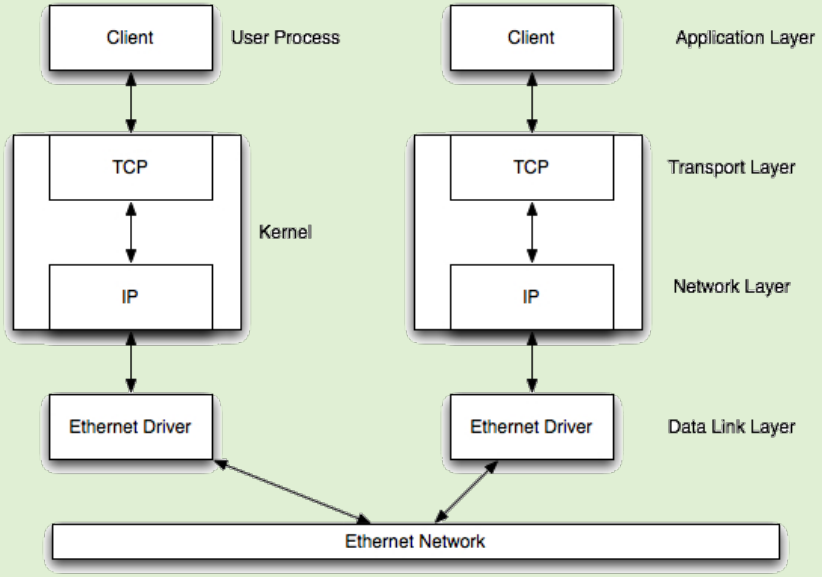
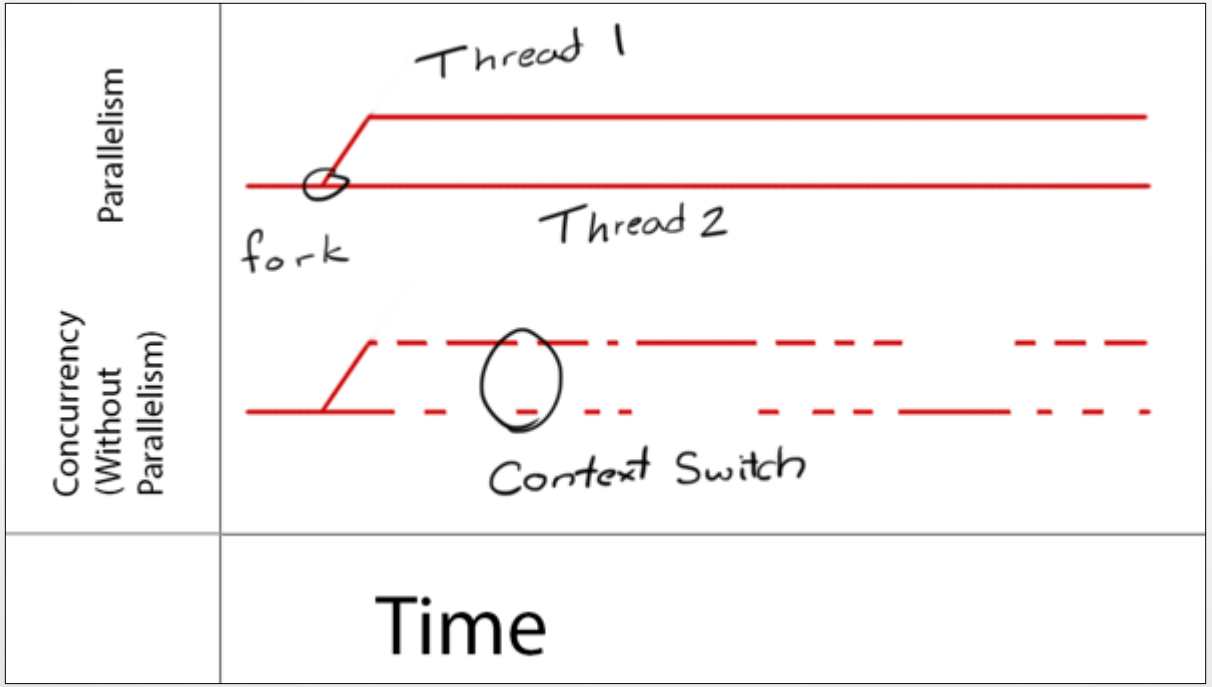
Taller de Programación en Redes
Stack TCP/IP - Sockets



Lic. en Sistemas de Información - Universidad Nacional de Luján
Dr. Gabriel Tolosa – tolosoft@unlu.edu.ar
Lic. Marcelo Fernández – fernandezm@unlu.edu.ar
Clase 3 - Febrero 2018

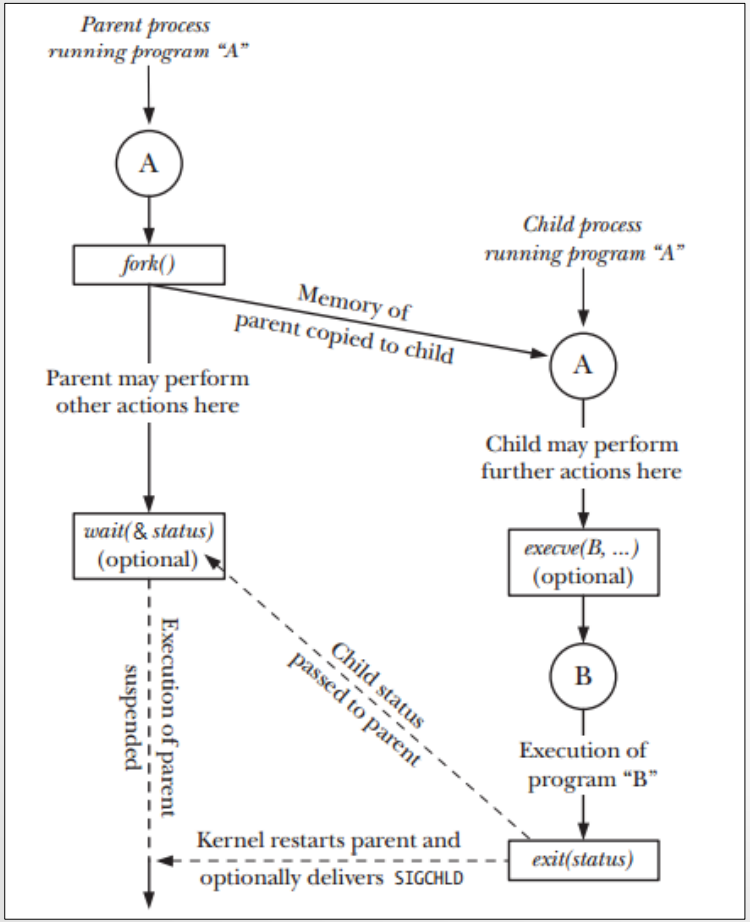
Multiprogramación, Concurrencia y Paralelismo

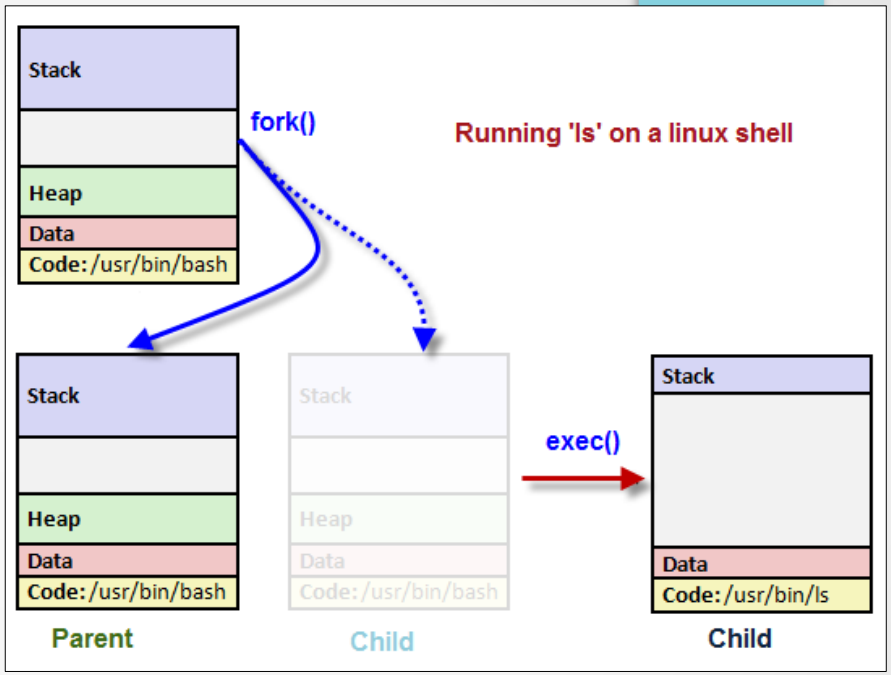
- Concurrencia (no Paralelismo [1])
- Procesos / Hilos



[1] https://blog.golang.org/concurrency-is-not-parallelism

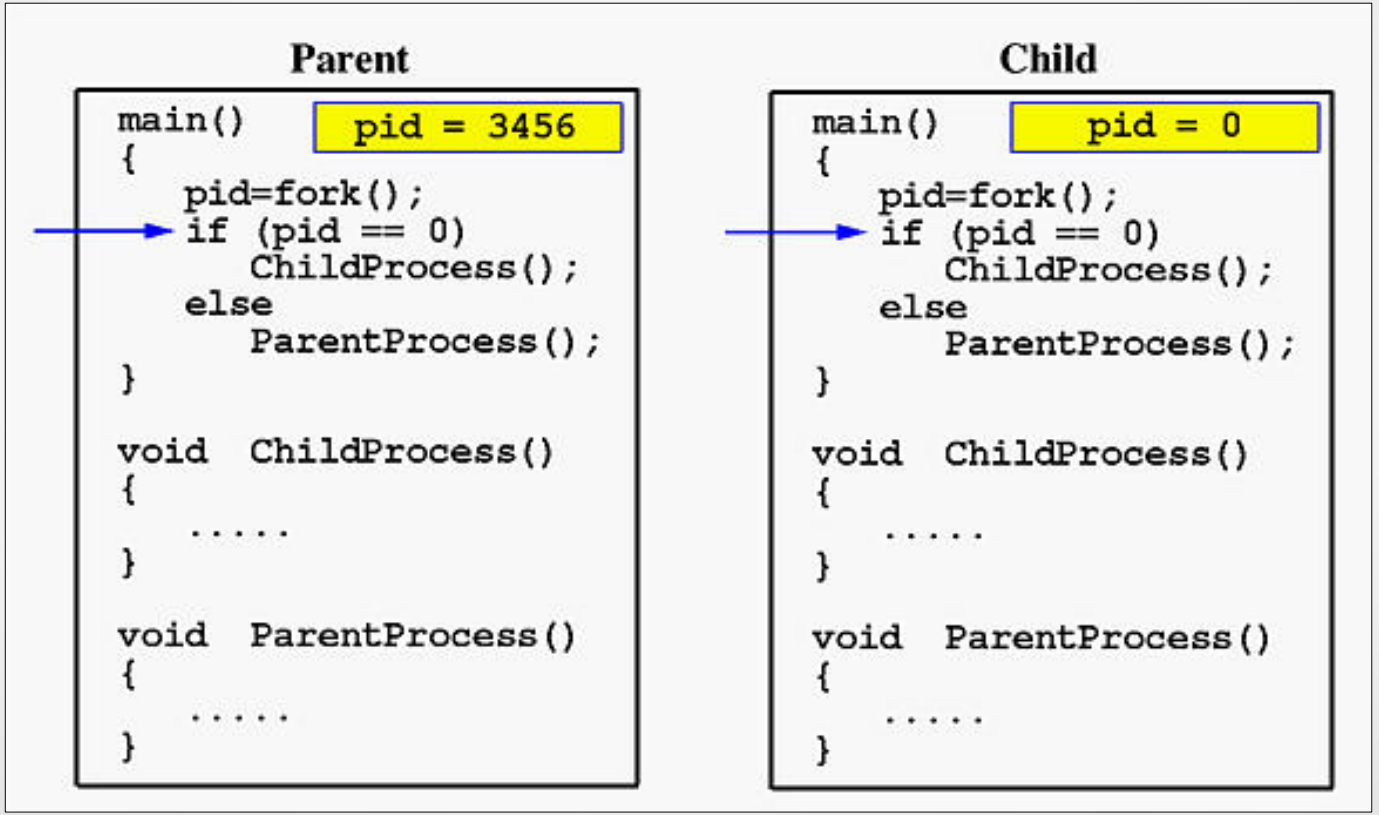
Forkeando un Proceso



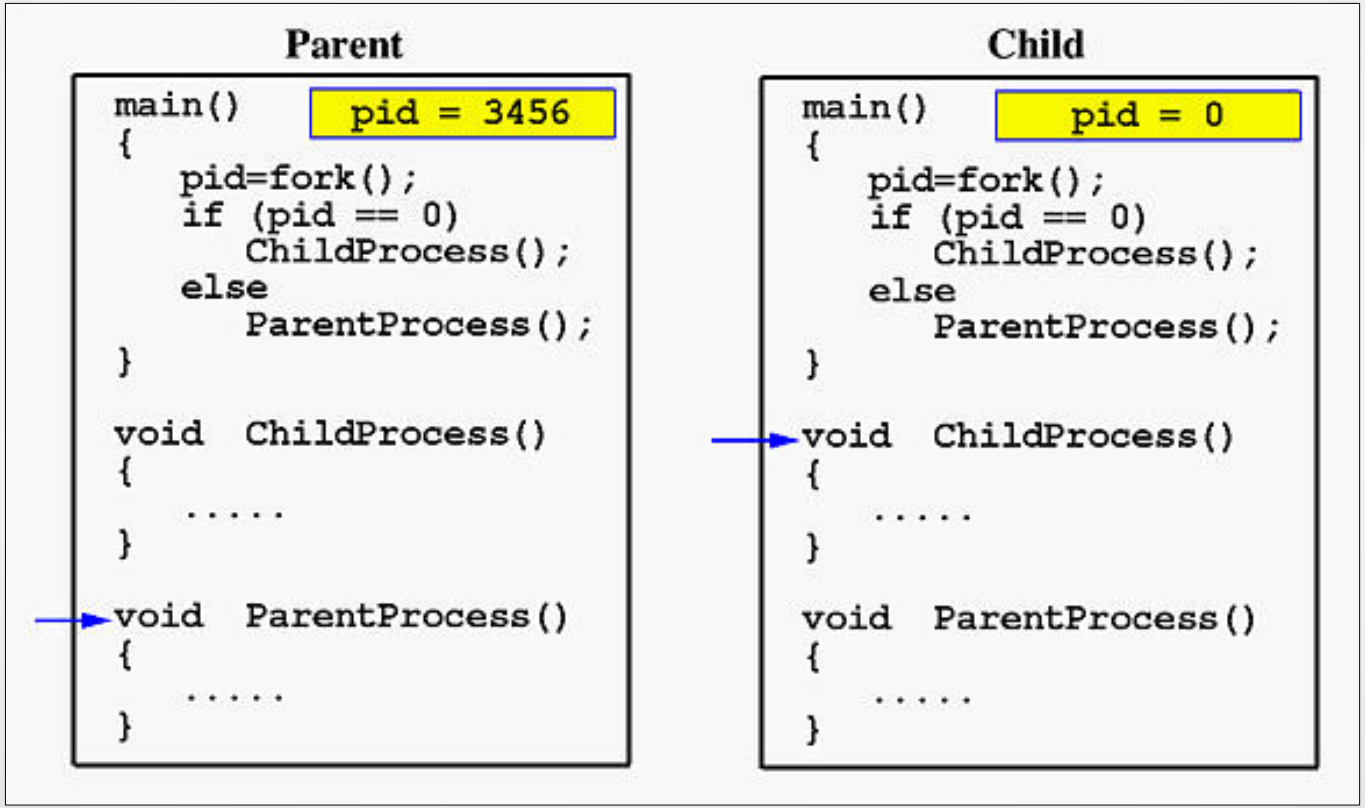


Tomado de:
- http://www.bogotobogo.com/Linux/linux_process_and_signals.php
- "The Linux Programming Interface: A Linux and UNIX System Programming Handbook", No Starch Press, 1Ed, ISBN-10: 1593272200

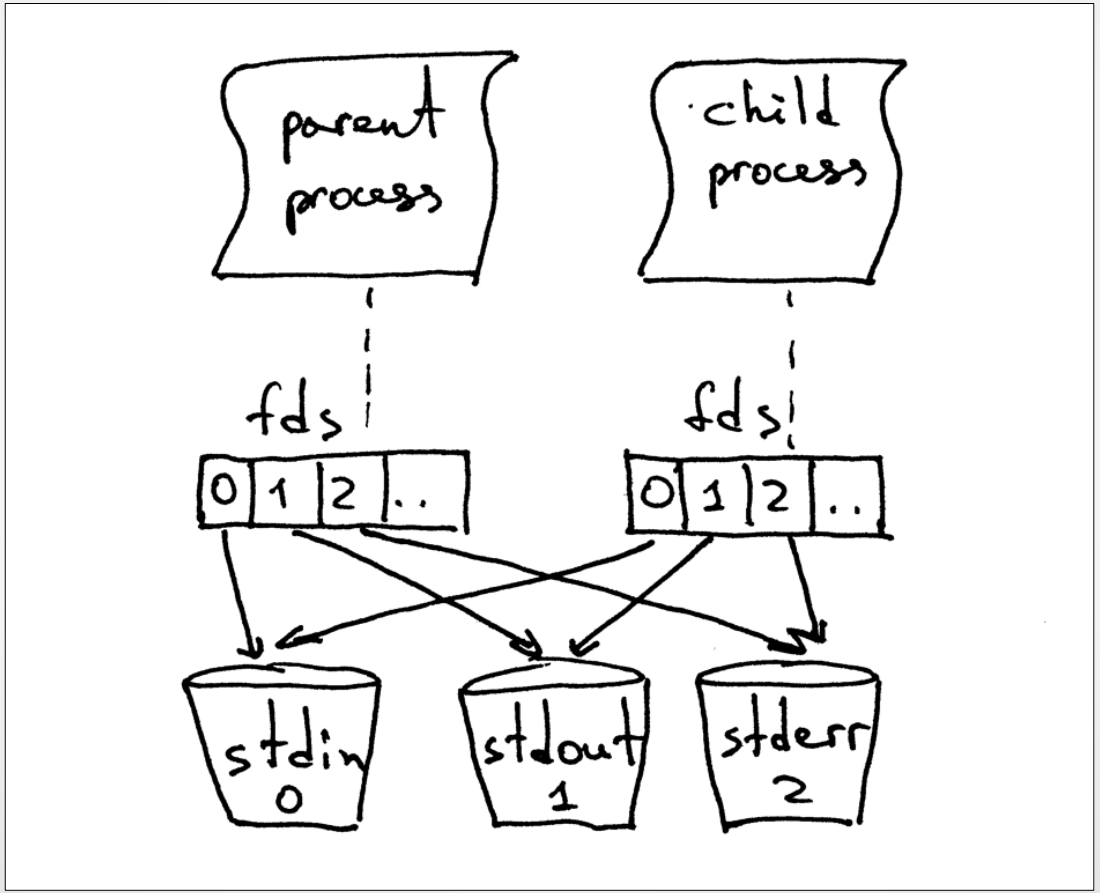
Forkeando un Proceso (2)



Forkeando un Proceso (2)



Forkeando un Proceso – File descriptors



Procesos – Vamos a los hechos

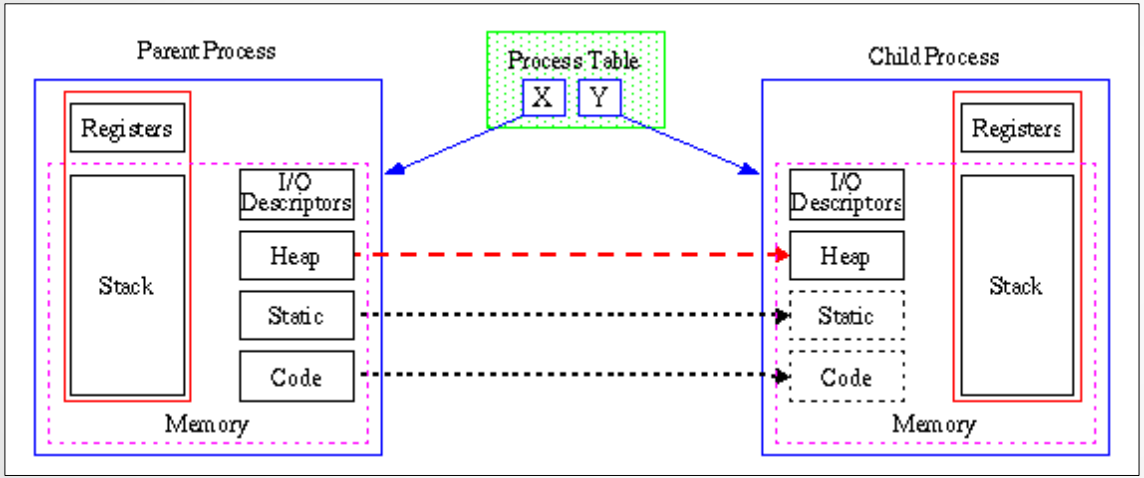
Ejemplos de código

Procesos y sockets - Práctica

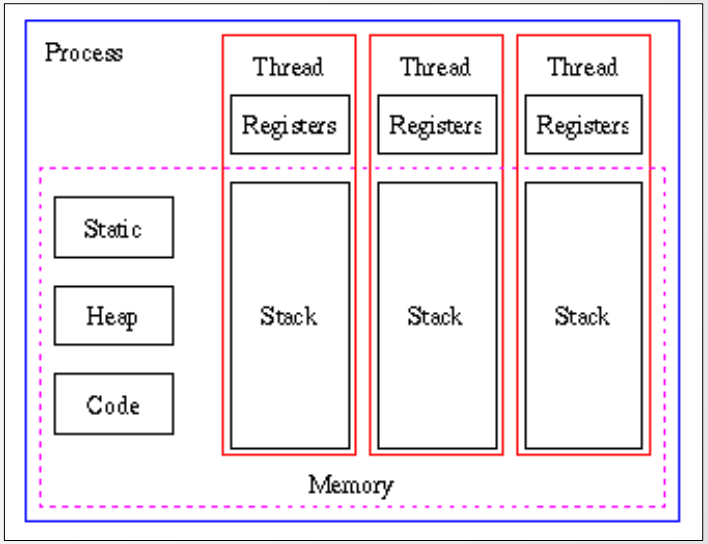
- Ejercicios Propuestos
  - Escriba un cliente HTTP que recupere y guarde en un directorio una página HTML (ambos recibidos como parámetro) y todos los objetos que contiene. Implemente – además – un caché de forma tal que si el objeto a recuperar existe y está actualizado no sea descargado nuevamente (use los headers HTTP correspondientes).
  - Implementar un Server HTTP multiproceso que sirva una página HTML a un navegador estándar, con los recursos asociados.
  - Agregarle al script anterior soporte de ejecución de scripts del lado del servidor.
  - Implementar un “Acelerador de descargas” HTTP.
  - Implementar un Proxy Server HTTP multiproceso.

Procesos vs. Hilos

Procesos



Hilos



Tomado de: https://www.w3.org/People/Frystyk/thesis/multithread.html