

```

Hello from child 15238
Hello from child 15239
Handler reaped child
Handler reaped child
Handler reaped child
CR
Parent processing input

```

code/ecf/signal2.c

```

1 void handler2(int sig)
2 {
3     int olderrno = errno;
4
5     while (waitpid(-1, NULL, 0) > 0) {
6         Sio_puts("Handler reaped child\n");
7     }
8     if (errno != ECHILD)
9         Sio_error("waitpid error");
10    Sleep(1);
11    errno = olderrno;
12 }

```

code/ecf/signal2.c

图 8-37 signal2: 图 8-36 的一个改进版本, 它能够正确解决信号不会排队等待的情况



练习题 8.8 下面这个程序的输出是什么?

code/ecf/signalprob0.c

```

1 volatile long counter = 2;
2
3 void handler1(int sig)
4 {
5     sigset_t mask, prev_mask;
6
7     Sigfillset(&mask);
8     Sigprocmask(SIG_BLOCK, &mask, &prev_mask); /* Block sigs */
9     Sio_putl(--counter);
10    Sigprocmask(SIG_SETMASK, &prev_mask, NULL); /* Restore sigs */
11
12    _exit(0);
13 }
14
15 int main()
16 {
17     pid_t pid;
18     sigset_t mask, prev_mask;
19
20     printf("%ld", counter);
21     fflush(stdout);
22
23     signal(SIGUSR1, handler1);
24     if ((pid = Fork()) == 0) {

```