



SISTEMAS OPERATIVOS

ACTIVIDAD DE APRENDIZAJE 5

TAREA 3

Profesor:

VIOLETA DEL ROCIO BECERRA VELAZQUEZ

VERDUZCO ROSALES LUIS ENRIQUE

223992388

Ingeniería en Computación

21/09/25

CUCEI DIVTIC D04

ÍNDICE

Creación y Terminación de Procesos.....	2
Diagrama de 5 Estados de un Proceso.....	3
Algoritmos de Planificación.....	4
Tabla de Algoritmos de Planificación.....	5
Elementos de la Tabla de Procesos (BCP).....	7
¿En qué consisten los algoritmos de planificación No Apropiativos?.....	8
Tiempos solicitados en el punto 9 del programa 3.....	8
¿Qué significa BCP?.....	9
Conclusión.....	9
Bibliografía.....	10

TABLA IMAGENES

Ilustración: Modelo de 5 Estados:.....	4
--	---

Creación y Terminación de Procesos

Formas de Creación de un Proceso:

- Inicio del sistema:

Cuando enciendes la computadora, el sistema operativo crea varios procesos para que todo funcione correctamente.

- Llamada al sistema por un proceso existente:

Un proceso que ya está en ejecución puede crear un nuevo proceso "hijo" para realizar una tarea específica. A esto se le conoce como *bifurcación* (fork).

- Solicitud de un usuario:

Cuando abres un programa, como un navegador web o un juego, estás pidiendo al sistema operativo que cree un nuevo proceso.

- Inicio de un trabajo por lotes:

En algunos sistemas, se pueden programar trabajos para que se ejecuten en segundo plano sin la intervención directa del usuario. Cada uno de estos trabajos se ejecuta como un proceso.

Formas de Terminación de un Proceso:

- Salida normal (voluntaria):

El proceso termina porque completó su tarea.

- Salida por error (voluntaria):

El proceso finaliza por sí mismo al detectar un error en su código.

- Error fatal (involuntaria):

El sistema operativo termina el proceso debido a un error grave, como intentar acceder a memoria no permitida.

- Terminación por otro proceso (involuntaria):

Un proceso (generalmente el "padre") puede enviar una señal para terminar a otro proceso (su "hijo"). Es como cuando usas el "Administrador de Tareas" para cerrar un programa que no responde.

Diagrama de 5 Estados de un Proceso

El diagrama de 5 estados describe el ciclo de vida de un proceso en el sistema operativo.

Definición de cada estado y sus transiciones:

- **Nuevo (New):**
Es el primer estado. El proceso se está creando, pero el sistema operativo todavía no lo ha aceptado en el grupo de procesos ejecutables.
 - **Transición:**
Una vez que el sistema operativo tiene los recursos y la capacidad, lo admite y pasa al estado de Listo.
- **Listo (Ready):**
El proceso tiene todo lo que necesita para ejecutarse (memoria, archivos, etc.) y solo está esperando a que el procesador (CPU) se desocupe para poder usarlo. Puede haber muchos procesos en esta cola.
 - **Transición:**
Cuando el planificador del sistema operativo le asigna el CPU, el proceso pasa al estado de ejecución.
- **Ejecución (Running):**
El proceso está actualmente usando el CPU y ejecutando sus instrucciones.
 - **Transiciones:**
 - Si termina su tarea, pasa al estado de Terminado.
 - Si necesita esperar por algo (como un dato del disco duro o una entrada del usuario), pasa al estado de Bloqueado.
 - Si se le acaba su "tiempo" asignado en el CPU (en sistemas con planificación apropiativa), el planificador lo regresa al estado de Listo para darle oportunidad a otro proceso.
- **Bloqueado (Blocked/Waiting):**
El proceso no puede continuar ejecutándose porque está esperando que ocurra un evento externo. Por ejemplo, esperar a que termine una operación de lectura de disco o a que el usuario presione una tecla.

Actividad de Aprendizaje 3

- Transición:
Una vez que el evento que esperaba ha ocurrido, el proceso vuelve al estado de Listo (no a Ejecución directamente, porque tiene que volver a esperar su turno para el CPU).
- Terminado (Terminated):
El proceso ha finalizado su ejecución y el sistema operativo está liberando los recursos que utilizaba.

Ilustración: Modelo de 5 Estados:



Algoritmos de Planificación

Definición de Algoritmos de Planificación:

Los algoritmos de planificación son las reglas y procedimientos que utiliza el sistema operativo para decidir cuál de todos los procesos que están en la cola de "Listos" será el siguiente en usar el procesador (CPU). Su objetivo principal es administrar el tiempo del CPU de manera eficiente y justa.

Políticas de Planificación:

Existen dos grandes políticas o enfoques para la planificación:

- No Apropiativa:

Una vez que un proceso obtiene el CPU, no se le puede quitar hasta que termine su tarea o hasta que voluntariamente lo suelte (por ejemplo, al entrar en estado de "Bloqueado"). Es como hacer una fila en el banco: no te quitan de la ventanilla hasta que termines tu trámite.

- Apropiativa:

El sistema operativo puede "quitarle" a la fuerza el CPU a un proceso en ejecución para dárselo a otro, generalmente uno con mayor prioridad o porque al primero se le acabó su "cuota de tiempo". Esto permite que el sistema sea más interactivo y que ningún proceso acapare el CPU.

Tabla de Algoritmos de Planificación

Algoritmo	Descripción	Política	Ventaja	Desventaja
FCFS (First-Come, First-Served)	El primer proceso que llega a la cola de listos es el primero que se ejecuta. Es como una fila normal.	No Apropiativa	Es muy simple de entender e implementar.	Un proceso muy largo puede hacer esperar a todos los demás (efecto convoy).
SJF (Shortest Job First)	Se selecciona el proceso que tenga el tiempo de ejecución estimado más corto.	No Apropiativa	Es óptimo para minimizar el tiempo de espera promedio.	Es difícil saber con exactitud cuánto durará un proceso. Puede causar inanición (procesos largos nunca se ejecutan).
SRT (Shortest Remaining Time)	Es la versión apropiativa de SJF. Si llega un proceso nuevo que necesita menos tiempo para terminar que el que le falta al proceso actual, se le da el paso.	Apropiativa	Mejora el tiempo de respuesta y es más justo que SJF.	Sigue teniendo el problema de la estimación y el riesgo de inanición. Tiene mayor sobrecarga por los cambios de contexto.

Actividad de Aprendizaje 3

RR (Round Robin)	A cada proceso se le asigna una pequeña unidad de tiempo llamada quantum. Se ejecuta por ese tiempo y si no termina, se va al final de la cola de listos.	Apropiativa	Es muy justo, todos los procesos tienen la oportunidad de ejecutarse. Ideal para sistemas de tiempo compartido.	El rendimiento depende mucho del tamaño del quantum. Un quantum muy pequeño genera mucha sobrecarga.
Planificación por Prioridades	A cada proceso se le asigna una prioridad. El CPU se le da al proceso con la prioridad más alta. Puede ser apropiativa o no apropiativa.	Ambas	Permite que los procesos más importantes se ejecuten primero. Es flexible.	Puede causar inanición si los procesos de baja prioridad nunca tienen la oportunidad de ejecutarse.
Colas Múltiples (Multilevel Queue)	Se dividen los procesos en diferentes colas, cada una con su propio algoritmo de planificación.	Ambas	Es muy flexible y se puede adaptar a diferentes tipos de procesos.	Es más complejo de diseñar e implementar.
Planificación Garantizada (Guaranteed)	Intenta darle a cada uno de los N procesos, $1/N$ del poder del CPU. El sistema lleva un registro de cuánto CPU ha consumido cada proceso.	Apropiativa	Proporciona un rendimiento predecible y justo para los usuarios.	Es difícil de implementar en la práctica.

Planificación por Lotería (Lottery)	A los procesos se les dan "boletos" para un sorteo del CPU. Los procesos más importantes reciben más boletos. El ganador del sorteo se ejecuta.	Apropiativa	Resuelve el problema de la inanición y es muy responsivo.	La aleatoriedad puede llevar a resultados no óptimos a corto plazo.
-------------------------------------	---	-------------	---	---

Elementos de la Tabla de Procesos (BCP)

La tabla de procesos es una estructura de datos que mantiene el sistema operativo para cada uno de los procesos existentes. El BCP (Bloque de Control de Proceso) contiene toda la información asociada a un proceso específico. Sus elementos clave son:

- **Identificador del Proceso (PID):** Un número único que identifica al proceso.
- **Estado del Proceso:** El estado actual en el que se encuentra (nuevo, listo, ejecución, bloqueado, terminado).
- **Contador de Programa (Program Counter):** La dirección de la próxima instrucción que el proceso debe ejecutar.
- **Registros del CPU:** Los valores de los registros del procesador (acumulador, registros de índice, etc.) para que el proceso pueda continuar donde se quedó.
- **Información de Planificación:** La prioridad del proceso, punteros a las colas de planificación, etc.
- **Información de Gestión de Memoria:** Punteros a las tablas de páginas o segmentos que definen la memoria asignada al proceso.
- **Información Contable:** Cuánto tiempo de CPU ha usado, límites de tiempo, etc.
- **Información del Estado de E/S:** Lista de los dispositivos de entrada/salida asignados al proceso y los archivos que tiene abiertos.

¿En qué consisten los algoritmos de planificación No Apropiativos?

Los algoritmos de planificación No Apropiativos son aquellos en los que, una vez que el procesador (CPU) se le asigna a un proceso, este proceso sigue utilizando el CPU sin interrupciones hasta que una de dos cosas suceda:

1. El proceso termina su ejecución por completo.
2. El proceso se bloquea voluntariamente porque necesita esperar un recurso, como un dato del disco o una entrada del teclado.

En resumen, el sistema operativo no puede "arrebatarle" el CPU a un proceso en ejecución. El proceso es el único que decide cuándo lo libera. Ejemplos claros son **FCFS** y **SJF**.

Explique cómo obtener los:

Tiempos solicitados en el punto 9 del programa 3

1. Tiempo de Llegada (TL)

Es simplemente el momento exacto en que un proceso llega y se pone en la cola de "Listos". Este es un dato que normalmente te dan para resolver el problema.

- Es un valor inicial, no se calcula. Por ejemplo: Proceso A llega en el segundo 0, Proceso B en el segundo 2.

2. Tiempo de Finalización (TF)

Es el momento exacto en que un proceso termina por completo su tarea y sale del sistema.

- Se determina observando en qué segundo exacto termina de ejecutarse el proceso según el algoritmo de planificación que estés usando.

3. Tiempo de Servicio (TS) o Ráfaga de CPU

Este es el tiempo total que un proceso necesita usar el CPU para completar su tarea. Es la cantidad de tiempo que el sistema tarda en atender un proceso si no hubiera interrupciones. Si sabes el tiempo total que un proceso estuvo en el sistema y le quitas lo que pasó esperando en la fila, el resultado es el tiempo que estuvo efectivamente usando el procesador.

- Tiempo de retorno (TR) - Tiempo de espera (TE)

4. Tiempo de Retorno (TR) o Turnaround Time

Es el tiempo total que un proceso pasa en el sistema, desde que llega hasta que termina. Incluye el tiempo que pasó esperando en la cola y el tiempo que pasó ejecutándose. Es el tiempo total desde que el proceso se formó en la fila hasta que terminó

- Se calcula con una resta simple:
$$TR = \text{Tiempo de Finalización (TF)} - \text{Tiempo de Llegada (TL)} \text{ ó } \text{Tiempo de espera (TE)} + \text{Tiempo de servicio (TS)}$$

5. Tiempo de Espera (TE) o Waiting Time

Es la suma de todo el tiempo que un proceso pasó en la cola de "Listos", esperando su turno para usar el CPU. No incluye el tiempo que estuvo ejecutándose ni el tiempo que estuvo bloqueado.

- Se calcula restando el tiempo que estuvo trabajando del tiempo total que estuvo en el sistema:
$$TE = \text{Tiempo de Retorno (TR)} - \text{Tiempo de Servicio (TS)}$$

6. Tiempo de Respuesta

Es el tiempo que transcurre desde que un proceso llega hasta que se ejecuta por primera vez. Es una medida de qué tan rápido el sistema responde a una nueva solicitud.

- Es un valor inicial, no se calcula.

¿Qué significa BCP?

BCP significa Bloque de Control de Proceso. En inglés se le conoce como PCB (Process Control Block). Es la estructura de datos fundamental que el sistema operativo utiliza para almacenar toda la información relacionada con un proceso específico. Es como el "expediente" o la "ficha de identificación" de cada proceso.

Conclusión

En resumen, lo que mejor entendí es que el sistema operativo necesita acomodar los programas que se están ejecutando, los cuales se llaman procesos. Para esto, utiliza algoritmos de planificación que son reglas para decidir qué proceso usa el procesador y en qué orden. Algunos algoritmos son no apropiativos, significa que el proceso no es interrumpido hasta que termina y otros son apropiativos, significa que

el sistema puede pausar un proceso para darle paso a otro. Para llevar una buena administración y no perder el control, toda la información de cada proceso se guarda en una ficha llamada BCP. Este sistema existe para lograr que muchos programas funcionen al mismo tiempo de manera eficiente.

Bibliografía

- Ascí, G. D. (s.f.). Planificación de CPU [Diapositivas de clase]. Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur. Recuperado de: <https://cs.uns.edu.ar/~gd/soyd/clasesgus/05-Planificaci%C3%B3ndeCPU.pdf>
- Contreras, J. (2017, 25 de septiembre). Planificación apropiativa vs no apropiativa [Presentación]. Prezi. Recuperado de: <https://prezi.com/2k7w0n8z8orz/planificacion-apropiativa-vs-no-apropiativa/>
- Departamento de Computación, Universidade da Coruña. (s.f.). *Sistemas Operativos: Procesos* [Diapositivas de clase]. Recuperado de: https://www.dc.fi.udc.es/~so-grado/2_PROCESOS.pdf
- Departamento de Computación, Universidade da Coruña. (s.f.). *Sistemas Operativos: Procesos y Planificación* [Diapositivas de clase]. Recuperado de: <https://www.dc.fi.udc.es/~so-grado/SO-Procesos-planif.pdf>
- Facultad de Ingeniería, Universidad de la República. (s.f.). *Planificación* [Diapositivas de clase]. Recuperado de: <https://www.fing.edu.uy/inco/cursos/sistoper/recursosTeoricos/6-SO-Teo-Planificacion.pdf>
- Facultad de Ingeniería, Universidad de la República. (s.f.). *Procesos* [Diapositivas de clase]. Recuperado de: <https://www.fing.edu.uy/inco/cursos/sistoper/recursosTeoricos/5-SO-Teo-Procesos.pdf>
- Ginzo, J. (2024, 21 de julio). *BCP o Bloque de Control de Proceso, ¿qué es?* Ginzo.tech. Recuperado de: <https://ginzo.tech/bcp-bloque-control-proceso/>
- Gong, S. (2024, 1 de abril). *Five-State Process Model*. Steven Gong's Notes. Recuperado de: <https://stevengong.co/notes/Five-State-Process-Model>
- González, F. J. (s.f.). *Tema 5: Planificación de procesos*. Departamento de Informática, Universidad de Valladolid. Recuperado de: <https://www.infor.uva.es/~fjgonzalez/apuntes/Tema5.pdf>

- Instituto Clavijero. (s.f.). 2.2. *Estados de los procesos*. Cursos Clavijero. Recuperado de: https://cursos.clavijero.edu.mx/cursos/182_so/modulo2/contenidos/tema2.2.html?opc=1
- LC Sistemas Operativos. (2015, 3 de febrero). *03.03 Algoritmos de Planificación*. WordPress.com. Recuperado de: <https://lcsistemasoperativos.wordpress.com/2015/02/03/03-03-algoritmos-de-planificacion/>
- Romero, S., & de Miguel, P. (2012). *Planificación del procesador (I)* [Diapositivas de clase]. Departamento de Arquitectura y Tecnología de Sistemas Informáticos, Universidad Politécnica de Madrid. Recuperado de: https://laurel.datsi.fi.upm.es/_media/docencia/asignaturas/soa/soa-planificacion-parte1.pdf
- Universitat Oberta de Catalunya. (2008). *Gestión de procesos*. Fundació per a la Universitat Oberta de Catalunya. Recuperado de: <https://openaccess.uoc.edu/server/api/core/bitstreams/038750e9-8f9a-457e-b551-bb1b0054cbb8/content>