPMENT CENTER(AFSC)
ATTN: LIBRARY/DOCUMENTS
ARNOLD AF STA TN 37389

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return