

Copies: Corbett Podlysky
Schwader Clark

RECEIVED
JUL 31 1973
J. H. SALTZER

Multics Pre-release User's Planning Guide

Introduction

The Honeywell standard Multics software product is scheduled for release the first quarter of 1974. Additional releases are scheduled approximately every 9 months thereafter. The software included in the first release will not be identical to the software currently being run at MIT. Customers who receive software prior to the first release of the standard product will receive copies of the system in use at MIT, with MIT owned software removed. These pre-release systems are not fully compatible with the standard product. The purpose of this document is to describe the differences between pre-release systems and the standard system. It is intended to serve as a planning guide for users who are going to develop programs using pre-release versions of the system.

File-
Multics
history

I/O System Changes

1. The file-manager used to support PL/I record I/O will be replaced by DIMS or access methods that support variable length sequential record files and indexed sequential record files. Files created by the file_manager will not be supported by the standard product.
2. The file_DIM will be replaced by a similar DIM, but will only support element sizes of 9 bits.
3. The I/O switch (ios_) will be replaced by an improved switch facility that will:
 1. permit use of a switch in multiple rings.
 2. permit record I/O and stream I/O through the same switch facility.
 3. give much better performance.

Programs that use the existing I/O switch will continue to work with the first release, but support of the existing switch will be removed by the second release.

4. The tape_DIM will be modified to write larger records in order to take advantage of the higher recording density of the MTS-500 tape subsystem. The DIM will continue to read tapes with the smaller record size; however, programs that depend on the record size will have to be changed.
5. Tape software will be enhanced by the addition of tape reel access control. This will cause a change in the semantics of the tape reel identifier argument used in an I/O system attach call.

Fortran Changes

1. Unformatted files will no longer be implemented as 256 word records written as a stream file. They will be implemented as variable length sequential record files using the new I/O switch and new sequential record DIM.
2. The Fortran compiler will be extended to support nearly all of the language supported by Fortran Y (GCOS) and will also include compiler options to accept card image input and convert from card image input to free-form input.
3. Performance of the I/O routines and the math library will be improved significantly.
4. The meaning of a repetition factor on a constant in a data statement will be changed to conform with the ANSI standard. 5*1ha will mean five occurrences of 1ha, rather than one occurrence of 5haaaaa.
5. All formatted files, except those written on units 6 and 42 will be written as nonprint files. Formatted files written on units 6 and 42 will be written as print files. The first character of a print file will be transliterated into a carriage control character. The "+" carriage control character will be implemented for all print files except those attached to "user_output".

The current implementation writes all formatted files as print files, and does not support the "+" carriage control character.

PL/I Changes

1. The <connected attribute>, <backwards attribute>, and <forwards attribute> will be deleted.
2. The maximum precision of decimal data will be reduced from 63 to 59 digits. This change results from changes made to the 5180 hardware.
3. The order of keyed records in a data set will be the order of ascending keys, not chronological order. This makes Multics PL/I more compatible with other implementations, and will be accompanied by a very major improvement in performance.

All record files will be read and written by the revised Multics I/O system. They can be attached and switched in the same manner as stream files. This change will allow record files to be written and read from tape and other supported I/O devices. ANY record files created by the file_manager will not be supported by PL/I after these changes are

installed.

4. The stream scan performed for list or data directed input will be modified to conform with the draft ANSI standard. The modification will only affect cases where the stream contained an arithmetic constant and the list item was a character-string variable. This change will also be accompanied by a major performance improvement. Output of items by list or data directed output will be modified to split items that don't fit on a line by using standard rules rather than our current rules.
5. If the second argument of the "mod" built-in function is a zero, the result will be the converted value of the first argument, not the zerodivide condition.
6. If the second argument of the "before" built-in function is not a null string and is not a substring within the first argument, the result will be the converted value of the first argument, rather than the null string.
7. Attribute set completion for file constants will no longer add implied attributes prior to evaluating user supplied <default statement>s. Implied attributes are added at file opening. This change only affects programs that contain <default statement>s with file description attributes in their <predicate>s.
8. An implicit declaration of an identifier will not give the <variable attribute> to the declaration. This change allows defaulting to builtin or constant and only affects programs that contain <default statement>s with the <variable attribute> in their <predicate>s, and also contain one or more implicit declarations.
9. The <base reference> of the <defined attribute> will no longer be optional, and will no longer require parentheses. The only programs affected are those that use a <default statement> to supply the <base reference>. This change makes Version II compatible with Version I in its treatment of the <base reference>.
10. Conversion of a bit-string longer than 71 bits to fixed-point binary will signal the size condition if any high order bits are ones. The current implementation ignores these bits.
11. A structure containing only packed members will not be packed if it is declared aligned. The current compiler incorrectly allocates storage for such structures.

The following changes are extensions to our current PL/I implementation and are included in the draft ANSI standard. Their inclusion in this update of Multics PL/I will not affect any existing programs:

11. String built-in functions will take aggregate arguments and yield aggregate results.
12. Pseudo-variables will allow aggregate arguments.
13. The finish condition will be implemented.
14. The "some" and "every" built-in functions will be implemented.
15. Pictures will be implemented and their definition will be altered to conform to the standard.
16. Label array constants and isub defined arrays can be passed as arguments, but are always passed by value. The current implementation gives a diagnostic for label arrays and incorrectly passes isub defined arrays by reference.

Signal Mechanism Changes

1. The fault interceptor will be modified to separate the single hardware detected overflow fault into: overflow, underflow or fixedoverflow condition. It will transform a fixedoverflow resulting from an EIS decimal instruction into a size condition. An illegal procedure fault resulting from an EIS character move or compare instruction whose second operand has a zero length will be treated by the fault interceptor as a nop if detection of the stringsize condition is disabled, and will otherwise be transformed into a stringsize condition.

These changes effectively change the hardware as seen by a program. The revised hardware image is more rational and better suited to the needs of PL/I.

2. The snap and system options of the PL/I on-statement will be supported by the new signal routine. The effect of snap will be to call a procedure named pl1_snap_ which in turn will call debug. pl1_snap_ will be documented as a user-replaceable routine.
3. The command loop will be modified so that:
 - a) An automatic hold after a quit or error condition will continue to be performed, but the automatic release after one command will be removed.

- b) If more than one invocation of the listener exists, the ready message will contain

"listener n.f"

where n is the invocation level of the most recently invoked listener, and f is the stack frame number of that invocation.

Note that when only one invocation of the listener exists, the ready message appears as it does today. The extra information given when a stacked listener exists serves to remind the user that he has a stacked listener and probably a suspended computation as a result of a quit or error condition.

To insure that all ready messages printed by the standard system will contain the listener level information, the `print_ready_message_procedure` will be modified to supply the extra information.

- c) The condition wall will not be established with each new listener.
4. The "cleanup" condition will be enabled as any other condition and will be signalled as a true condition. The condition will be signalled when control is returned past a procedure activation that has established a "cleanup" on-unit. If multiple procedure activations have enabled on-units, the condition will be signalled as control attempts to return past each procedure activation, thus producing the desired effect.
5. The "any_other" condition name will represent any condition. An on-unit established for the "any_other" condition will respond to any condition for which an on-unit has not been established in that stack frame, and is effectively a condition wall or default handler. A signal of the "any_other" condition will signal a condition whose name is "any_other".
6. When "signal_" reaches the end of the stack in a given ring, it will test the condition to determine if it is one for which the system default on-unit can return control or signal another condition. If the default on-unit can do neither, a crawl-out occurs; otherwise control remains in the current ring.

this change makes it possible to have PL/I subsystems in inner-rings. Without the change, a simple endpage or stringsize condition causes a crawl-out.

7. The linker will be modified to not signal "linkage_error" unless there is a true linkage error. It will not signal a condition in order to implement the trap-before-link or trap-at-first-reference features.
8. The unwinder and signal_ will be reimplemented and separated from each other. Likewise the default error handler will be reimplemented and cleaned up. Error messages issued by the default error handler will be improved.
9. The data structures passed by a signalling program to the responding on-unit have been given a standard format which will facilitate the writing of user supplied default error handlers, as well as permitting a cleaner implementation of the system supplied default error handler.
10. The "finish" condition will be signalled just before a process terminates (crashes are the only case not included). It has a default on-unit that closes all open files. The process will continue to run if the user supplied on-unit does not return control. However, in some cases a subsequent process termination will be fatal. Process termination due to bumps or resource limit stops will allow a grace period before killing the process.
11. The start and release commands will take effect immediately, rather than at the end of the command line on which they occur.
12. The "stack" condition will be signalled by the firm when it detects a reference to the last 4 pages of the stack.

Runoff Changes

The runoff command will be replaced by another version that is coded in pL/I. That version will be slightly incompatible with the current version. Specifically it will:

1. Not produce a .chars segment.
2. Not have a symbol table.
3. Not have a footnote facility.
4. May not perform as well in the initial release. Later releases will perform as well or better than the current version.

Debugger Changes

The debug command will be replaced by a new command that is designed to be easier to use and more powerful. The new command is totally source program oriented and does not require the user to have knowledge of the structure of his object program. The new debugger may not be ready for the first release of the system. If not, it will be included in the second release.

ALM Changes

The ALM assembler will be replaced in a future release of the system. The new assembler will be faster than ALM and will contain a good macro facility. Users are advised not to use ALM because support for it will be dropped in some later version of the system.

BASIC Changes

The Dartmouth BASIC subsystem in use at MIT is not owned by Honeywell and will not be released or supported by Honeywell. The standard Multics system will contain a BASIC compiler designed and written for Multics. It supports a version of the BASIC language that is essentially compatible with the Dartmouth version 6 language, with the exception of those language features that depend on features of the Dartmouth system.

IPC Changes

In a future release of the system, the inter-process communication facility will be replaced or extensively revised. Users are advised not to use IPC, but instead should synchronize processes by means of the `set_lock` subroutine.

Mail Command Changes

The mail command and the inter-process message facility (`accept_message`, `send_message`, etc.) will be combined into a single facility. The mail command and the current inter-process message facility will not be supported after the second release. The new commands will store messages in a ring 1 message segment and will not use the contents of the existing mailbox or `con_msgs` segments.

Storage System Changes

1. As part of the modifications to simplify the access control mechanism, the CACL (Common Access Control List) facility is

being removed. The commands that maintain CACL's will be removed, the programs that compute access will be modified to use only the ACL (Access control List), and those supervisor entries that can be called to maintain CACL's and ACL's will be modified. If they are called to operate on a CACL they will return a status code.

The major purpose of CACL's was to supply a default ACL. That function has been replaced by an "initial ACL". Users of a system that contains support of CACL's can use the map_cacl command to transform their CACL's into ACL's and initial ACL's.

2. The max length attribute will be used to replace append permission for the control of the upper bound on a segment's size. Under the current implementation, sufficient max length and append permission are needed to increase a segment beyond its current length. The system will be changed to not require the presence of the append attribute. For compatibility with current programs, entry points to the supervisor will continue to accept an attempt to set append access. These entry points will not support such calls in the second release.
3. The maximum length of a segment will be increased from 64K to 256K. This will have the effect that certain incorrectly coded programs that used to get an out-of-bounds fault will instead operate in some unspecified manner because of address wrap-around. Users can prevent this from happening by the appropriate setting of the max length attribute.
4. It will no longer be necessary to have write access to a segment in order to delete it. A segment can be deleted if and only if the safety switch is off and the user has modify access to the directory containing the segment.

Because of the enhancements being made to the storage system, some of the commands that are used to obtain and set information pertaining to the contents of the storage system will be extended to accommodate new attributes and the existence of multi-segment files.

Administrative Software Changes

A number of enhancements to the administrative software will add and change some of the practices and procedures used by administrators. Among these are:

1. The addition of tape reel management software will require all tape reels for which access control is to be performed to be registered on the system.

2. A new facility to aid the system Administrator in defining new projects will be used to replace the current tools.
3. The method of adding names to the access control lists of gates to the supervisor will be changed.
4. To make new system installation easier, the startup procedures which are controlled by the system administrator will be changed.

Operational Software Changes

Because of the modifications of the system to operate with new peripheral equipment (MIS-500, DSS-191, Unit Record Controller, System Control Center) various operating procedures of BOS and Multics will be changed. These changes will also imply changes to some of the BOS configuration cards.

Enhancements to the I/O daemon facility to add new operational features and the control of remote printers will cause some of the operating procedures associated with the I/O daemon to change.

Some of the tools used as part of crash recovery and dump analysis will be changed to make operation simpler.

Modifications to the system to support very large storage requirements and the removable hierarchy facility will lead to new operational procedures. In particular, the current backup, reload, and retrieval facility will be replaced by a more efficient and reliable mechanism in a future release.

The following commands are being removed:

```
v5basic
basic
basic_run
basic_system
print_dartmouth_library
sat_dartmouth_library
decam
v2pl1_abs
setcacl
listcacl
delatecacl
```

The following subroutines are being removed:

```
hcs_$iel_dir_trea_
file_manager_
```

Additional subroutines and commands may be removed or replaced as part of the previously described changes.