

Universidad de Guadalajara  
CUCEI (Ciencias Exactas e Ingenierías)  
Departamento de ciencias computacionales  
Materia: Sistemas operativos  
Profesor: Violeta del Rocío Becerra Velazquez

**Gómez Rubio Alexia**

Código:219551644  
Carrera: Ingeniería en computación  
Sección: D04

Tarea 3  
Tema: Funcionamiento de un sistema operativo  
Fecha: 15 Septiembre 2024

## Tabla de contenido

|   |    |
|---|----|
| Creación de procesos .....                    | 3  |
| Terminación de procesos.....                  | 3  |
| Diagrama de 5 estados (imagen).....           | 5  |
| Definición de cada estado .....               | 6  |
| Transiciones válidas de los estados .....     | 6  |
| Algoritmos de Planificación.....              | 7  |
| Políticas de planificación .....              | 7  |
| Tabla de algoritmos .....                     | 8  |
| Elementos de la tabla de procesos (BCP) ..... | 10 |
| Preguntas.....                                | 11 |
| Conclusión .....                              | 12 |
| Referencias bibliográficas.....               | 12 |

## Creación de procesos

Cuando se añade un proceso a los que ya está administrando el sistema operativo, hay que construir las estructuras de datos que se utilizan para gestionar y controlar el proceso y asignar el espacio de direcciones que va a utilizar dicho proceso. Estas acciones constituyen la creación de un nuevo proceso.

Al proceso que solicita el servicio Create-Process se le denomina proceso padre, y al proceso que es creado mediante este servicio, proceso hijo. Este mecanismo de generación de procesos tiene como consecuencia que las relaciones de parentesco entre los procesos existentes en un sistema tengan estructura de árbol.

Las formas más comunes de la creación de procesos son:

- ❖ **Nueva tarea en un sistema de proceso por lotes:** El sistema operativo está provisto de un flujo de control de trabajos por lotes, generalmente para cinta o disco. Cuando el sistema operativo se prepara para tomar un nuevo trabajo, leerá la próxima secuencia de órdenes de control de trabajos.
- ❖ **Nueva conexión interactiva:** Un usuario entra en el sistema desde una terminal.
- ❖ **Comandos del usuario:** los usuarios pueden crear nuevos procesos por medio de comandos en las terminales y ejecutarlos.
- ❖ **Nuevo proceso creado por el SO para dar un servicio:** El sistema operativo puede crear un proceso para llevar a cabo una función de parte de un programa usuario, sin que el usuario tenga que esperar (por ejemplo, imprimir).
- ❖ **Un proceso generado por otro ya existente:** Con afán de modularidad o para aprovechar el paralelismo, un programa de usuario puede ordenar la creación de una serie de procesos.

## Terminación de procesos

Un trabajo por lotes debe incluir una instrucción de detención halt o una llamada explícita a un servicio del sistema operativo para indicar su terminación. Un proceso puede terminar por sí mismo, o bien puede ser terminado por otro proceso, que generalmente sólo puede ser su proceso padre, por decisión del usuario o porque tiene un error.

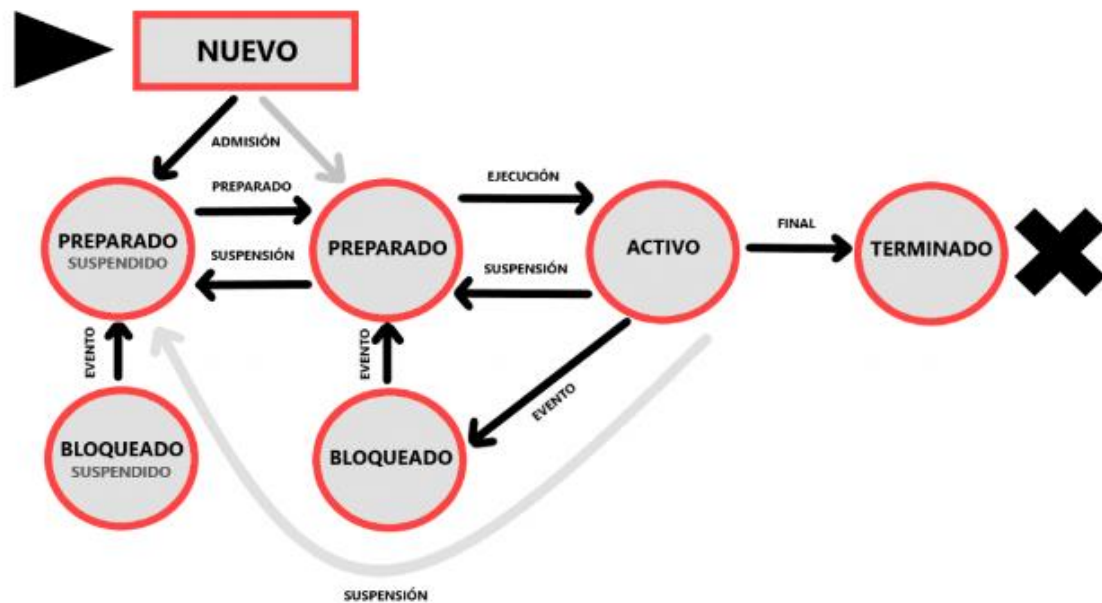
Todas estas acciones provocan al final una petición de servicio al SO para terminar el proceso demandante. Además, una serie de errores pueden llevarnos a la terminación de un proceso.

Las formas más comunes de la terminación de procesos son:

- ❖ **Terminación normal:** Un proceso termina de ejecutar su conjunto de instrucciones y finaliza.
- ❖ **Tiempo límite excedido:** El proceso requiere más tiempo para completar su ejecución del que el sistema establece como máximo.

- ❖ **No disponibilidad de memoria:** Un proceso necesita más memoria de la que el sistema puede proporcionar.
- ❖ **Violación de límites:** Ocurre cuando un proceso trata de acceder a una posición de memoria a la que no puede hacerlo.
- ❖ **Error de protección:** Un proceso intenta utilizar un recurso o un archivo para el que no tiene permiso o trata de utilizarlo de forma incorrecta.
- ❖ **Decisión del usuario:** El usuario puede terminar el programa en cualquier momento.
- ❖ **Error aritmético:** Aparece si el proceso intenta hacer un cálculo prohibido como la división por cero o trata de almacenar un número mayor del que el hardware acepta.
- ❖ **Superación del tiempo máximo de espera por un recurso:** En este caso, el proceso se encuentra a la espera de obtener un recurso o de que tenga lugar un determinado evento durante un tiempo que alcanza el límite establecido.
- ❖ **Fallo de dispositivo I/O:** Se produce por un error en la entrada o la salida, tal como la incapacidad de encontrar un archivo o la ocurrencia de un fallo de lectura o escritura después de un número máximo de intentos.
- ❖ **Instrucción no válida:** Se produce si un proceso intenta ejecutar una instrucción inexistente.
- ❖ **Intento de acceso a una instrucción privilegiada:** Se presenta si un proceso intenta utilizar una instrucción reservada para el SO.
- ❖ **Mal uso de los datos:** Un elemento de dato no está inicializado o es de un tipo equivocado para la operación que se pretende realizar.
- ❖ **Intervención del operador o del SO:** Por alguna razón, el operador o el SO termina con el proceso. Por ejemplo, si se considera comprometido el rendimiento del sistema.
- ❖ **Finalización del proceso padre:** Cuando un proceso padre finaliza el SO puede diseñarse para terminar automáticamente con todos sus descendientes.
- ❖ **Solicitud del proceso padre:** Un proceso padre tiene normalmente autoridad para terminar con cualquiera de sus hijos.

Diagrama de 5 estados (imagen)



### Definición de cada estado

- ❖ **Estado Nuevo:** este estado corresponderá a procesos que acaban de ser definidos pero que aún no han sido admitidos por el sistema operativo como procesos ejecutables. Para estos procesos se habrán realizado ciertas tareas de gestión interna como la asignación de un identificador y la creación de algunas estructuras de control.
- ❖ **Estado Listo:** En este estado se encontrarán aquellos procesos que dispongan de todos los recursos necesarios para comenzar o proseguir su ejecución y se encuentran a la espera de que se les conceda el control del procesador, ya fueron admitidos/aceptados, pero todavía no hacen nada. (pertenece a la memoria principal).
- ❖ **Estado de Ejecución:** En este estado se encuentra el proceso que tiene el control del procesador. Dado que se considerarán arquitecturas que disponen de un único procesador, en un instante determinado sólo un proceso puede encontrarse en este estado. (pertenece a la memoria principal).
- ❖ **Estado Bloqueado:** En este estado se encuentran aquellos procesos que carecen de algún recurso necesario para su ejecución, siendo este recurso distinto del procesador, o bien se encuentran a la espera de que tenga lugar un determinado evento. (pertenece a la memoria principal).
- ❖ **Estado Terminado:** A este estado pertenecen aquellos procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso alcanza este estado cuando llega al punto normal de terminación, cuando se abandona debido a un error irreparable o cuando un proceso con la debida autoridad hace que termine su ejecución. En este punto, el proceso ya no es susceptible de ser elegido para ejecutarse. Sin embargo, el SO conserva cierta información asociada con él para su posible utilización, bien por otras aplicaciones como programas de utilidad para el análisis de la historia y rendimiento del proceso, o bien por parte del SO con fines estadísticos. Una vez extraída esta información, el SO ya no necesita mantener más datos relativos al proceso y éstos se borran del sistema.
- ❖ **Listo y suspendido:** Son aquellos procesos que tuvieron algún error, pero están listos para regresar a listos y continuar con su proceso para terminar su ejecución. (pertenece a la memoria secundaria)
- ❖ **Bloqueado y suspendido:** son procesos que tuvieron que salir de su proceso, pero esperan que se resuelva la interrupción para regresar a bloqueo y de ahí a listos para continuar con su proceso y terminar la ejecución. (pertenece a la memoria secundaria).

### Transiciones válidas de los estados

- ❖ **Transición a Nuevo:** Se crea un nuevo proceso para ejecutar un programa.

- ❖ **Transición Nuevo-Listo:** Esta transición tiene lugar cuando el SO está preparado para aceptar o admitir un proceso más. Se tendrán en cuenta las restricciones derivadas de la capacidad de la memoria y que no haya tantos procesos activos como para degradar el rendimiento.
- ❖ **Transición Listo-Ejecución:** Esta transición se produce cuando el SO selecciona un nuevo proceso para ejecutar en función de su política de planificación.
- ❖ **Transición Ejecución-Listo:** La razón más común para esta transición es que el proceso que está en ejecución ha alcanzado el tiempo máximo permitido de ejecución ininterrumpida. Hay otras causas alternativas que no están implementadas en todos los SO como la expropiación de un proceso en favor de otro más prioritario. Otra situación, muy extraordinaria, que origina esta transición es que un proceso ceda voluntariamente el control del procesador.
- ❖ **Transición Ejecución-Bloqueo:** Un proceso realiza esta transición cuando queda a la espera por la concesión de un determinado recurso o por la ocurrencia de un determinado suceso.
- ❖ **Transición Bloqueado-Listo:** Tiene lugar si a un proceso bloqueado se le concede el recurso solicitado u ocurre el suceso por el que estaba esperando.
- ❖ **Transición Listo-Terminado:** Puede ocurrir si, por ejemplo, un proceso padre decide en un momento determinado finalizar la ejecución de sus procesos hijos. Si alguno de dichos procesos se encontraba en estado preparado realizará esta transición. Otra razón puede ser debido a un requisito de memoria que es denegado.
- ❖ **Transición Bloqueado-Terminado:** Un proceso hijo puede realizar esta transición por la misma razón que la anterior. Otra causa puede ser que el proceso supere el tiempo máximo de espera por un recurso y el sistema operativo decida entonces terminarlo (es la razón más habitual)

## Algoritmos de Planificación.

Un algoritmo de planificación determina cómo se planifican los programas de aplicación para manejar la carga en las colas. El enfoque básico para controlar un sistema IMS en línea con colas cargadas es permitir que la demanda controle la planificación de programas en un número razonable de regiones de mensajes. Estos algoritmos se utilizan para calcular los recursos que consume otro algoritmo o conjunto de algoritmos (programa) al realizar una determinada tarea.

## Políticas de planificación

**Apropiativas:** En los algoritmos de planificación apropiativa, un proceso puede ser interrumpido y retirado de la CPU incluso si no ha terminado, para dar lugar a otro proceso de mayor prioridad o necesidad.

**No Apropiativas:** En estos algoritmos, una vez que un proceso obtiene la CPU, no puede ser interrumpido hasta que finalice por completo o entre en estado de espera.

## Tabla de algoritmos

| Algoritmos                      | Descripción   | Política       | Ventaja   | Desventaja  |
|---------------------------------|---|----------------|---|---|
| Round Robin (RR)                | Es un algoritmo de planificación que se utiliza en sistemas operativos y aplicaciones para gestionar la ejecución de procesos o tareas. Su enfoque principal es distribuir el tiempo de CPU de manera justa entre todas las tareas pendientes.                      | Apropiativa    | Facilidad en el manejo<br>Las propiedades no cambian<br>Respuesta rápida  | Procesos de duración variable<br>Después de cierto tiempo el algoritmo se vuelve complejo                                 |
| First come, first server (FCFS) | Este algoritmo sigue la regla de los procesos son ejecutados en el orden que llegan a la cola de procesos listos, una vez que el procesador le es asignado a un proceso este lo mantiene hasta que termina o se bloquea (por ejemplo, al generar un pedido de E/S). | No apropiativa | Fácil de implementar<br>Es útil en sistemas pequeños<br>No puede ser interrumpido una vez que comienza con el proceso | Tarda mucho en completar sus procesos<br>Este efecto da como resultado una menor utilización de la CPU y del dispositivo. |
| Shortest Remaining Time (SRT)   | El planificador siempre elige el proceso que tiene el tiempo restante de procesamiento esperado más corto.  | Apropiativa    | Ofrece un buen tiempo de respuesta<br>No produce overhead<br>Produce tiempo de retorno                                | Puede ser algo tardado<br>Es complejo de configurar<br>No tiene control de retorno  |
| Shortest job fist (SJF)         | Este algoritmo se basa en una política de programación que selecciona el proceso de espera con el menor tiempo de ejecución para ejecutar a continuación.   | No apropiativo | Escoge el ciclo más pequeño<br>Se puede comprobar si el algoritmo es optimo   | No se puede implementar a nivel de programación de CPU a corto plazo.   |



|  |   |                                   |   |   |
|--|---|-----------------------------------|---|---|
|  |   |                                   |   | Favorece a los procesos más cortos  |
| Prioridades  | En este tipo de planificación a cada proceso se le asigna una prioridad siguiendo un criterio determinado, y de acuerdo con esa prioridad será el orden en que se atienda cada proceso.   | Apropiativa                       | Prioriza procesos críticos<br><br>Los procesos de alta prioridad se pueden ejecutar indefinidamente                             | Puede causar inanición para procesos de baja prioridad  |
| Colas múltiples  | Este algoritmo permite que los procesos se puedan dividir en diferentes clases donde cada clase tiene sus propias necesidades de programación.  | Apropiativa<br><br>No apropiativa | Soporta la sobrecarga<br><br>Adaptable a las necesidades del sistema  | Complejo de implementar<br><br>Difícil de ajustar a los parámetros entre colas  |
| Programación de colas de retroalimentación multinivel (MFQS) | La programación de CPU es como la programación de cola multinivel (MLQ), pero en este proceso puede moverse entre las colas, es decir los procesos se asignan permanentemente a una cola al ingresar al sistema y no se permite que los procesos se muevan entre colas. | Apropiativa                       | Bajo costo de programación<br><br>Permite que diferentes procesos se muevan entre diferentes colas.                             | No es flexible<br><br>También requiere algún medio de selección de valores para todos los parámetros para definir el mejor planificador |
| Completely Fair Scheduler (CFS)                              | es un algoritmo planificador desarrollado con la meta de maximizar el uso de la CPU con las diferentes tareas que se lanzan en un sistema Linux   | Apropiativa                       | Proporciona equidad total entre procesos<br>mantiene baja latencia.<br><br>Adaptabilidad<br><br>Soporte para entornos multicore | Puede penalizar procesos de CPU intensiva en favor de interactivos.<br><br>Complejo en la implementación                                |

## Elementos de la tabla de procesos (BCP)

El BCP es clave ya que permite que el sistema operativo de soporte a múltiples procesos, así como también proporcionar multiprogramación.

- ❖ **Identificación del proceso** (Process Identifier, sus siglas en inglés PID): es única para el proceso y es lo que permite distinguirlo del resto de procesos, pudiendo almacenar el identificador del proceso padre que lo creó y el identificador del usuario, dependiendo del sistema.
- ❖ **Estado del proceso**: si el proceso está actualmente en ejecución o en otro estado, como listo, suspendido, parado o zombi. Los estados varían según el tipo de sistema operativo.
- ❖ **Información de planificación**: se refiere a cómo de prioritarios son el resto de los procesos, y es el evento por el cual el proceso va a esperar mientras está bloqueado. Solo en caso de que se use dicho algoritmo para planificación de CPU.
- ❖ **Punteros a memoria**: en estos se incluyen todos los punteros al código del programa y datos referentes a este proceso, cualquier bloque de memoria compartido con otros procesos o si el proceso usa memoria virtual. A su vez almacena punteros a la pila y el montículo del proceso. Así que describe los segmentos de memoria que han sido asignados al proceso, mostrando los espacios de direcciones o los límites de memoria asignados al proceso.
- ❖ **Datos de contexto**: se encuentran presentes en el registro del procesador mientras el proceso este en ejecución. Esto permite almacenar el valor de todos los registros del procesador, las banderas de estado, señales, entre otras. En resumen, corresponden a todos los datos necesarios para poder ejecutar un proceso en el momento que el sistema operativo lo indique.
- ❖ **Información de estado E/S y recursos asignados**: aquí se incluyen las peticiones de E/S pendientes, dispositivos de E/S asignados a ese proceso, una lista de ficheros en uso, entre otros. Así como una lista de recursos asignados donde involucran descriptores de archivos y sockets abiertos.
- ❖ **Comunicación entre procesos**: cuenta con distintos indicadores, señales y mensajes relacionados con la comunicación entre dos procesos independientes.
- ❖ **Información de auditoría**: aquí es posible incluir la cantidad de tiempo de procesador y de tiempo de reloj utilizados, así como también los límites de tiempo, registros contables, entre otros como las estadísticas del proceso y valores de registro de CPU que también es usado en el cambio de contexto. Y los datos del propietario (owner).
- ❖ **Signals pendientes**: que se encuentren almacenados en un mapa de bits.

## Preguntas

### 1. Describa en qué consisten los algoritmos de planificación No Apropiativos.

Son los algoritmos que no permiten la expulsión de procesos para ejecutar uno nuevo, a menos de que tenga un error o lo interrumpan forzosamente.

### 2. Explique cómo obtener cada uno de los tiempos solicitados en el punto 9 del programa 3 (actividad de aprendizaje 6).

**a. Tiempo de Llegada:** Hora en la que el proceso entra al sistema.

\*No tiene formula, pero se considera desde que empieza el proceso.

**b. Tiempo de Finalización:** Hora en la que el proceso termino.

\*No tiene formula, pero es cuando se termina el proceso.

**c. Tiempo de Retorno:** Tiempo total desde que el proceso llega hasta que termina.

$$TR = TF - TL$$

**d. Tiempo de Respuesta:** Tiempo transcurrido desde que llega hasta que es atendido por primera vez.

\*Tiempo en que tarda en ser ejecutado.

**e. Tiempo de Espera:** Tiempo que el proceso ha estado esperando para usar el procesador.

$$TE = TR - TS$$

**f. Tiempo de Servicio:** Tiempo que el proceso ha estado dentro del procesador. (Si el proceso termino su ejecución normal es el TME, de no ser así es el tiempo transcurrido)

$$TS = TR - TE$$

### 3. ¿Qué significa BCP?

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 en particular o que está en ejecución, permite al sistema operativo gestionar y controlar los procesos de manera eficiente.

## Conclusión

El diagrama de los cinco estados no lo conocía, pero con la explicación de la maestra y la investigación que realice me quedo mas claro y pude entender mejor cada una de las fases, de igual forma el como se clasifican los algoritmos si son apropiativos (los procesos que pueden ser obligados a salir) y los no apropiativos (los que no pueden ser obligados a salir a menos de que sean forzados a terminar o tengan un erro).

El bloque de control de procesos es muy importante para agrupar la información de los procesos, permitiendo al sistema operativo gestionar y controlar los procesos de manera eficiente.

## Referencias bibliográficas

de Luis R Castellanos, V. T. las E. (2015, February 3). *03.03. Algoritmos de planificación*. Sistemas Operativos.

<https://lcsistemasoperativos.wordpress.com/2015/02/03/03-03-algoritmos-de-planificacion/>

*Tema 2.2 Creación y terminación de procesos - Sistemas Operativos - Instituto*

*Consortio Clavijero*. (n.d.). Edu.Mx. Retrieved August 29, 2024, from

[https://cursos.clavijero.edu.mx/cursos/182\\_so/modulo2/contenidos/tema2.2.html?opc=1](https://cursos.clavijero.edu.mx/cursos/182_so/modulo2/contenidos/tema2.2.html?opc=1)

*Tema 2.3.2 Modelo de cinco estados - Tema 2.3 Estados de un proceso - Sistemas*

*Operativos - Instituto Consortio Clavijero*. (n.d.). Edu.Mx. Retrieved August

29, 2024, from

[https://cursos.clavijero.edu.mx/cursos/182\\_so/modulo2/contenidos/tema2.3.2.html](https://cursos.clavijero.edu.mx/cursos/182_so/modulo2/contenidos/tema2.3.2.html)

*Políticas de planificación de procesos. Ventajas y desventajas.* (n.d.). Blogspot.com.

Retrieved August 29, 2024, from <https://gestion-de-procesos-informaticos.blogspot.com/2017/06/politicas-de-planificacion-de-procesos.html>

Ginzo Technologies. (2021, June 24). BCP o Bloque de control del proceso. *GINZO*

*TECHNOLOGIES SL.* <https://ginzo.tech/bcp-bloque-control-proceso/>