SETL Newsletter 88 Nov. 10, 1972

A scheme for BALMSETL measurements. J. Schwartz

The crucial impact which run time will have on the possibility of expanding the SETLB user community gives particular importance to measurements which can clarify our understanding of the factors currently affecting run time. Dave Shields and Kent Curtis will be generating a systematic series of timing measurements for all the important SETL operations, as applied to 'typical' data. This newsletter will suggest a larger-scale measurement which can probably be run quite easily and which should substantially increase our understanding of the typical pattern of actions involved during SETLB runs. Specifically, I propose to measure the number of times each of the BALMSETL primitives is invoked during runs of the programs currently in the SETLB test library. For this, the following technique, patterned after the approach to software paging outlined in Newsletter 86, would probably be appropriate.

For each routine whose entries are to be counted, create a dummy routine which catches calls to it, and increments a counter before passing along the call. The counts can be kept in a BALM vector *meascts*; external names of subroutines should be kept also, in an associated BALM vector *measnms*. Procedures themselves will be kept in a vector *measrouts*. Counting will be set up by calling the following macro:

```
measure(procd(x),name) means
do measrouts(globalctr) = procd, meascts(globalctr)=0,
   measnms(globalctr)=name, globalctr = globalctr + 1,
procd = proc(x), sub = measrouts([present value of globalctr]),
 result = sub(x), meascts([present value of globalctr])
   = meascts([present value of globalctr]) + 1,
  return(result) end end.
```

Initiation of a run to be measured would be preceded by a sequence of statements (in SETLB) such as

```
measure (procl(onearg),'procl');
measure(proc2(threeargs), 'proc2'); etc.
```

which might be read in with the material for the standard SETLB prefix.
Before exit, an obvious procedure

```
dmpmeas;
```

should be executed; this will print out the measurements collected.