NAME

perfmonctl − interface to PMU

SYNOPSIS

#include <syscall.h>
#include <perfmon.h>

long perfmonctl(int *fd***, int** *cmd***, void *****arg***, int** *narg***);**

DESCRIPTION

perfmonctl system call provides an interface to PMU (performance monitoring unit). PMU consists of PMD (performance monitoring data) registers and PMC (performance monitoring control) registers, where are gathered the hardware statistic.

perfmonctl will apply a function *cmd* to input arguments *arg.* The number of arguments is defined by input variable *narg*. *fd* specifies the perfmon context to operate on.

The implemented commands *cmd* are:

PFM_CREATE_CONTEXT
set up a context
perfmonctl(int *fd***, PFM_CREATE_CONTEXT , pfarg_context_t *****ctxt***, 1);**

The *fd* parameter is ignored. A new context is created as specified in *ctxt* and its file descriptor is returned in *ctxt->ctx_fd*.

The file descriptor, apart from passing it to **perfmonctl**, can be used to read event notifications (type **pfm_msg_t**) using the **read**(2) system call. Both **select**(2) and **poll**(2) can be used to wait for event notifications.

The context can be destroyed using the **close**(2) system call.

PFM_WRITE_PMCS
set PMC registers
perfmonctl(int *fd***, PFM_WRITE_PMCS , pfarg_pmc_t *****pmcs***, n);**

PFM_WRITE_PMDS
set PMD registers
perfmonctl(int *fd***, PFM_WRITE_PMDS , pfarg_pmd_t *****pmds***, n);**

PFM_READ_PMDS
read PMD registers
perfmonctl(int *fd***, PFM_READ_PMDS , pfarg_pmd_t *****pmds***, n);**

PFM_START
start monitoring
perfmonctl(int *fd***, PFM_START , arg , 1);**
perfmonctl(int *fd***, PFM_START , NULL , 0);**

PFM_STOP
stop monitoring
perfmonctl(int *fd***, PFM_START , NULL , 0);**

PFM_LOAD_CONTEXT
attach the context to a thread
perfmonctl(int *fd***, PFM_LOAD_CONTEXT ,pfarg_load_t *****largs***,*1);*

PFM_UNLOAD_CONTEXT
detach the context from a thread
perfmonctl(int *fd***, PFM_UNLOAD_CONTEXT , NULL , 0);**

PFM_RESTART
restart monitoring after recieving an overflow notification
perfmonctl(int *fd***, PFM_RESTART , NULL , 0);**

PFM_CREATE_EVTSETS
create or modify event sets
perfmonctl(int *fd***, PFM_CREATE_EVTSETS, pfarg_setdesc_t *desc , n);**

PFM_DELETE_EVTSETS
delete event sets
perfmonctl(int *fd***, PFM_DELETE_EVTSET, pfarg_setdesc_t *desc , n);**

PFM_GETINFO_EVTSETS
get information about event sets
perfmonctl(int *fd***, PFM_GETINFO_EVTSETS, pfarg_setinfo_t *info, n);**

RETURN VALUE

performctl returns zero when the operation is successful. On error -1 is returned and an error code is set in **errno**.

AVAILABILITY

This syscall is implemented only on the IA-64 architecture since kernel 2.6.

SEE ALSO

gprof(1), The perfmon2 interface specification