**Multiproceso:** se puede ejecutar más de un proceso a la vez. La CPU va alternando los procesos muy rápido.

**Programación multiproceso:** varios procesos ejecutándose sobre el mismo código.

**Proceso:** programa en ejecución.

**1.2.1 Estados de un proceso.**

* **En ejecución:** el proceso se está ejecutando.
* **Bloqueado:** el proceso no hace nada hasta que ocurra un evento.
* **Listo:** el proceso está parado temporalmente y listo para ejecutarse en cuanto pueda.

**Concurrencia:** varios procesos en ejecución.

**Programa:** conjunto de instrucciones con datos de entrada para obtener una salida.

Dos procesos son concurrentes si el segundo se ejecuta después de que empiece el primero y antes de que este termine.

**Programación paralela:** si los procesos se ejecutan a la vez (más de un procesador o núcleo).

Para que los programas se ejecuten concurrentemente de forma correcta se necesitan **mecanismos de comunicación y sincronización entre procesos.**

**Programación concurrente:** se encarga de especificar la ejecución concurrente de las acciones de un programa y con los problemas que conlleva.

**1.3.2 Beneficios de la programación concurrente.**

* **Mejor aprovechamiento de la CPU.**
* **Velocidad de ejecución.**
* **Solución a problemas de naturaleza concurrente:** algunos ejemplos son los sistemas de control. Tecnologías web. Aplicaciones basadas en GUI. Simulación. SGBD.

En un sistema monoprocesador los procesos comparten la misma memoria, forma de comunicar y sincronizar procesos se hace con variables compartidas. En la multiprogramación un proceso abandona el procesador y otro lo ocupa.

En un sistema multiprocesador se puede tener un proceso en cada procesador. Pueden ser fuertemente acoplados si comparten memoria o débilmente acoplados si no. Multiproceso es la gestión de varios procesos en un sistema multiprocesador.

**1.3.3 Programas concurrentes.**

Programa concurrente define un conjunto de acciones que pueden ser ejecutadas a la vez.

**Bernstein:**

* **Conjunto de lectura:** las instrucciones con variables que hay que leer.
* **Conjunto de escritura:** las instrucciones con variables que hay que escribir.

**1.3.4 Problemas de la programación concurrente.**

* **Exclusión mutua:** a que 2 procesos accedan a la misma variable a la vez. La región crítica es para que un proceso acceda a la variable mientras los otros esperan.
* **Condición de sincronización:** que un proceso necesite que otro avance para que pueda continuar su ejecución.

Herramientas para manejar la concurrencia: la región crítica. Los semáforos, buzones, sucesos, monitores…

**1.4.1 Programación Paralela.**

**Programa paralelo:** programa concurrente diseñado para multiprocesador. El problema se divide en partes y estas se ejecutan a la vez.

En un sistema multiprocesador hay 2 modelos de programación paralela:

* Modelo de memoria compartida: la memoria se comparte, los valores escritos se pueden leer por cualquier procesador.
* Modelo de paso de mensajes: cada procesador tiene su memoria. Los procesadores hacen peticiones de información al que la tenga.

Los sistemas multiprocesadores comparten la memoria y los sistemas multicomputadores cada uno tiene su propia memoria.

Los clusters son sistemas de procesamiento paralelo que utiliza varios ordenadores enlazados por una red que simulan un único ordenador.

**Ventajas del procesamiento paralelo:**

* Ejecución simultánea de tareas.
* Disminuye el tiempo de ejecución de una aplicación.
* Resolución de problemas complejos.
* Utilización de recursos no locales como internet.
* Disminución de costes respecto a supercomputadores.

**Desventajas del procesamiento paralelo:**

* Más difícil de desarrollar, compilar y escribir.
* Mayor consumo de energía.
* Mayor complejidad para acceder a los datos.
* Comunicar y sincronizar las tareas.

**1.4.2 Programación distribuida.**

En un sistema distribuido los componentes se comunican por una red a través de mensajes. Tiene las siguientes consecuencias:

* Concurrencia.
* Inexistencia del reloj global.
* Fallos independientes.

En un sistema distribuido hay 3 modelos de para la comunicación de procesos:

* Sockets: Proporcionan los puntos extremos para la comunicación.
* Llamadas de procedimientos remotos o RPC: un cliente llama a un procedimiento en el servidor.
* Invocación remota de objetos: un objeto de un proceso puede llamar métodos de otro objeto en otro proceso.

**Ventajas los sistemas distribuidos:**

* Se pueden compartir recursos y datos.
* Capacidad de crecimiento incremental.
* Mayor flexibilidad de carga de trabajo.
* Alta disponibilidad.
* Soporte de aplicaciones.

**Desventajas los sistemas distribuidos:**

* Aumento de la complejidad.
* Problemas con la red cómo pérdida de mensajes.
* Problemas de seguridad como ddos.

RESUMEN TO GUAPO:

* **Programación Concurrente:** Varios elementos de proceso que trabajan de forma conjunta en la resolución de un problema. En un único procesador.
* **Programación paralela:** programación concurrente para la resolución de un problema en varios procesadores.
* **Programación Distribuida:** programación paralela distribuida en la red. Se usan mensajes.