

```

/* sig_sample3.c */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>

int ntimes = 0;

void main()
{
    pid_t pid, ppid;
    void p_action(int), c_action(int);
    static struct sigaction pact, cact;

    pact.sa_handler = p_action;
    sigaction(SIGUSR1, &pact, NULL);

    switch (pid = fork()) {
        case -1:
            perror("synchro");
            exit(1);
        case 0:
            cact.sa_handler = c_action;
            sigaction(SIGUSR1, &cact, NULL);

            ppid = getppid();

            for(;;) {
                sleep(1);
                kill(ppid, SIGUSR1);
                pause();
            }

        default:
            for(;;) {
                pause();
                sleep(1);
                kill(pid, SIGUSR1);
            }
    }
}

```

```
void p_action(int sig)
{
    fprintf(stderr, "Parent caught signal #%d\n", ++ntimes);
}
```

```
void c_action(int sig)
{
    fprintf(stderr, "Child caught signal #%d\n", ++ntimes);
}
```