

**NAME**

**watchdog** - software and hardware watchdog facility

**SYNOPSIS**

```
#include <sys/watchdog.h>
```

*void*

```
watchdog_fn(void *private, u_int cmd, int *error);
```

```
EVENTHANDLER_REGISTER(watchdog_list, watchdog_fn, private, 0);
```

```
EVENTHANDLER_DEREGISTER(watchdog_list, eventhandler_tag);
```

**DESCRIPTION**

To implement a watchdog in software or hardware, only a single function needs to be written and registered on the global *watchdog\_list*.

The function must examine the *cmd* argument and act on it as follows:

If *cmd* is zero, the watchdog must be disabled and the *error* argument left untouched. If the watchdog cannot be disabled, the *error* argument must be set to EOPNOTSUPP.

Else the watchdog should be reset and configured to a timeout of  $(1 \ll (cmd \& WD\_INTERVAL))$  nanoseconds or larger and the *error* argument be set to zero to signal arming of a watchdog.

If the watchdog cannot be configured to the proposed timeout, it must be disabled and the *error* argument left as is (to avoid hiding the arming of another watchdog).

There is no specification of what the watchdog should do when it times out, but a hardware reset or similar "drastic but certain" behaviour is recommended.

**SEE ALSO**

watchdog(4)

**AUTHORS**

The **watchdog** facility and this manual page was written Poul-Henning Kamp <phk@FreeBSD.org>.