

Programación multiproceso (Programación de servicios y procesos)

Ejecutables. Procesos. Servicios

Uno de los conceptos más importantes en los sistemas operativos es el proceso que se define como un programa en ejecución. Los programas son un conjunto de archivos que están almacenados en algún dispositivo de almacenamiento (disco duro, USB, etc.) y que por sí solos no tienen ningún funcionamiento, pero cuando ese conjunto de archivos se ejecutan entonces pasan a ser un proceso.

Los procesos que se ejecutan pueden estar en primer plano o segundo plano: los de primer plano interactúan con los usuarios, y los de segundo plano o demonios realizan una función específica sin tener que actuar con el usuario (aunque podrían hacerlo).

Los procesos en segundo plano que realizan distintas funciones, algunas relacionadas con el sistema operativo y otras no, se denominan servicios, y se están ejecutando permanentemente en el sistema.

Más información

- <https://www.jesusninoc.com/07/07/7-gestion-de-procesos-e-n-powershell/>
- <https://www.jesusninoc.com/07/07/7-gestion-de-procesos-e-n-powershell/#Servicios>
- <https://www.jesusninoc.com/03/09/abrir-chrome-en-modo-in>

Ejercicios

- <https://www.jesusninoc.com/?s=powershell+procesos>
 - <https://www.jesusninoc.com/?s=powershell+servicios>
 - <https://www.jesusninoc.com/02/17/ejercicios-de-java-lanzar-programa-que-ejecute-ping/>
 - <https://www.jesusninoc.com/02/17/ejercicios-de-java-lanzar-programa-que-ejecute-un-ping-dos-veces/>
-

Estados de un proceso

Los procesos cuando se ejecutan pueden atravesar distintos estados desde que se inician hasta que finalizan. Cuando un proceso se inicia, primero se mete en una cola de trabajos; cuando es admitido por el sistema, pasa a una cola de procesos que están preparados y esperando para ejecutarse; cuando el procesador asigna tiempo de ejecución, el proceso pasa de estar preparado a ejecución, y cuando el proceso necesita alguna señal o dato, pasa al estado bloqueado (se lo introduce en la cola de bloqueados).

Más información

- https://www.jesusninoc.com/07/07/7-gestion-de-procesos-en-powershell/#Tabla_de_procesos
- https://www.jesusninoc.com/07/07/7-gestion-de-procesos-en-powershell/#Estados_de_un_proceso

Ejercicios

- <https://www.jesusninoc.com/?s=powershell+estados+proceso>
-

Hilos

Un proceso tiene miniprocesos que se llaman hilos, la razón de dividir un proceso en unidades más pequeñas es que la mayoría de los procesos están realizando varias acciones a la vez (leer del disco, esperar un clic del ratón, etc.) y puede ser que alguna de esas acciones se bloquee y deje bloqueado al proceso completo.

Utilizando los hilos, el bloqueo sólo se produce en una parte y no todo el proceso, esto quiere decir que otros hilos se pueden seguir ejecutando sin problema.

Algunas características de los hilos:

- Los procesos crean hilos.
 - Los hilos tienen prioridad.
 - Se pueden ver los hilos que se ejecutan.
 - Se pueden ver las prioridades de los hilos.
 - Se pueden relacionar hilos con procesos y servicios.
-

Más información

- <https://www.jesusninoc.com/07/07/7-gestion-de-procesos-e-n-powershell/#Hilos>
-

Ejercicios

- <https://www.jesusninoc.com/01/10/ejercicios-de-powershell-ejercicios-sobre-hilos/>
 - [Crear un hilo implementando el interfaz Runnable](#)
 - [Crear un hilo heredando de la clase Thread](#)
-

Programación concurrente

La programación concurrente consiste en que los procesos comparten el mismo tiempo pero no se ejecutan en el mismo instante.

Programación paralela

La programación paralela consiste en ejecutar en el mismo instante varios procesos en los diferentes procesadores o núcleos. La programación paralela es imposible en ordenadores con un procesador de un solo núcleo.

Programación distribuida

La programación distribuida consiste en utilizar sistemas en los que los componentes hardware o software se encuentran en ordenadores unidos mediante la red y se comunican mediante el paso de mensajes. Es importante indicar que los sistemas distribuidos no están sincronizados.

Comunicación entre procesos

En ocasiones los procesos tienen que comunicarse unos con otros pasándose información, la forma de comunicarse entre procesos se puede realizar mediante, por ejemplo, zonas de memoria compartida o pipes.

Existen algunas situaciones que el sistema operativo debe controlar para que la comunicación entre procesos sea correcta. Una de ellas es asegurar que si un proceso utiliza una zona o archivo compartido, no deje a otros procesos utilizar esos recursos a la vez, es decir, excluir del acceso a otros, esto se conoce como exclusión mutua. La solución es controlar el acceso a la región crítica, que se puede definir como la parte de un programa en la cual se accede a un recurso compartido y donde no debe haber más de un proceso accediendo simultáneamente. Habrá, por tanto, que ordenar y controlar el acceso que hacen los programas a esa región crítica, no puede haber dos procesos simultáneamente en sus regiones críticas», entonces si un programa entra en la sección crítica, no permitirá el acceso a otro programa.

Una tubería es una sección de memoria compartida que los procesos utilizan para la comunicación. El proceso que crea una tubería es el servidor de tuberías. Un proceso que se conecta a una tubería es un cliente de tubería. Un proceso escribe información en la tubería, luego el otro proceso lee la información de la tubería.

Más información

- https://www.jesusninoc.com/07/07/7-gestion-de-procesos-en-powershell/#Comunicacion_entre_procesos
- <https://www.jesusninoc.com/07/10/programacion-de-memorias-compartidas-en-linux/>
- <https://www.jesusninoc.com/11/22/uso-de-las-funciones-ftok-shmget-shmat-y-shmctl/>
- <https://www.jesusninoc.com/02/06/comunicar-dos-procesos-en-powershell-mediante-el-uso-de-tuberias-pipes-un-proceso-crea-y-escribe-en-la-tuberia-y-otro-proceso-lee-la-informacion/>
- <https://www.jesusninoc.com/02/06/comunicar-dos-procesos->

[en-powershell-mediante-el-uso-de-tuberias-pipes-un-proceso-crea-y-lee-en-la-tuberia-y-otro-proceso-escribe-la-informacion/](#)

- [https://www.jesusninoc.com/05/26/utilizar-la-tuberia-pipe-o-en-wsl-windows-subsystem-for-linux/](#)
 - [https://www.jesusninoc.com/02/27/crear-y-utilizar-una-tuberia-sin-nombre-en-linux/](#)
 - [https://www.jesusninoc.com/02/27/crear-y-utilizar-una-tuberia-con-nombre-en-linux-fifo/](#)
-

Ejercicios

- [https://www.jesusninoc.com/11/22/uso-de-las-funciones-ftok-shmget-shmat-y-shmctl-ejemplo/](#)
 - [https://www.jesusninoc.com/11/22/uso-de-las-funciones-ftok-shmget-shmat-y-shmctl-ejemplo/](#)
 - [https://www.jesusninoc.com/06/25/utilizar-zonas-de-memoria-compartida-en-linux-mediante-wsl-desde-powershell/](#)
 - [https://www.jesusninoc.com/04/05/resumen-de-las-operaciones-con-memoria-compartida-en-php/](#)
-

Gestión de procesos

El sistema operativo se encarga de: crear y destruir procesos, suspender y reanudar procesos, y sincronizar y comunicar procesos.

Más información

- [https://www.jesusninoc.com/07/07/7-gestion-de-procesos-en-powershell/#Crear_procesos](#)

- https://www.jesusninoc.com/07/07/7-gestion-de-procesos-en-powershell/#Parar_procesos
-

Ejercicios

- [Create operating system processes](#)
 - [Create and destroy operating system processes](#)
 - [Crear un proceso utilizando Runtime](#)
 - [Crear un proceso utilizando ProcessBuilder](#)
 - [Lanzar un programa del sistema operativo como argumento desde la línea de comandos](#)
 - [Ejecutar y guardar el comando 'arp' utilizando ProcessBuilder](#)
-

Sincronización entre procesos

La sincronización consiste en ejecutar de forma correcta y ordenada los procesos que están en el sistema y de esa forma se evitan errores.

Programación de aplicaciones multiproceso

Dependiendo del lenguaje de programación y el sistema operativo en el que se quiera programar habrá que hacer la programación de una forma u otra.