Proyecto FPMADdigital

Recursos digitales y multimedia para Formación Profesional

Comunidad de Madrid

Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial

CONSEJERÍA DE EDUCACIÓN, UNIVERSIDADES, CIENCIA Y PORTAVOCÍA

Unión Europea

Fondo Social Europeo

"El FSE invierte en tu futuro"

Financiado como parte de la respuesta de la Unión a la pandemia de COVID-19

FPMADdigital
Recursos digitales y multimedia para Formación Profesional

CFGS Desarrollo de Aplicaciones Multiplataforma

módulo profesional

0490 - Programación de Servicios y Procesos

unidad didáctica

01 Programación multiproceso

resultados de aprendizaje

01 Desarrolla aplicaciones compuestas por varios procesos

Comunidad de Madrid
Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial
CONSEJERÍA DE EDUCACIÓN, UNIVERSIDADES, CIENCIA Y PORTAVOCÍA

Unión Europea
Fondo Social Europeo
"El FSE invierte en tu futuro"
Financiado como parte de la respuesta de la Unión a la pandemia de COVID-19

Resultados de aprendizaje y unidades didácticas

Resultados de aprendizaje y unidades didácticas

RESULTADOS DE APRENDIZAJE					UNIDAD DIDÁCTICA
1	2	3	4	5	
X					1.- Programación multiproceso
	X				2.- Programación concurrente y asíncrona
		X			3.- Programación de comunicaciones en red
			X		4.- Generación de servicios en red
				X	5.- Técnicas de programación segura

Unidades didácticas y materiales asociados

Unidades didácticas y materiales multimedia

RRAA					UU.DD	Material Multimedia
1	2	3	4	5		
X					1.- Programación multiproceso	1.1 Contenidos básicos 1.2 Ejemplos aplicados
	X				2.- Programación concurrente y asíncrona	2.1 Contenidos básicos 2.2 Ejemplos aplicados
		X			3.- Programación de comunicaciones en red	3.1 Contenidos básicos 3.2 Ejemplos aplicados
			X		4.- Generación de servicios en red	4.1 Contenidos básicos 4.2 Ejemplos aplicados
				X	5.- Técnicas de programación segura	5.1 Contenidos básicos 5.2 Ejemplos aplicados

Repositorios de materiales y prácticas

Repositorio de materiales y prácticas

Todos los proyectos mostrados, así como otros materiales utilizados en las unidades didácticas los podrás encontrar completos en:

https://github.com/joseluisgs/FP-NextGen-ProgramacionServiciosProcesos

Cualquier error o propuestas de mejora se publicarán en el repositorio indicado.
Gracias por tu colaboración.

Contenidos

1. Aplicaciones y Procesos
2. Gestión de Procesos
3. Programación de Procesos

Aplicaciones y Procesos

Aplicaciones y Procesos

Una **aplicación** es un tipo de programa informático, diseñado como herramienta para resolver de manera automática un problema específico del usuario.

Un **ejecutable** es un fichero que contiene el código binario o interpretado que será ejecutado en un ordenador.

Un **proceso** es un programa en ejecución. Pero, es más que eso, un proceso en el sistema operativo (SO), es una unidad de trabajo completa; y, el SO gestiona los distintos procesos que se encuentren en ejecución en el equipo. Un **proceso** existe mientras que se esté ejecutando una aplicación. Es más, la ejecución de una aplicación, puede implicar que se arranquen varios procesos en nuestro equipo; y puede estar formada por varios ejecutables y librerías.

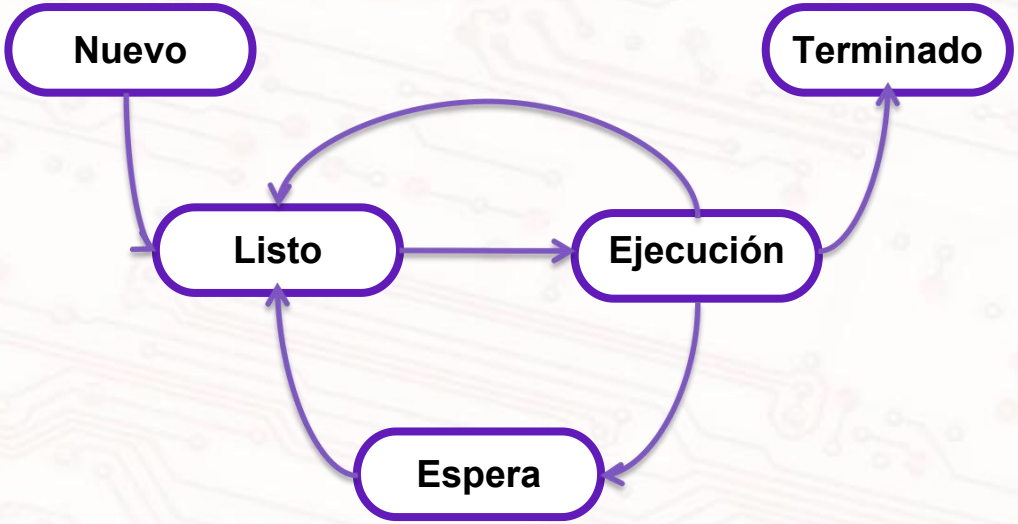
Gestión de Procesos

Tipos de procesos

- **Por lotes**. Están formados por una serie de tareas, de las que el usuario sólo está interesado en el resultado final. Por ejemplo: enviar a imprimir varios documentos, escanear nuestro equipo en busca de virus...
- **Interactivos**. Aquellas tareas en las que el proceso interactúa continuamente con el usuario y actúa de acuerdo a las acciones que éste realiza, o a los datos que suministra. Por ejemplo: un procesador de textos, una aplicación formada por formularios que permiten introducir datos en una base de datos...
- **Tiempo real**. Tareas en las que es crítico el tiempo de respuesta del sistema. Por ejemplo: el ordenador de a bordo de un automóvil, reaccionará ante los eventos del vehículo en un tiempo máximo que consideramos correcto y aceptable. Otro ejemplo, son los equipos que controlan los brazos mecánicos en los procesos industriales de fabricación.

Estados de un proceso

1. **Nuevo**. Proceso nuevo, creado.
2. **Listo**. Proceso que está esperando la CPU para ejecutar sus instrucciones.
3. En **Ejecución**. Proceso que actualmente, está en turno de ejecución en la CPU.
4. **Espera/Bloqueado**. Proceso que está a la espera de que finalice una E/S.
5. **Suspendido**. Proceso que se ha llevado a la memoria virtual para liberar, un poco, la RAM del sistema.
6. **Terminado**. Proceso que ha finalizado y ya no necesitará más la CPU.



PCB: Bloque de Control del Proceso

El **bloque de control del proceso** es un registro especial donde el sistema operativo agrupa toda la información que necesita conocer respecto a un proceso particular.
Identificador de proceso.

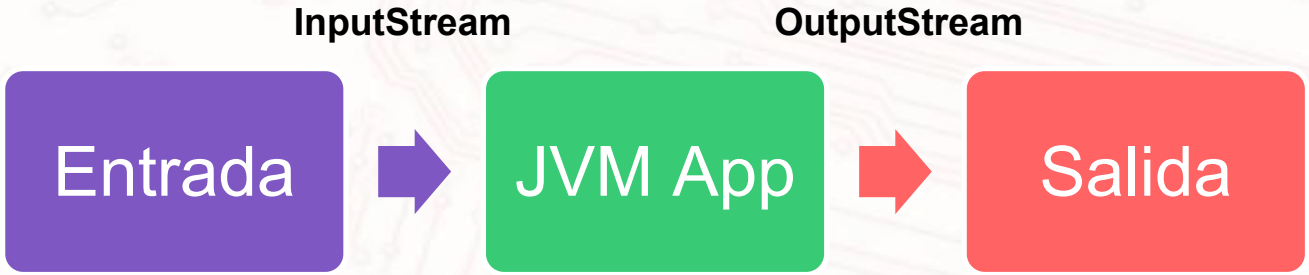
- **Estado del proceso:** listo, en espera, bloqueado.
- **Contador de programa**: dirección de la próxima instrucción a ejecutar.
- **Valores de registro de CPU**: se utilizan también en el cambio de contexto.
- **Espacio de direcciones** de memoria.
- **Prioridad** en caso de utilizarse dicho algoritmo para planificación de CPU.
- **Lista de recursos asignados** (incluyendo descriptores de archivos y sockets abiertos).
- **Estadísticas** del proceso.
- Datos del **propietario** (owner).
- **Permisos** asignados.
- **Señales** (Signals) pendientes de ser servidas. (Almacenados en un mapa de bits).

Programación de Procesos

Programación de Procesos

Usaremos:

- **ProcessBuilder**: para crear el manejador del proceso.
- **InputStream**: stream de datos entrantes al proceso.
- **OutputStream**: stream de datos salientes del proceso.



```java
public class Main {
    public static void main(String[] args) {
        try {

            // Pasandoles datos a un proceso
            Process p=Runtime.getRuntime().exec ("grep java");

            // La entrada de su información es OutputStream
            OutputStream out = p.getOutputStream();
            PrintWriter pw =new PrintWriter(new OutputStreamWriter(out));
            pw.println("Me gusta PSP y java");
            pw.println("Soy un crack de java");
            pw.println("No se me escapa ni una");
            pw.println("Pedazo de clase de java");
            pw.println("java y los procesos me quieren");
            pw.close();

            // Ya le hemos pasado la información, ahora leemos su salida
            InputStream in=p.getInputStream();
            BufferedReader br=new BufferedReader(new InputStreamReader(in));
            String linea;
            while ((linea = br.readLine()) != null)
                System.out.println(linea);
            br.close();

            // Esperamos a que termine
            p.waitFor();

            System.out.println("valor de salida " + p.exitValue());

        } catch (IOException ex) {
            System.err.println("Error al ejecutar el proceso");
        } catch (InterruptedException ex) {
            System.err.println("Error en wait for");
        }
    }
}
```

Conclusiones

¡Vamos con la práctica!

"Menos del 10% del código tienen que ver directamente con el propósito del sistema; el resto tiene que ver con la entrada y salida, validación de datos, mantenimiento de estructuras de datos y otras labores domésticas"
- Mary shaw

Comunidad de Madrid
Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial
CONSEJERÍA DE EDUCACIÓN, UNIVERSIDADES, CIENCIA Y PORTAVOCÍA

Unión Europea
Fondo Social Europeo
"El FSE invierte en tu futuro"
Financiado como parte de la respuesta de la Unión a la pandemia de COVID-19