

Published: 6/7/66

Identification

Instruction-by-instruction interpretive execution of programs.

monitor

D. B. Wagner

Purpose

Monitor accepts requests which cause certain areas of programs, whenever entered, to be executed interpretively instead of being allowed to run freely. Used in conjunction with the tracer command it allows very tight control of the execution of a program which is causing trouble.

Usage

The command

monitor

causes monitor to begin reading requests from the console. The user may type any of the requests listed below or any of the "control" requests (if, else, do, end) described in BX.10.00. He might also type macro invocations (in the same form as in the command language: see BX.1.01) which expand to sequences of these requests. If a line received by monitor (after macro expansion) is not recognizable as a request, it is treated as a command. The line is given to the Shell, which gives an appropriate diagnostic if it is not a command either. \*

Requests to Monitor

The request

setmon location to location

causes arrangements to be made so that all execution of instructions in the block specified is done under strict supervision. Whenever control reaches a location in a block specified in a setmon request, an interpreter gains control of the process (by the same mechanism as that used by the access event in breaker) and executes the machine instructions interpretively. At every instruction a call to the tracer entry tracer\$report (see BX.10.02) is made with appropriate arguments. Name is a character-string expression which is to be used as the first (identification) argument in these calls. It is expected that interpretive execution of machine instructions will take roughly 50x normal execution time.

The request

resetmon name

resets any setmon's which have been given for the specified name.

The request

exit

causes monitor to return to its caller, normally the Shell.