

```
1: // $Id: sigactionintr.cpp,v 1.6 2012-06-06 14:30:48-07 - - $
2:
3: //
4: // Program does nothing except echo input line by line.
5: // However, at every SIGINT (Ctrl/C), it prints a message
6: // acknowledging the Ctrl/C, but the program does not abort
7: // as is usual without a signal.
8: //
9:
10: #include <cerrno>
11: #include <csignal>
12: #include <cstdlib>
13: #include <iostream>
14: #include <stdexcept>
15:
16: #include <signal.h>
17: #include <unistd.h>
18:
19: using namespace std;
20:
21: string progame;
22:
23: string basename (const string &arg) {
24:     return arg.substr (arg.find_last_of ('/') + 1);
25: }
26:
27: void syserror (const string &object) {
28:     throw runtime_error (object + ": " + strerror (errno));
29: }
30:
31: void print_pid() {
32:     static int pid = getpid();
33:     cout << "[" << pid << "]" << progame << ": ";
34: }
35:
36: extern "C" {
37:     static volatile int action_count = 0;
38:     void action_handler (int signo) {
39:         ++action_count;
40:         print_pid();
41:         cout << "Caught signal " << signo
42:             << ", action_count " << action_count << endl;
43:     }
44: }
45:
46: void shell() {
47:     // Fake shell. Just loop printing messages. Watch for SIGINT.
48:     cout << "Entering shell." << endl;
49:     for (int count = 0; ; ++count) {
50:         sleep (1);
51:         print_pid();
52:         cout << "Shell count " << count
53:             << ", action_count " << action_count << endl;
54:         if (action_count > 9) throw runtime_error ("action_count > 9");
55:     }
56: }
57:
58: int main (int argc, char **argv) {
59:     argc = argc; // warning: unused parameter 'argc'
60:     progame = basename (argv[0]);
61:
62:     try {
63:
64:         // Initialize signal handler.
```

```
65:     struct sigaction newaction, oldaction;
66:     sigemptyset (&newaction.sa_mask);
67:     newaction.sa_handler = action_handler;
68:     newaction.sa_flags = 0;
69:     int sigaction_rc = sigaction (SIGINT, &newaction, &oldaction);
70:     if (sigaction_rc < 0) syserror ("sigaction (SIGINT)");
71:     cout << progname << ": " << "signal (SIGINT) = " << sigaction_rc
72:         << endl;
73:
74:     cout << "Just before calling shell." << endl;
75:     shell();
76:
77: }catch (exception &error) {
78:     cerr << progname << ": " << error.what() << endl;
79:     return EXIT_FAILURE;
80: }
81: return EXIT_SUCCESS;
82: }
83:
```

```
1: bash-3.2$ ./sigactionintr
2: sigactionintr: signal (SIGINT) = 0
3: Just before calling shell.
4: Entering shell.
5: [26113]sigactionintr: Shell count 0, action_count 0
6: [26113]sigactionintr: Shell count 1, action_count 0
7: [26113]sigactionintr: Shell count 2, action_count 0
8: [26113]sigactionintr: Caught signal 2, action_count 1
9: [26113]sigactionintr: Shell count 3, action_count 1
10: [26113]sigactionintr: Shell count 4, action_count 1
11: [26113]sigactionintr: Shell count 5, action_count 1
12: [26113]sigactionintr: Caught signal 2, action_count 2
13: [26113]sigactionintr: Shell count 6, action_count 2
14: [26113]sigactionintr: Shell count 7, action_count 2
15: [26113]sigactionintr: Shell count 8, action_count 2
16: [26113]sigactionintr: Caught signal 2, action_count 3
17: [26113]sigactionintr: Shell count 9, action_count 3
18: [26113]sigactionintr: Shell count 10, action_count 3
19: [26113]sigactionintr: Caught signal 2, action_count 4
20: [26113]sigactionintr: Shell count 11, action_count 4
21: [26113]sigactionintr: Caught signal 2, action_count 5
22: [26113]sigactionintr: Shell count 12, action_count 5
23: [26113]sigactionintr: Shell count 13, action_count 5
24: [26113]sigactionintr: Caught signal 2, action_count 6
25: [26113]sigactionintr: Shell count 14, action_count 6
26: [26113]sigactionintr: Caught signal 2, action_count 7
27: [26113]sigactionintr: Shell count 15, action_count 7
28: [26113]sigactionintr: Shell count 16, action_count 7
29: [26113]sigactionintr: Shell count 17, action_count 7
30: [26113]sigactionintr: Caught signal 2, action_count 8
31: [26113]sigactionintr: Shell count 18, action_count 8
32: [26113]sigactionintr: Caught signal 2, action_count 9
33: [26113]sigactionintr: Shell count 19, action_count 9
34: [26113]sigactionintr: Caught signal 2, action_count 10
35: [26113]sigactionintr: Shell count 20, action_count 10
36: sigactionintr: action_count > 9
37: bash-3.2$ exit
```