NAME

g_post_event, **g_waitfor_event**, **g_cancel_event** - GEOM events management

SYNOPSIS

#include <geom/geom.h>

int
g_post_event(*g_event_t *func, void *arg, int flag, ...*);

int
g_waitfor_event(*g_event_t *func, void *arg, int flag, ...*);

void
g_cancel_event(*void *ref*);

*struct g_event **
g_alloc_event(*int flag*);

void
g_post_event_ep(*g_event_t *func, void *arg, struct g_event *ep, ...*);

DESCRIPTION

The GEOM framework has its own event queue to inform classes about important events. The event queue can be also used by GEOM classes themselves, for example to work around some restrictions in the I/O path, where sleeping, heavy weight tasks, etc. are not permitted.

The **g_post_event**() function tells the GEOM framework to call function *func* with argument *arg* from the event queue. The *flag* argument is passed to malloc(9) for memory allocations inside of **g_post_event**(). The only allowed flags are M_WAITOK and M_NOWAIT. The rest of the arguments are used as references to identify the event. An event can be canceled by using any of the given references as an argument to **g_cancel_event**(). The list of references has to end with a NULL value.

The **g_waitfor_event**() function is a blocking version of the **g_post_event**() function. It waits until the event is finished or canceled and then returns.

The **g_post_event_ep**() function posts the event with a pre-allocated *struct g_event*. An event may be pre-allocated with **g_alloc_event**().

The **g_cancel_event**() function cancels all event(s) identified by *ref*. Cancellation is equivalent to calling the requested function with requested arguments and argument *flag* set to EV_CANCEL.

RESTRICTIONS/CONDITIONS

g_post_event():

The argument *flag* has to be M_WAITOK or M_NOWAIT.

The list of references has to end with a NULL value.

g_waitfor_event():

The argument *flag* has to be M_WAITOK or M_NOWAIT.

The list of references has to end with a NULL value.

The **g_waitfor_event**() function cannot be called from an event, since doing so would result in a deadlock.

g_alloc_event():

The argument *flag* has to be M_WAITOK or M_NOWAIT.

The returned *struct g_event* * must be freed with **g_free**() if **g_post_event_ep**() is not called.

RETURN VALUES

The **g_post_event**() and **g_waitfor_event**() functions return 0 if successful; otherwise an error code is returned.

EXAMPLES

Example of a function called from the event queue.

```
void
example_event(void *arg, int flag)
{

        if (flag == EV_CANCEL) {
                printf("Event with argument %p canceled.\n", arg);
                return;
        }

        printf("Event with argument %p called.\n", arg);
}
```

ERRORS

Possible errors for the **g_post_event**() function:

[ENOMEM] The *flag* argument was set to M_NOWAIT and there was insufficient memory.

Possible errors for the **g_waitfor_event**() function:

[EAGAIN] The event was canceled.

[ENOMEM] The *flag* argument was set to M_NOWAIT and there was insufficient memory.

SEE ALSO

geom(4), DECLARE_GEOM_CLASS(9), g_access(9), g_attach(9), g_bio(9), g_consumer(9), g_data(9), g_geom(9), g_provider(9), g_provider_by_name(9), g_wither_geom(9)

AUTHORS

This manual page was written by Pawel Jakub Dawidek <*pjd@FreeBSD.org*>.