File

Honeywell

INTERACTIVE PROGRAMMING ENVIRONMENT

MULTICS

SOFTWARE

To the user, a computer system is only as productive as it is accessible. Based on human engineering design concepts, the Multics System uses interactive, remote terminal access – the most natural and convenient mode for the user – as the primary mode for Multics.

With the advent of Multics, the computer is to be measured, not merely by hardware speeds, but by how well it helps solve a problem – from the very inception of that problem to its best solution. Rather than wait for computer availability in a batch mode and submit many concurrent jobs, a Multics user prepares, compiles, and checks out his programs in one continuous interactive terminal session.

For Novice Or Advanced User

The Multics interactive programming environment provides a complete range of facilities that satisfies both the novice user and the professional programmer. Both are able to enjoy appropriate software tools and both can work on the same system, protected by advanced hardware/software security features.

For Virtually Any Application

The Multics user interface provides an environment for a nearly unlimited scope of applications regardless of size, complexity, or storage requirements. The multiprocessing, multiprogramming capabilities of Multics and its diversity of languages and utility routines provide the user with all the support he needs.

SYSTEM FEATURES

Some of the more important features of the Multics interactive programming environment include:

- The process, a unique concept
- Flexibility of environment shaping
- Information sharing in the Multics Virtual Memory and Storage System
- Sophisticated language processors
- Extensive support facilities and tools
- Powerful command processor
- Software protection (security)
- Special user interfaces

The Process, A Unique Concept

When a user first accesses (logs into) the Multics System, he is allocated system resources in an environment termed a "process." Specifically, the process is dynamically assigned space within the virtual memory (address space) and other system resources as required. As a result, each user views his process as if it were the only one in the system. In this environment, the user's address space dynamically grows and shrinks as his program requirements expand and contract – and the activity is totally transparent to the user and under control of the shared operating system on the Model 6180. The system creates a process at login time and destroys it at logout time on behalf of each user. There is no user direction or intervention. The user executes his programs and system commands in coexistence with the processes of all other logged-in users under the multiprogramming control of the Multics System.

Flexibility of Environment Shaping

The administration of a typical Multics System includes one system administrator and many project administrators. The project administrator defines the initial process for users under his project. He may give a user maximum flexibility by allowing him complete control in creating his own initial process, or he may limit the user's initial process by restricting his access to various software functions.

The initial process, then, defines the range of access each user has to system software functions. If the user has complete control of his own process environment, he may change that environment and still be within the normal operating conventions of the system.

Part of the user's interactive environment is a special segment called the initial working directory in the storage hierarchy. This directory provides a path of communication to other segments in the hierarchy. This path is always known to the system; the user need not specify the potentially long succession of pathnames to his other segments. From his working directory, he need only reference the symbolic entry name of his program and data segments. This shorthand method of using the symbolic entry name (i.e., relative pathname) simplifies file handling for the user. In addition, the user can change his base of operation to the working directory of another user (with that user's permission) to use the files to which he has been given access.

Information Sharing in the Multics Virtual Memory And Storage System

All procedures and data are organized within the Multics Storage System and its associated Virtual Memory. Within the storage system are complete system facilities that provide the user extensive control over file manipulation and file sharing. A user may specify the individuals who have access to his files. Access can be given to one user, to a group of users (project), to a particular class of users (interactive or absentee[1]), or to no user at all. In addition, levels of protection or "domains of access" can be specified as further control over the same files.

The Multics Storage System is supported by a powerful virtual memory, totally transparent and available to the user as he needs it. "His" virtual memory dynamically expands and contracts as user requirements and system resources grow and shrink. Programmers no longer need be concerned about overlaying or partitioning program modules to satisfy limited core memory resources. Instead, programmers can concentrate on program synthesis and on developing the most efficient algorithm to solve their particular problem.

(continued)

Specifications may change as design improvements are introduced.

Sophisticated Language Processors

The Multics System includes several language processors. Foremost among these is an exhaustive and trend-setting Multics PL/I compiler which is identical for both Multics system programmers and applications programmers. The present Multics PL/I compiler has undergone several major design iterations to become perhaps the most stable and reliable PL/I compiler in existence. This is the same PL/I compiler that was used to produce the Multics Operating System software itself, 95 percent of which is written in the PL/I language.

A complete Multics FORTRAN compiler is also available to satisfy any FORTRAN requirement as well as facilitate software transferability from other computer systems.

For those users who find it necessary to write portions of their software in the language of the host computer, Multics includes the ALM (Assembly Language for Multics) assembler. Like the Multics compilers, the assembler supports all system requirements for inter-program communication.

The Multics compilers will optionally generate a symbol table that permits a user to completely check out his program at the original symbolic name level with the aid of the debug command.

Extensive Support Facilities and Tools

Stable and reliable software components within the Multics Operating System provide numerous utility and support functions. Foremost among these are the Multics text editors. These text editors have undergone several design iterations to become extremely reliable and very sensitive to human engineering requirements. One major text editor (edm) is line-oriented while another (qedx) is context- and buffer-oriented.

The software package known as debug permits a user to analyze and correct a compiled program at both the original symbolic name level and the more specific machine-register level.

Performance-measurement tools permit the user to analyze his program's behavior so that optimum applications software can readily be developed.

Interuser communication facilities, both immediate and deferred, permit online messages to be transmitted among users. In addition, online documentation facilities provide the user with document preparation tools. These same tools are used by the system to inform the user of system capabilities and facilities.

The command line has been designed to provide as sophisticated a flexibility as any user might possibly require with both a commutative and associative syntax form.

The Powerful Command Processor

The command processor, the means by which a user communicates his requirements to the system, accepts input from a console, interprets the user's request, and invokes the software component to perform the desired function. The software component can be either system- or user-supplied: there is no distinction at the command level. The command processor allows recursive, iterative commands and the imbedding of function calls in the command line.

The command processor is a shared, replaceable module, written in PL/I. Therefore, if the administrator desires, a user can be instructed to interface with a special version of the command processor (possibly user-created), thereby limiting the software requests or commands available to him. The design of the command processor thus permits an extremely wide range of interfaces to all system facilities either on a controlled or open-ended basis.

Software Protection (Security)

Superimposed on all software in Multics are "domains of access" or rings of protection through which software may or may not be accessed. There can be up to eight levels of privilege (rings). The initial level for a user is determined when he accesses the system. The access or ring level changes by calling more privileged procedures and is enforced by hardware. This protection mechanism or ring structure further refines the normal read, write, and execute attributes associated with all files in the system.

Special User Interfaces

Included within Multics are special user interfaces that permit the development of other operating systems, protected subsystems, or limited service facilities. All these capabilities have been exercised in Multics and require no unusual tools or additional software.

SUMMARY OF FEATURES

- The Multics interactive programming environment provides facilities for both the novice and advanced user, for a wide range of applications.
- The user's virtual memory (address space) dynamically changes as his program and data requirements change.
- A unique process environment exists for each user, and this environment can be reshaped.
- The working directory concept and the use of symbolic entry names for segments simplifies file handling.
- Files are protected by user-specified access controls and by levels of protection or "domains of access."
- Multics includes several language processors including Multics PL/I, Multics FORTRAN, and ALM.
- The support facilities of Multics include text editors, program debugging aids, performance measurement tools, interuser communication facilities, and online documentation.
- The Multics command processor allows a wide range of interfaces to all system facilities either on a controlled or open-ended basis.
- Included within Multics are special user interfaces that permit the development of other operating systems, protected subsystems, or limited service facilities.

[1] Multics batch.

The Multics Operating System and PL/I Compiler are coded systems supported by documentation, periodic program maintenance, and, where feasible, improvements to the current versions, provided they are not modified by the user. Training and certain software enhancements are available from Honeywell at additional cost.

HONEYWELL INFORMATION SYSTEMS

In the U.S.A.: 200 Smith Street, MS 061, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario

7207
5473 Printed in U.S.A.