# Fork & Multiprocessing Synchronization

Hüseyin Çakır

github.com/caesarsalad

ckrhuseyin48@gmail.com

MSKU - CENG 2034 - 2020

# Getting Started

https://github.com/caesarsalad/ceng2034_os_labs

OS for DevOps or Backend Developers

- Linux (Ubuntu 18.04)
- Python3
- Git
- Github Account

Operating System Concepts, 9th Edition by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne

http://wiki.netseclab.mu.edu.tr/index.php?title=CENG_2034:_Operating_Systems

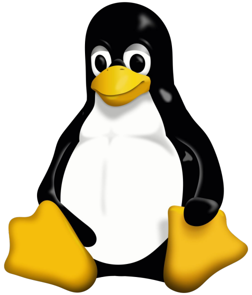
gitlab.com/cakirhuseyin

github.com/caesarsalad

ckrhuseyin48@gmail.com

http://pages.cs.wisc.edu/~remzi/OSTEP/

# Fork



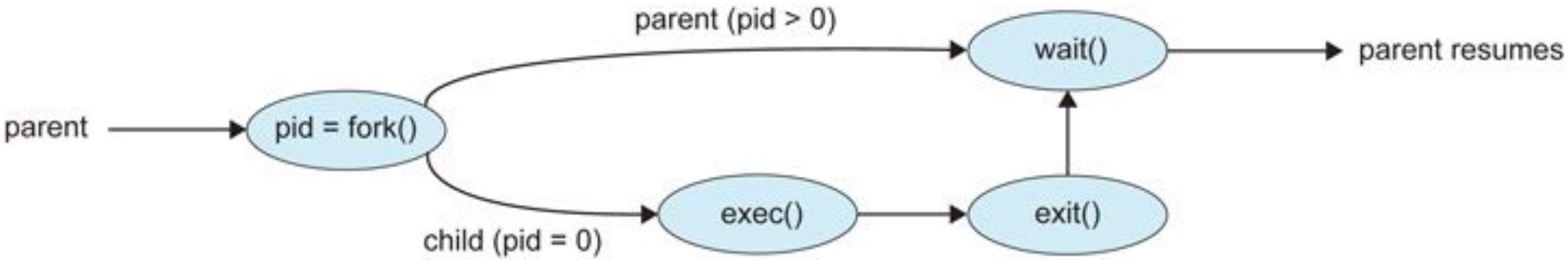
- os.fork() in Python provides the implementation of the system call fork().
- os.fork() when called from a program it creates a new process.
- Thus executing os.fork() creates two processes: A parent process and a child process.
- The newly created child process is the **exact replica of the parent process**.
- If successful, the return value of the os.fork() call in the parent program will be the process id of the child process. In case of any failure the return value will be -1. For the successful case in the child process the **return value will be zero**.

# Wait

The wait() method of os module in Python enables a parent process to synchronize with the child process. i.e, To wait till the child process exits and then proceed.

Return Value:

A Python tuple consisting of the existing child process id and a sixteen-bit integer to denote exit status.



# Resources

- “Premature optimization is the root of all evil (or at least most of it) in programming.” D. Knuth
- https://pythontic.com/modules/os/fork
- https://code-maven.com/parallel-processing-using-fork-in-python
- https://docs.python.org/3/library/os.html#process-management
- https://en.wikipedia.org/wiki/Fork_bomb
-