Published: 01/09/68

Identification

Segment-Usage Metering
N. I. Morris, M. A. Padlipsky

Purpose

In order to determine which system segments are being used most frequently, the following scheme for gathering statistics has been evolved. The same mechanism is also used for gathering timing data on page faults and segment faults.

Discussion

The basic tools used in this effort are a modified version of Fault Interceptor Module (FIM) and a special interrupt handler for the alarm clock interrupt. It is intended that the "special" FIM be used in all system initialization bootload runs until Initial Multics, at least, and possibly thereafter.

Special handling is accorded to three cases: missing page fault, missing segment fault, and alarm clock interrupt. For the first two, a count and a running time total will be kept. The alarm clock interrupt is a somewhat more elaborate issue. Basically, the sampling of segment usage takes place as follows: the calendar alarm clock is set to run out after a certain period of time (initially, 10 milliseconds will be the period); when the alarm clock interrupt occurs, the alarm clock interrupt handler will record which segment was being executed at the time by incrementing a count of the number of times that segment has been encountered when the timer ran out, and then the clock will be reset to run out again--at which time the segment encountered then will be counted, and so on.

Data Base

A wired-down segment, <segment_meter_table>, serves as the data base for statistics-gathering. It contains two blocks of five words each for page and segment faults, a running count of the number of alarm clock interrupts handled (which gives the length of the run), and an array of some 2200(10) words for storing segment counts. The page and segment fault blocks contain the count, the running time total, an "in-process" switch, and the time the last fault occurred, in the following format:

Location	Contents
0-1	Calendar clock time last page fault was taken
2-3	Calendar clock time last segment fault was taken
4	Running time total for page faults (in microseconds)
5	Running time total for segment faults (in microseconds)
6	Number of page faults processed
7	Number of segment faults processed
10	Non-zero if segment fault currently being processed
11	Non-zero if page fault currently being processed
12	Total number of clock interrupts handled (run time in 10's of milliseconds)
13	Non-zero if metering is being performed.

(Note that when segment faults are taken during the handling of segment faults, the time the last fault occurred is not overwritten, thus arriving at the total segment fault time. Also, segment fault time is adjusted not to include time needed to process page faults while handling a segment fault.) Each word in the segment count array from location 14 to the end contains the count for the correspondingly-numbered segment.

Monitoring

Procedure print_statistics is furnished to cause the information in <segment_meter_table> to be converted to character strings and printed on the on-line printer. The calling sequence is:

```
call print_statistics;
```

Calls may be made from appropriate points in the Multics Initializer, or, for that matter, through the Shell. Print_statistics will set the switch at location 13(8) in <segment_meter_table> to zero while operating, so that metering will not be performed during this time; it will set the switch back to 1 on exit.