

**NAME**

**pmc\_name\_of\_capability**, **pmc\_name\_of\_class**, **pmc\_name\_of\_cputype**, **pmc\_name\_of\_disposition**, **pmc\_name\_of\_event**, **pmc\_name\_of\_mode**, **pmc\_name\_of\_state** - human readable names for numeric constants used by `pmc(3)` and `hwpmc(4)`

**LIBRARY**

Performance Counters Library (`libpmc`, `-lpmc`)

**SYNOPSIS**

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

```
const char *
```

```
pmc_name_of_capability(enum pmc_caps pc);
```

```
const char *
```

```
pmc_name_of_class(enum pmc_class pc);
```

```
const char *
```

```
pmc_name_of_cputype(enum pmc_cputype ct);
```

```
const char *
```

```
pmc_name_of_disposition(enum pmc_disp pd);
```

```
const char *
```

```
pmc_name_of_event(enum pmc_event pe);
```

```
const char *
```

```
pmc_name_of_mode(enum pmc_mode pm);
```

```
const char *
```

```
pmc_name_of_state(enum pmc_state ps);
```

**DESCRIPTION**

These convenience functions translate numeric constants used by the Performance Counters Library (`libpmc`, `-lpmc`) to `const char *` pointers to human readable representations of their arguments.

Function **pmc\_name\_of\_capability()** translates a PMC capability flag given in argument `pc` to a human readable string. PMC capabilities are described in `pmc(3)`.

Function **pmc\_name\_of\_class()** translates the PMC class value specified in argument `pc` to a human

readable name. PMC classes are described in `pmc(3)`.

Function `pmc_name_of_cputype()` translates the CPU type value specified in argument *ct* to a human readable name. CPU types known to the library are described in `pmc(3)`.

Function `pmc_name_of_disposition()` translates the PMC row disposition specified in argument *pd* to a human readable name. PMC row dispositions are described in `hwpmc(4)`.

Function `pmc_name_of_event()` translates the PMC event number specified by argument *pe* to a string. PMC event names are documented in section *EVENT SPECIFIERS* of `pmc(3)`.

Function `pmc_name_of_mode()` translates the PMC mode specified by argument *pm* to a human readable string. PMC modes are described in `pmc(3)`.

Function `pmc_name_of_state()` translates the value of argument *ps* to a human readable name.

## IMPLEMENTATION NOTES

The returned pointers point to static storage inside the PMC library and should not be freed by the caller.

## RETURN VALUES

These functions return a non-NULL pointer on successful completion. In case of an error, a NULL pointer 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:

[EINVAL]           The function argument specified an invalid value.

## SEE ALSO

`pmc(3)`, `pmc_cpuinfo(3)`, `pmc_pmcinfo(3)`, `hwpmc(4)`