

code/ecf/procmask2.c

```

1 void handler(int sig)
2 {
3     int olderrno = errno;
4     sigset_t mask_all, prev_all;
5     pid_t pid;
6
7     Sigfillset(&mask_all);
8     while ((pid = waitpid(-1, NULL, 0)) > 0) { /* Reap a zombie child */
9         Sigprocmask(SIG_BLOCK, &mask_all, &prev_all);
10        deletejob(pid); /* Delete the child from the job list */
11        Sigprocmask(SIG_SETMASK, &prev_all, NULL);
12    }
13    if (errno != ECHILD)
14        Sio_error("waitpid error");
15    errno = olderrno;
16 }
17
18 int main(int argc, char **argv)
19 {
20     int pid;
21     sigset_t mask_all, mask_one, prev_one;
22
23     Sigfillset(&mask_all);
24     Sigemptyset(&mask_one);
25     Sigaddset(&mask_one, SIGCHLD);
26     Signal(SIGCHLD, handler);
27     initjobs(); /* Initialize the job list */
28
29     while (1) {
30         Sigprocmask(SIG_BLOCK, &mask_one, &prev_one); /* Block SIGCHLD */
31         if ((pid = Fork()) == 0) { /* Child process */
32             Sigprocmask(SIG_SETMASK, &prev_one, NULL); /* Unblock SIGCHLD */
33             Execve("/bin/date", argv, NULL);
34         }
35         Sigprocmask(SIG_BLOCK, &mask_all, NULL); /* Parent process */
36         addjob(pid); /* Add the child to the job list */
37         Sigprocmask(SIG_SETMASK, &prev_one, NULL); /* Unblock SIGCHLD */
38     }
39     exit(0);
40 }

```

code/ecf/procmask2.c

图 8-40 用 sigprocmask 来同步进程。在这个例子中，父进程保证在相应的 deletejob 之前执行 addjob

8.5.7 显式地等待信号

有时候主程序需要显式地等待某个信号处理程序运行。例如，当 Linux shell 创建一个后台作业时，在接收下一条用户命令之前，它必须等待作业终止，被 SIGCHLD 处理程序回收。

图 8-41 给出了一个基本的思路。父进程设置 SIGINT 和 SIGCHLD 的处理程序，然后