# COMP3500: Project 4 Part 4 - Managing Process State

**Exercise 1 (Plickers):** Where can you find the following prototypes?

int execv(const char \*prog, char \*const \*args);

pid\_t fork(void);

A. ~/cs161/src/include/syscall.h

B. ~/cs161/src/include/unistd.h

C. ~/cs161/src/kern/include/syscall.h

D. ~/cs161/src/kern/include/unistd.h

**Exercise 2 (Plickers):** Where can you find the sys\_fork() prototype?

A. ~/cs161/src/include/syscall.h

B. ~/cs161/src/include/unistd.h

C. ~/cs161/src/kern/include/syscall.h

D. ~/cs161/src/kern/include/unistd.h

**Exercise 3 (Plickers, Wheeldecide):** Design Question 1: What is your decision? Why?

1. Single-threaded B. Multiple-threaded processes

**Exercise 4 (Wheeldecide):** Data Structure Question 1:   
 What fields should be added in the thread struct? (**Hints:** two items)

**Exercise 5 (Wheeldecide):** Design Question 2: How to allocate (i.e., assign) PIDs?

**Exercise 6 (Wheeldecide):** Data Structure Question 3: Why we need a lock for PID management?