

SISTEMA OPERATIVO

ACTIVIDAD DE
APRENDIZAJE 4

JESSICA ALEXANDRA MAGAÑA
SALCEDO

215616229

Ingeniería en computación

18/02/2024

Departamento de ciencias
computacionales

Centro Universitario de Ciencias Exactas
e Ingenierías

Índice

PORTADA	1
Indice	2
Introducción	3
Creación y terminación de procesos	4
Diagrama de 5 estados	5
Algoritmo de planificación	7
Tabla de procesos. (BCP)	9
Preguntas sobre la investigación	10
Conclusión	11
Bibliografía	12

Introducción

La creación y terminación de procesos son aspectos fundamentales en el funcionamiento de cualquier sistema operativo. Cuando se crea un nuevo proceso, ya sea para ejecutar una tarea específica o para modularizar el trabajo, se inician una serie de acciones para su gestión y control. Por otro lado, la terminación de procesos puede ocurrir por diversas razones, como la finalización exitosa de la tarea, errores en la ejecución o decisiones del sistema operativo.

En este contexto, los algoritmos de planificación juegan un papel crucial en la asignación eficiente de recursos de CPU a los procesos en ejecución. Estos algoritmos, basados en diferentes políticas, determinan qué proceso ejecutar a continuación, teniendo en cuenta factores como la prioridad, el tiempo de ejecución y la disponibilidad de recursos.

Además, el Bloque de Control de Procesos (BCP) es una estructura de datos esencial utilizada por los sistemas operativos para gestionar información detallada sobre cada proceso en ejecución. Contiene elementos como la identificación del proceso, su estado, registros del procesador, información de planificación y recursos asignados.

Creación y Terminación de Procesos

La creación y terminación de procesos son elementos clave en el funcionamiento de cualquier sistema operativo. Cuando se añade un nuevo proceso al conjunto que ya está administrando el sistema operativo, se realizan diversas acciones para su gestión y control:

Creación de Procesos:

1. **Nueva tarea en un sistema de proceso por lotes:** El sistema operativo toma un nuevo trabajo de la cola de trabajos por lotes y construye las estructuras de datos necesarias para su gestión.
2. **Nueva conexión interactiva:** Cuando un usuario se conecta al sistema desde una terminal, se crea un nuevo proceso para manejar la interacción.
3. **Nuevo proceso creado por el SO para dar un servicio:** El sistema puede iniciar un proceso para ejecutar una tarea en nombre del usuario, como la impresión de un documento, sin requerir la intervención directa del usuario.
4. **Proceso generado por otro ya existente:** Un programa de usuario puede ordenar la creación de nuevos procesos para modularizar el trabajo o aprovechar el paralelismo.

Terminación de Procesos:

1. **Terminación normal:** El proceso completa su ejecución de manera exitosa y finaliza.
2. **Tiempo límite excedido:** Si un proceso excede el límite de tiempo establecido para su ejecución, se termina.
3. **No disponibilidad de memoria:** Si un proceso requiere más memoria de la disponible, se termina debido a la falta de recursos.
4. **Violación de límites y error de protección:** Se termina un proceso si intenta acceder a memoria o recursos de forma no autorizada.
5. **Error aritmético:** Ocurre cuando el proceso realiza operaciones aritméticas prohibidas, como división por cero.
6. **Superación del tiempo máximo de espera por un recurso:** Si un proceso queda en espera de un recurso durante más tiempo del permitido, se termina.
7. **Fallo de dispositivo I/O:** Se termina un proceso si se producen errores persistentes en operaciones de entrada/salida.

8. **Instrucción no válida e intento de acceso a una instrucción privilegiada:** Un proceso se termina si intenta ejecutar instrucciones inexistentes o reservadas para el sistema operativo.
9. **Mal uso de los datos:** Si un proceso utiliza datos de manera incorrecta, puede ser terminado.
10. **Intervención del operador o del SO:** El operador o el sistema operativo pueden terminar un proceso si compromete el rendimiento del sistema.
11. **Finalización del proceso padre y solicitud del proceso padre:** Un proceso padre puede terminar automáticamente con todos sus procesos hijos al finalizar, o solicitar la terminación de un proceso hijo.

Diagrama de 5 estados

En el modelo de cinco estados en la gestión de procesos en sistemas operativos, cada proceso puede encontrarse en uno de los siguientes estados, cada uno con características y funcionalidades específicas:

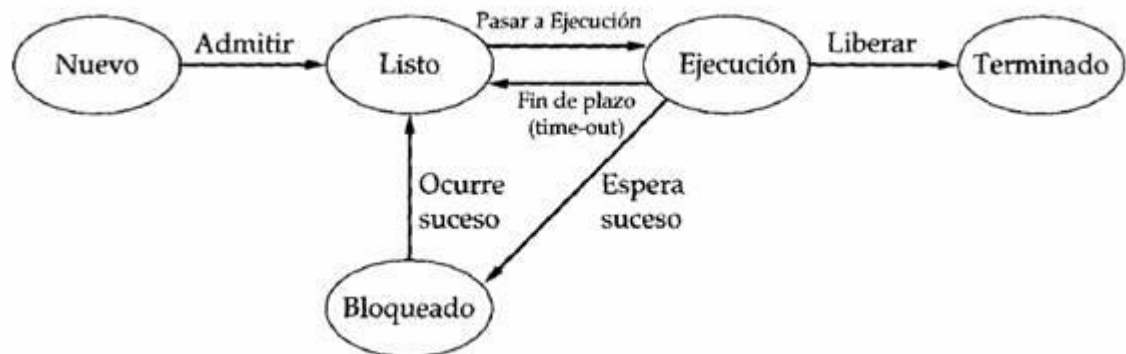
1. **Nuevo:** En este estado, los procesos han sido definidos recientemente pero aún no han sido admitidos por el sistema operativo como procesos ejecutables. Se han realizado tareas de gestión interna, como asignación de identificadores y creación de estructuras de control, pero el proceso aún no ha comenzado su ejecución.
2. **Listo o Preparado:** Los procesos en este estado tienen todos los recursos necesarios para su ejecución y están listos para comenzar o continuar su ejecución. Sin embargo, están a la espera de que se les conceda el control del procesador. Este estado es crucial para la planificación de procesos, ya que aquí se encuentran los procesos disponibles para la ejecución.
3. **Ejecución:** Cuando un proceso tiene el control del procesador, se encuentra en este estado. Durante este estado, el proceso está realizando sus instrucciones y utilizando los recursos del sistema. En sistemas con un único procesador, solo un proceso puede estar en este estado en un momento dado.
4. **Bloqueado:** Los procesos en este estado carecen de algún recurso necesario para su ejecución, que no sea el procesador, o están esperando que ocurra un evento específico. Esto puede ser debido a la espera de la disponibilidad de un recurso de entrada/salida, como la lectura de datos desde un disco, o la espera de que otro proceso termine de ejecutarse.
5. **Terminado:** Los procesos en este estado han finalizado su ejecución y han sido excluidos del grupo de procesos ejecutables. Pueden haber alcanzado su punto de terminación normal, haber sido abandonados debido a un error

irrecuperable, o haber sido terminados por un proceso con la autoridad necesaria.

Las transiciones entre estos estados son gestionadas por el sistema operativo y ocurren en respuesta a diferentes eventos. Estas transiciones incluyen:

- **Transición a Nuevo:** Ocurre cuando se crea un nuevo proceso para ejecutar un programa.
- **Transición Nuevo-Preparado:** Se produce cuando el sistema operativo está listo para aceptar un nuevo proceso y lo mueve al estado preparado.
- **Transición Preparado-Ejecución:** El sistema operativo selecciona un proceso preparado para ejecutar según su política de planificación y lo mueve al estado de ejecución.
- **Transición Ejecución-Preparado:** Ocurre cuando un proceso en ejecución ha alcanzado el tiempo máximo permitido o ha sido expropiado y se devuelve al estado preparado.
- **Transición Ejecución-Bloqueo:** Un proceso pasa a este estado cuando requiere un recurso que no está disponible o está esperando la ocurrencia de un evento.
- **Transición Bloqueado-Preparado:** Cuando el recurso requerido está disponible o el evento esperado ocurre, el proceso pasa de bloqueado a preparado.
- **Transición Preparado-Terminado:** Un proceso en estado preparado puede ser terminado por decisión del sistema operativo o del proceso padre.
- **Transición Bloqueado-Terminado:** Un proceso bloqueado puede ser terminado si excede el tiempo máximo de espera por un recurso o si es terminado por el sistema operativo.

Estas transiciones son fundamentales para el funcionamiento del sistema operativo, ya que determinan cómo se gestionan los procesos y cómo se asignan los recursos del sistema. La eficiente gestión de estos estados y transiciones garantiza un rendimiento óptimo del sistema operativo.



Algoritmos de Planificación

Los algoritmos de planificación son parte fundamental de los sistemas operativos y se encargan de tomar decisiones sobre qué proceso o tarea ejecutar a continuación en una computadora. Estos algoritmos se basan en una serie de políticas y reglas para asignar recursos de manera eficiente y equitativa, teniendo en cuenta factores como la prioridad de los procesos, el tiempo de ejecución, entre otros. 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.

Políticas de Planificación

1. **Apropiativas (Preemptive):** En este tipo de política, el sistema operativo puede interrumpir la ejecución de un proceso en cualquier momento para dar paso a otro proceso de mayor prioridad. Esto significa que un proceso puede ser suspendido, o "preempted", para permitir que otro proceso se ejecute, lo que garantiza una mayor capacidad de respuesta y una gestión más eficiente de los recursos.
2. **No Apropiativas (Non-preemptive):** En este caso, un proceso se ejecuta hasta que llega a un estado de bloqueo, termina su ejecución o cede voluntariamente el control de la CPU. No se permite la interrupción de un proceso en ejecución, lo que significa que el proceso en curso tiene el control exclusivo de la CPU hasta que completa su tarea o se bloquea. Este enfoque puede ser más simple de implementar, pero puede resultar en una menor capacidad de respuesta del sistema.

Algoritmo	Descripción	Política	Ventaja	Desventaja
RR (Round Robin)	Asigna un quantum de tiempo a cada proceso en ejecución y luego pasa al siguiente proceso en una cola circular.	Apropiativa	Fácil de implementar, buena capacidad de respuesta	Puede generar un alto tiempo de respuesta para tareas largas

Actividad de aprendizaje 4

FCFS (First Come, First Served)	Ejecuta los procesos en el orden en que llegan a la cola de listos.	No Apropriativa	Simple de entender e implementar	Puede generar un fenómeno de "inversión de prioridades" si un proceso corto se encuentra detrás de uno largo
SRT (Shortest Remaining Time)	Selecciona el proceso con el tiempo de ejecución restante más corto.	Apropiativa	Minimiza el tiempo de espera para procesos cortos	Puede generar inanición para procesos largos
SJF (Shortest Job First)	Asigna prioridad al proceso más corto en términos de tiempo de CPU restante.	No Apropriativa	Minimiza el tiempo de respuesta promedio	Requiere conocimiento previo de los tiempos de ejecución de los procesos
Prioridades	Asigna una prioridad a cada proceso y ejecuta primero aquellos con mayor prioridad.	Apropiativa o No Apropriativa	Permite dar preferencia a procesos críticos o importantes	Puede generar inanición si los procesos de baja prioridad nunca se ejecutan
Colas Múltiples	Divide los procesos en diferentes colas según su prioridad o características y aplica un algoritmo de planificación diferente a cada cola.	Puede ser apropiativa o no apropiativa según la política en cada cola	Permite ajustar la planificación según las necesidades específicas de cada tipo de proceso	Puede aumentar la complejidad del sistema y la sobrecarga de gestión

SJRF (Shortest Job Remaining First)	Similar a SJF, pero en este caso se evalúa el tiempo restante de los procesos en la cola de listos en lugar de su tiempo total de ejecución.	Apropiativa	Minimiza el tiempo de espera promedio	Puede generar inanición para procesos largos si llegan constantemente procesos más cortos
Multinivel	Los procesos se agrupan en niveles y se asigna a cada nivel un algoritmo de planificación diferente. Los procesos pueden moverse entre niveles según ciertas reglas.	Puede ser apropiativa o no apropiativa según la política de cada nivel	Ofrece flexibilidad al permitir diferentes políticas de planificación para diferentes tipos de procesos	Puede aumentar la complejidad de la implementación y requerir un ajuste cuidadoso de los parámetros de configuración

Tabla de procesos. (BCP)

El Bloque de Control de Procesos (BCP), también conocido como Bloque de Control de PCB (Proceso, en inglés), es una estructura de datos fundamental en los sistemas operativos para representar y gestionar la información asociada a un proceso. Los elementos típicos que componen un BCP son los siguientes:

1. **Identificación del Proceso:** Un identificador único que permite diferenciar entre diferentes procesos en el sistema.
2. **Estado del Proceso:** Indica en qué fase del ciclo de vida del proceso se encuentra, como listo, en ejecución, bloqueado, terminado, etc.
3. **Contador de Programa (PC):** Contiene la dirección de la próxima instrucción a ejecutar para el proceso.
4. **Registros del Procesador:** Almacenan el estado de los registros de la CPU del proceso, como el acumulador, los registros de índice, el contador de programa, etc.

5. **Información de Planificación:** Datos relacionados con la planificación del proceso, como su prioridad, tiempo de ejecución restante, tiempo de espera, etc.
6. **Información de Recursos:** Contiene detalles sobre los recursos asignados al proceso, como los dispositivos de E/S, archivos abiertos, memoria asignada, etc.
7. **Información de Propiedad:** Información sobre el usuario o grupo propietario del proceso, permisos de acceso, etc.
8. **Lista de Bloqueo (si aplica):** Si el proceso está bloqueado esperando algún recurso, esta lista contiene los detalles de los recursos esperados y el estado de bloqueo.

Preguntas sobre la investigación

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

Los algoritmos de planificación no apropiativos, también conocidos como algoritmos de planificación por turnos, se caracterizan por asignar un recurso de CPU a un proceso y no cambiarlo hasta que dicho proceso haya completado su ejecución o haya sido bloqueado. En estos algoritmos, el proceso en ejecución no puede ser interrumpido voluntariamente por el sistema operativo; solo cede la CPU después de que haya terminado su ejecución o cuando se produce un evento que lo bloquea, como una solicitud de E/S. Ejemplos de algoritmos de planificación no apropiativos incluyen el algoritmo de planificación FCFS (First-Come, First-Served) y el algoritmo de planificación SJF (Shortest Job First).

2. ¿Qué significa BCP?

BCP significa "Bloque de Control de Procesos". Es una estructura de datos utilizada por los sistemas operativos para administrar y mantener información sobre los procesos en ejecución. El BCP contiene detalles importantes sobre cada proceso, como su identificación, estado actual, registros del procesador, información de planificación, recursos asignados, información de propiedad, lista de bloqueo, entre otros. El BCP es fundamental para el funcionamiento del sistema operativo, ya que permite realizar operaciones como la conmutación de procesos, la planificación de CPU, la gestión de recursos y la protección del sistema.

Conclusión

En conclusión, la creación y terminación de procesos son elementos esenciales en la operación de cualquier sistema operativo. La capacidad de crear y gestionar procesos de manera eficiente permite que el sistema maneje tareas de forma ordenada y aproveche los recursos disponibles de manera óptima. La terminación de procesos, ya sea por finalización exitosa o por eventos inesperados, es crucial para liberar recursos y mantener la estabilidad del sistema.

Los algoritmos de planificación desempeñan un papel vital en la asignación de recursos de CPU, determinando qué proceso se ejecutará a continuación. Estos algoritmos, basados en diversas políticas, buscan maximizar la utilización de recursos, minimizar los tiempos de espera y garantizar una respuesta rápida a las solicitudes del usuario.

El Bloque de Control de Procesos (BCP) proporciona una estructura de datos fundamental para el seguimiento y la gestión de información detallada sobre cada proceso en ejecución. El BCP permite al sistema operativo llevar a cabo operaciones como la conmutación de procesos, la planificación de CPU y la gestión de recursos de manera efectiva.

En conjunto, la creación y terminación de procesos, junto con los algoritmos de planificación y el uso del BCP, son aspectos críticos para el funcionamiento eficiente de un sistema operativo. Una gestión adecuada de estos elementos garantiza un rendimiento óptimo del sistema, una respuesta rápida a las demandas del usuario y una mayor estabilidad operativa.

Actividad de aprendizaje 4

Bibliografía

https://cursos.clavijero.edu.mx/cursos/147_poti/modulo2/contenidos/tema2.2.html

"Operating System Concepts" de Abraham Silberschatz, Peter B. Galvin y Greg Gagne.