
Programación de Servicios y Procesos

Publicación 1.0

Oscar Gomez

25 de September de 2014

1. Programación multiproceso	1
1.1. Ejecutables. Procesos. Servicios.	1
1.2. Hilos.	2
1.3. Programación concurrente.	2
1.4. Programación paralela y distribuida.	2
1.5. Creación de procesos.	2
1.6. Comunicación entre procesos.	3
1.7. Gestión de procesos.	3
1.8. Comandos para la gestión de procesos en sistemas libres y propietarios.	3
1.9. Sincronización entre procesos.	3
1.10. Programación de aplicaciones multiproceso.	3
1.11. Documentación	3
1.12. Depuración.	3
2. Programación multihilo	5
2.1. Recursos compartidos por los hilos.	6
2.2. Estados de un hilo. Cambios de estado.	6
2.3. Elementos relacionados con la programación de hilos. Librerías y clases.	6
2.4. Gestión de hilos.	6
2.5. Creación, inicio y finalización.	6
2.6. Sincronización de hilos.	6
2.7. Información entre hilos. Intercambio.	6
2.8. Prioridades de los hilos.	6
2.9. Gestión de prioridades.	6
2.10. Compartición de información entre hilos.	6
2.11. Programación de aplicaciones multihilo.	6
2.12. Documentación.	6
2.13. Depuración.	6
3. Programación de comunicaciones en red	7
3.1. Comunicación entre aplicaciones.	8
3.2. Roles cliente y servidor.	8
3.3. Elementos de programación de aplicaciones en red. Librerías.	8
3.4. Funciones y objetos de las librerías.	8

3.5.	Sockets.	8
3.6.	Tipos de sockets. Características.	8
3.7.	Creación de sockets.	8
3.8.	Enlazado y establecimiento de conexiones.	8
3.9.	Utilización de sockets para la transmisión y recepción de información.	8
3.10.	Programación de aplicaciones cliente y servidor.	8
3.11.	Utilización de hilos en la programación de aplicaciones en red.	8
3.12.	Depuración.	8
4.	Generación de servicios en red	9
4.1.	Protocolos estándar de comunicación en red a nivel de aplicación	10
4.2.	Librerías de clases y componentes.	10
4.3.	Utilización de objetos predefinidos.	10
4.4.	Propiedades de los objetos predefinidos.	10
4.5.	Métodos y eventos de los objetos predefinidos.	10
4.6.	Establecimiento y finalización de conexiones.	10
4.7.	Transmisión de información.	10
4.8.	Programación de aplicaciones cliente.	10
4.9.	Programación de servidores.	10
4.10.	Implementación de comunicaciones simultáneas.	10
4.11.	Documentación.	10
4.12.	Depuración.	10
4.13.	Monitorización de tiempos de respuesta.	10
5.	Utilización de técnicas de programación segura	11
5.1.	Prácticas de programación segura.	11
5.2.	Criptografía de clave pública y clave privada.	11
5.3.	Principales aplicaciones de la criptografía.	11
5.4.	Protocolos criptográficos.	11
5.5.	Política de seguridad.	11
5.6.	Programación de mecanismos de control de acceso.	11
5.7.	Encriptación de información.	11
5.8.	Protocolos seguros de comunicaciones.	11
5.9.	Programación de aplicaciones con comunicaciones seguras.	11
5.10.	Pruebas y depuración.	11

Programación multiproceso

1.1 Ejecutables. Procesos. Servicios.

1.1.1 Ejecutables

Un ejecutable es un archivo con la estructura necesaria para que el sistema operativo pueda poner en marcha el programa que hay dentro. En Windows, los ejecutables suelen ser archivos con la extensión .EXE.

Sin embargo, Java genera ficheros .JAR o .CLASS. Estos ficheros *no son ejecutables* sino que son archivos que el intérprete de JAVA (el archivo `java.exe`) leerá y ejecutará.

El intérprete toma el programa y lo traduce a instrucciones del microprocesador en el que estemos, que puede ser x86 o un x64 o lo que sea. Ese proceso se hace “al instante” o JIT (Just-In-Time).

Un EXE puede que no contenga las instrucciones de los microprocesadores más modernos. Como todos son compatibles no es un gran problema, sin embargo, puede que no aprovechemos al 100 % la capacidad de nuestro micro.

1.1.2 Procesos

Es un archivo que está en ejecución y bajo el control del sistema operativo. Un proceso puede atravesar diversas etapas en su “ciclo de vida”. Los estados en los que puede estar son:

- En ejecución: está dentro del microprocesador.
- Pausado/detenido/en espera: el proceso tiene que seguir en ejecución pero en ese momento el S.O tomó la decisión de dejar paso a otro.
- Interrumpido: el proceso tiene que seguir en ejecución pero *el usuario* ha decidido interrumpir la ejecución.
- Existen otros estados pero ya son muy dependientes del sistema operativo concreto.

1.1.3 Servicios

Un servicio es un proceso que no muestra ninguna ventana ni gráfico en pantalla porque no está pensado para que el usuario lo maneje directamente.

Habitualmente, un servicio es un programa que atiende a otro programa.

1.2 Hilos.

Un hilo es un concepto más avanzado que un proceso: al hablar de procesos cada uno tiene su propio espacio en memoria. Si abrimos 20 procesos cada uno de ellos consume 20x de memoria RAM. Un hilo es un proceso mucho más ligero, en el que el código y los datos se comparten de una forma distinta.

Un proceso no tiene acceso a los datos de otro procesos. Sin embargo un hilo sí accede a los datos de otro hilo. Esto complicará algunas cuestiones a la hora de programar.

1.3 Programación concurrente.

La programación concurrente es la parte de la programación que se ocupa de crear programas que pueden tener varios procesos/hilos que colaboran para ejecutar un trabajo y aprovechar al máximo el rendimiento de sistemas multinúcleo.

1.4 Programación paralela y distribuida.

Dentro de la programación concurrente tenemos la paralela y la distribuida:

- En general se denomina “programación paralela” a la creación de software que se ejecuta siempre en un solo ordenador (con varios núcleos o no)
- Se denomina “programación distribuida” a la creación de software que se ejecuta en ordenadores distintos y que se comunican a través de una red.

1.5 Creación de procesos.

En Java es posible crear procesos utilizando algunas clases que el entorno ofrece para esta tarea. En este tema, veremos en profundidad la clase `ProcessBuilder`.

El ejemplo siguiente muestra como lanzar un proceso de Acrobat Reader:

```
public class LanzadorProcesos {  
    public void ejecutar(String ruta) {  
  
        ProcessBuilder pb;
```

```
        try {
            pb = new ProcessBuilder(ruta);
            pb.start();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        String ruta=
            "C:\\Program Files (x86)\\Adobe\\Reader 11.0\\Reader\\Ac
        LanzadorProcesos lp=new LanzadorProcesos();
        lp.ejecutar(ruta);
        System.out.println("Finalizado");
    }
}
```

1.6 Comunicación entre procesos.

1.7 Gestión de procesos.

1.8 Comandos para la gestión de procesos en sistemas libres y propietarios.

1.9 Sincronización entre procesos.

1.10 Programación de aplicaciones multiproceso.

1.11 Documentación

1.12 Depuración.

Programación multihilo

- 2.1 Recursos compartidos por los hilos.**
- 2.2 Estados de un hilo. Cambios de estado.**
- 2.3 Elementos relacionados con la programación de hilos. Librerías y clases.**
- 2.4 Gestión de hilos.**
- 2.5 Creación, inicio y finalización.**
- 2.6 Sincronización de hilos.**
- 2.7 Información entre hilos. Intercambio.**
- 2.8 Prioridades de los hilos.**
- 2.9 Gestión de prioridades.**
- 2.10 Compartición de información entre hilos.**
- 2.11 Programación de aplicaciones multihilo.**
- 2.12 Documentación.**
- 2.13 Depuración.**

Programación de comunicaciones en red

3.1 Comunicación entre aplicaciones.

3.2 Roles cliente y servidor.

**3.3 Elementos de programación de aplicaciones en red.
Librerías.**

3.4 Funciones y objetos de las librerías.

3.5 Sockets.

3.6 Tipos de sockets. Características.

3.7 Creación de sockets.

3.8 Enlazado y establecimiento de conexiones.

3.9 Utilización de sockets para la transmisión y recepción de información.

3.10 Programación de aplicaciones cliente y servidor.

3.11 Utilización de hilos en la programación de aplicaciones en red.

8.12 Depuración. Capítulo 3. Programación de comunicaciones en red

Generación de servicios en red

4.1 Protocolos estándar de comunicación en red a nivel de aplicación

4.1.1 Telnet

4.1.2 FTP

4.1.3 HTTP

4.1.4 POP3

4.1.5 SMTP

4.2 Librerías de clases y componentes.

4.3 Utilización de objetos predefinidos.

4.4 Propiedades de los objetos predefinidos.

4.5 Métodos y eventos de los objetos predefinidos.

4.6 Establecimiento y finalización de conexiones.

4.7 Transmisión de información.

4.8 Programación de aplicaciones cliente.

4.9 Programación de servidores.

4.10 Implementación de comunicaciones simultáneas

Utilización de técnicas de programación segura

5.1 Prácticas de programación segura.

5.2 Criptografía de clave pública y clave privada.

5.3 Principales aplicaciones de la criptografía.

5.4 Protocolos criptográficos.

5.5 Política de seguridad.

5.6 Programación de mecanismos de control de acceso.

5.7 Encriptación de información.

5.8 Protocolos seguros de comunicaciones.

5.9 Programación de aplicaciones con comunicaciones seguras.

5.10 Pruebas y depuración.