

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>

void catch_ctlc (int sig_num)
{ printf("You cannot kill this process");
  printf ("\nEnter an integer (enter 0 to terminate): ");
  fflush(stdout);
}

void catch_user (int sig_num)
{
  printf ("A user signal is caught %d\n", sig_num);
  printf ("\nEnter an integer (enter 0 to terminate): ");
  fflush(stdout);
}

int main()
{
  int pid, input;

  // bind an interrupt bit to an interrupt handler function
  signal(SIGINT, catch_ctlc);

  pid = fork();
  if (pid > 0)
  { printf ("Created child process %d\n", pid);
    sleep (5); // let child process get a chance to run
    kill (pid, SIGRTMIN);
    usleep (5000000); // = sleep (5), sleeps for 5 second
    kill (pid, SIGRTMAX);
    wait ();
  }
  else
  { printf ("This is the child process %d\n", getpid());

    signal(SIGRTMIN, catch_user);
    signal(SIGRTMAX, catch_user);

    do
    { printf ("Enter an integer (enter 0 to terminate): ");
      scanf ("%d", &input);
      printf ("Square of %d = %d\n", input, input*input);
    } while (input != 0);
  }
  return (1);
}
```