



# Programación de Sistemas

## CCPG1008

---

Federico Domínguez, PhD.

Unidad 6 – Sesión 2: Concurrencia con procesos

# Contenido

---

Concurrencia con procesos

Ejemplo aplicación echo

# La forma más fácil de implementar concurrencia con procesos es usando las funciones *fork*, *execve* y *wait* en conjunto.

---

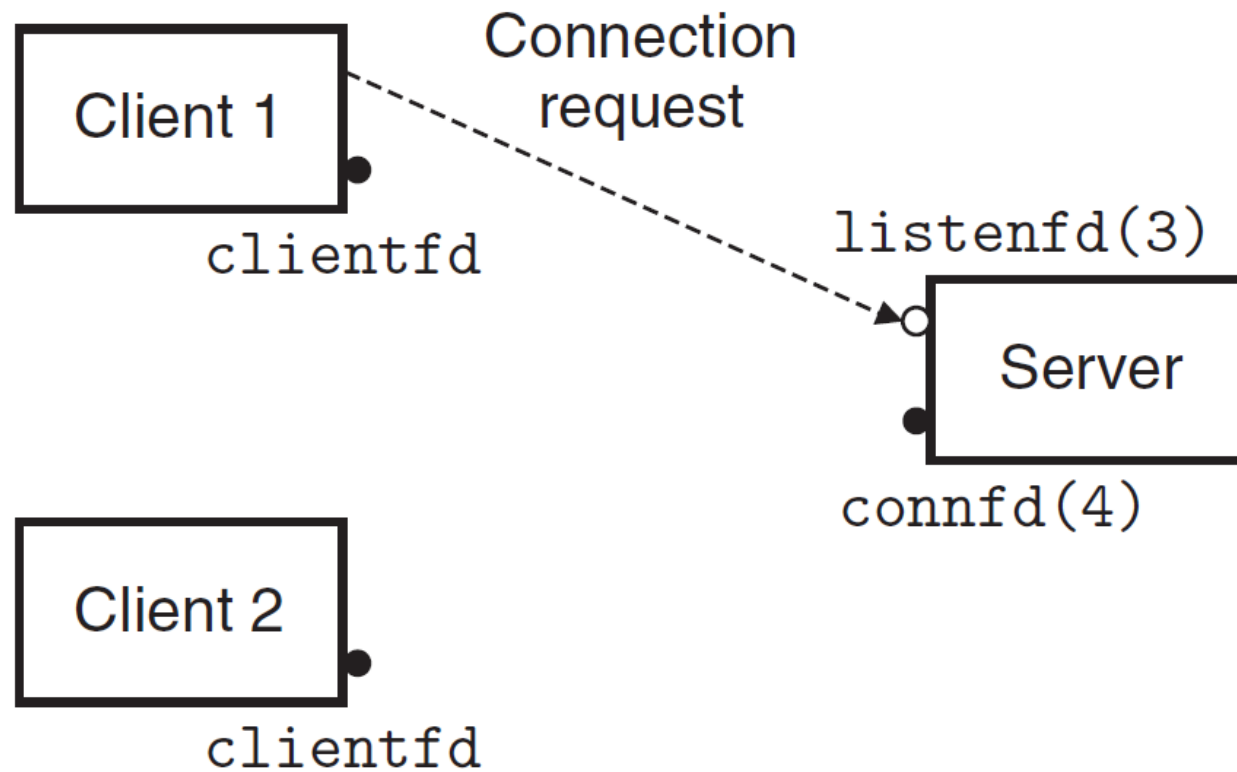
La función *fork* crea el proceso nuevo.

La función *execve* carga un programa nuevo en este proceso. (si fuese necesario)

La función *wait* espera y recoge el proceso hijo creado para liberar los recursos asignados.

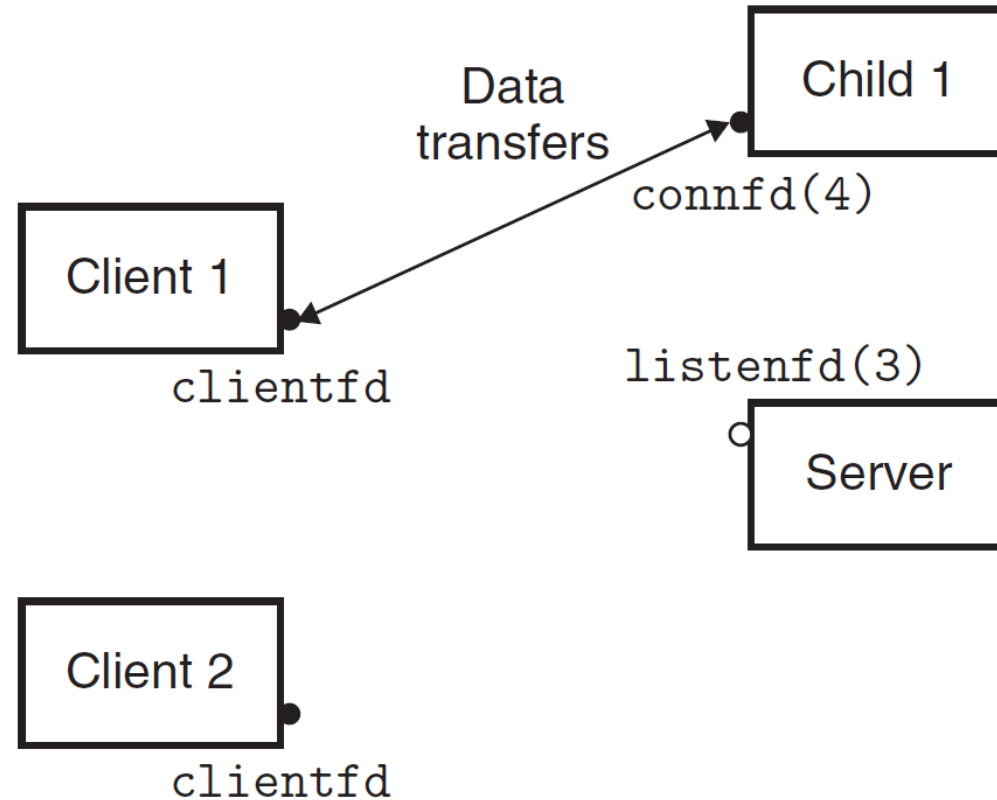
# Ejemplo de concurrencia con sockets

---



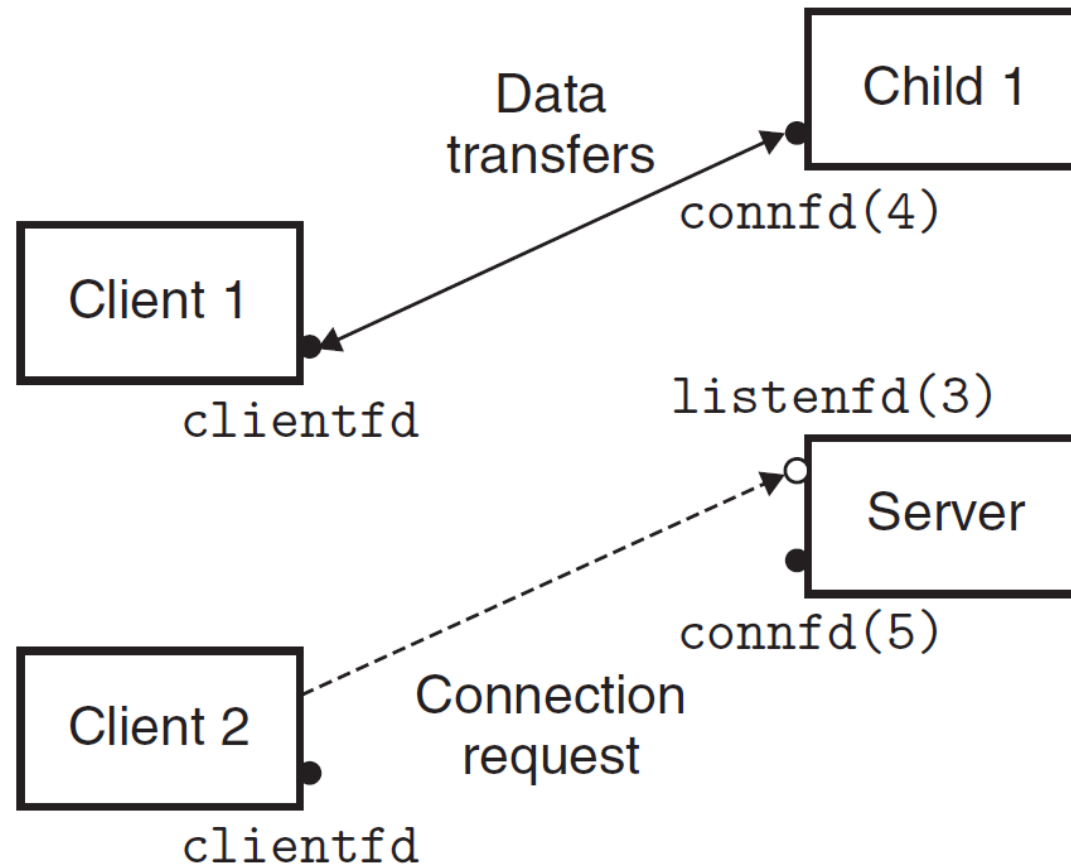
Al aceptar una conexión, el servidor crea un proceso hijo con *fork*. El servidor cierra el descriptor de conexión y el proceso hijo cierra el descriptor de escucha.

---



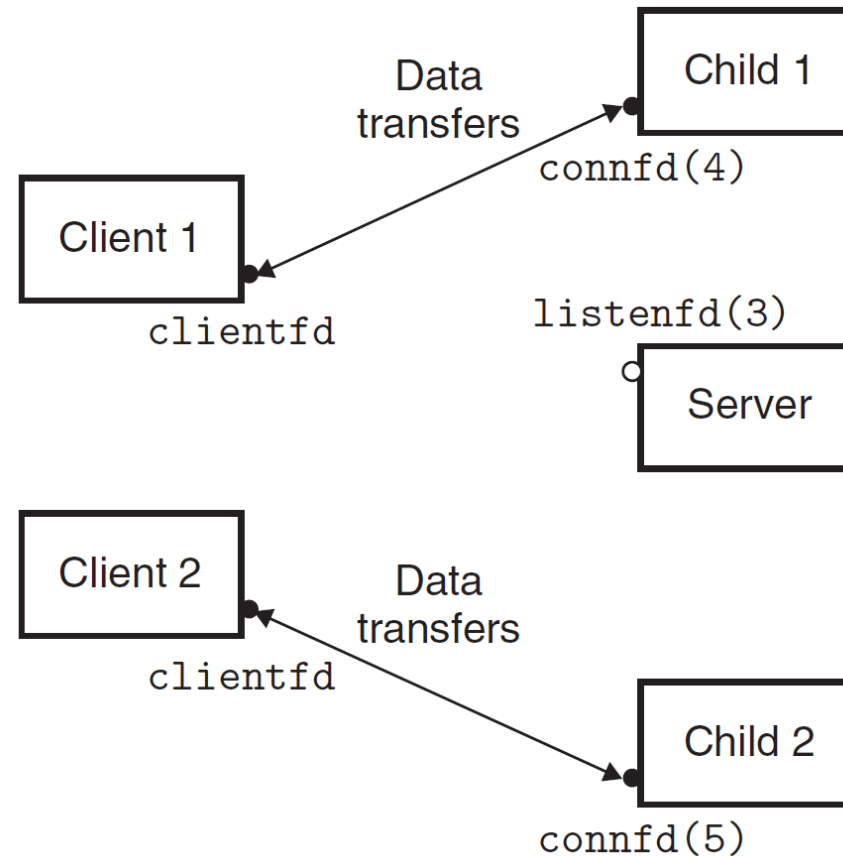
De esta forma, el proceso hijo se encarga de la transferencia de datos con el cliente mientras el proceso padre espera por nuevas conexiones.

---



# Al establecerse una conexión, se repite el proceso.

---



# Ejemplo de la aplicación echo

---

```
Signal(SIGCHLD, sigchld_handler);
listenfd = Open_listenfd(port);
while (1) {
    connfd = Accept(listenfd, (SA *) &clientaddr, &clientlen);
    if (Fork() == 0) {
        Close(listenfd); /* Child closes its listening socket */
        echo(connfd);    /* Child services client */
        Close(connfd);   /* Child closes connection with client */
        exit(0);         /* Child exits */
    }
    Close(connfd); /* Parent closes connected socket (important!) */
}
```



# Ejemplo de la aplicación echo

---

```
void sigchld_handler(int sig)
{
    while (waitpid(-1, 0, WNOHANG) > 0)
        ;
    return;
}
```

# Referencias

---

Libro guía *Computer Systems: A programmers perspective*. Secciones 12.0-1