

## CS483 – Lesson 05 fork() and execvp() in C

### Before you Begin

1. Create a new CLion project called Lsn05. To simplify your task, you can implement each of the exercises in main.c one after another.

### Example

1. **fork():** In your main() function, use for to create a copy of this process and print message detailing whether or not execution is in the parent process or the child (see example code posted to course web site).
2. **execvp():** Modify the child portion to overwrite the currently executing process with a new process using execvp(). The process (with arguments) you want to load and execute is -> `ls -lF -a`

### Exercises

1. **fork() and execvp():** Using the example's presented in class create a new process using fork() (a child) and have the child execute the following command, "man -k pid". The parent process should wait() until the child has completed execution of the "man" command and then print a message stating the parent has ended prior to termination.
2. **fgets() and prep for execvp():** Use fgets to get a command to execute from the user. The command may have zero or more arguments. Create the char \* array required by execvp() from the user's input. Note: strtok() from last lesson is required.

### Challenge Exercise

1. **Ex02 continued - fork() and execvp():** Extend exercise 2 by creating a new process using fork() and then have the newly created child process execute the given command. The parent process should wait() until the child has completed execution of the command.