

title: Quiz 3, Operating Systems, 2023-2024

date: 2 Dec-17 Dec 2023

1. In a small C code, create 4 different processes with the right usage of `fork()`, where each process will receive a character the user types in. After all processes have acquired their character, they print the summary in ascending order and exit. Their order originates from their global process id (PID).

```
""  
I am child process 1 and my character was c.  
I am child process 2 and my character was h.  
I am child process 3 and my character was 4.  
I am child process 4 and my character was d.
```

Your code should be split in at least two different `.c` files.

In order to be able to share/communicate the saved keys from each child process you can use either pipes/shared memory or write to file (e.g. each process writes to a `$pid\log`, and parent process reads). You can refer to the example codes of Lab 4, for examples regarding pipes. If everything fails, as a last resort, each child can print it's own key. Try to at least experiment with `sleep()` to make the printing sync properly according to the order mentioned above.

2. Also provide a Makefile that

a) compiles your code into a single executable with `make all`,

b) cleans reproducible and intermediate files with `make clean` and

c) installs the binary into `/home/deadpool/quiz-3/bins/`, for user `deadpool` with `make install`. You can use either Make or CMake.

All files should live under `/home/deadpool/quiz-3` on the CSAL VM, for user `deadpool`. E.g. `/home/deadpool/quiz-3/fork_1.c`, `/home/deadpool/quiz-3/fork_2.c`, `/home/deadpool/quiz-3/Makefile`

For the e-learning part, create a tarball (tar.gz) that contains the aforementioned contents of folder quiz-3 (no binaries though) and upload it.