**Algoritmos de planificación**

En la planificación de procesos existe un elemento que apoya en dicha planificación conocido como cola de planificación. La cola de planificación agrupa a todos los procesos que solicitan el uso de algún recurso. Sobre las colas de planificación se implementan algoritmos de planificación.

Un algoritmo de planificación permite seleccionar un proceso bajo algún criterio de planificación de la cola de planificación para asignarle algún recurso requerido. Los sistemas operativos pueden utilizar comúnmente los siguientes algoritmos de planificación:

* Algoritmo primero en llegar primero en servirse (FCFS)
* Algoritmo primero el trabajo más corto (SJF) //Nunca se ha implementado en un S.O.
* Algoritmo de prioridad //Da prioridad a cada proceso
* Algoritmo por turno circular (Round Robin)

Estos algoritmos de planificación pueden analizarse su eficiencia mediante algún criterio de análisis, los criterios de análisis comúnmente empleados son los siguientes:

* Aprovechamiento de la CPU: Este criterio se basa se basa en el porcentaje de uso de la CPU por los ´procesos, un aprovechamiento óptimo se encuentra en el rango del 70% al 90%
* Rendimiento: Este criterio se basa en el número de procesos ejecutados en cierta unidad de tiempo, a mayor número de procesos ejecutados mejor aprovechamiento
* Tiempo de espera: Este criterio se basa en el tiempo que un proceso debe esperar en alguna cola de planificación
* Tiempo de respuesta: Este criterio se basa en el tiempo transcurrido hasta el momento en que se genera la primera respuesta en la ejecución de un proceso
* Tiempo de retorno: Este criterio se basa en el tiempo total que un proceso tarda desde su inicio hasta su terminación, se calcula a través de la siguiente expresión

TRET = TESP + TMEM + TE/S +TEJE

Donde:

TESP: Tiempo de espera del proceso

TMEM: Tempo de memoria

TE/S: Tiempo de acceso a entrada/salida

TEJE: Tiempo de duración del proceso

En la planificación de procesos se identifican dos tipos de planificación según la forma de uso del recurso asignado a un proceso, estos tipos son:

* Planificación expropiativa: En este tipo de planificación un proceso de mayor importancia puede interrumpir a otro proceso que esté utilizando un recurso, quitándole el recurso el proceso de mayor importancia.
* Planificación no expropiativa: En este tipo de planificación un proceso bajo ninguna circunstancia podrá ser interrumpido por cualquier otro proceso, aun cuando sea un proceso de mayor importancia el que sdesea interrumpir

El análisis de los algoritmos de planificación, se utiliza una herramienta gráfica para ubicar los procesos en simulación de una cola de planificación, esta herramienta gráfica es un diagrama de GANTT, el cual permite asociar a los procesos sus tiempos de duración.

**Algoritmo Primero en Llegar Primero en Servirse (FCFS) (Estrictamente no expropativo)**

Este algoritmo considera como criterio de planificación el orden de llegada de los procesos, es decir el primer proceso en aparecer será el primer proceso en ser seleccionado. Los pasos para analizar el algoritmo son los siguientes

1. Construir el diagrama de GANTT de acuerdo al órden de llegada de los procesos
2. Calcular el tiempo de espera de cada proceso a través de la siguiente expresión:

Donde:

es el último tiempo de aparición en el diagrama del proceso

Lapso de tiempo previo de aparición en el diagrama del proceso

Tiempo de llegada del proceso

1. Calcular el tiempo de espera promedio del grupo de procesos mediante la siguiente expresión

Donde:

es el tiempo de espera del proceso

N es el número de procesos en el grupo

**Ejemplo:** *Considere el siguiente grupo de procesos*

|  |  |  |
| --- | --- | --- |
| **Proceso** | **Duración** | **Tiempo de llegada** |
| *1* | *10 U.T.* | *0 U.T.* |
| *2* | *5 U.T.* | *0 U.T.* |
| *3* | *7 U.T.* | *0 U.T.* |
| *4* | *2 U.T.* | *0 U.T.* |
| *5* | *4 U.T.* | *0 U.T.* |

Calcular el tiempo de espera de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando el algoritmo de FCFS

>>Solución

1. Diagrama de GANTT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proceso 1 | Proceso 2 | Proceso 3 | Proceso 4 | Proceso 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 10 | 15 | 22 | 24 | 28 | U.T. |

1. Tiempos de espera
2. Tiempo de espera promedio

NOTAS: Se recomienda este algoritmo si se tiene soporte para procesamiento por lotes.

Las características principales del algoritmo FCFS son las siguientes

* Únicamente permite planificación no expropiativa
* Genera normalmente el peor tiempo de espera promedio entre todos los algoritmos de planificación
* Solo se implementa en sistemas operativos con soporte de procesamiento por lotes

**Algoritmo primero el trabajo más corto (SJF)**

Este algoritmo considera como criterio de planificación la duración de los procesos es decir se selecciona primero al proceso cuya duración sea la más corta. Los pasos para analizar el algoritmo son los siguientes:

1. Construir el diagrama de GANTT de acuerdo a la duración más corta de los procesos
2. Calcular el tiempo de espera de cada proceso utilizando la expresión dada en el paso 2 del algoritmo FCFS
3. Calcular el tiempo de espera promedio del grupo de procesos utilizando la expresión dada en el paso 3 del algoritmo FCFS

**Ejemplo:** *Considere el mismo grupo de procesos usados en el algoritmo FCFS, calculando el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando el algoritmo SJF*

*>>Solución:*

1. *Diagrama de GANTT*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proceso 4 | Proceso 5 | Proceso 2 | Proceso 3 | Proceso 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 6 | 11 | 18 | 28 | U.T. |

1. *Tiempos de espera*
2. Tiempo de espera promedio

Las características principales del algoritmo SJF son las siguientes:

* Permite planificación tanto expropiativa como no expropiativa
* Genera el mejor tiempo de espera promedio entre todos los algoritmos de planificación
* No se implementa en los sistemas operativos debido a la complejidad en su funcionamiento

Como ejemplo de la planificación expropiativa del algoritmo SJF, considere el siguiente cálculo de procesos

|  |  |  |
| --- | --- | --- |
| **Proceso** | **Duración** | **Tiempo llegado** |
| 1 | 10 U.T. | 0 U.T. |
| 2 | 20 U.T. | 10 U.T. |
| 3 | 2 U.T. | 12 U.T. |
| 4 | 3 U.T. | 15 U.T. |
| 5 | 5 U.T. | 18 U.T. |

Calcular el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando el algoritmo SJF con planificación apropiativa

>>Solución:

1. Diagrama de GANTT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Proceso 1 | Proceso 3 | Proceso 1 | Proceso 4 | Proceso 5 | Proceso 2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 12 | 14 | 17 | 20 | 25 | 45 | U.T. |

1. Tiempos de espera

NOTA: Este algoritmo solo se usa como referencia no está implementado en ningún sistema operativo.

**Algoritmo por prioridad**

Este algoritmo considera como criterio de planificación la prioridad asociada al proceso, es decir se seleccionará primero a los procesos con la prioridad más alta. Los pasos para analizar el algoritmo son los siguientes:

1. Construir el diagrama de Gantt de acuerdo a la prioridad más alta de los procesos
2. Calcular el tiempo de espera de cada proceso utilizando la expresión dada en el paso 2 del algoritmo FCFS
3. Calcular el tiempo de espera promedio del grupo de procesos utilizando la expresión dada en el paso 3 del algoritmo FCFS

**Ejemplo:** *Considere el mismo grupo de procesos utilizado en el ejemplo del algoritmo FCFS con las siguientes prioridades*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Proceso*** | ***Duración*** | ***Prioridad*** | ***Tiempo de llegada*** |
| *1* | *10 U.T.* | *4* | *0 U.T.* |
| *2* | *7 U.T.* | *3* | *0 U.T.* |
| *3* | *5 U.T.* | *3* | *0 U.T.* |
| *4* | *2 U.T.* | *1* | *0 U.T.* |
| *5* | *4 U.T.* | *2* | *0 U.T.* |

Calcule el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando el algoritmo por prioridad, siendo 1 la prioridad más alta

>>Solución

1. *Diagrama de GANTT*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proceso 4 | Proceso 5 | Proceso 2 | Proceso 3 | Proceso 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 6 | 13 | 18 | 28 | U.T. |

1. Tiempos de espera

1. Tiempo De espera promedio

Las características principales del algoritmo por prioridad son las siguientes

* Permite planificación tanto expropiativa como no expropiativa
* Genera un tiempo de espera promedio normalmente eficiente
* Es uno de los algoritmos implementado en los siguientes sistemas operativos actuales

**Algoritmo por turno (Round Robin)**

Este algoritmo considera como criterio de planificación el orden de llegada de los procesos (idéntico al algoritmo FCFS), agregando un lapso de tiempo en el cual se le permite a un proceso utilizar el recurso asignado, este lapso de tiempo es conocido como quantum. Los pasos para analizar el algoritmo son los siguientes:

1. Construir el diagrama de Gantt de acuerdo al orden de llegada de los procesos y al cuanto asignado
2. Calcular el tiempo de espera de cada proceso utilizando la expresión dada en el paso 2 del algoritmo FCFS
3. Calcular el tiempo de espera promedio del grupo de procesos utilizando la expresión dada en el paso 3 del algoritmo FCFS

**Ejemplo:** *Considere el mismo grupo de procesos usado en el ejemplo del algoritmo FCFS, calcule el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando el algoritmo por turno circular con un cuanto de 4 U.T.*

*>>Solución*

1. *Diagrama de GANTT*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proceso 1 | Proceso 2 | Proceso 3 | Proceso 4 | Proceso 5 | Proceso 1 | Proceso 2 | Proceso 3 | Proceso 1 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 4 | 8 | 12 | 14 | 18 | 22 | 25 | 26 | 28 | U.T. |

1. Tiempos de espera

1. Tiempo De espera promedio

Las características principales del algoritmo por turno circular son las siguientes

* Permite únicamente planificación expropiativa
* Genera un tiempo de espera promedio considerado como aceptable pro su equidad
* En todos los sistemas operativos actuales se implementa

**Colas de múltiples niveles**

Los algoritmos de planificación se implementan en un sistema operativo a través de una estructura conocida como cola de múltiples niveles. Una cola de múltiples niveles consiste en una división de una cola de planificación asociada a un recurso en subcolas. Existen dos tipos de colas de múltiples niveles:

* **Cola de múltiples niveles simple**: Esta cola se caracteriza porque una vez que un proceso se asigna a una subcola bajo ninguna circunstancia puede cambiar a otra subcola (No era posible cambiar de una subcola a otra la cual tuviese un algoritmo de planificación más eficiente)
* **Cola de múltiples niveles retroalimentada**: Esta cola se caracteriza por que un proceso asignado a una subcola puede cambiar a otra subcola bajo algún criterio. (A diferencia de las colas de múltiples niveles simples estas si permiten cambiar de subcolas)

Sobre cada subcola de la cola de múltiples niveles es posible implementar un algoritmo de planificación particular, además existe un algoritmo de planificación encargado de planificar todas las colas de forma global.

Esta estructura de colas múltiples niveles es la manera en que los sistemas operativos actuales implementan los algoritmos de planificación en la práctica

# 2.3 Comunicación entre procesos

Los procesos desde el punto de vista de la comunicación pueden clasificarse en dos tipos:

* Procesos independientes: estos procesos se caracterizan porque su ejecución no requiere comunicarse con otros procesos para completar dicha ejecución
* Procesos cooperativos: Estos procesos se caracterizan porque su ejecución requiere comunicarse con otros procesos para completar dicha ejecución

Los procesos cooperativos requieren del apoyo del sistema operativo para llevar a cabo su comunicación. Los sistemas operativos proporcionan un conjunto de mecanismos de comunicación entre procesos conocidos como IPC (Comunicación Inter Procesos). Los IPCs

* Tuberías: Son los archivos creados y controlados internamente por el sistema operativo, sirven como búferes en los cuales se almacenarán los datos a comunicar. Implementa primitivas para la lectura y escritura de datos, los datos a comunicar se manejan como mensajes sin formato. Únicamente soportan comunicación unidireccional con una conectividad uno a uno. (Esta solo se realiza entre procesos)
* Memoria compartida: Es una región de memoria creada y controlada por el sistema operativo asociada a un proceso, la región de memoria creada se anexa al espacio de direcciones del proceso que utilizará la memoria compartida. La lectura y escrituro de datos en la memoria se realiza a través de acceder a la direcciona de memoria requerida mediante algún apuntador. La memoria compartida puede ser accedida por todos los procesos que lo soliciten, siendo necesario la utilización de mecanismos de sincronización para asegurar la consistencia de datos compartidos ente procesos (El proceso que necesita la memoria compartida necesita pedir primeramente permiso al SO para poder utilizar memoria compartida además el proceso debe anexar la memoria compartida a su ruta de direcciones, una vez realizado lo anterior puede escribir y leer sobre esta región de memoria. Para la lectura y escritura se requiere un apuntador a la dirección de la región de memoria. También será necesario tener algoritmos que permiten tener una sincronización con los procesos.)
* Sockets: Son similares en funcionamiento a las tuberías, sin embargo, la implementación se apoya en el administrador de red, permitiendo tanto una comunicación local como remota entre procesos, es decir los procesos a comunicar se pueden encontrar en la misma computadora o en computadoras diferentes. Implementan primitivas para enviar y recibir información mediante el paso de mensajes con formato. Los mensajes mantienen un formato especificado por el protocolo de red utilizado según la capa de red en el que el socket se implemente. La comunicación soportada es bidireccional con una conectividad uno a muchos. ()

**Sincronización de procesos:**

La comunicación entre procesos involucra frecuentemente el acceso a recursos compartidos entre los procesos, según la secuencia de acceso al recurso compartido se tendrá consistencia o inconsistencia en la información manipulada en el recurso compartido. Para garantizar un acceso correcto y consistente de la información compartida entre los procesos, los sistemas operativos ofrecen un conjunto de mecanismos de sincronización los cuales son:

* Semáforos
* Monitores

# 2.4 Semáforos:

Los semáforos son un mecanismo de sincronización consistente en una variable conocida como variable del semáforo, sobre esta variable es posible realizar dos operaciones relacionadas con la prueba y acceso a la variable del semáforo, y con la liberación de la variable del semáforo.

Las dos operaciones que se efectúan sobre la variable del semáforo se llevan a cabo de forma atómica. Los sistemas operativos ofrecen llamadas al sistema para la creación y operación de los semáforos tanto propias como por terceros

(De manera atómica quiere decir, si hay algún problema interno con la ejecución se deshacen todas las operaciones previas y se regresa hasta un punto en el cual sirve nuevamente o bien la variable es estable. Hay dos tipos de semáforos los binarios y los n-ários pero los que siempre se usan son los binarios. Los n-arios se usan si y solo si hay varios procesos que ocupan la variable de semáforo)

# 2.5 Monitores:

Los monitores son un mecanismo de sincronización basado en un objeto, este objeto implementa internamente atributos y métodos para llevar a cabo la creación y manipulación de las variables de sincronización asociadas al monitor. Los monitores usualmente son implementados externamente al sistema operativo al sistema operativo a través de una biblioteca de clases en algún lenguaje de programación orientado a objetos.

La finalidad de estos mecanismos de sincronización es garantizar un aspecto conocido como exclusión mutua entre procesos, la exclusión mutua consiste en garantizar que un solo proceso a la vez puede estar en su sección crítica. La sección crítica es un conjunto de instrucciones que manipulan algún recurso compartido por el proceso.

# 2.6 Problemas clásicos de comunicación

La comunicación entre procesos puede generar algunos que han sido considerados como clásicos pro los sistemas operativos, algunos de estos problemas clásicos son los siguientes:

* Productor / Consumidor (Se tiene un buffer limitado [Del productor] el consumidor extrae del buffer. Se pueden producir dos problemas
* 1 que el productor sea más rápido que el consumidor como se tiene un buffer limitado puede desbordarse y hay información perdida
* 2 que el consumidor sea más rápido que el productor, como el consumidor es más rápido puede que se termine la información y el consumidor se quede ocioso)
* Lectores / Escritores (Se relaciona más con archivos. Se debe evitar que todas las lecturas se realicen antes que las escrituras, para tener la información más “actual” o bien consistente.)
* Cena de filósofos (Pueden estar en dos estados, pensando o comiendo, para comer se tiene un cubierto que se comparte con el filósofo que está a su lado. El punto es que ningún filósofo muera de hambre ya que por mera lógica al estar compartiendo los cubiertos si un filósofo se tarda demasiado en comer y un filósofo que está a su lado pasa del estado pensando al de comer y quien tiene el cubierto se tarda demasiado entonces el que necesita el cubierto podría morir de hambre, el punto aquí es evitar esto)

Estos problemas de comunicación son resueltos mediante el uso de algún mecanismo de sincronización de los mencionados anteriormente, o a través de cualquier otro mecanismo del que disponga el sistema operativo.

Para acceder a la sección crítica debemos activar la sincronización, una vez termina se libera la sincronización y entra el siguiente proceso nuevamente activando la sincronización y así sucesivamente.

**TAREA: Resolver en equipo los siguientes ejercicios**

1. **Considere el siguiente grupo de procesos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Proceso | Duración | Prioridad | Tiempo llegada |
| 1 | 20 U.T. | 3 | 0 U.T. |
| 2 | 2 U.T. | 1 | 5 U.T. |
| 3 | 5 U.T. | 2 | 6 U.T. |
| 4 | 15 U.T. | 3 | 10 U.T. |
| 5 | 7 U.T. | 3 | 50 U.T. |
| 6 | 1 U.T. | 2 | 56 U.T. |
| 7 | 4 U.T. | 1 | 57 U.T. |
| 8 | 11 U.T. | 3 | 70 U.T. |
| 9 | 3 U.T. | 2 | 76 U.T. |
| 10 | 2 U.T. | 1 | 77 U.T. |

Calcular el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizando:

1. Algoritmo FCFS
2. Algoritmo SJF expropiativo y no expropiativo
3. Algoritmo por prioridad expropiativo y no expropiativo(El valor de 1 es la prioridad más alta)
4. Algoritmo por turno circular con un cuanto de 5 U.T.
5. **Considere el siguiente grupo de procesos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Proceso | Duración | Prioridad | Tiempo llegada |
| 1 | 8 U.T. | 3 | 0 U.T. |
| 2 | 3 U.T. | 3 | 3 U.T. |
| 3 | 15 U.T. | 4 | 9 U.T. |
| 4 | 5 U.T. | 1 | 13 U.T. |
| 5 | 2 U.T. | 3 | 15 U.T. |
| 6 | 1 U.T. | 2 | 16 U.T. |
| 7 | 25 U.T. | 5 | 45 U.T. |
| 8 | 17 U.T. | 3 | 55 U.T. |
| 9 | 6 U.T. | 4 | 60 U.T. |
| 10 | 2 U.T. | 1 | 64 U.T. |
| 11 | 10 U.T. | 3 | 70 U.T. |
| 12 | 3 U.T. | 2 | 73 U.T. |

Calcular el tiempo de espera de cada proceso y el tiempo de espera promedio del grupo de procesos utilizado

1. Algoritmo FCFS
2. Algoritmo SJF expropiativo y no expropiativo
3. Algoritmo por prioridad expropiativo y no expropiativo (El valor de 1 es la prioridad más alta)
4. Algoritmo por turno circular con un cuanto de 3 U.T.

Fecha de entrega: Próxima clase