Signal8 Demonstration

Files

The files for this demonstration can be found in the rlogin cluster in the directory

`/web/courses/cs3214/spring2014/butta/examples/signal-demo/signal8`

The files are `esh-sys-utils.c esh-sys-utils.h Makefile receiver.c`

The `“make” command` by default will create an executable named `receiver`. The `receiver` executes an infinite loop, receiving and reacting to various signals.

Purpose

The purposes of this demonstration are

- to see how to control when signals may be received by exploring the effect of blocking and unblocking signals
- to see what happens when a signal is sent to a process that is currently blocking that signal
- to see how to determine which signals are currently blocked and which are waiting to be delivered

Note

For this demonstration you will need two windows on the same machine on the rlogin cluster. To do this:

- ssh to the rlogin cluster (assume that you are connected to the machine “`birch`”)
- ssh to the rlogin cluster again (assume this time you are connected to the machine “`pawpaw`”.
- in the window for the “`pawpaw`” machine use the command “`ssh birch`”. This will connect to the “`birch”` machine

For purposes of discussion below one window will be referred to as the `run` window and the other window will be referred to as the `command` window. Almost all of the work will be done in the `command` window.

Steps

1. Use the `Makefile` to create the executable program `receiver` using the command `“make”`.
2. At the shell prompt execute the `receiver` program in the `run` window. Output in this window will be produced as a result of the following steps.
3. In the `command` window use the command “`ps –s`” to display status information about active processes. Note the process id of the `receiver` process. Refer to this

process id as `<rpid>`. In the steps that follow it is only the `receiver` process that is of interest. To show only the process status information for the receiver you can use the command “`ps –s –p <rpid>`”.

4. For the receiver process note the columns labeled `PENDING`, `BLOCKED`, and `IGNORED`. Answer question 1.
5. Use the command “`man 7 signal`” and note the list of signals and their corresponding signal numbers. For example, `SIGQUIT` is signal number 3. If multiple numbers are shown use the middle number. For example, `SIGSTOP` is signal number 19. Answer question 2.
6. In the command window use the command “`kill –SIGUSR1 <rpid>`” to send the `SIGUSR1` signal to the `receiver` process.
7. Use the command “`ps –s –p <rpid>`” to observe the status of the `receiver` process. Note the column labeled `PENDING`. Answer question 3.
8. In the command window use the command “`kill –SIGCONT <rpid>`” to send the `SIGCONT` signal to the `receiver` process. Observe the output in the `run` window.
9. Use the command “`ps –s –p <rpid>`” to observe the status of the `receiver` process. Answer question 4.
10. In the command window use the command “`kill –SIGUSR2 <rpid>`” to send the `SIGUSR2` signal to the `receiver` process.
11. Use the command “`ps –s –p <rpid>`” to observe the status of the `receiver` process. Note the column labeled `PENDING`.
12. Repeat steps 10 and 11 two or three times and answer question 5.
13. In the command window use the command “`kill –SIGCONT <rpid>`” to send the `SIGCONT` signal to the `receiver` process. Observe the output in the `run` window.
14. Use the command “`ps –s –p <rpid>`” to observe the status of the `receiver` process. Answer question 6 and 7.
15. Send additional `SIGUSR1` and `SIGUSR2` signals to the `receiver` process and observe the output in the `run` window.
16. Terminate the receiver process by using a `control-c` in the `run` window.

Questions

Based on your observations, answer these questions.

1. In step 4 what is the value of the `BLOCKED` status for the `receiver`. The `BLOCKED` status is shown as a bit string (each displayed digit is a byte) with the rightmost bit being considered 1. Which two bits are set in `BLOCKED`? Each bit in `BLOCKED` that is set means that the corresponding signal is currently blocked.
2. Which two signals are initially blocked by the receiver process?
3. The `PENDING` status information is coded the same way as the `BLOCKED` status information. Which signal(s) is pending? Explain why this signal is pending.
4. What has changed in the `PENDING` and `BLOCKED` status information? Can you hypothesize what effect the `SIGCONT` had on the receiver process that explains this

change and the output observed in step 8? You can look at the `receiver.c` code to confirm your hypothesis.

5. How did the `PENDING` status information change with the sending of each `SIGUSR2` signal? Hypothesize what happens as each signal is sent.
6. What has changed in the `PENDING` and `BLOCKED` status information? Can you hypothesize what effect the `SIGCONT` had on the receiver process that explains this change and the output observed in step 8? You can look at the `receiver.c` code to confirm your hypothesis.
7. Based on the output in the run window how many `SIGUSR2` signals were delivered to the `receiver` process? Can you explain this behavior?