

TO: Distribution

MSB-104

FROM: R.A. Freiburghouse

DATE: May 31, 1973

SUBJECT: The New Signal Mechanism

The new signal mechanism proposed in MSB-95 generated considerable discussion and numerous counter proposals or suggestions. We have reviewed those comments and have modified our original proposal accordingly.

The purpose of this document is to briefly describe the changes we plan to make to the signal mechanism. Implementation of some of these changes has already begun and all are scheduled for installation this year.

Several of the most controversial proposals of MSB-95 have been changed or dropped. This was due to well supported arguments given by several people. It is unfortunate that past design decisions were made without similar public exposure and comment.

1. The fim will be modified to separate the single hardware detected overflow fault into: overflow, underflow or fixed-overflow 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 fim as a nop if detection of the string-size 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 p~~l~~1\_snap\_ which in turn will call debug. p~~l~~1\_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 or an "any" 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. This change will be made at the same time as the "finish" and "stack" conditions are implemented. Users who have their own default error handlers will have to make only one set of changes to process or pass-on these new conditions.
  5. The "any" condition name will represent any condition. An on-unit established for the any condition will respond to all 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" condition will signal a condition whose name is "any".
  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 string-size 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 so-called pre-signal or ripple-up of signals will not be implemented at this time. It is an extension which probably should be made only if full facilities to use an inner ring as a monitor are provided by the system. The fim will continue to signal in the faulting ring and software detected conditions will not pass through ring 0.
11. 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.
12. The start and release commands will take effect immediately, rather than at the end of the command line on which they occur.
13. The "stack" condition will be signalled by the fim when it detects a reference to the last 4 pages of the stack.