(Meter) MPS Metering Package (EHS, 2/22/73)

(introduction) Introduction

This note describes an interim set of procedures for monitoring the behavior of MPS programs. The metering facilities are currently based upon counts and collections of counts, called profiles. The implementation of metering uses counters which are incremented each time control reaches designated points in the code for a program.

(procedures) Procedures Available

Metering is based upon dynamically allocated counters and upon patches, which increment those counters, temporarily inserted into the object code. Counters are manipulated by procedures declared within the metering package. Each procedure performs some action upon some counter or set of counters. The name of each metering procedure is the concatenation of a prefix describing the action and a suffix partially specifying the range of that action.

Actions

Begin

Allocates a counter, inserts code to increment the counter, and sets the value of the counter to zero.

Zero

Sets the value of an existing counter to zero.

Print

Prints the value of an existing counter with format and identification appropriate for the quantification requested.

End

Removes the code to increment an existing counter and deallocates that counter.

Ranges

EntryCount

Specifies the (internal) entry point of a designated procedure body.

Counts

Specifies the (internal) entry points of all procedure bodies within a designated program module.

Profile

Specifies all "interesting" statements in a procedure body.

An MPL statement is "interesting" if the corresponding source statement is the first MPL statement occurring within an NLS plex (or, less precisely, the first MPL statement occurring after an increase in the depth, relative to the NLS tree structure, of the source text).

Profiles

Specifies all "interesting" statements of all procedure

bodies within a designated program module.

When a range specification requires the designation of a procedure body, an actual procedure is required as a parameter, and it designates the (code) body of that procedure. When the specification requires the designation of a program module, an address is required as a parameter. That parameter must be the address of a word within the stack or data segment of some process, and it designates the program module containing the code for that process [see (docmps,debugger,ProcessID)].

Note that counts are accumulated over all uses of the metered code, i.e., over the equivalence classes of procedure activations or processes which are defined by the sharing of code [see, however, (,appendix)].

The following combinations of action and range are provided by the procedures in the standard metering package:

(BeginEntryCount) PROCEDURE(PROCEDURE p)

(ZeroEntryCount) PROCEDURE(PROCEDURE p)

(EntryCount) PROCEDURE(PROCEDURE p)
%returns the value of the specified counter%

(EndEntryCount) PROCEDURE(PROCEDURE p)

(BeginCounts) PROCEDURE(addr)

(ZeroCounts) PROCEDURE(addr)

(PrintCounts) PROCEDURE(addr)

(EndCounts) PROCEDURE(addr)

(BeginProfile) PROCEDURE(PROCEDURE p)

(ZeroProfile) PROCEDURE(PROCEDURE p)

(PrintProfile) PROCEDURE(PROCEDURE p)

(EndProfile) PROCEDURE(PROCEDURE p)

(BeginProfiles) PROCEDURE(addr)

(ZeroProfiles) PROCEDURE(addr)

(PrintProfiles) PROCEDURE(addr)

(EndProfiles) PROCEDURE(addr)

(use) Use

Initialization of the metering package requires running a process created from the file <MPS>METER.MPS; this, in turn, creates a private subprocess from the file <MPS>STMTMAP.MPS. Interpretation of the debugging command

meter()

initializes the metering package, and the names of the metering procedures become known in the debugger's standard context.

The output routines which operate upon an entire program module print an identification of the module and of the procedure associated with each count or set of counts.

The routines which display profiles also print an outline of the program. Each NLS statement is represented by a vertical bar

indented to show the relative location of that statement in the NLS tree structure. Counts appear immediately to the right of those vertical bars which mark the positions of "interesting" statements.

The outline produced for each plex usually ends with the last element of that plex containing an MPS statement, i.e., trailing or subordinate declarations or comments are likely to be omitted.

A count applies only to the first MPL phrase or statement within that NLS statement with which it is associated. Interpretation of such counts thus requires inspection of the corresponding source text. If reasonable coding conventions have been used, the frequency counts for all basic blocks within a procedure often can be deduced from those which, by the (heuristically chosen) definition above, are "interesting."

Note that the counting mechanism is not disabled during, e.g., the printing of counts or the interpretation of debugging commands. This must be taken into account when analyzing basic system modules such as the debugger or string package [see, however, (,appendix)].

(appendix) Selective Metering

A somewhat more selective metering package is available in the file <MPS>SMETER.MPS. In this package, each counter is associated with a unique process; otherwise, specifications and behavior are identical to those of the standard package.

If the specification of the counter location requires a procedure as a parameter, the process is the one in which the actual parameter was declared.

If the specification requires an address as a parameter, the process is the one with the data or stack segment containing the address.

In conjunction with an appropriate process structure, selective metering permits the examination of the use of procedures declared within a particular process.

Processes sharing code from the same program module can be metered simultaneously and independently.

The selective metering package is somewhat larger and degrades execution speed slightly more than the standard metering package.