```
/*********************************************************/
/*** Sample program demonstrating the sending of signals ***/
/*********************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <signal.h>

/* The signal handler for the child process */
void childSigHandler ( int sig )
{
   if (sig == SIGUSR1) {
      printf("+++ Child : Received signal SIGUSR1 from parent...\n");
      sleep(1);
   } else if (sig == SIGUSR2) {
      printf("+++ Child : Received signal SIGUSR2 from parent...\n");
      sleep(5);
   }
   exit(0);
}

int main ()
{
   int pid;

   pid = fork();                                /* Spawn the child process */
   if (pid) {
                                                        /* Parent process */
      int t;
      srand((unsigned int)time(NULL));
      t = 2 + rand() % 4;
      printf("+++ Parent: Going to sleep for %d seconds\n", t);
      sleep(t);    /* Sleep for some time before sending a signal to child */
      t = 1 + rand() % 2;
      printf("+++ Parent: Going to send signal SIGUSR%d to child\n", t);
      kill(pid, (t == 1) ? SIGUSR1 : SIGUSR2);    /* Send signal to child */
      wait(NULL);                                  /* Wait for child to exit */
      printf("+++ Parent: Child exited\n");

   } else {
                                                          /* Child process */
      signal(SIGUSR1, childSigHandler);        /* Register SIGUSR1 handler */
      signal(SIGUSR2, childSigHandler);        /* Register SIGUSR2 handler */
      while (1) sleep(1);  /* Sleep until a signal is received from parent */

   }

   exit(0);
}
```