

**NAME**

**pmc\_disable**, **pmc\_enable** - administrative control of hardware performance counters

**LIBRARY**

Performance Counters Library (libpmc, -lpmc)

**SYNOPSIS**

```
#include <pmc.h>
```

*int*

```
pmc_disable(int cpu, int pmc);
```

*int*

```
pmc_enable(int cpu, int pmc);
```

**DESCRIPTION**

These functions allow specific hardware performance monitoring counters in a system to be disabled and enabled administratively. The hardware performance counters available on each CPU are numbered using small non-negative integers, in a system dependent manner. Disabled counters will not be available to applications for use.

The invoking process needs to have the PRIV\_PMC\_MANAGE privilege to perform these operations.

Function **pmc\_disable**() disables the hardware counter numbered by argument *pmc* on CPU number *cpu*.

Function **pmc\_enable**() enables the hardware counter numbered by argument *pmc* on CPU number *cpu*.

**IMPLEMENTATION NOTES**

Hardware PMCs that are currently in use by applications cannot be disabled. Allocation of a process scope software PMC marks all hardware PMCs in the system with the same *pmc* number as being in-use.

**RETURN VALUES**

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

**ERRORS**

A call to these functions may fail with the following errors:

[EBUSY]           Function **pmc\_disable**() specified a hardware PMC is currently in use.

- [EINVAL]           Arguments *cpu* or *pmc* were invalid.
- [ENXIO]            Argument *cpu* specified a disabled or absent CPU.
- [EPERM]            The current process lacks sufficient privilege to perform this operation.

**SEE ALSO**

pmc(3), pmc\_cpuintro(3), pmc\_pmcinfo(3), hwpmc(4), pmccontrol(8), priv\_check(9)