

Specification

To write a C program that produces an executable that can send and receive messages to/from another process. We will use two processes that are running the same executable code. You are going to use conditional compilation flags to allow you to create two versions of this program. One (the default) uses two signals (SIGUSR1 and SIGUSR2) and the other (conditionally compiled as `-DSINGLE`) uses only one signal (SIGUSR1)

When invoked the program reports its own process ID and then expects to read from standard input the process ID of the process with which it will communicate. Subsequently, each line of input is treated as a separate message. For example:

```
% covertsigs
Own PID:
15310 14252
This is my first message
This is my second message
.
%
```

The name of the executable is `covertsigs`. Here 14252 is the first line typed by the user which is the PID of the process to communicate with. Then the messages are entered one-at-a-time. A line with a single period terminates the program (note that it does *not* terminate the execution of the other process (14252) with which we were communicating). The receiving process reports the received messages prefixing them with a `!` exclamation mark symbol if it is confident it received it correctly and with a `?` question mark symbol if it has detected an error (yes it is possible to have errors -- read on). For example

```
% covertsigs
Own PID:
14252 15310
! This is my first message
! This is my second message
```

but it could have been (due to errors in the first message):

```
% covertsigs
Own PID:
14252 15310
? ThiWE%@
! This is my second message
```

or any other combination that might results from errors and such. The reason we could have a message in error is because,

as you will discover, the signals are not always reliable and also because it depends on how you decide to code the messages using signal(s).

In the interest of readability, the example presented above had one process sending and one receiving. You are expected though to have, at each process, a mix of messages received and messages sent. Consider for example a session transcript like this:

```
% covertsigs
Own PID:
13333 13338
! Hi!
Hi to you too
! how's the
weather Fine
? same h3254dsg
453 Did you say
"same here" ! yes
```

Essentially it is a two person chat channel (but generally of low speed and subject to errors).