

# Fork y Thread

## Múltiples instancias y comunicación

Esteban De La Fuente Rubio  
esteban@delaf.cl  
L<sup>A</sup>T<sub>E</sub>X

Universidad Andrés Bello

13 oct 2011

# Tabla de contenidos

- 1 Introducción
- 2 Fork
  - Funciones librería C
  - Ventajas y desventajas
- 3 Thread
  - Funciones librería C
  - Ventajas y desventajas

# Introducción

- En un sistema multiusuario se hace necesario poder atender simultáneamente a diferentes usuarios.
- Lo anterior implica permitir que un mismo proceso pueda ser ejecutado varias veces dependiendo de las solicitudes.
- Un servidor debe poder atender múltiples peticiones al mismo tiempo.

## Introducción (2)

- Procesos pueden trabajar colaborativamente en múltiples tareas (o cálculos) para un objetivo en común.
- Es necesario, por lo tanto, tener comunicados a dichos procesos.
- Se verán 2 formas de ejecutar múltiples instancias de un proceso: fork y thread.

# Fork

- Corresponde a un proceso creando otro, una bifurcación de un proceso.
- El proceso nuevo es otro proceso, con su propio PID y tiene su propia memoria.
- Se puede crear un proceso diferente o bien se puede crear un proceso idéntico.
- Ejemplo de system calls: exec o fork.

# exec()

- Ejecuta otro proceso indicado como parámetro a la función.
- La salida del proceso es mostrada por la pantalla.
- Si se quiere recuperar la salida del proceso usar popen.

# popen()

- Inicia un proceso creando una tubería haciendo fork para ejecutar el proceso mediante una shell.
- Tubería es unidireccional (solo lectura o solo escritura).
- Luego se procesa como si de un archivo se tratase.

# fork()

- Proceso padre e hijo tienen el mismo código para ejecución sin embargo se ejecutan de manera separada.

```
if ((processID = fork()) < 0) { /* error */ }  
else if (processID > 0) { /* proceso padre */ }  
else if (processID == 0) { /* proceso hijo */ }
```

# Ventajas

- Más simple de entender y utilizar que hebras.
- Más seguro ya que cada proceso tiene su propio espacio de ejecución.
- Si un proceso cae no afecta al resto (se puede hacer kill).

# Desventajas

- Cada proceso requiere su propio espacio de memoria, por lo que este debe ser creado y eliminado cada vez que se hace fork.
- Mayor uso de memoria, ya que se tiene replicado el proceso.
- Comunicación entre procesos es complicada, ya que no se comparte memoria se deben utilizar métodos externos (como tuberías o sockets).

# Thread

- Considerados como “procesos de peso ligero” (Light Weight Processes (LWPs)).
- Un thread es un hilo de ejecución de un proceso.
- Cada hilo es una ejecución independiente, pero que comparten espacio de memoria con el hilo principal.

## Thread (2)

Se comparte:

- Instrucciones.
- La mayoría de los datos.
- Descriptores de archivos.
- Mismo PID.
- Señales y manejadores de señales (problema con kill -9).

## Thread (3)

Es único:

- Thread ID.
- Registros
- Stack.
- Prioridad (planificación independiente).

# Funciones librería C

- `pthread_create()`: crea una nueva hebra.
- `pthread_detach()`: marca la hebra como separada, se liberarán los recursos cuando termine.
- `pthread_self()`: obtiene el ID de la hebra.

# Ejemplo

```
typedef struct {
    int contador;
} ThreadArgs;
pthread_t threadID;
ThreadArgs *threadArgs = malloc(sizeof(ThreadArgs));
threadArgs->contador = 0;
pthread_create(&threadID, NULL, ThreadMain,
              (void *) threadArgs);
void *ThreadMain (void *threadArgs) {
    pthread_detach(pthread_self());
    while(1) { args->contador++; sleep(1); }
    return NULL;
}
```

# Exclusión mutua

- Se debe utilizar un semáforo o mutex para evitar los problemas de la concurrencia.

```
typedef struct {  
    int contador;  
    pthread_mutex_t *mutex;  
} ThreadArgs;  
pthread_mutex_lock (args->mutex);  
args->contador++;  
pthread_mutex_unlock (args->mutex);
```

# Ventajas

- Todos los hilos comparten el mismo espacio de memoria.
- Comunicación entre procesos rápida al compartir memoria.
- Tienen menos sobrecarga que fork ya que comparten memoria.
- Rápidos para iniciar y terminar.
- Eficientes en sistemas con multi procesador.

# Desventajas

- Todos los hilos comparten el mismo espacio de memoria.