NAME

readprofile – read kernel profiling information

SYNOPSIS

readprofile [options]

VERSION

This manpage documents version 2.0 of the program.

DESCRIPTION

The **readprofile** command uses the */proc/profile* information to print ascii data on standard output. The output is organized in three columns: the first is the number of clock ticks, the second is the name of the C function in the kernel where those many ticks occurred, and the third is the normalized 'load' of the procedure, calculated as a ratio between the number of ticks and the length of the procedure. The output is filled with blanks to ease readability.

OPTIONS

–a, **––all**
: Print all symbols in the mapfile. By default the procedures with reported ticks are not printed.

–b, **––histbin**
: Print individual histogram-bin counts.

–i, **––info**
: Info. This makes **readprofile** only print the profiling step used by the kernel. The profiling step is the resolution of the profiling buffer, and is chosen during kernel configuration (through 'make config'), or in the kernel's command line. If the **–t** (terse) switch is used together with **–i** only the decimal number is printed.

–m, **––mapfile** *mapfile*
: Specify a mapfile, which by default is */usr/src/linux/System.map*. You should specify the map file on cmdline if your current kernel isn't the last one you compiled, or if you keep System.map elsewhere. If the name of the map file ends with '.gz' it is decompressed on the fly.

–M, **––multiplier** *multiplier*
: On some architectures it is possible to alter the frequency at which the kernel delivers profiling interrupts to each CPU. This option allows you to set the frequency, as a multiplier of the system clock frequency, HZ. Linux 2.6.16 dropped multiplier support for most systems. This option also resets the profiling buffer, and requires superuser privileges.

–p, **––profile** *pro-file*
: Specify a different profiling buffer, which by default is */proc/profile*. Using a different pro-file is useful if you want to 'freeze' the kernel profiling at some time and read it later. The */proc/profile* file can be copied using 'cat' or 'cp'. There is no more support for compressed profile buffers, like in **readprofile-1.1,** because the program needs to know the size of the buffer in advance.

–r, **––reset**
: Reset the profiling buffer. This can only be invoked by root, because */proc/profile* is readable by everybody but writable only by the superuser. However, you can make **readprofile** set-user-ID 0, in order to reset the buffer without gaining privileges.

–s, ––counters
: Print individual counters within functions.

–v, **––verbose**
: Verbose. The output is organized in four columns and filled with blanks. The first column is the RAM address of a kernel function, the second is the name of the function, the third is the number of clock ticks and the last is the normalized load.

–V, **––version**
: Display version information and exit.

−h, **−−help**
Display help text and exit.

EXAMPLES

Browse the profiling buffer ordering by clock ticks:
readprofile | sort -nr | less

Print the 20 most loaded procedures:
readprofile | sort -nr +2 | head -20

Print only filesystem profile:
readprofile | grep _ext2

Look at all the kernel information, with ram addresses:
readprofile -av | less

Browse a ‘frozen’ profile buffer for a non current kernel:
readprofile -p ~/profile.freeze -m /zImage.map.gz

Request profiling at 2kHz per CPU, and reset the profiling buffer:
sudo readprofile -M 20

BUGS

readprofile only works with a 1.3.x or newer kernel, because */proc/profile* changed in the step from 1.2 to 1.3

This program only works with ELF kernels. The change for a.out kernels is trivial, and left as an exercise to the a.out user.

To enable profiling, the kernel must be rebooted, because no profiling module is available, and it wouldn’t be easy to build. To enable profiling, you can specify "profile=2" (or another number) on the kernel commandline. The number you specify is the two-exponent used as profiling step.

Profiling is disabled when interrupts are inhibited. This means that many profiling ticks happen when interrupts are re-enabled. Watch out for misleading information.

FILES

/proc/profile A binary snapshot of the profiling buffer.
/usr/src/linux/System.map The symbol table for the kernel.
/usr/src/linux/* The program being profiled :-)

AVAILABILITY

The readprofile command is part of the util-linux package and is available from Linux Kernel Archive ⟨https://www.kernel.org/pub/linux/utils/util-linux/⟩.