

UNIVERSIDAD AUTONOMA DE CHIHUAHUA
FACULTAD DE INGENIERIA
INGENIERIA EN CIENCIAS DE LA COMPUTACION

SISTEMAS OPERATIVOS
INTEGRANTES:

AMERICA GAUDALUPE MARTINEZ CANO 348810

FATIMA MONSERRATH DUARTE PEREZ 353324

DIEGO ALEJANDRO MARTINEZ CANO 353198

PROFE:

MIGUEL ANGEL MARTINEZ LOPEZ

PROCESOS DE SISTEMAS
OPERATIVOS

Tabla de contenido

Introducción	3
Jerarquía de Procesos	4
Niveles de Jerarquía de Procesos	4
Importancia de la Jerarquía de Procesos	5
Control y Gestión de Procesos	5
Jerarquía de Capas	6
Conclusión	8
Planificación de Procesos	9
Niveles de Planificación	10
Objetivos de la Planificación	11
Criterios de Planificación	12
Planificación Apropriativa y No Apropriativa	14
Tipos de Planificación	15
Conclusión	25
Políticas de Procesos	26
Importancia de las Políticas de Procesos	27
Aplicación de las Políticas de Procesos en la Gestión de Procesos	28
Conclusión	29

INTRODUCCIÓN

El concepto central de cualquier Sistema Operativo es el de proceso: una abstracción de un programa en ejecución también llamada tarea. En un sistema operativo, un proceso es un programa en ejecución que ha sido asignado por el sistema operativo a una o varias unidades de procesamiento (por ejemplo, una CPU). Un proceso es una entidad activa que realiza tareas específicas, que pueden ser de entrada, salida, procesamiento o comunicación, y que requiere de recursos del sistema para su ejecución, como memoria, CPU, dispositivos de entrada y salida, y otros recursos del sistema.

Cada proceso tiene su propio espacio de memoria y conjunto de registros, que se utilizan para almacenar la información necesaria para su ejecución. Además, cada proceso tiene su propio identificador único, que lo distingue de los demás procesos en el sistema operativo. Los procesos pueden ser creados por el sistema operativo o por otros procesos en ejecución.

El sistema operativo es responsable de la gestión y coordinación de los procesos en el sistema, lo que incluye su creación, eliminación, suspensión y reanudación, así como la asignación de recursos del sistema. El sistema operativo también es responsable de la planificación de procesos, es decir, la forma en que se determina qué proceso debe ejecutarse en cada momento y cómo se distribuyen los recursos del sistema entre los distintos procesos.

En resumen, un proceso en un sistema operativo es un programa en ejecución que realiza tareas específicas y que requiere de recursos del sistema para su ejecución. El sistema operativo es responsable de la gestión y coordinación de los procesos en el sistema, incluyendo su creación, eliminación, suspensión y reanudación, y la planificación de procesos para garantizar el rendimiento y la eficiencia del sistema.

JERARQUÍA DE PROCESOS

En cuanto a las jerarquías de procesos es necesario señalar que los Sistemas Operativos deben disponer de una forma de crear y destruir procesos cuando se requiera durante la operación, teniendo además presente que los procesos pueden generar procesos hijos mediante llamadas al Sistema Operativo, pudiendo darse ejecución en paralelo.

La jerarquía de procesos es un concepto fundamental en el funcionamiento de un sistema operativo. Es una estructura organizativa en la que los procesos se organizan en una jerarquía de árbol. En esta jerarquía, cada proceso tiene un padre y puede tener uno o varios hijos, lo que permite una estructura jerárquica de los procesos en el sistema. En este ensayo, exploraremos en profundidad la jerarquía de procesos, su importancia y su papel en el funcionamiento del sistema operativo.

Niveles de Jerarquía de Procesos

La jerarquía de procesos se divide en tres niveles: el nivel superior, los procesos de usuario y los procesos de sistema. En la cima de la jerarquía se encuentra el proceso "init" o "systemd", que es el primer proceso que se inicia cuando se arranca el sistema operativo. Este proceso es responsable de iniciar otros procesos del sistema operativo, y se encarga de establecer el entorno necesario para que los procesos puedan ejecutarse adecuadamente.

Por debajo del proceso "init", se encuentran los procesos de nivel superior, que son procesos esenciales para el funcionamiento del sistema operativo. Estos procesos incluyen servicios del sistema, como el administrador de red, el administrador de energía, el administrador de impresión, entre otros. Estos procesos son cruciales para garantizar un funcionamiento estable y seguro del sistema operativo.

Por debajo de los procesos de nivel superior, se encuentran los procesos de usuario. Estos procesos son iniciados por los usuarios del sistema operativo y se ejecutan en el contexto de un usuario específico. Los procesos de usuario pueden

tener uno o varios procesos hijos, lo que permite la ejecución de múltiples procesos por parte de un usuario.

Importancia de la Jerarquía de Procesos

La jerarquía de procesos es esencial para la gestión de recursos en el sistema operativo. Los procesos de nivel superior, como los servicios del sistema, tienen acceso privilegiado a los recursos del sistema y están diseñados para garantizar que los recursos se utilicen de manera eficiente y segura. Los procesos de usuario, por otro lado, tienen acceso limitado a los recursos del sistema y están diseñados para garantizar que un usuario no pueda afectar el funcionamiento de otros usuarios o del sistema operativo en general.

La jerarquía de procesos también permite la gestión de procesos en el sistema operativo. Cada proceso tiene una prioridad asignada, lo que permite al sistema operativo determinar qué proceso se ejecutará en un momento dado. Si un proceso de nivel superior necesita más recursos, el sistema operativo puede reasignar recursos de otros procesos para garantizar que todos los procesos puedan ejecutarse correctamente.

La jerarquía de procesos también permite la gestión de recursos compartidos, como la memoria. Si un proceso necesita más memoria, el sistema operativo puede reasignar memoria de otros procesos que estén utilizando menos memoria. Esto ayuda a garantizar que todos los procesos tengan acceso a la memoria necesaria para funcionar correctamente.

Control y Gestión de Procesos

La jerarquía de procesos también permite al sistema operativo controlar y gestionar los procesos. Si un proceso se vuelve inestable o consume demasiados recursos, el sistema operativo puede matarlo o restringir su acceso a los recursos del sistema. Esto ayuda a garantizar la estabilidad y seguridad del sistema operativo.

Además, la jerarquía de procesos permite la gestión de procesos en segundo plano. Cuando un proceso de usuario se inicia, puede ejecutarse en segundo plano

mientras el usuario realiza otras tareas. Esto permite a los usuarios realizar varias tareas simultáneamente sin afectar el rendimiento del sistema operativo.

La jerarquía de procesos también permite la comunicación entre procesos. Los procesos pueden comunicarse entre sí mediante señales y mensajes, lo que permite la coordinación y el intercambio de información entre los procesos. Además, la jerarquía de procesos es importante para la eficiencia del sistema operativo. Si el sistema operativo no puede gestionar correctamente los recursos y procesos, esto puede llevar a una disminución en el rendimiento del sistema y a problemas de estabilidad.

Es importante destacar que la jerarquía de procesos no es estática, sino que puede cambiar a medida que los procesos se inician y se cierran. El sistema operativo debe ser capaz de adaptarse y gestionar estos cambios de manera efectiva para mantener la estabilidad y eficiencia del sistema.

También es importante mencionar que la jerarquía de procesos puede variar según el sistema operativo utilizado. Si bien los principios básicos de la jerarquía de procesos son similares en diferentes sistemas operativos, la implementación puede diferir en función del diseño y la arquitectura del sistema operativo.

Jerarquía de Capas

Es una generalización del modelo de estructura simple para un sistema monolítico. Consiste en organizar el s. o. como una jerarquía de capas, cada una construida sobre la inmediata inferior.

El primer sistema con este esquema fue el THE (Technische Hogeschool Eindhoven):

- Capa 0
 - Trabaja con la asignación del procesador.
 - Alterna entre los procesos cuando ocurren las interrupciones o expiran los cronómetros.
 - Proporciona la multiprogramación básica

- Capa 1
 - Administra la memoria.
 - Asegura que las páginas (porciones de memoria) requeridas de los procesos lleguen a memoria cuando fueran necesarias.
- Capa 2
 - Administra la comunicación entre cada proceso y la consola del operador.
 - Por sobre esta capa, cada proceso tiene su propia consola de operador
- Capa 3
 - Controla los dispositivos de e/s y almacena en buffers los flujos de información entre ellos.
 - Por sobre la capa 3 cada proceso puede trabajar con dispositivos abstractos de e/s en vez de con dispositivos reales.
- Capa 4
 - Aloja los programas del usuario.
 - Los programas. del usuario no tienen que preocuparse por el proceso, memoria, consola o control de e/s.
- Capa 5
 - Localiza el proceso operador del sistema.

5 - Operador
4 - Programas del Usuario
3 - Control de Entrada / Salida
2 - Comunicaciones Operador - Proceso
1 - Administración de la Memoria y del Disco
0 - Asignación del Procesador y Multiprogramación

Una generalización más avanzada del concepto de capas se presentó con “Multics (Multiplexed Information and Computing Service)

- Presenta una estructura en anillos concéntricos, siendo los interiores los privilegiados.
- Un procedimiento de un anillo exterior, para llamar a un procedimiento de un anillo interior, debe hacer el equivalente a una llamada al sistema.

Conclusión

La jerarquía de procesos es un componente esencial del sistema operativo que permite la gestión de recursos, la gestión de procesos y la comunicación entre procesos. La jerarquía de procesos permite al sistema operativo controlar y gestionar los procesos de manera efectiva, lo que garantiza la estabilidad y seguridad del sistema operativo. La jerarquía de procesos también permite a los usuarios realizar varias tareas simultáneamente y ejecutar procesos en segundo plano sin afectar el rendimiento del sistema operativo. Además, la jerarquía de procesos permite la comunicación entre procesos, lo que permite la coordinación y el intercambio de información entre los procesos.

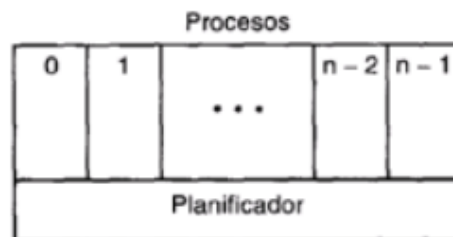
En resumen, la jerarquía de procesos es un componente esencial del sistema operativo que permite la gestión de recursos, la gestión de procesos y la comunicación entre procesos. La jerarquía de procesos es crucial para garantizar la estabilidad y seguridad del sistema operativo y permitir a los usuarios realizar varias tareas simultáneamente.

PLANIFICACIÓN DE PROCESOS

La planificación de procesos es una parte fundamental de cualquier sistema operativo moderno. Es la manera en que el sistema decide qué proceso debe ejecutarse en cada momento, y cómo se distribuyen los recursos del sistema entre los distintos procesos.

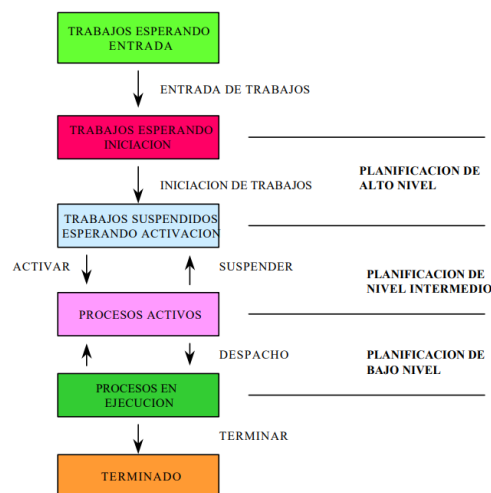
Cuando más de un proceso es ejecutable desde el punto de vista lógico, el Sistema Operativo debe decidir cuál de ellos debe ejecutarse en primer término. El Planificador es la porción del Sistema Operativo que decide y el Algoritmo de Planificación es el utilizado. Los principales “criterios” respecto de un buen algoritmo de planificación son la equidad, la eficacia, el tiempo de respuesta, el tiempo de regreso y el rendimiento.

Cada proceso es único e impredecible, es decir que pueden requerir intensivamente operaciones de Entrada/Salida o intensivamente CPU; el planificador del Sistema Operativo no tiene la certeza de cuánto tiempo transcurrirá hasta que un proceso se bloquee, ya sea por una operación de Entrada / Salida o por otra razón. Para evitar que un proceso se apropie de la CPU un tiempo excesivo, los equipos poseen un dispositivo que provoca una interrupción en forma periódica, por ejemplo 60 Hz, o sea sesenta veces por segundo. En cada interrupción del reloj el Sistema Operativo decide si el proceso que se está ejecutando continúa o si el proceso agotó su tiempo de CPU y debe suspenderse y ceder la CPU a otro proceso.



Niveles de Planificación

- Planificación de alto nivel
 - También se denomina Planificación de trabajos.
 - Determina a qué trabajos se les va a permitir competir activamente por los recursos del sistema, lo cual se denomina Planificación de admisión.
- Planificación de nivel intermedio
 - Determina a qué procesos se les puede permitir competir por la CPU.
 - Responde a fluctuaciones a corto plazo en la carga del sistema y efectúa “suspensiones” y “activaciones” (“reanudaciones”) de procesos.
 - Debe ayudar a alcanzar ciertas metas en el rendimiento total del sistema.
- Planificación de bajo nivel
 - Determina a qué proceso listo se le asigna la CPU cuando esta queda disponible y asigna la CPU al mismo, es decir que “despacha” la CPU al proceso.
 - La efectúa el Despachador del Sistema Operativo, el que opera muchas veces por segundo y reside siempre en el almacenamiento primario.



Objetivos de la planificación

Los objetivos de la planificación del procesador son los siguientes e involucran a los conceptos detallados seguidamente:

- Ser justa:
 - Todos los procesos son tratados de igual manera.
 - Ningún proceso es postergado indefinidamente.
- Maximizar la capacidad de ejecución:
 - Maximizar el número de procesos servidos por unidad de tiempo.
- Maximizar el número de usuarios interactivos que reciban unos tiempos de respuesta aceptables:
 - En un máximo de unos segundos.
- Ser predecible:
 - Un trabajo dado debe ejecutarse aproximadamente en la misma cantidad de tiempo independientemente de la carga del sistema.
- Minimizar la sobrecarga:
 - No suele considerarse un objetivo muy importante.
- Equilibrar el uso de recursos:
 - Favorecer a los procesos que utilizarán recursos infrautilizados.
- Equilibrar respuesta y utilización:
 - La mejor manera de garantizar buenos tiempos de respuesta es disponer de los recursos suficientes cuando se necesitan, pero la utilización total de recursos podrá ser pobre.
- Evitar la postergación indefinida:
 - Se utiliza la estrategia del “envejecimiento”.
 - Mientras un proceso espera por un recurso su prioridad debe aumentar, así la prioridad llegará a ser tan alta que el proceso recibirá el recurso esperado.
- Asegurar la prioridad:
 - Los mecanismos de planificación deben favorecer a los procesos con prioridades más altas.

- Dar preferencia a los procesos que mantienen recursos claves:
 - Un proceso de baja prioridad podría mantener un recurso clave, que puede ser requerido por un proceso de más alta prioridad.
 - Si el recurso es no apropiativo, el mecanismo de planificación debe otorgar al proceso un tratamiento mejor del que le correspondería normalmente, puesto que es necesario liberar rápidamente el recurso clave.
- Dar mejor tratamiento a los procesos que muestren un “comportamiento deseable”:
 - Un ejemplo de comportamiento deseable es una tasa baja de paginación.
- Degradarse suavemente con cargas pesadas:
 - Un mecanismo de planificación no debe colapsar con el peso de una exigente carga del sistema.
 - Se debe evitar una carga excesiva mediante las siguientes acciones:
 - × No permitiendo que se creen nuevos procesos cuando la carga ya es pesada.
 - × Dando servicio a la carga más pesada al proporcionar un nivel moderadamente reducido de servicio a todos los procesos.

Criterios de Planificación

Para realizar los objetivos de la planificación, un mecanismo de planificación debe considerar lo siguiente:

- La limitación de un proceso a las operaciones de Entrada / Salida: cuando un proceso consigue la CPU, ¿la utiliza solo brevemente antes de generar una petición de Entrada/Salida?
- La limitación de un proceso a la CPU: cuando un proceso obtiene la CPU, ¿tiende a usarla hasta que expira su tiempo?
- Si un proceso es por lote (batch) o interactivo: los usuarios interactivos deben recibir inmediato servicio para garantizar buenos tiempos de respuesta.

- ¿Qué urgencia tiene una respuesta rápida?: por ejemplo, un proceso de tiempo real de un sistema de control que supervise una refinería de combustible requiere una respuesta rápida, más rápida que la respuesta requerida por un proceso en lotes (batch) que deberá entregarse al día siguiente.
- La prioridad de un proceso: a mayor prioridad mejor tratamiento.
- Frecuentemente un proceso genera fallos (carencias) de página:
 - Probablemente los procesos que generan pocos fallos de página hayan acumulado sus “conjuntos de trabajo” en el almacenamiento principal.
 - Los procesos que experimentan gran cantidad de fallos de página aún no han establecido sus conjuntos de trabajo.
 - Un criterio indica favorecer a los procesos que han establecido sus conjuntos de trabajo.
 - Otro criterio indica favorecer a los procesos con una tasa alta de fallos de página ya que rápidamente generarán una petición de Entrada / Salida.
- Frecuentemente un proceso ha sido apropiado por otro de más alta prioridad, lo cual significa lo siguiente:
 - A menudo los procesos apropiados deben recibir un tratamiento menos favorable.
 - Cada vez que el Sistema Operativo asume la sobrecarga para hacer ejecutar este proceso, el corto tiempo de ejecución antes de la apropiación no justifica la sobrecarga de hacer ejecutar al proceso en primer lugar.
- ¿Cuánto tiempo de ejecución real ha recibido el proceso?: un criterio considera que debe ser favorecido un proceso que ha recibido muy poco tiempo de CPU.
- ¿Cuánto tiempo adicional va a necesitar el proceso para terminar?: los tiempos promedio de espera pueden reducirse priorizando los procesos que requieren de un tiempo de ejecución mínimo para su terminación, pero pocas

veces es posible conocer la cantidad de tiempo adicional que cada proceso necesita para terminar

Planificación Apropiativa y No Apropiativa

Las Disciplinas de Planificación pueden ser Apropiativas o No Apropiativas.

Las principales características de la planificación apropiativa son las siguientes:

- Es útil cuando los procesos de alta prioridad requieren atención rápida.
- Es importante para garantizar buenos tiempos de respuesta en sistemas interactivos de tiempo compartido.
- Tiene su costo en recursos, ya que el intercambio de contexto implica sobrecarga y además requiere mantener muchos procesos en el almacenamiento principal, en espera de la CPU, lo que también implica sobrecarga.

Las principales características de la planificación no apropiativa son las siguientes:

- Significa que los trabajos “largos” hacen esperar a los trabajos “cortos”.
- Logra más equidad en el tratamiento de los procesos.
- Logra hacer más predecibles los tiempos de respuesta puesto que los trabajos nuevos de prioridad alta no pueden desplazar a los trabajos en espera.

El diseño de un mecanismo apropiativo hace necesario considerar las arbitrariedades de casi cualquier esquema de prioridades, debido a que muchas veces las propias prioridades no son asignadas de forma significativa. El mecanismo debería ser sencillo pero efectivo y significativo.

Disciplina	Descripción
“Apropiativa”	Una vez que se le ha otorgado la cpu a un proceso, <i>le puede ser retirada</i>
“No Apropiativa”	Una vez que se le ha otorgado la cpu a un proceso, <i>no le puede ser retirada</i>

Tipos de Planificación

Planificación a Plazo Fijo

Ciertos trabajos se planifican para ser terminados en un tiempo específico o plazo fijo. Es una planificación compleja debido a los siguientes factores:

- El usuario debe suministrar anticipadamente una lista precisa de recursos necesarios para el proceso, pero generalmente no se dispone de dicha información.
- La ejecución del trabajo de plazo fijo no debe producir una grave degradación del servicio a otros usuarios.
- El sistema debe planificar cuidadosamente sus necesidades de recursos hasta el plazo fijo, lo que se puede complicar con las demandas de recursos de nuevos procesos que ingresen al sistema.
- La concurrencia de varios procesos de plazo fijo (activos a la vez) puede requerir métodos sofisticados de optimización.
- La administración intensiva de recursos puede generar una considerable sobrecarga adicional.

Planificación Garantizada

Se establecen compromisos de desempeño con el proceso del usuario, por ejemplo, si existen “n” procesos en el sistema, el proceso del usuario recibirá cerca del “1/n” de la potencia de la CPU.

El sistema debe tener un registro del tiempo de CPU que cada proceso ha tenido desde su entrada al sistema y del tiempo transcurrido desde esa entrada.

Con los datos anteriores y el registro de procesos en curso de ejecución, el sistema calcula y determina qué procesos están más alejados por defecto de la relación “1/n” prometida y prioriza los procesos que han recibido menos CPU de la prometida.

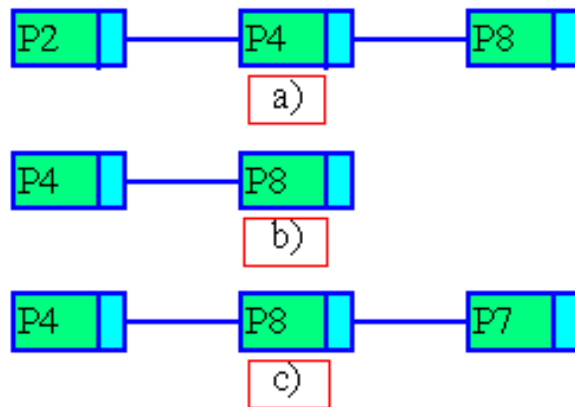
Planificación del Primero en Entrar Primero en Salir (FIFO)

Es muy simple, los procesos se despachan de acuerdo con su tiempo de llegada a la cola de listos. Una vez que el proceso obtiene la CPU, se ejecuta hasta terminar, ya que es una disciplina “no apropiativa”.

Puede ocasionar que procesos largos hagan esperar a procesos cortos y que procesos no importantes hagan esperar a procesos importantes. Es más predecible que otros esquemas. No puede garantizar buenos tiempos de respuesta interactivos.

Suele utilizarse integrado a otros esquemas, por ejemplo, de la siguiente manera:

- Los procesos se despachan con algún esquema de prioridad.
- Los procesos con igual prioridad se despachan “FIFO”.



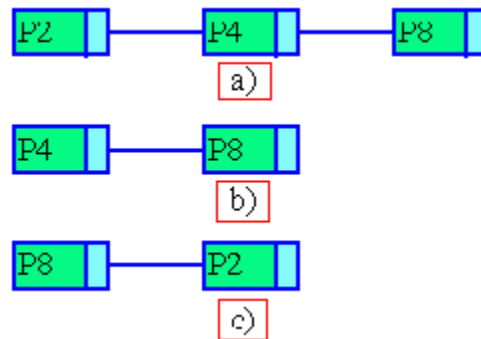
Planificación de Asignación en Rueda (RR: Round Robin)

Los procesos se despachan en “FIFO” y disponen de una cantidad limitada de tiempo de CPU, llamada “división de tiempo” o “cuanto”.

Si un proceso no termina antes de expirar su tiempo de CPU ocurren las siguientes acciones:

1. La CPU es apropiada.
2. La CPU es otorgada al siguiente proceso en espera.
3. El proceso apropiado es situado al final de la lista de listos.

Es efectiva en ambientes de tiempo compartido. La sobrecarga de la apropiación se mantiene baja mediante mecanismos eficientes de intercambio de contexto y con suficiente memoria principal para los procesos.



Tamaño del Cuanto o Quantum

La determinación del tamaño del cuanto es decisiva para la operación efectiva de un sistema computacional.

Los interrogantes son: ¿cuánto pequeño o grande?, ¿cuánto fijo o variable? y ¿Cuánto igual para todos los procesos de usuarios o determinado por separado para cada uno de ellos?

Si el cuanto se hace muy grande, cada proceso recibe todo el tiempo necesario para llegar a su terminación, por lo cual la asignación en rueda ("RR") degenera en "FIFO".

Si el cuanto se hace muy pequeño, la sobrecarga del intercambio de contexto se convierte en un factor dominante y el rendimiento del sistema se degrada, puesto que la mayor parte del tiempo de CPU se invierte en el intercambio del procesador (cambio de contexto) y los procesos de usuario disponen de muy poco tiempo de CPU.

El cuanto debe ser lo suficientemente grande como para permitir que la gran mayoría de las peticiones interactivas requieran de menos tiempo que la duración del cuanto, es decir que el tiempo transcurrido desde el otorgamiento de la CPU a un proceso hasta que genera una petición de Entrada / Salida debe ser menor que el cuanto establecido, de esta forma, ocurrida la petición la CPU pasa a otro proceso

y como el cuanto es mayor que el tiempo transcurrido hasta la petición de Entrada / Salida, los procesos trabajan al máximo de velocidad, se minimiza la sobrecarga de apropiación y se maximiza la utilización de la Entrada / Salida.

El cuanto óptimo varía de un sistema a otro y con la carga, siendo un valor de referencia 100 mseg (cien milisegundos).

Planificación del Trabajo Más Corto Primero (SJF)

Es una disciplina no apropiativa y por lo tanto no recomendable en ambientes de tiempo compartido. El proceso en espera con el menor tiempo estimado de ejecución hasta su terminación es el siguiente en ejecutarse.

Los tiempos promedio de espera son menores que con "FIFO". Los tiempos de espera son menos predecibles que en "FIFO". Favorece a los procesos cortos en detrimento de los largos.

Tiende a reducir el número de procesos en espera y el número de procesos que esperan detrás de procesos largos. Requiere un conocimiento preciso del tiempo de ejecución de un proceso, lo que generalmente se desconoce. Se pueden estimar los tiempos en base a series de valores anteriores.

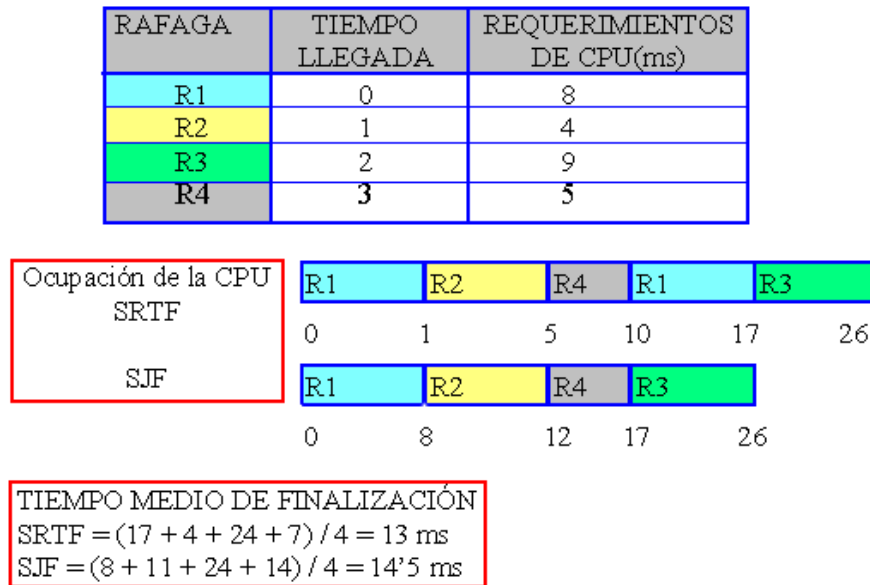
Planificación del Tiempo Restante Más Corto (SRT)

Es la contraparte apropiativa del SJF. Es útil en sistemas de tiempo compartido. El proceso con el tiempo estimado de ejecución menor para finalizar es el siguiente en ser ejecutado.

Un proceso en ejecución puede ser apropiado por un nuevo proceso con un tiempo estimado de ejecución menor. Tiene mayor sobrecarga que la planificación SJF. Debe mantener un registro del tiempo de servicio transcurrido del proceso en ejecución, lo que aumenta la sobrecarga. Los trabajos largos tienen un promedio y una varianza de los tiempos de espera aún mayor que en SJF.

La apropiación de un proceso a punto de terminar por otro de menor duración recién llegado podría significar un mayor tiempo de cambio de contexto (administración del procesador) que el tiempo de finalización del primero.

Al diseñarse los Sistemas Operativos se debe considerar cuidadosamente la sobrecarga de los mecanismos de administración de recursos comparándola con los beneficios esperados.



Ejemplo de SRTF Y SJF

Planificación el Siguiente con Relación de Respuesta Máxima (HRN)

Corrige algunas de las debilidades del SJF, tales como el exceso de perjuicio hacia los procesos (trabajos) largos y el exceso de favoritismo hacia los nuevos trabajos cortos. Es una disciplina no apropiativa.

La prioridad de cada proceso está en función no sólo del tiempo de servicio del trabajo, sino que también influye la cantidad de tiempo que el trabajo ha estado esperando ser servido. Cuando un proceso ha obtenido la CPU, corre hasta terminar.

Las prioridades, que son dinámicas, se calculan según la siguiente fórmula, donde p_r es la "prioridad", t_e es el "tiempo de espera" y t_s es el "tiempo de servicio"

$$p_r = \frac{(t_e + t_s)}{t_s}$$

Planificación por Prioridad

Considera factores externos al proceso. Las ideas centrales son que cada proceso tiene asociada una prioridad y que el proceso ejecutable con máxima prioridad es el que tiene el permiso de ejecución.

Los procesos de alta prioridad podrían ejecutar indefinidamente, ya que el planificador del sistema puede disminuir la prioridad del proceso en ejecución en cada interrupción del reloj. Las prioridades también pueden ser asignadas dinámicamente por el sistema para lograr ciertas metas relacionadas con el procesador o la Entrada / Salida.

Los procesos limitados por la Entrada / Salida (requerimientos intensivos de Entrada/Salida) ocupan mucho de su tiempo en espera de operaciones de Entrada/Salida, por lo tanto:

- Deben tener prioridad para usar la CPU y efectuar la siguiente petición de Entrada/Salida, ya que se ejecutará (la operación de Entrada / Salida) en paralelo con otro proceso que utilice la CPU.
- Si deben esperar mucho tiempo a la CPU estarán ocupando memoria por un tiempo innecesario.

Un algoritmo sencillo consiste en establecer que la prioridad sea $1/f$, donde f es la fracción del último cuanto utilizado por el proceso. Un proceso que utilice 2 mseg (dos milisegundos) de su cuanto de 100 mseg (cien milisegundos) tendrá prioridad 50 (cincuenta). Un proceso que se ejecutó 50 mseg antes del bloqueo tendrá prioridad 2. Un proceso que utilizó todo el cuanto tendrá prioridad 1.

Frecuentemente los procesos se agrupan en “Clases de Prioridad”, en cuyo caso se utiliza la Planificación con Prioridades entre las clases y con Round Robin (RR) dentro de cada clase. Si las prioridades no se reajustan en algún momento, los procesos de las clases de prioridad mínima podrían demorarse indefinidamente.

Hay muchos criterios para definir la prioridad, por ejemplo:

- Según categoría del usuario.

- Según tipo de proceso: sistema, interactivo, o por lotes; o bien, intensivo en CPU o intensivo en E/S.
- Según cuanto hayan ocupado la CPU hasta el momento
- Para evitar que un proceso de baja prioridad sea postergado en demasía, aumentar prioridad mientras más tiempo lleve esperando: envejecimiento (aging).
- Para evitar que un proceso de alta prioridad ejecute por demasiado tiempo, se le puede ir bajando la prioridad.

Parámetro	Linux	Windows
Clases de planificación	3	2
1. Prioridades normales (dinámicas)	40; de -20 a 19	15; de 1 a 15
2. Prioridades de tiempo real FCFS (fijas)	100; de 0 a 99	16; de 16 a 31
3. Prioridades de tiempo real con rodaja (fijas)	100; de 0 a 99	--
Orden de importancia de la prioridad	Baja→Alta	Alta→Baja

Colas de Retroalimentación de Niveles Múltiples

Proporcionan una estructura para lograr los siguientes objetivos:

- Favorecer trabajos cortos.
- Favorecer trabajos limitados por la Entrada / Salida para optimizar el uso de los dispositivos de Entrada / Salida.
- Determinar la naturaleza de un trabajo lo más rápido posible y planificar el trabajo (proceso) en consecuencia.

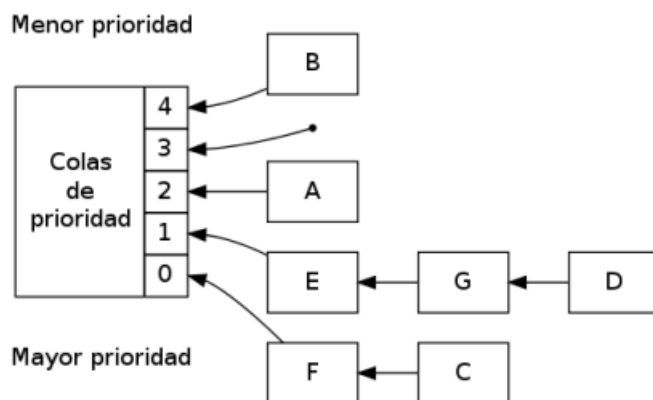
Un nuevo proceso entra en la red de línea de espera al final de la cola superior. Se mueve por esta cola “FIFO” hasta obtener la CPU. Si el trabajo termina o abandona la CPU para esperar por la terminación de una operación de Entrada/Salida o la terminación de algún otro suceso, el trabajo abandona la red de línea de espera.

Si su cuanto expira antes de abandonar la CPU voluntariamente, el proceso se coloca en la parte trasera de la cola del siguiente nivel inferior. El trabajo recibe servicio al llegar a la cabeza de esta cola si la primera está vacía.

Mientras el proceso continúe consumiendo totalmente su cuanto, en cada nivel, continuará moviéndose hacia el final de las colas inferiores. Generalmente hay una cola en la parte más profunda a través de la cual el proceso circula en asignación de rueda hasta que termina.

Existen esquemas en los que el cuanto otorgado al proceso aumenta a medida que el proceso se mueve hacia las colas de los niveles inferiores, en tal caso, cuanto más tiempo haya estado el proceso en la red de línea de espera, mayor será su cuanto cada vez que obtiene la CPU y no podrá obtener la CPU muy a menudo debido a la mayor prioridad de los procesos de las colas superiores.

Un proceso situado en una cola dada no podrá ser ejecutado hasta que las colas de los niveles superiores estén vacías. Un proceso en ejecución es apropiado por un proceso que llegue a una cola superior. Es un mecanismo adaptable, es decir que se adapta a cargas variables.



Política Versus Mecanismo de Planificación

Puede ocurrir que haya procesos con muchos procesos hijos ejecutándose bajo su control, por ejemplo, un proceso en un DBMS con procesos hijos atendiendo funciones específicas, tales como, análisis de interrogantes, acceso a discos, etc.

Es posible que el proceso principal (padre) pueda identificar la importancia (o criticidad) de sus procesos hijos, pero los planificadores analizados no aceptan datos de los procesos de usuario relativos a decisiones de planificación.

La solución es separar el mecanismo de planificación de la política de planificación, para ello se parametriza el algoritmo de planificación y los parámetros pueden ser determinados por medio de procesos del usuario; así el mecanismo está en el núcleo del Sistema Operativo pero la política queda establecida por un proceso del usuario.

Planificación de Dos Niveles

Los esquemas analizados hasta ahora suponen que todos los procesos ejecutables están en la memoria principal.

Si la memoria principal es insuficiente, ocurrirá lo siguiente:

- Habrá procesos ejecutables que se mantengan en disco.
- Habrá importantes implicaciones para la planificación, tales como las siguientes:
 - El tiempo de alternancia entre procesos para traer y procesar un proceso del disco es considerablemente mayor que el tiempo para un proceso que ya está en la memoria principal.
 - Es más eficiente el intercambio de los procesos con un planificador de dos niveles.

El esquema operativo de un planificador de dos niveles es como sigue:

1. Se carga en la memoria principal cierto subconjunto de los procesos ejecutables.
2. El planificador se restringe a ellos durante cierto tiempo.
3. Periódicamente se llama a un planificador de nivel superior para efectuar las siguientes tareas:
 - a. Eliminar de la memoria los procesos que hayan permanecido en ella el tiempo suficiente.
 - b. Cargar a memoria los procesos que hayan estado en disco demasiado tiempo.

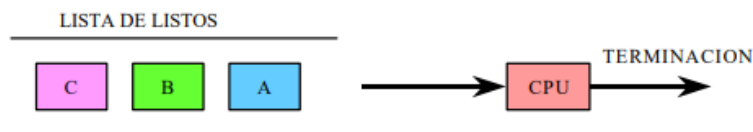
4. El planificador de nivel inferior se restringe de nuevo a los procesos ejecutables que se encuentren en la memoria.
5. El planificador de nivel superior se encarga de desplazar los procesos de memoria a disco y viceversa.

Los criterios que podría utilizar el planificador de nivel superior para tomar sus decisiones son los que se indican a continuación:

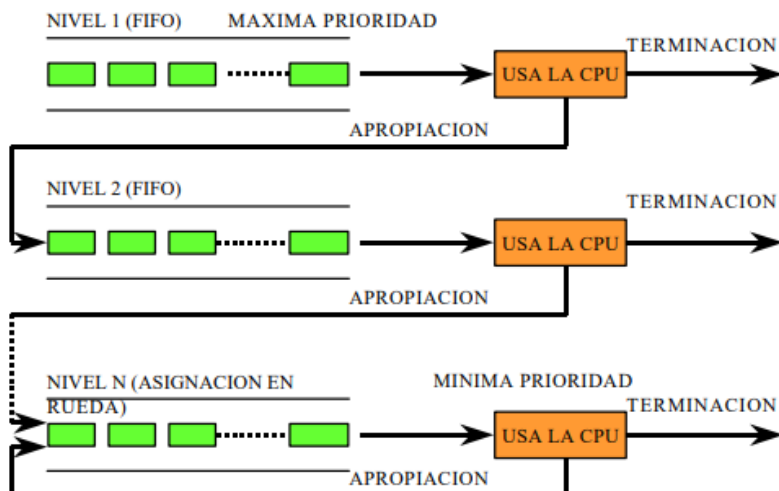
- ¿Cuánto tiempo ha transcurrido desde el último intercambio del proceso?
- ¿Cuánto tiempo de CPU ha utilizado recientemente el proceso?
- ¿Qué tan grande es el proceso? (generalmente los procesos pequeños no causan tantos problemas en este sentido).
- ¿Qué tan alta es la prioridad del proceso?

El planificador de nivel superior podría utilizar cualquiera de los métodos de planificación analizados.

PLANIFICACION PRIMERO EN ENTRAR PRIMERO EN SALIR



COLAS DE RETROALIMENTACION DE NIVELES MULTIPLES



Conclusión

En conclusión, la planificación de procesos es una parte vital de cualquier sistema operativo moderno. Es la manera en que el sistema decide qué proceso debe ejecutarse en cada momento y cómo se distribuyen los recursos del sistema entre los distintos procesos. Existen varios algoritmos de planificación de procesos que los sistemas operativos pueden utilizar dependiendo de las necesidades específicas del sistema y los procesos en ejecución. Sin embargo, el objetivo principal de cualquier algoritmo de planificación de procesos es maximizar el rendimiento del sistema y garantizar una respuesta rápida y sin interrupciones para los usuarios.

Es importante destacar que la planificación de procesos no es una tarea fácil, ya que debe tener en cuenta una gran cantidad de factores, como la prioridad de los procesos, la cantidad de recursos que necesitan, la cantidad de procesos en ejecución, entre otros. Además, la planificación de procesos debe ser lo suficientemente flexible como para adaptarse a las condiciones cambiantes del sistema en tiempo real.

POLITICAS DE PROCESOS

Las políticas de procesos son fundamentales en la administración de sistemas operativos, ya que permiten determinar cómo se asignan y administran los recursos del sistema. Las políticas de procesos en los sistemas operativos son esenciales para garantizar que los procesos se ejecuten de manera eficiente y efectiva, lo que ayuda a maximizar el rendimiento del sistema operativo. En este ensayo, se discutirán las políticas de procesos en los sistemas operativos, su importancia y cómo se aplican en la gestión de procesos.

Las políticas de procesos se refieren a las reglas y estrategias utilizadas por los sistemas operativos para administrar los procesos y sus recursos. Las políticas de procesos pueden variar según el sistema operativo y el entorno en el que se está ejecutando el sistema. Algunas de las políticas de procesos comunes en los sistemas operativos incluyen:

Asignación de recursos: la política de asignación de recursos se refiere a cómo se asignan los recursos del sistema a los procesos. Esto incluye recursos como CPU, memoria, E/S, etc. La asignación de recursos puede ser dinámica o estática. En la asignación dinámica de recursos, los recursos se asignan en función de las necesidades de los procesos. En la asignación estática de recursos, los recursos se asignan de manera predeterminada y se mantienen fijos.

Prioridad de procesos: la política de prioridad de procesos determina qué proceso tiene prioridad sobre otros procesos. Los procesos pueden tener diferentes niveles de prioridad en función de su importancia y urgencia. Los procesos de alta prioridad obtienen más recursos del sistema y tienen una mayor probabilidad de ejecutarse antes que los procesos de baja prioridad.

Planificación de procesos: la política de planificación de procesos se refiere a cómo se programan los procesos para su ejecución. El planificador de procesos decide qué proceso se ejecutará a continuación y durante cuánto tiempo se ejecutará.

Control de errores: la política de control de errores se refiere a cómo se manejan los errores y las excepciones en los procesos. Esto incluye cómo se recuperan los procesos de errores y cómo se informan los errores a los usuarios.

Importancia de las políticas de procesos

Las políticas de procesos son esenciales para la gestión eficiente de los recursos del sistema y la ejecución de procesos en los sistemas operativos. Las políticas de procesos permiten la asignación efectiva de recursos del sistema a los procesos, lo que ayuda a maximizar el rendimiento del sistema. Además, las políticas de procesos permiten la programación de procesos para su ejecución y la gestión de excepciones y errores.

Las políticas de procesos también son importantes para la seguridad del sistema operativo. La asignación adecuada de recursos del sistema ayuda a prevenir la sobrecarga del sistema y la falta de recursos. La priorización de procesos también ayuda a garantizar que los procesos críticos obtengan los recursos necesarios para ejecutarse de manera efectiva. Además, la política de control de errores permite la gestión efectiva de errores y excepciones, lo que ayuda a prevenir la corrupción de datos y otros problemas de seguridad.

Además, las políticas de procesos también pueden ser influenciadas por las características de hardware del sistema. Por ejemplo, en sistemas con múltiples núcleos de procesador, se puede utilizar la política de asignación de procesos para asegurarse de que los procesos críticos tengan asignados núcleos específicos, lo que puede mejorar el rendimiento y la eficiencia del sistema.

Por otro lado, las políticas de procesos también pueden ser adaptadas para satisfacer las necesidades específicas de diferentes tipos de aplicaciones. Por ejemplo, para aplicaciones de tiempo real, se puede utilizar una política de planificación en tiempo real para asegurarse de que los procesos críticos se ejecuten dentro de los límites de tiempo establecidos. Por otro lado, para aplicaciones de procesamiento en lote, se puede utilizar una política de planificación

basada en prioridades para asegurarse de que los trabajos se ejecuten en el orden correcto.

Además, la gestión de procesos es especialmente importante en sistemas con múltiples usuarios y múltiples procesos. En estos sistemas, las políticas de procesos son cruciales para garantizar que todos los procesos obtengan los recursos que necesitan sin afectar la capacidad de otros procesos para operar eficientemente. Además, las políticas de procesos pueden ayudar a garantizar la seguridad del sistema operativo al limitar los recursos disponibles para procesos no autorizados o maliciosos.

Aplicación de las políticas de procesos en la gestión de procesos

Las políticas de procesos se aplican en la gestión de procesos de varias maneras.

- En primer lugar, la política de asignación de recursos determina cómo se asignan los recursos del sistema a los procesos. La asignación dinámica de recursos puede garantizar que los procesos reciban los recursos que necesitan en tiempo real, lo que ayuda a maximizar la eficiencia del sistema. Además, la asignación estática de recursos puede ayudar a garantizar que los recursos críticos estén disponibles para los procesos que los necesitan.
- En segundo lugar, la política de prioridad de procesos permite que los procesos críticos obtengan los recursos que necesitan para ejecutarse de manera efectiva. Los procesos críticos a menudo tienen una mayor prioridad que los procesos no críticos, lo que les permite obtener los recursos necesarios para ejecutarse de manera efectiva. La priorización de procesos también puede ayudar a garantizar que los procesos importantes se ejecuten en un momento adecuado para maximizar su eficiencia máxima a la hora de ejecutarlos.
- En tercer lugar, la política de planificación de procesos se utiliza para programar la ejecución de procesos. La planificación de procesos puede ser

en tiempo real o basada en prioridades. En la planificación en tiempo real, los procesos se programan en función del momento en que se reciben y de la disponibilidad de recursos del sistema. En la planificación basada en prioridades, los procesos se programan en función de su nivel de prioridad.

- Por último, la política de control de errores permite la gestión efectiva de errores y excepciones en los procesos. Los errores y excepciones pueden ser manejados de varias maneras, como informar al usuario, reiniciar el proceso o cambiar la asignación de recursos del sistema.

Conclusión

En resumen, las políticas de procesos son fundamentales en la administración de sistemas operativos. Las políticas de procesos permiten la asignación eficiente de recursos del sistema a los procesos, la priorización de procesos, la programación de procesos y la gestión de errores y excepciones. Las políticas de procesos también son importantes para la seguridad del sistema operativo y para maximizar el rendimiento del sistema. Los desarrolladores de sistemas operativos deben comprender la importancia de las políticas de procesos y cómo se aplican en la gestión de procesos para garantizar un sistema operativo de calidad.