

Exercise Sheet 4

In this exercise sheet you will work with processes and signals.

Help: Remember that a process can be created with `fork()`, `vfork()`, and in Linux also with `clone()`. Choose the most convenient and adequate one for every situation. A parent process can wait for its children using `wait()` or `waitpid()`.

Task 1

Write a program in which the parent process creates exactly 9 child processes and waits for them to finish. Every child process finishes immediately after being created.

Task 2

Write a program which creates exactly 16 child processes. After that, the parent process prints on the standard output the message: **16 child processes has been created**. Every child process simply prints its `pid` on the standard output. Analyse the obtained output. Is the order of the messages (parent and child ones) consistent? Can the order of these messages be predicted? What does it depend on?

Task 3

Write a program in which the parent process creates exactly 1 child. After creating the child, the behaviour of the **parent** process is as follows: it sends the signal `SIGUSR1` to the child every 5 seconds. To implement this behaviour the parent process must use the following system calls: `alarm()` and `pause()`. After sending the signal `SIGUSR1` three times, the fourth time the signal `SIGUSR2` is sent to the child. After this, the parent waits for the child to finish.

The behaviour of the **child** is as follows: it waits until it is interrupted by any signal. If the signal received is `SIGUSR1` it prints a message to the standard output. If the signal received is `SIGUSR2` then it finishes. In addition, during the first 15 seconds of its execution, the signal `SIGUSR2` should be blocked.

Students are expected to check the behaviour of `alarm()` and `pause()` in the man pages.

Task 4

Write a program in which the parent creates exactly 1 child process. The child process should print its `pid` to the standard output and then finish. The parent process waits until it is sure that the child has terminated. For this students are required to check in the man page of `fork` to see whether a parent is notified of child termination via any signal. The parent terminates after it has waited for the child process.