

Homework 3:

Part A:

Use *man* to find out more about: **fork()**, **execve()** commands in Unix.

1. Use *man* to find more about built-in commands: **exec()**, **wait()**, **kill**
2. Use Internet sources and give an overview of the command that is used in Windows for creating a process.
3. In a Unix environment, execute **parent.c**, **child.c** and **orphan.c** as follows:

Child and parent:

- compile the child and parent:

gcc parent.c -o parent

gcc child.c -o child

- run the parent in the current directory (the parent after the fork will call the child)

Don't worry about warning messages.

./parent

Orphan:

- compile and run the orphan:

gcc orphan.c -o orphan

./orphan

Observe and understand the programs' output.

Extensively comment the output of the programs by relating the theory discussed in class, the meaning of the covered commands and the program listings.

PART B: Synchronization

A. Consider the **2nd attempt** (from the lecture notes). ***Is the "No Starvation" condition satisfied?***

Hint: in your proof you might want to check if there is a particular execution sequence by which a process might be able to use the CS over and over, while the other process is starving in the while loop.

B. Prove that the Peterson Solution is correct by showing that all 3 conditions for a correct solution to the Critical Section Problem are respected.

Hint: you can use the textbook comments but your proof should be clearer and more detailed.