1. #include <unistd.h>
2. #include <sys/inotify.h>
3. #include <stdio.h>
4. #include <error.h>
5. #include <errno.h>
6. #include <string.h>
8. #define ERROR(text) error(1, errno, "%s", text)
10. **struct** EventMask {
11. **int**        flag;
12. **const** **char** \*name;
14. };
16. **int** freadsome(**void** \*dest, **size\_t** remain, **FILE** \*file)
17. {
18. **char** \*offset = (**char**\*)dest;
19. **while** (remain) {
20. **int** n = fread(offset, 1, remain, file);
21. **if** (n==0) {
22. **return** -1;
23. }
25. remain -= n;
26. offset += n;
27. }
28. **return** 0;
29. }
31. **int** main(**int** argc, **char** \*argv[])
32. {
33. **const** **char** \*target;
34. **if** (argc == 1) {
35. target = ".";
36. } **else** {
37. target = argv[1];
38. }
40. EventMask event\_masks[] = {
41. {IN\_ACCESS        , "IN\_ACCESS"}        ,
42. {IN\_ATTRIB        , "IN\_ATTRIB"}        ,
43. {IN\_CLOSE\_WRITE   , "IN\_CLOSE\_WRITE"}   ,
44. {IN\_CLOSE\_NOWRITE , "IN\_CLOSE\_NOWRITE"} ,
45. {IN\_CREATE        , "IN\_CREATE"}        ,
46. {IN\_DELETE        , "IN\_DELETE"}        ,
47. {IN\_DELETE\_SELF   , "IN\_DELETE\_SELF"}   ,
48. {IN\_MODIFY        , "IN\_MODIFY"}        ,
49. {IN\_MOVE\_SELF     , "IN\_MOVE\_SELF"}     ,
50. {IN\_MOVED\_FROM    , "IN\_MOVED\_FROM"}    ,
51. {IN\_MOVED\_TO      , "IN\_MOVED\_TO"}      ,
52. {IN\_OPEN          , "IN\_OPEN"}          ,
54. {IN\_DONT\_FOLLOW   , "IN\_DONT\_FOLLOW"}   ,
55. {IN\_EXCL\_UNLINK   , "IN\_EXCL\_UNLINK"}   ,
56. {IN\_MASK\_ADD      , "IN\_MASK\_ADD"}      ,
57. {IN\_ONESHOT       , "IN\_ONESHOT"}       ,
58. {IN\_ONLYDIR       , "IN\_ONLYDIR"}       ,
60. {IN\_IGNORED       , "IN\_IGNORED"}       ,
61. {IN\_ISDIR         , "IN\_ISDIR"}         ,
62. {IN\_Q\_OVERFLOW    , "IN\_Q\_OVERFLOW"}    ,
63. {IN\_UNMOUNT       , "IN\_UNMOUNT"}       ,
64. };
66. **int** monitor = inotify\_init();
67. **if** ( -1 == monitor ) {
68. ERROR("monitor");
69. }
71. **int** watcher = inotify\_add\_watch(monitor, target, IN\_ALL\_EVENTS);
72. **if** ( -1 == watcher  ) {
73. ERROR("inotify\_add\_watch");
74. }
76. **FILE** \*monitor\_file = fdopen(monitor, "r");
77. **char** last\_name[1024];
78. **char** name[1024];
80. /\* event:inotify\_event -> name:char[event.len] \*/
81. **while** (**true**) {
82. inotify\_event event;
83. **if** ( -1 == freadsome(&event, **sizeof**(event), monitor\_file) ) {
84. ERROR("freadsome");
85. }
86. **if** (event.len) {
87. freadsome(name, event.len, monitor\_file);
88. } **else** {
89. sprintf(name, "FD: %d\n", event.wd);
90. }
92. **if** (strcmp(name, last\_name) != 0) {
93. puts(name);
94. strcpy(last\_name, name);
95. }
97. /\* 显示event的mask的含义 \*/
98. **for** (**int** i=0; i<**sizeof**(event\_masks)/**sizeof**(EventMask); ++i) {
99. **if** (event.mask & event\_masks[i].flag) {
100. printf("\t%s\n", event\_masks[i].name);
101. }
102. }
103. }
104. **return** 0;
105. }