



Centro Universitario de Ciencias Exactas e Ingenieras.

Departamento para la integración ciber-humana.

Sistemas operativos.

Becerra Velázquez Violeta Rocío.

**Yáñez Salazar Saul Emanuel.**

220656484.

Ingeniería en computación.

D04.

**Administración de procesos.**

Domingo de 9 febrero de 2025.

## Creación de Procesos

Los procesos pueden ser creados en diversas situaciones, entre las cuales destacan:

**Inicialización del Sistema:** Al arrancar el sistema operativo, se generan varios procesos esenciales para su funcionamiento.

**Llamadas al Sistema desde un Proceso Existente:** Un proceso en ejecución puede invocar una llamada al sistema para crear otro proceso. Por ejemplo, en sistemas UNIX, la llamada `fork()` permite que un proceso padre genere un proceso hijo.

**Solicitud del Usuario:** Los usuarios pueden iniciar nuevos procesos, ya sea mediante la ejecución de comandos en una terminal o al interactuar con interfaces gráficas.

**Inicio de Trabajos por Lotes:** En sistemas de procesamiento por lotes, la creación de procesos se desencadena al iniciar un nuevo trabajo. (FING, s.f.)

## Terminación de Procesos

La finalización de un proceso puede ocurrir por diversas razones, entre las cuales se incluyen:

**Terminación Normal (Voluntaria):** El proceso completa su ejecución y finaliza de manera natural.

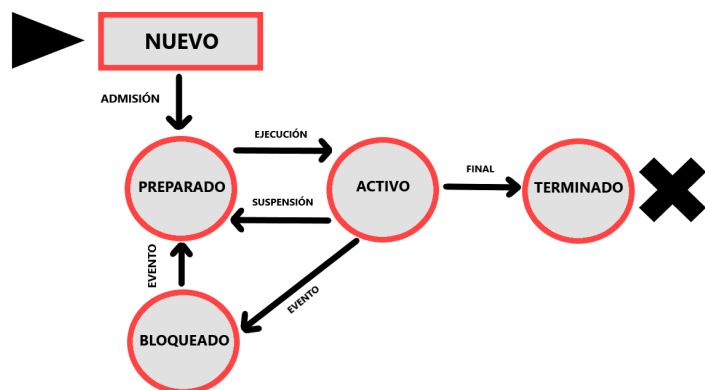
**Terminación por Error (Voluntaria):** El proceso detecta una condición de error y decide finalizar.

**Error Fatal (Involuntaria):** El proceso encuentra un error crítico que le impide continuar, como una instrucción ilegal o una referencia de memoria inválida.

**Terminación por Otro Proceso (Involuntaria):** Un proceso puede ser finalizado por otro, generalmente su proceso padre, mediante llamadas al sistema como `kill` en UNIX o `TerminateProcess` en Windows.

### Diagrama de procesos.

Los procesos pueden pasar por 5 estados, el primero es preparado, después de esto se puede ir activo para ejecutarse, y este a su vez puede regresar para un estado de suspensión, estando activo el proceso se pueden bloquear, y una vez que el bloqueo termine puede volver al



estado de preparado, y, por último, estando en el estado activo, este proceso se puede terminar.

### **Algoritmos de planificación.**

Los algoritmos de planificación en sistemas operativos determinan cómo se asigna el tiempo de CPU entre los procesos.

**Primero en Llegar, Primero en Servir (FCFS):** Atiende los procesos en orden de llegada. Puede generar tiempos de espera altos si un proceso largo precede a otros más cortos

**Trabajo Más Corto Primero (SJF):** Prioriza los procesos con menor tiempo estimado de ejecución. Minimiza el tiempo de espera promedio, pero requiere conocer la duración de los procesos de antemano

**Tiempo Restante Más Corto (SRTF):** Variante apropiativa de SJF, en la que un proceso puede ser interrumpido si llega otro con menor tiempo restante de ejecución

**Round Robin (RR):** Asigna un intervalo de tiempo fijo (**quantum**) a cada proceso y lo intercambia en cola si no finaliza. Es eficiente en sistemas de tiempo compartido, pero el tamaño del quantum afecta el rendimiento

**Planificación por Prioridades:** Asigna prioridades a los procesos y ejecuta los de mayor prioridad primero. Puede ser apropiativa o no. Si no se gestiona bien, puede causar inanición de procesos con baja prioridad.

**Colas Multinivel:** Separa procesos en colas según prioridad o tipo de proceso. Cada cola puede tener su propio algoritmo de planificación. (Silberschatz, 2018) (Stallings, 2018)

### ***Definición de Algoritmos de planificación***

Los algoritmos de planificación en sistemas operativos son estrategias utilizadas para administrar la ejecución de procesos dentro de un sistema computacional, con el objetivo de optimizar el uso de los recursos y garantizar un rendimiento eficiente. Estos algoritmos deciden el orden en que los procesos acceden a la CPU y otros recursos, considerando criterios como equidad, tiempo de respuesta, eficiencia y prioridad. (Morales, 2016)

### **Métodos de planificación de procesos.**

#### **Planificación No Apropiativa**

En esta política, un proceso que obtiene la CPU la mantiene hasta que termine o entre en estado de espera (por ejemplo, esperando entrada/salida). La CPU no puede ser retirada de un proceso en ejecución para asignársela a otro, incluso si un proceso con mayor prioridad llega al sistema.

**Ventajas:**

Más simple de implementar.

No requiere una gestión compleja de interrupciones.

Reduce el overhead del cambio de contexto.

**Desventajas:**

Puede llevar a tiempos de espera elevados para procesos cortos si un proceso largo se ejecuta primero (como en **FCFS**).

No es adecuada para sistemas en tiempo real, ya que procesos urgentes pueden quedar bloqueados esperando.

**Planificación Apropiativa**

En esta política, la CPU puede ser retirada de un proceso en ejecución y asignada a otro proceso más prioritario. Esto permite que el sistema responda dinámicamente a nuevas tareas o cambios en las prioridades.

**Ventajas:**

Mayor capacidad de respuesta en sistemas multitarea.

Es adecuada para sistemas en tiempo real y entornos interactivos.

Evita la inanición de procesos al permitir la interrupción de procesos largos.

**Desventajas:**

Mayor overhead debido a los cambios de contexto frecuentes.

Puede generar mayor complejidad en la implementación del sistema operativo. (Morales, 2016).

Algoritmo	Descripción	Política	Ventaja.	Desventaja.
<b>Round Robin</b>	Asigna a cada proceso un intervalo de tiempo fijo (quantum) y alterna entre ellos en orden circular	Apropiativa	Garantiza equidad y tiempos de repuesta adecuados en sistemas interactivos.	Si el quantum es muy corto, aumenta el overhead por cambios de contexto.

<b>First-come, first-served.,</b>	Atiende los procesos en el orden de ejecución.	No apropiativa.	Simple de implementar y justo para los procesos en orden de llegada.	Pueden causar problemas de convoy effect, donde procesos cortos esperar a que terminen procesos largos.
<b>Shortest remaining time.</b>	Variante de SJF que selecciona el proceso con el menor tiempo restante de ejecución.	Apropiativa.	Minimiza el tiempo de espera promedio.	Puede provocar inanición de procesos largos si llegan muchos procesos cortos.
<b>Shortes Job First.</b>	Ejecuta el primero el proceso con el menor tiempo de ejecución estimado.	No apropiativa	Minimiza el tiempo promedio de ejecución.	Difícil de implementar en la práctica, ya que requiere conocer la duración de los procesos de antemano.
<b>Planificación por prioridad.</b>	Asigna prioridades a los procesos y ejecuta primero los de mayor prioridad.	Puede ser apropiativa o no apropiativa.	Permite ejecutar primero tareas críticas o importantes.	Puede causar inanición en procesos de baja prioridad si no se implementa envejecimiento.
<b>Colas múltiples.</b>	Divide los procesos en varias colas según su prioridad o tipo.	Puede ser apropiativa, o no apropiativa.	Permite gestionar distintos tipos de procesos de manera adecuada.	Complejo de administrar y configurar adecuadamente.
<b>Planificación por lotería.</b>	Asigna tickets a los procesos, y selecciona aleatoriamente cual se ejecutará, con mayor probabilidad procesos con más tickets.	Apropiativa	Proporciona equidad probabilística en la asignación de la CPU.	Puede ser determinista y generar tiempos de espera variables.

<b>Fair-share scheduling.</b>	Distribuye el tiempo de CPU equivalente entre usuarios o grupos en lugar de entre procesos individuales.	Puede ser apropiativa o no apropiativa.	Garantiza que un usuario con muchos procesos no monopolice la CPU.	Puede penalizar a usuarios con más procesos exigentes, afectando su rendimiento.
-------------------------------	--	---	--	--

(Planificacoin de procesos. , s.f.)

La Tabla de Procesos o Bloque de Control de Procesos es una estructura de datos clave en los sistemas operativos, que almacena toda la información necesaria para gestionar y controlar un proceso.

### Elementos del BCP

- **Identificación del proceso (PID):** Un identificador único asignado al proceso.
- **Estado del proceso:** Indica si el proceso está **nuevo, listo, en ejecución, bloqueado o terminado**.
- **Contador de programa (PC):** Dirección de la próxima instrucción a ejecutar.
- **Registros de CPU:** Incluye registros generales, de pila, de estado, etc., que deben guardarse y restaurarse en cambios de contexto.
- **Espacio de direcciones de memoria:** Información sobre la memoria asignada al proceso (segmentación, paginación, tablas de páginas, etc.).
- **Información de planificación:** Incluye prioridad, colas de planificación y otros datos utilizados por el planificador.
- **Información de gestión de memoria:** Detalles sobre segmentos de código, datos y pila del proceso.
- **Información de entrada/salida (E/S):** Dispositivos de E/S asignados, archivos abiertos, permisos y buffers.
- **Información sobre comunicación entre procesos (IPC):** Canales, mensajes y señales usadas en la sincronización entre procesos.
- **Tiempo de CPU utilizado:** Registro del tiempo de procesamiento consumido.
- **Identificadores de procesos padre e hijos:** Relación jerárquica entre procesos en el sistema.
- **Recursos asignados:** Información sobre semáforos, archivos y dispositivos utilizados. (Ginzo, 2021)

### Describe en qué consisten los algoritmos de planificación No Apropiativos.

Los algoritmos de planificación no apropiativos son aquellos en los que una vez que un proceso obtiene el control de la CPU, no puede ser interrumpido hasta que este mismo finalice su ejecución o realice una operación que lo bloquee como una entrada/salida.

### **Cálculos sobre tiempos en procesos.**

Tiempo de llegada: Este tiempo es en el que proceso entra al sistema, generalmente como parte de los datos de entrada.

Tiempo de finalización: Es el tiempo en el que el proceso termina su ejecución  $t$ , también suele ser proporcionado. Pero en caso de que no, se puede resolver con la ecuación del tiempo de retorno. Donde:

$$T_f = T_r + T_L$$

Tiempo de retorno: Representa el tiempo total que el proceso paso en el sistema, se puede calcular como:

$$T_{\text{retorno}} = T_{\text{finalización}} - T_{\text{Llegada}}$$

Tiempo de espera: Es el tiempo que el proceso ha estado en la cola de listos sin ejecutarse, obtiene restando el tiempo de servicio al tiempo de retorno.

$$T_{\text{espera}} = T_{\text{retorno}} - T_{\text{servicio}}$$

Tiempo de servicio: Es el tiempo total que el proceso ha estado usando la CPU. Se puede calcular sumando todos los periodos en los que el proceso estuvo en ejecución.

### **¿Qué significa BCP?**

BCP significa Bloque de Control de Procesos. Es una estructura de datos que almacena toda la información relevante sobre un proceso en ejecución dentro de un sistema operativo. El BCP es fundamental para gestionar y controlar el ciclo de vida de un proceso sin perder datos importantes.

### **Conclusiones.**

La distribución y administración de la ejecución de procesos en un sistema operativo es un pilar fundamental para garantizar un uso eficiente y equitativo de los recursos computacionales. Dado que la capacidad de procesamiento es limitada y los sistemas deben gestionar múltiples procesos simultáneamente, es imprescindible contar con estrategias y mecanismos que regulen el ciclo de vida de cada proceso.

La planificación de procesos juega un papel crucial en la asignación del tiempo de CPU, priorizando tareas en función de criterios como tiempo de llegada, prioridad, tiempo de servicio y requerimientos del sistema. Algoritmos de planificación como FCFS, SJF, Round Robin, y Planificación por Prioridades permiten distribuir los procesos de manera organizada, evitando bloqueos y optimizando el rendimiento del sistema.

Además, la administración de la memoria y el manejo de interrupciones son aspectos esenciales en la ejecución de procesos. El sistema operativo debe garantizar que cada proceso tenga acceso a la memoria necesaria sin interferencias, empleando técnicas como la paginación y segmentación para optimizar el uso de los recursos. Asimismo, el manejo adecuado de interrupciones y cambios de contexto permite una transición fluida entre procesos, minimizando la latencia y maximizando la eficiencia del CPU

## Bibliografía

- FING. (s.f.). *Procesos: descripcion y control*. . Obtenido de Sistemas operativos:  
<https://www.fing.edu.uy/inco/cursos/sistoper/recursosTeoricos/5-SO-Teo-Procesos.pdf>
- Ginzo. (24 de Junio de 2021). *Bloque de control del proceso*. . Obtenido de Ginzo Tech:  
<https://ginzo.tech/bcp-bloque-control-proceso/>
- Morales, J. (21 de Septiembre de 2016). *Planificacion de procesos*. . Obtenido de  
<http://jmoral.es/blog/planificacion-procesos>:  
<http://jmoral.es/blog/planificacion-procesos>
- Planificacoin de procesos*. . (s.f.). Obtenido de Sistop:  
[https://sistop.gwolf.org/html/03\\_planificacion\\_de\\_procesos.html](https://sistop.gwolf.org/html/03_planificacion_de_procesos.html)
- Silberschatz, A. G. (2018). *Operating System Concepts(10th ed)*. Wiley.
- Stallings, W. (2018). *Operating Systems: Internal and Design Principles (9th ed)* . Pearson. .



