```c
/*
 * cgroup_event_listener.c - Simple listener of cgroup events
 *
 * Copyright (C) Kirill A. Shutemov <kirill@shutemov.name>
 */

#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <libgen.h>
#include <limits.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#include <sys/eventfd.h>

#define USAGE_STR "Usage: cgroup_event_listener <path-to-control-file> <args>\n"

int main(int argc, char **argv)
{
    int efd = -1;
    int cfd = -1;
    int event_control = -1;
    char event_control_path[PATH_MAX];
    char line[LINE_MAX];
    int ret;

    if (argc != 3) {
        fputs(USAGE_STR, stderr);
        return 1;
    }

    cfd = open(argv[1], O_RDONLY);
    if (cfd == -1) {
        fprintf(stderr, "Cannot open %s: %s\n", argv[1],
                strerror(errno));
        goto out;
    }

    ret = snprintf(event_control_path, PATH_MAX, "%s/cgroup.event_control",
            dirname(argv[1]));
    if (ret >= PATH_MAX) {
        fputs("Path to cgroup.event_control is too long\n", stderr);
        goto out;
    }

    event_control = open(event_control_path, O_WRONLY);
    if (event_control == -1) {
        fprintf(stderr, "Cannot open %s: %s\n", event_control_path,
                strerror(errno));
        goto out;
    }

    efd = eventfd(0, 0);
    if (efd == -1) {
        perror("eventfd() failed");
        goto out;
    }

    ret = snprintf(line, LINE_MAX, "%d %d %s", efd, cfd, argv[2]);
    if (ret >= LINE_MAX) {
        fputs("Arguments string is too long\n", stderr);
        goto out;
    }
    ret = write(event_control, line, strlen(line) + 1);
    if (ret == -1) {
        perror("Cannot write to cgroup.event_control");
        goto out;
    }

    while (1) {
        uint64_t result;

        ret = read(efd, &result, sizeof(result));
        if (ret == -1) {
            if (errno == EINTR)
                continue;
            perror("Cannot read from eventfd");
            break;
        }
        assert(ret == sizeof(result));

        ret = access(event_control_path, W_OK);
        if ((ret == -1) && (errno == ENOENT)) {
                puts("The cgroup seems to have removed.");
                ret = 0;
                break;
        }

        if (ret == -1) {
            perror("cgroup.event_control "
                    "is not accessable any more");
            break;
        }

        printf("%s %s: crossed\n", argv[1], argv[2]);
    }

out:
    if (efd >= 0)
        close(efd);
    if (event_control >= 0)
        close(event_control);
    if (cfd >= 0)
        close(cfd);

    return (ret != 0);
}
```