

## NAME

**pmcannotate** - sources printout with inlined profiling

## SYNOPSIS

**pmcannotate** [-a] [-h] [-k *pathname*] [-l *level*] *pmcout.out* *binaryobj*

## DESCRIPTION

The **pmcannotate** utility can produce both C sources or assembly sources of a program with a line-by-line based profiling. The profiling information is retrieved through a *pmcstat*(8) raw output while the program operations are retrieved through the *objdump*(1) tool.

When calling **pmcannotate** the raw output is passed through the *pmcout.out* argument, while the program is passed through the *binaryobj* argument.

As long as **pmcannotate** relies on *objdump*(1) and *pmcstat*(8) to work, it will fail if one of them is not available.

## OPTIONS

The following options are available:

- a Shows the program profiling inlined in the assembly code only. No C information involving C sources is provided.
- h Prints out information about the usage of the tool.
- l *level*  
Changes the lower bound (expressed in percentage) for traced functions that will be printed out in the report. The default value is 0.5%.
- k *kerneldir*  
Set the pathname of the kernel directory to argument *kerneldir*. This directory specifies where **pmcannotate** should look for the kernel and its modules. The default is */boot/kernel*.

## LIMITATIONS

As long as **pmcannotate** relies on the *objdump*(1) utility to retrieve the C code, the program needs to be compiled with debugging options. Sometimes, in particular with heavy optimization levels, the *objdump*(1) utility embeds the code of inlining functions directly in the callers, making an output difficult to read. The x86 version reports the sampling from *pmcstat* collecting the following instruction in regard of the interrupted one. This means that the samples may be attributed to the line below the one of interest.

**SEE ALSO**

objdump(1), pmcstat(8)

**AUTHORS**

Attilio Rao <*attilio@FreeBSD.org*>