NAME

profil – execution time profile

SYNOPSIS

#include <unistd.h>

void profil(unsigned short **buff*, **unsigned int** *bufsiz*, **unsigned int** *offset*, **unsigned int** *scale*);

DESCRIPTION

profil( ) provides CPU-use statistics by profiling the amount of CPU time expended by a program. **profil( )** generates the statistics by creating an execution histogram for a current process. The histogram is defined for a specific region of program code to be profiled, and the identified region is logically broken up into a set of equal size subdivisions, each of which corresponds to a count in the histogram. With each clock tick, the current subdivision is identified and its corresponding histogram count is incremented. These counts establish a relative measure of how much time is being spent in each code subdivision. The resulting histogram counts for a profiled region can be used to identify those functions that consume a disproportionately high percentage of CPU time.

buff is a buffer of *bufsiz* bytes in which the histogram counts are stored in an array of **unsigned short int**.

offset, *scale*, and *bufsiz* specify the region to be profiled.

offset is effectively the start address of the region to be profiled.

scale, broadly speaking, is a contraction factor that indicates how much smaller the histogram buffer is than the region to be profiled. More precisely, *scale* is interpreted as an unsigned 16-bit fixed-point fraction with the decimal point implied on the left. Its value is the reciprocal of the number of bytes in a subdivision, per byte of histogram buffer. Since there are two bytes per histogram counter, the effective ratio of subdivision bytes per counter is one half the scale.

Several observations can be made:

- the maximal value of *scale*, **0xffff** (approximately 1), maps subdivisions 2 bytes long to each counter.
- the minimum value of *scale* (for which profiling is performed), **0x0002** (1/32,768), maps subdivision 65,536 bytes long to each counter.
- the default value of *scale* (currently used by **cc −qp**), **0x4000**, maps subdivisions 8 bytes long to each counter.

The values are used within the kernel as follows: when the process is interrupted for a clock tick, the value of *offset* is subtracted from the current value of the program counter (pc), and the remainder is multiplied by *scale* to derive a result. That result is used as an index into the histogram array to locate the cell to be incremented. Therefore, the cell count represents the number of times that the process was executing code in the subdivision associated with that cell when the process was interrupted.

scale can be computed as (*RATIO* ∗ **0200000L**), where *RATIO* is the desired ratio of *bufsiz* to profiled region size, and has a value between 0 and 1. Qualitatively speaking, the closer *RATIO* is to 1, the higher the resolution of the profile information.

bufsiz can be computed as (*size_of_region_to_be_profiled* ∗ *RATIO*).

SEE ALSO

exec(2), **fork**(2), **times**(2), **monitor**(3C), **prof**(5)

NOTES

Profiling is turned off by giving a *scale* of 0 or 1, and is rendered ineffective by giving a *bufsiz* of 0. Profiling is turned off when an **exec**(2) is executed, but remains on in both child and parent processes after a **fork**(2). Profiling is turned off if a *buff* update would cause a memory fault.