Fil

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

File No.: 1L21

7207
5473 Printed in U.S.A.

fd

Honeywell

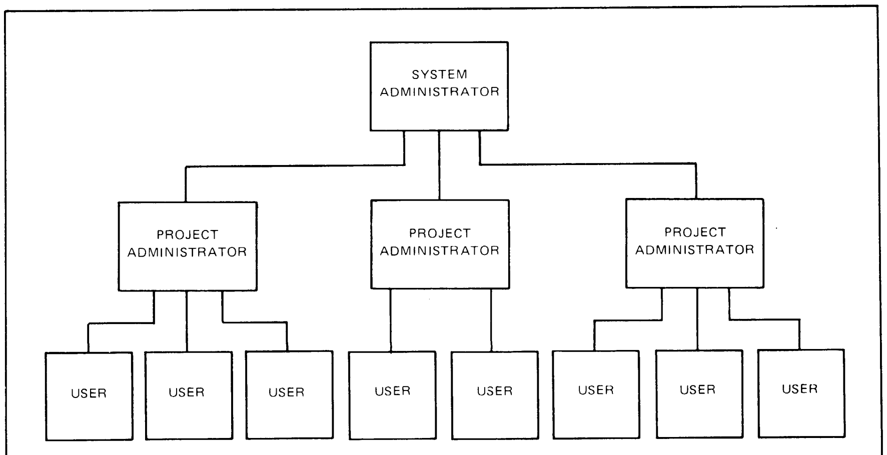
# ADMINISTRATIVE FUNCTIONS

MULTICS

SOFTWARE

Multics is a large-scale, time-sharing system that multiplexes system resources among the jobs of many users. Because the administration of system resources in any large time-sharing system involves various time-consuming clerical tasks and control measures, Multics features a decentralization capability that makes these tasks easier to perform and offers the user better service for each dollar spent.

Multics administration defines three distinct levels of responsibility: *system, project,* and *user*. These levels are arranged like a pyramid (see illustration) with system administration at the top. A system administrator allocates system resources among the projects within his system, and a project administrator allocates these resources among the users on his project. Each user, in turn, can manage his own data through storage management and access controls. Because detailed supervisory tasks are shared and administrative bottlenecks are eliminated, the decentralized structure of Multics offers advantages to both the administrators and the users.



Example of Administration Hierarchy

## SYSTEM ADMINISTRATION

The system administrator has several system management capabilities exclusively his own.

**Resource Control**

In the area of resource control, he:

- Establishes the system configuration and operating parameters.
- Defines which projects are on the system and how much secondary storage quota each is allowed.
- Assigns projects to a load control group so each one can have a fair share of system access.
- Names project administrators and sets project attributes that each project administrator can distribute among his users.

**Accounting Procedures**

In the area of accounting procedures, he:

- Maintains records of system usage so projects can be billed for only the actual use of resources—not the estimated use.
- Can set prices on interactive (cpu time for eight shifts, memory usage, and connect time) and absentee[1] (cpu time for four queues, memory usage, and connect time) usage, I/O (for four queues), disk storage, user registration fee per project, etc.
- Can use either the standard Multics accounting procedures or procedures of his own design.
- Can access accounting records of the whole system.
- Can generate usage reports and bills for each project at any time without interrupting Multics service.

**Security**

For security purposes, the system administrator:

- Registers projects, users, and initial user passwords.
- Uses an advanced security feature of Multics called ring protection to define the rings in which a project can operate. Ring protection is an extension and refinement of the basic access control.

## PROJECT ADMINISTRATION

The project administrator also has several system management capabilities exclusively his own. The system administrator performs these functions if a project administrator is not assigned.

**Resource Control**

In the area of resource control, the project administrator:

- Works with the resources allocated to him by the system administrator.
- Adds and deletes system-registered users from his project at will.
- Can add anonymous users (users on his project who are not registered on the system).
- Distributes the secondary storage among his users.
- Sets access to his users' directories.

(continued)

Specifications may change as design improvements are introduced.

- Can perform all his administration functions remotely from his own terminal. (Any change made by the project administrator takes effect *immediately.)*

### Resource Usage Monitoring

In order to monitor resource usage, the project administrator:

- Has access to all system accounting data concerning his project.
- Has access to system programs to get month-to-date usage figures for each user on his project.
- Can determine effective usage of the system by comparing the cpu and connect time by shift.
- Can monitor the secondary storage usage (actual usage versus assigned quota) of each user on his project.

### Environment Shaping

Finally, in order to shape the overall environment, he:

- Can give all users full system capabilities or can restrict any user on his project to either the Limited Service System or a closed subsystem. (A closed subsystem may duplicate a totally different system so that the user would need little, if any, knowledge of Multics.)
- Can, if given authority by the system administrator, give any user on his project special attributes. For example, he may allow a user guaranteed access to the system when he logs in or allow him to be preempted by another user.
- May set user dollar limits by month or shift or set user cutoff limits by date or dollar charge. If a logged-in user exceeds his limit, he is given a three-minute warning before being logged off the system.

## USER CAPABILITIES

There are several administrative functions available to the user.

### Resource Usage Monitoring

To monitor resource usage, the user:

- Checks his own use of resources, including total dollar usage, cpu and connect time by shift, I/O usage, and absentee usage.

### Access Control

- Specifies precisely to whom and with what access mode (read, write, or execute), part or all of his data is available.
- Can grant or revoke access at any time with a simple command, which goes into effect *immediately*.

### User Management

To manage usage, the user:

- Can create directories under his own home directory for his convenience. For example, he may wish to group programs or data either by category or under the same access control.
- May change his password or default project at log-in time if given authority by the system administrator.
- May specify how control will be transferred to his process at log-in time if given authority by the project administrator.

## FEATURES

In summary Multics administration offers a variety of features:

### Resource Distribution

- System configuration and operating parameters
- Disk quota
- Special attributes for users and projects

### Accounting

- Pricing
- Billing operations
- Adjustable update periods

### Usage Control

- Dollar limits by shift, month, or absolute
- Load control groups
- Real time versus cpu time (governor)
- Disk storage
- Dollar charge (absolute by project or user)
- Cutoffs by user, date, or dollar charge
- Project and user registration

### Environment Shaping

- Full Multics
- Limited Service System
- Closed subsystems

### Access Control and Security

- Password initially set by system administrator, changeable by user
- Person ID set by system administrator
- Project ID set by system administrator
- Person allowed on project by project administrator
- Access control by user and mode
- Ring protection

---

[1] Multics batch

HONEYWELL INFORMATION SYSTEMS

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

file

**Honeywell**

# TEXT PROCESSING

MULTICS

SOFTWARE

Among its many utility capabilities, Multics provides a multitude of text editing and formatting options that give the user flexibility in creating and maintaining error-free text and in producing properly formatted output. The Multics text processing routines include two text editors and a text formatting routine.

## TEXT EDITING

The standard Multics text editor, edm (editor for Multics), features 22 requests that can be learned quickly without having to spend time learning special computer languages. Other than familiarity with the edm requests and terminals, no special knowledge is needed.

Another editing routine, called qedx, offers a wider latitude of applications. Based on a powerful text editing language called QED, qedx offers a basic set of editing requests that, like edm, allows the user to create and edit ASCII files. In addition, the user wishing more than that basic ability can advance to another level of file manipulation and control. In fact, for the more advanced user, qedx becomes a relatively powerful pseudo-programming language.

### Common Features

There are many outstanding features common to both edm and qedx. For example, a data or source-program segment requiring editing can be virtually any size. The user doesn't need to know its size, physical or logical location, or manipulation by the Multics System.

For another example, if Multics system commands were executed during edm or qedx editing, the buffers being edited would not be affected. This allows the user to do his text editing concurrently with other program and data processing operations.

Editing requests in edm and qedx are convenient. For simplicity, most requests are abbreviated to just one character (e.g., *p* for *print,* and *c* for *change).* Some requests, such as *change,* can be directed to affect any range of lines, from one to all (globally). A string of characters can be added to the beginning or end of a line of text without having to reference any existing character string in the line.

Both edm and qedx can be invoked in a variety of ways, thus increasing their flexibility and range of applications. They may be called from the terminal, from a command list stored in a segment, or from a program; in addition, qedx can be called recursively.

To make editing simpler, segments need to be referenced only once in most cases. This is also true of character strings when the same string is to be used in several different editing requests. Both edm and qedx "remember" which segment is being edited as well as what character string is involved in a search.

All of these common features add up to fast, efficient editing on a Multics System.

### Features Specific to edm

- Verbose mode—prints helpful information for the new user. This mode can be inhibited by the user when he gains more experience and wishes to increase his editing speed.
- Line addressing—uses an imaginary pointer. Special edit requests manipulate this pointer forward or backward from its current position to the beginning or end of the segment, or to a content-indicated line.

### Features Specific to qedx

- qedx provides three basic methods of addressing:
  - by line number (an absolute search for a given line number).
  - by content (an absolute search for a character string).
  - by number of lines or content *relative* to a given line.

  Most importantly, these addressing methods can be combined in a single qedx request. For example, the request *+10/INTRODUCTION/p* would direct the computer search to begin 10 lines down from the current line of text, and to continue from that point to the first line containing the string INTRODUCTION. That line would be printed. Combining addressing methods provides the user speed and flexibility in editing his segments.
- In qedx, each edit request includes an address; there are no special requests needed to manipulate the pointer.
- qedx provides six error messages to inform the user of invalid addressing or of failure to find the specified character string.
- qedx provides temporary buffers that can be used to store parts of segments being edited, or to store editing macros built by the user. Editing control over a current text segment could be passed to one of these buffers which in turn would execute a series of editing requests on the text segment. The use of macros to automatically edit segments saves the user a great deal of time and is the best way to use the total power of qedx. A pseudo-program can be written once; then used over and over again in editing different material. A change in the user's standard editing practices would require only a one-time change to the particular editing macro involved; all his text would be changed automatically by using this new macro.
- For user convenience, some ASCII characters are given special significance in issuing qedx requests. When these characters are also part of a character string being searched for, the special significance usually afforded them can be masked, and the search performed without confusion.

(Continued)

Specifications may change as design improvements are introduced.

Order No.: AK56, Rev. 0

## TEXT FORMATTING

Multics ability to edit a file for correct content is only one feature. Editing it for proper formatting is another. The Multics utility routine for text formatting, called runoff, provides many options that give the user complete control over the way his text looks when it is visually displayed.

There are two steps in the formatting process. First, the text is edited with an editing routine such as edm or qedx to place the formatting control characters in the text. These control characters are selected from among the many offered by runoff.

Then, when the runoff command is issued, the control arguments imbedded in the text are executed. Also at this time, additional control arguments—of a more general nature—can be specified. These arguments might include the output device to be used, the method of handling page breaks, page numbering, "time-outs," or other general control considerations.

The general conditions of outputting are thus defined only once, in a list of control arguments given with the runoff command. This allows the same text to be handled in different ways with a minimum amount of user time involved in changing parameters.

### Imbedded Control Options

Some of the specific options for imbedded controls include:

- Control arguments can be imbedded in the text as separate lines beginning with a period or a blank space. Tab characters are converted into an appropriate string of blanks or line skips.
- The maximum number of characters per input or output line is 361. This permits 120 underlined characters plus the new line character.
- Expressions can be used in any request that requires a number argument and can be either arithmetic or string. Arithmetic operators include multiply, divide, plus, minus, equal, greater than, less than, boolean AND-OR-NOT, and others. Parentheses can be used for grouping. Either octal or decimal numbers can be used along with string constants. String variables can also be defined and substituted.
- Special built-in symbols in runoff are provided for footnote and equation numbering.
- Standard default arguments are used when none are specified.
- Margins can be set so that lines are left- or right-adjusted.
- Instructions can be given to print numeric values instead of character string values, or vice versa.
- The next line of text can be designated to begin at the top of a new page.
- The next line of text can be designated to begin on a new line of output. (i.e., a paragraph break)
- A line of text can be centered on the page.
- Spacing between lines can be specified as single, double, or some multiple.
- Headers and footers can be specified for all pages, or for even or odd pages only. Various control commands to manage these headers and footers can also be used. The page number can be set to print in the header or footer with the page number variable and automatically incremented by runoff after each page is created. Up to 20 lines of header and footer information can be printed on each page.
- Controls can be specified for equation lines.
- A Multics system command line can be imbedded in the text to be executed by the Multics supervisor when the segment is run off. For example, the user may wish to enter new information (e.g., variables or additional commands) at execution time. After the command is executed, control is returned to runoff.
- Special controls are available for creating footnotes and placing them at the bottom of a particular page.
- Options are provided to specify whether a line should be completely right-justified by inserting blanks or by merely filling the line with as many complete words as possible.
- A different named segment can be inserted in the middle of a text segment being edited. Segments can be merged; parts of segments, rearranged.
- Indenting can be specified for the next line of text or for all lines following.
- Labels or references can be defined for future control words.
- Runoff can be instructed to print n lines literally and not look for control lines or interpret any special characters.
- Line length can be set for all following lines.
- The top and bottom margins can be changed to a specified number of lines or incremented +n or -n from the current margins as set.
- Margins can be set between header, text, and foot copy.
- A block of lines can be reserved for a diagram, illustration, or photo.
- The text can be printed exactly as it appears in the text segment (same indenting, formatting, etc.).
- An odd page can be forced, as when a new chapter is desired.
- A new page can be forced and the page number incremented by n.
- The page length can be set to n lines or incremented by +n or -n lines.
- Arabic numerals can be converted to Roman.

### Command Options

These arguments can appear in the list following the runoff command:

- Naming the segment or segments to be processed and the device to which printing is directed, including specific features of the device such as the font being used.
- Flagging of designated key characters so that they will be printed as blanks, to be replaced later by special symbols. As the lines containing key characters are output, they are also written into a reminder segment, and their page and line numbers recorded for future reference.
- Setting the page number from which printing is to begin.
- Invoking a word hyphenation procedure.
- Setting an overall indentation control to center printed copy on the paper.
- Invoking or suppressing page breaks.
- Printing source line numbers in the left margin of the output.
- Presetting the initial page number and having subsequent pages renumbered.
- Setting an argument string for the internal parameter segment used by runoff.
- Specifying the number of preliminary passes that runoff will make over the

(Continued)

text to establish the value of certain symbols defined at various points in the input segment. The value of these symbols will be used as arguments to the final pass when output is produced.

- Directing output to a user-named segment rather than a printing device.
- Stopping the printing at the end of each page, to wait for a carriage return signal from the user before beginning the next.
- Specifying the page after which the printing will terminate.
- Specifying that printing of output should stop and not continue until a carriage return key is pressed.

## ADDITIONAL FEATURES OF THE MULTICS SYSTEM

The Multics System, which Honeywell feels is one of the most advanced computer systems in the world, offers its users many outstanding features including public-utility-like system reliability, virtual memory and hierarchical file storage, controlled sharing of information among many simultaneous users, data security enforced by both hardware and software, and the interactive terminal as its primary mode of access. Because of these features, especially the latter, the text processing user is among those who profit most from the Multics System.

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.

The Other Computer Company:

**Honeywell**

HONEYWELL INFORMATION SYSTEMS

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

File No.: 1L51

7238
5473 Printed in U.S.A.