

# Hardware, Microcontroladores y Sistemas Embebidos

---

## Práctica de Laboratorio 3

Contenido involucrado con Sincronización de Tareas y Manejo de I/O

### Ejercicio 1 - Procesos

- fork() [https://linuxhint.com/fork\\_linux\\_system\\_call\\_c/](https://linuxhint.com/fork_linux_system_call_c/)
- wait() <https://linuxhint.com/wait-system-call-in-c/>
- exec() <https://linuxhint.com/linux-exec-system-call/>

<https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdf>

### Ejercicio 2 - Mutex

- Ejemplo del uso de una sección crítica y condiciones de carrera
- Uso de un candado en una sección crítica
- Uso de candados (POSIX)

<https://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf>

### Ejercicio 3 - Semáforos

- Uso de semáforos (POSIX)
- Sem\_wait()
- Sem\_post()

<https://linuxhint.com/posix-semaphores-with-c-programming/>  
<https://pages.cs.wisc.edu/~remzi/OSTEP/threads-sema.pdf>

### Ejercicio 4 - USB

- Ver por medio de la línea de comandos contenidos de un USB o un disco duro
- Comando mount

<https://pages.cs.wisc.edu/~remzi/OSTEP/file-devices.pdf>

## Ejercicio 5 - Cambiar Salida Estándar

- printf()
  - Int
  - Double
  - String
- Redirigir a un archivo

<https://www.programiz.com/c-programming/c-input-output>

## Ejercicio 6 - Cambiar Entrada Estándar

- scanf()
- sscanf()
- fscanf()
- fgets()
- Leer parámetros consola (argc, argv)

<https://www.programiz.com/c-programming/c-input-output>

## Ejercicio 7 - Manejo Archivos

- fopen()
- fwrite()
- fclose()

<https://www.programiz.com/c-programming/c-file-input-output>

<http://www.simplyembedded.org/archives/filesystems-with-the-raspberry-pi/>

## Links Utilidad

1. [How to Debug C Program using gdb in 6 Simple Steps](#)
2. [Strings in C \(With Examples\)](#)
3. [7 Ways to Determine the File System Type in Linux \(Ext2, Ext3 or Ext4\)](#)
4. [Learn to Code with C - free Raspberry Pi book — The MagPi magazine](#)
5. <https://www.programiz.com/c-programming/c-input-output>
6. <https://www.programiz.com/c-programming/c-file-input-output>
7. [remzi-arpacidusseau/ostep-projects: Projects for an undergraduate OS course](#)
8. [remzi-arpacidusseau/ostep-code: Code from various chapters in OSTEP](#)  
[\(http://www.ostep.org\)](http://www.ostep.org)
9. [Interlude: Process API](#)
10. [The Abstraction: The Process](#)
11. [Scheduling: Introduction](#)
12. [Laboratory: Tutorial](#)
13. [Segmentation fault](#)
14. [Debugging Under Unix: gdb Tutorial](#)
15. [GDB Tutorial - A Walkthrough with Examples](#)
16. [Intro to Threads and Processes in Python | by Brendan Fortuner](#)
17. [threading — Thread-based parallelism — Python 3.10.0 documentation](#)
18. [multiprocessing — Process-based parallelism — Python 3.10.0 documentation](#)
19. [What are the differences between the threading and multiprocessing modules?](#)
20. [Threads in Python - Datacamp](#)
21. [An Intro to Threading in Python – Real Python](#)
22. [System call](#)