

# Proceso (informática)

Un **proceso**, en informática, puede entenderse informalmente como un programa en ejecución. Formalmente un proceso es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados".<sup>1</sup>

Para entender mejor lo que es un proceso y la diferencia entre un programa y un proceso, A. S. Tanenbaum propone la analogía "Un científico computacional con mente culinaria hornea un pastel de cumpleaños para su hija; tiene la receta para un pastel de cumpleaños y una cocina bien equipada con todos los ingredientes necesarios, harina, huevo, azúcar, leche, etc." Situando cada parte de la analogía se puede decir que la receta representa el programa (el algoritmo), el científico computacional es el procesador y los ingredientes son las entradas del programa. El proceso es la actividad que consiste en que el científico computacional vaya leyendo la receta, obteniendo los ingredientes y horneando el pastel.

Cada proceso tiene su contador de programa, registros y variables, aislados de otros procesos, incluso siendo el mismo programa en ejecución 2 veces. Cuando este último caso sucede, el sistema operativo usa la misma región de memoria de código, debido a que dicho código no cambiará, a menos que se ejecute una versión distinta del programa.

Los procesos son gestionados por el sistema operativo y están formados por:

- Las instrucciones de un programa destinadas a ser ejecutadas por el microprocesador.
- Su estado de ejecución en un momento dado, esto es, los valores de los registros de la unidad central de procesamiento para dicho programa.
- Su memoria de trabajo (memoria crítica), es decir, la memoria que ha reservado y sus contenidos.
- Otra información que permite al sistema operativo su planificación.

Un proceso se rige en pequeñas porciones, conocidas como páginas, y cada proceso tiene su propia tabla de paginación, fungiendo como una optimización del sistema operativo ante los fallos de página.

Esta definición varía ligeramente en el caso de sistemas operativos multihilo, donde un proceso consta de uno o más hilos, la memoria de trabajo (compartida por todos los hilos) y la información de planificación. Cada hilo consta de instrucciones y estado de ejecución.

Los procesos son creados y eliminados por el sistema operativo, así como también este se debe hacer cargo de la comunicación entre procesos, pero lo hace a petición de otros procesos (interrupción o tiempo de reloj). El mecanismo por el cual un proceso crea otro proceso se denomina bifurcación (*fork*). El proceso de arranque de GNU/Linux inicia con un solo proceso (init) y después comienza a crear los hilos necesarios para tener el sistema listo para su uso. Los nuevos procesos pueden ser independientes y no compartir el espacio de memoria con el proceso que los ha creado o ser creados en el mismo espacio de memoria.

En los sistemas operativos multihilo es posible crear tanto hilos como procesos. La diferencia estriba en que un proceso solamente puede crear hilos para sí mismo y en que dichos hilos comparten toda la memoria reservada para el proceso.

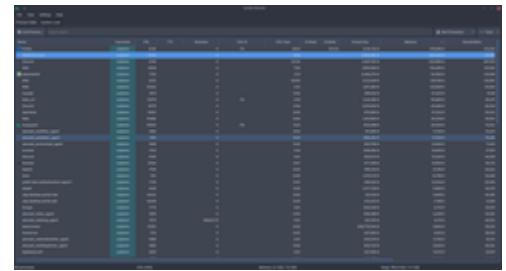


tabla de proceso

Los procesos pueden ser cooperativos o independientes. Dos o más procesos pueden cooperar mediante señales de forma que uno obliga a detenerse a los otros hasta que reciban una señal para continuar.

- Se usa una variable de tipo semáforo para sincronizar los procesos.
- Si un proceso está esperando una señal, se suspende hasta que la señal se envíe.
- Se mantiene una cola de procesos en espera en el semáforo.
- La forma de elegir los procesos de la cola en espera es mediante una política first in first out.

La sincronización explícita entre procesos es un caso particular del estado "bloqueado". En este caso, el suceso que permite desbloquear un proceso no es una operación de entrada/salida, sino una señal generada a propósito por el programador desde otro proceso.

Hay cuatro eventos principales que provocan la creación de procesos:

- El arranque del sistema.
- La ejecución, desde un proceso, de una llamada al sistema para la creación de otro proceso.
- Una petición de usuario para crear un proceso.
- El inicio de un trabajo por lotes.

Los procesos pueden contener uno o más hilos, haciendo más eficiente las tareas, asimismo la complejidad de los algoritmos de sincronización, ya que podría ocurrir la condición de carrera muy a menudo, inclusive los indeseados interbloqueos.

## Índice

**Creación de un proceso**

**Terminación de un proceso**

**Estados de un proceso**

**Tipos de procesos**

**Véase también**

**Referencias**

**Bibliografía**

## Creación de un proceso

---

Hasta el día de hoy existen sólo 4 formas de crear un proceso:

- Arranque del sistema.
- En la ejecución, desde un proceso, de una llamada al sistema para la creación del proceso.
- Una petición deliberada del usuario para crear un proceso.
- El inicio de un trabajo por lotes.

La forma de creación de procesos en Unix es a través de una llamada al sistema fork la cual creará un proceso hijo en total semejanza al padre, hasta que el recién proceso decida cambiar su imagen en memoria, incluso obtener sus propios descriptores de archivos abiertos.

## Terminación de un proceso

---

El ciclo de vida de un proceso es fácil, depende de la creación, la ejecución de instrucciones y la terminación. Cabe señalar que un proceso en el transcurso de su ciclo puede estar en diferentes estados.

- Salida normal.
- Salida por error.
- Error fatal.
- Eliminado por otro proceso.

## Estados de un proceso

---

Los estados de un proceso obedecen a su participación y disponibilidad dentro del sistema operativo y surgen de la necesidad de controlar la ejecución de cada proceso. Los procesadores sólo pueden ejecutar un solo proceso a la vez, turnándolos para el uso de este. Existen procesos no apropiativos o cooperativos que básicamente ocupan todo el tiempo del procesador hasta que ellos deciden dejarlo. Los procesos apropiativos son aquellos que ocupan por un período de tiempo el procesador hasta que una interrupción o señal llega al procesador para hacer el cambio de proceso, a esto se le conoce como cambio de contexto.

Los posibles estados que puede tener un proceso son ejecución, bloqueado y listo:

- Ejecución, es un proceso que está haciendo uso del procesador.
- Bloqueado, No puede ejecutarse hasta que un evento externo sea llevado a cabo.
- Listo, ha dejado disponible al procesador para que otro proceso pueda ocuparlo.

Las posibles transiciones son 4. La primera se realiza cuando el sistema operativo determina que el proceso no puede continuar justo en ese momento, en algunos sistemas se puede hacer una llamada al sistema "pause" para pasar al estado bloqueado, en Unix cuando el proceso está leyendo datos provenientes de una canalización o de un archivo especial (terminal) y no hay entrada disponible, el proceso se bloquea de forma automática.

Las transiciones 2 y 3 son llevadas a cabo por el planificador de procesos, siendo que el proceso no tiene conocimiento de este. La transición 2 se da cuando el planificador de procesos decide que el proceso ya estuvo el tiempo suficiente en ejecución y debe dar paso a la ejecución de otros procesos (adquieran tiempo del procesador). La transición 3 se realiza cuando todos los procesos han ocupado tiempo del procesador y debe retomarse el primer proceso.

La transición 4 ocurre cuando se produce un evento externo por el que un proceso estaba en espera, por ejemplo, introducir datos desde la terminal. Si no hay otro proceso en ejecución en ese instante, la transición 3 se activa y el proceso comienza a ejecutarse; también podría pasar al estado de "listo" y esperar un momento para iniciar la ejecución.

## Tipos de procesos

---

Existen dos tipos de procesos, aquellos que se ejecutan en modo kernel y aquellos que se ejecutan en modo usuario. Los primeros son más lentos por las llamadas al sistema que realizan, sin embargo, son más seguros por la integridad que representan. Cuando hablamos de los procesos de usuario, podemos decir que el sistema operativo podría no ser multiproceso, ya que se vale de librerías (como pthread) para hacer un multiplexado y dar la apariencia de trabajar como multiproceso.

Podría pensarse en otra clasificación, como son los procesos en primer plano y procesos en segundo plano. Los primeros interactúan con el usuario, es decir, el usuario proporciona los datos que el proceso utilizará. Los segundos, son creados para tareas bien definidas y no necesitan la intervención del usuario, por ejemplo, se puede tener un proceso en segundo plano para revisar la temperatura del disco duro constantemente, éstos también son conocidos como demonios.<sup>[*cita requerida*]</sup><sup>[2]</sup>

## Véase también

---

- Memoria virtual
- Multiproceso
- Unix
- Paginación

## Referencias

---

1. Stallings, William (2005). *Sistemas operativos: aspectos internos y principios de diseño* (5ª edición). Pearson Prentice Hall. p. 109. ISBN 978-84-205-4462-5.
2. Jesús., Niño Camazón, (1 de enero de 2011). «Gestión de procesos» (<https://www.worldcat.org/oclc/852653978>). *Sistemas operativos monopuseto*. Editex. p. p. 215. ISBN 8497719719. OCLC 852653978 (<https://www.worldcat.org/oclc/852653978>).

## Bibliografía

---

- Tanenbaum, Andrew S. (2009). *Sistemas operativos modernos* (3ª edición). Pearson Prentice Hall. ISBN 978-607-442-046-3.

---

Obtenido de «[https://es.wikipedia.org/w/index.php?title=Proceso\\_\(informática\)&oldid=129111949](https://es.wikipedia.org/w/index.php?title=Proceso_(informática)&oldid=129111949)»

---

Esta página se editó por última vez el 7 sep 2020 a las 21:27.

El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; pueden aplicarse cláusulas adicionales. Al usar este sitio, usted acepta nuestros términos de uso y nuestra política de privacidad. Wikipedia® es una marca registrada de la Fundación Wikimedia, Inc., una organización sin ánimo de lucro.