Unix System Programming

B-PSU-200

Bootstrap

navy

2.0

EPITECH.
L'ECOLE DE L'INNOVATION ET DE
L'EXPERTISE INFORMATIQUE

Bootstrap

binary name:	process_info, kill_it, who_sig_me, signal_me
repository name:	PSU_navy_bootstrap_$ACADEMICYEAR
repository rights:	ramassage-tek
language:	C
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

This Bootstrap is a introduction to the way signals function in a Unix system.

The Unix kernel "informs" processes through signals in order to transmit potential problems (SIGSEGV for a segmentation error for instance).
Each signal is assigned a default behavior.
Read the related man pages in order to understand each signal's procces' behavior.

Numerous system calls exist to handle signals.
We will purposely not cover a large part of them, so that you can have fun researching them yourself.

EPITECH.
L'ECOLE DE L'INNOVATION ET DE L'EXPERTISE INFORMATIQUE

Exercise 1

Write a program named **process_info**
that displays the following process information: process ID, parent process ID
and process group ID.

For instance:

```
./process_info
PID: 18975
PPID: 18954
PGID: 18973
```

man getpid

Exercise 2

Write a program named **kill_it** that sends the SIGQUIT signal to the process whose PID is passed as parameter.

For instance:

```
./kill_it 4754
```

kill man page. In order to test it, use the ps program to obtain the PID of a program (firefox, xeyes etc.)

Exercise 3

Write a program named **who_sig_me**
that, for each received signal, displays its name and the PID of the emitter process.
The program takes the list of signals to be rerouted as parameter. If a signal's rerouting fails, an error message is displayed.

For instance:

Terminal

```
~/B-PSU-200> ./who_sig_me 10 12 > stdout.log &
[1] 1585
~/B-PSU-200> kill -USR1 1585
~/B-PSU-200> kill -USR2 1585
~/B-PSU-200> kill -USR1 1585
~/B-PSU-200> kill -QUIT 1585
~/B-PSU-200> cat -e stdout.log
Signal User defined signal 1 received from 1586
Signal User defined signal 2 received from 1587
Signal User defined signal 1 received from 1588
```

Terminal

```
~/B-PSU-200> ./who_sig_me 12 9 > stdout.log &
[2] 1590
~/B-PSU-200> kill -USR2 1590
~/B-PSU-200> kill -KILL 1590
~/B-PSU-200> cat -e stdout.log
Unable to handle Killed signal
Signal User defined sigal 2 received from 1591
```

sigaction and strsignal man pages. Scan the system's .h in order to understand the number that is associated with each signal.

Exercise 4

Write a program named **signal_me** that counts the number of times it receives the SIGUSR1 and SIGUSR2 signals.
This program must display a summary when it receives the SIGQUIT signal, before exiting.

Terminal

```
∼/B-PSU-200> ./signal_me > stdout.log &
[1] 12985
∼/B-PSU-200> kill -USR1 12985
∼/B-PSU-200> kill -USR1 12985
∼/B-PSU-200> kill -USR2 12985
∼/B-PSU-200> kill -USR2 12985
∼/B-PSU-200> kill -USR1 12985
∼/B-PSU-200> kill -QUIT 12985
∼/B-PSU-200> cat -e stdout.log
SIGUSR1:  3
SIGUSR2:  2
```