

**NAME**

monitor – prepare execution profile

**SYNOPSIS**

```
monitor(lowpc, highpc, buffer, bufsize)  
int lowpc( ), highpc( ), buffer[ ], bufsize;
```

**DESCRIPTION**

*Monitor* is an interface to the system's profile entry (II). *Lowpc* and *highpc* are the names of two functions; *buffer* is the address of a (user supplied) array of *bufsize* integers. *Monitor* arranges for the system to sample the user's program counter periodically and record the execution histogram in the buffer. The lowest address sampled is that of *lowpc* and the highest is just below *highpc*. For the results to be significant, especially where there are small, heavily used routines, it is suggested that the buffer be no more than a few times smaller than the range of locations sampled.

To profile the entire program, it is sufficient to use

```
extern etext;  
...  
monitor(2, &etext, buf, bufsize);
```

*Etext* is a loader-defined symbol which lies just above all the program text.

To stop execution monitoring and write the results on the file *mon.out*, use

```
monitor(0);
```

Then, when the program exits, *prof* (I) can be used to examine the results.

It is seldom necessary to call this routine directly; the **-p** option of *cc* is simpler if one is satisfied with its default profile range and resolution.

**FILES**

mon.out

**SEE ALSO**

*prof* (I), *profil* (II), *cc* (I)