

INTRODUCCIÓN

Sistema Operativo

Definición

- Software que permite la abstracción de los recursos de Hardware a las aplicaciones y se encarga de la administración de los mismos.
- Programa que controla la ejecución de los programas de aplicación y que actúa como interfaz entre las aplicaciones del usuario y el hardware.

Como una máquina extendida

Tiene como objetivo crear buenas abstracciones para después implementar y administrar los objetos abstractos entonces creados.

Como un administrador de recursos

Tiene como objetivo proporcionar una asignación ordenada y controlada de los procesadores, memorias y dispositivos de E/S, entre los diversos programas que compiten por estos recursos.

Incluye el multiplexaje de recursos en dos formas:

- En tiempo
Programas y usuarios toman turnos para utilizarlo (cola de impresión).
- En espacio
Cada cliente obtiene una parte del recurso (división de memoria).

Modos de ejecución

- Modo kernel
 - El sistema operativo tiene acceso completo a todo el hardware y puede ejecutar cualquier instrucción que la máquina sea capaz de ejecutar.
- Modo usuario
 - Sólo un subconjunto de las instrucciones de máquina es permitido. En particular, las instrucciones que afectan el control de la máquina o que se encargan de la E/S están prohibidas para los programas en modo usuario.

Arranque de la computadora

1. El BIOS inicia su ejecución.

- Hace pruebas para ver cuánta RAM hay instalada y si el teclado u otros dispositivos básicos responden. También configura nuevos dispositivos.
- Determina el dispositivo de arranque disco a bootear
- Lee, coloca en memoria y ejecuta el primer sector del dispositivo contiene un programa que examina la tabla de particiones y además, lee el bootloader

2. Sistema operativo consulta a BIOS la información de configuración.

- Comprueba drivers y los carga en el kernel
- Inicializa tablas, crea procesos en segundo plano y arranca el inicio de sesión

Tipos de Sistemas Operativos

De mainframe

- Orientados al procesamiento de muchos trabajos que requieren muchas operaciones E/S a la vez.
- Ofrecen 3 tipos de servicios
- Procesamiento por lotes
 - Procesa trabajos de rutina sin que haya un usuario interactivo presente.
- Procesamiento de transacciones
 - Manejan grandes cantidades de pequeñas peticiones
- Tiempo compartido
 - Permiten que varios usuarios remotos ejecuten trabajos en la computadora al mismo tiempo.

Servidores web, servidores para sitios de e-commerce, servidores para transacciones B2B

De servidores

- Dan servicio a varios usuarios de forma simultánea a través de una red y les permiten compartir recursos de hardware y software.
- Ejecutados en servidores, estaciones de trabajo o mainframes.

Solaris, Linux, Windows Server 2022

De multiprocesadores

Son sistemas de varias CPU.

- Variación de los **sistemas operativos de servidores** con características especiales para la *comunicación, conectividad y consistencia*.

Windows, Linux

De computadoras personales

Tienen como objetivo proporcionar buen soporte para un sólo usuario.

- Utilizados para el procesamiento de texto, hojas de cálculo y el acceso a internet.

Windows, Linux, Macintosh

De computadoras de bolsillo

Computadora de bolsillo: Computadora que cabe en los bolsillos y realiza una pequeña variedad de funciones.

- Proporcionan telefonía, fotografía digital y otras funciones.
- Ejecutan aplicaciones desarrolladas por terceros.

Se diferencian con las PC's porque no tienen discos duros de gran tamaño.

Android, iOS, Symbian OS, Palm OS

Integrados

Operan en computadoras que controlan dispositivos sin software instalado por el usuario. Quemado en la ROM

Hornos de microondas, televisiones, autos,..., reproductores MP3.

De nodos sensores

Redes de pequeños nodos.

Nodos: Pequeñas computadoras intercomunicadas con una estación base usando comunicación inalámbrica.

- Utilizados para proteger perímetros, resguardar fronteras, detectar incendios, medir temperaturas.
- Ejecuta un OS pequeño, que responde a eventos externos o realiza mediciones periódicas.
- Sus programas se queman en la fabricación.

De tiempo real

Tienen el tiempo como un parámetro clave.

- Deben proveer garantías absolutas de que cierta acción ocurrirá en un instante determinado.

Aeronáutica, milicia, control de procesos industriales.

De tarjetas inteligentes

- Tienen severas restricciones de poder de procesamiento y memoria.
- Pueden realizar *pagos electrónicos* o varias funciones.
- Administra y protege los recursos.
- Se energizan mediante contactos con el lector o mediante inducción.

Multiplexar: Enviar varias señales mediante un medio de transmisión.

PROCESO

Definicion

- Programa en ejecución. Asociado a un espacio de memoria y a un conjunto de recursos
- Contenedor Posee toda la información que necesita un programa para correr
- Instancia de un programa en ejecución, incluyendo los valores actuales del contador de programa, los registros y las variables.

Espacios de direcciones

Los procesos tienen asignados espacios de direcciones limitados en memoria principal que es en donde lee y escribe información, el espacio de direcciones contiene el programa ejecutable, los datos del programa y su pila además de toda la información necesaria para ejecutar el programa.

En esencia, un proceso es un recipiente que guarda toda la información necesaria para ejecutar un programa.

Tabla de procesos

Es un arreglo (o lista enlazada) de estructuras, una para cada proceso que se encuentre actualmente en existencia que contiene toda su información en el momento.

En las tablas de proceso se guarda toda la información del mismo para que puede cambiar de estados y retornar sin problemas.

Llamadas al sistema

Existen llamadas al sistema que interactúan con los procesos y/o crean

Hay otras llamadas al sistema de procesos disponibles para solicitar más memoria (o liberar la memoria sin utilizar), esperar a que termine un proceso hijo y superponer su programa con uno distinto

La creación de un proceso se hace mediante las llamadas al sistema `fork()` y `execv()`

- `fork()` que crea un proceso hijo, idéntico al padre
- `exec()` la cual hace que toda su imagen de núcleo completa se sustituya por el archivo nombrado en su primer parámetro

Señales

Las señales son la analogía en software de las interrupciones de hardware y se pueden generar mediante una variedad de causas además de la expiración de los temporizadores

UID y GID

Los usuarios tienen IDs conocidas como UID los procesos contienen la UID del que los creó, los hijos de dichos procesos tienen la misma UID que el padre.

Los usuarios pueden ser miembros de grupos con su respectiva GID.

Existe una UID llamada Superuser que tiene poderes especiales y puede violar muchas reglas de protección.

Rutas de direcciones

El nombre de ruta también conocido como dirección es la ubicación dentro de la jerarquía de directorios se lee de el directorio raíz en adelante

Directorio de trabajo

Todos los procesos tienen un directorio de trabajo que son aquellos que no comienzan con barra diagonal.

Los procesos pueden modificar su directorio de trabajo mediante llamadas al sistema.

Descriptor de archivos

Para proveer un lugar en donde se puedan mantener los archivos, la mayoría de los sistemas operativos tienen el concepto de un directorio como una manera de agrupar archivos.

Existen llamadas al sistema para crear archivos, directorios y eliminar archivos de un directorio.

La jerarquía de los directorios difiere en tamaño, protección y tiempo de vida de la de procesos. Son más extensas y se pueden acceder sin ser del mismo grupo aparte del propietario también suelen durar mucho más tiempo.

Montar y desmontar

UNIX utiliza un sistema de archivos montado en el que si un dispositivo de almacenamiento externo es incorporado al sistema, no es accesible hasta que este sea montado, en el sistema de archivos asignándole un punto de montaje en un directorio al que luego hará referencia el contenido del dispositivo.

Los contenidos anteriores del directorio no son accesibles mientras el dispositivo este montado

Archivos especiales

Los dispositivos E/S en UNIX son representados como archivos de los cuales se puede efectuar llamadas al sistema para modificarlos o leer su contenido.

- **Archivos especiales de bloque:** transmiten datos en bloques (paquetes)
- **Archivos especiales de carácter:** transmiten un flujo de datos continuo

Tuberías

Una tubería o pipe es un tipo de pseudo archivo que se puede utilizar para conectar dos procesos.

La comunicación se da mediante la escritura y lectura de la tubería la cual reacciona como un archivo de entrada o salida.

La única manera de diferenciar entre un archivo normal y una tubería es mediante una llamada al sistema especial

Proteccion

En UNIX existen diferentes tipos de implementaciones para mejorar la seguridad como por ejemplo en el caso de los archivos se utiliza una serie de bits para definir los permisos del archivo y a quien se lo permite realizar.

Shell

El shell no forma parte de el sistema operativo hace de interprete entre el usuario y el sistema operativo a no ser que el usuario tenga interfaz grafica.

La GUI es el la interfaz de usuario que se ejecuta arriba del sistema operativos

Estructura de un sistema operativo

Los seis diseños son:

- Sistemas monolíticos
- Sistemas de capas
- Microkernels
- Sistemas cliente-servidor
- Máquinas virtuales
- Exokernels

Sistemas Monolíticos

En este tipo de sistema operativo todo el sistema operativo se ejecuta en un solo programa en modo kernel.

El sistema operativo se escribe como una colección de procedimientos, enlazados entre sí en un solo programa binario ejecutable extenso.

Sistema de capas

En este tipo de sistema operativo el funcionamiento esta basado en 6 capas

Microkernels

Este tipo de sistema operativo tiene como objetivo tener un kernel más ligero porque mientras los módulos sean mas grandes es mas probable que se ejecuten errores, por lo que se busca es que los errores ocurran en modo usuario donde los programas tienen menos poder y no lleven a una falla mayor.

Modelo Cliente-Servidor

Este tipo de sistema operativo es una variación del microkernel en la que los procesos se diferencian en cliente y servidor donde el cliente simplemente decide que se hace y el servidor lo recibe y ejecuta

Un ejemplo de la actualidad son las paginas web donde el cliente envía peticiones y el servidor responde

Conceptos

- **Concurrencia:**
- **Multiprogramación:** Es la conmutación de programa a programa que la CPU realiza lo que se llama multiprogramación
- **Pseudoparalelismo:**
- **Mutinúcleo:** Un multiprocesador contiene multiples CPUs o nucleos que puede ejecutar tareas individualmente

Creacion de procesos

Hay cuatro eventos principales que provocan la creación de procesos.

- El arranque del sistema.
- La ejecución, desde un proceso, de una llamada al sistema para creación de procesos.
- Una petición de usuario para crear un proceso.
- El inicio de un trabajo por lotes.

Terminacion de procesos

- Salida normal (voluntaria).
- Salida por error (voluntaria).
- Error fatal (involuntaria).
- Eliminado por otro proceso (involuntaria)

Jerarquía:

Los procesos hijos son los procesos creados por otro proceso creando una jerarquía.

Toda la jerarquía de procesos forma un grupo de procesos.

No existe la jerarquía en Windows.

Estados de un proceso

- En ejecución (en realidad está usando la CPU en ese instante).
- Listo (ejecutable; se detuvo temporalmente para dejar que se ejecute otro proceso).
- Bloqueado (no puede ejecutarse sino hasta que ocurra cierto evento externo).

Planificación.

Definición

- El planificador de procesos es el encargado de decidir qué proceso se va a ejecutar a continuación y lo selecciona a través del algoritmo de planificación.

Comportamiento de un proceso:

- Podemos diferenciar dos tipos de procesos:
 - Los limitados a E/S que tienen tiempos de ejecución en CPU cortos con constantes interrupciones de E/S
 - Los limitados a cálculos que tienen tiempos de ejecución en CPU largos con algunas interrupciones de E/S

- Tipos de algoritmos de planificación
 - **No Apropiativos:** Selecciona un proceso para ejecutarlo y después sólo deja que se ejecute hasta que el mismo se bloquea o hasta que libera la CPU en forma voluntaria.
 - **Apropiativos:** Selecciona un proceso y deja que se ejecute por un máximo de tiempo fijo

Categorías de los algoritmos de planificación

- **Procesamiento por lotes** (*No apropiativo*)
 - No hay usuarios que esperen impacientemente en sus terminales para obtener una respuesta rápida a una petición corta.
 - Reduce la conmutación de procesos y por ende, mejora el rendimiento.

Conmutación: Cambiar de un proceso a otro durante su ejecución.
--

- **Interactivo** (*Apropiativo*)
 - La apropiación es esencial para evitar que un proceso acapare la CPU y niegue el servicio a los demás.
- **De tiempo real** (*No apropiativo, de tiempos cortos*)
 - Sólo ejecutan programas destinados para ampliar la aplicación en cuestión.

Metas de los algoritmos de planificación

Todos los sistemas

- Equidad - Otorgar a cada proceso una parte justa de la CPU
- Aplicación de políticas - Verificar que se lleven a cabo las políticas establecidas
- Balance - Mantener ocupadas todas las partes del sistema

Sistemas de procesamiento por lotes

- Rendimiento - Maximizar el número de trabajos por hora
- Tiempo de retorno - Minimizar el tiempo entre la entrega y la terminación
- Utilización de la CPU - Mantener ocupada la CPU todo el tiempo

Sistemas interactivos

- Tiempo de respuesta - Responder a las peticiones con rapidez más importante
- Proporcionalidad - Cumplir las expectativas de los usuarios

El planificador puede intervenir en los casos en los que el tiempo de retraso se deba a una mala elección en el orden de procesos

Sistemas de tiempo real

- Cumplir con los plazos - Evitar perder datos
- Predictibilidad - Evitar la degradación de la calidad en los sistemas multimedia

Planificación en sistemas de procesamiento por lotes. Ejemplos

Primero en entrar, primero en ser atendido

- No es apropiativo. La CPU se asigna a los procesos en el orden en el que la solicitan
- Si el proceso en ejecución se bloquea, el primer proceso en la cola se ejecuta a continuación
- Cuando un proceso bloqueado pasa al estado listo, al igual que un trabajo recién llegado, se coloca al final de la cola

El trabajo más corto primero

- No es apropiativo
- Prioriza los trabajos más cortos primero por lo que más trabajos/tiempo

El menor tiempo restante a continuación

- Una versión apropiativa del algoritmo tipo el trabajo más corto primero es el menor tiempo restante a continuación
- Consiste en que si el nuevo trabajo necesita menos tiempo para terminar que el proceso actual, éste se suspende y el nuevo trabajo se inicia.

Planificación en sistemas interactivos

Ejemplos de algoritmos de planificación de sistemas interactivos:

Planificación por turno circular

- Utiliza el turno circular, a cada proceso se le asigna un quantum, durante el que se le permite ejecutarse. Si el proceso se bloquea antes de transcurrido el quantum se realiza la conmutación de la CPU.
- El tiempo del quantum puede determinar cuanto se desperdiciara en la conmutación

Planificación por prioridad

- Consiste en la asignación de un orden jerárquico basado en niveles de prioridad, el algoritmo puede realizar variaciones de los niveles para que los procesos mas importantes no se ejecuten infinitamente y den lugar al resto.
- También utiliza un quantum para realizar las conmutaciones de procesos al proceso con la prioridad mas alta.

Múltiples colas

- Este algoritmo consiste en la creación de niveles y cada una tiene un numero de quantums para su ejecución determinados. Al un proceso terminar todos sus quantums sube un nivel y sus números de quantums incrementa hasta que el mismo termine, entonces los procesos CPU dependientes funcionan mas eficientemente y no se desperdicia tiempo haciendo conmutaciones de procesos

El proceso más corto a continuación

- En este algoritmo hay que realizar estimaciones basadas en comportamiento anterior para poder saber cual es el procesos con el tiempo de ejecución estimado mas corto y ejecutarlo a continuación.

Planificación por sorteo

- Los procesos se dividen en niveles de prioridad y los mas prioritarios tienen mas números para ganar el sorteo por lo que tienen mas probabilidad de ganar los recursos que requieren.

Planificación por partes equitativas

- La CPU se reparte de manera equitativa entre los usuarios por lo que si un usuario tiene muchos procesos y el otro unos cuantos ambos tendrán el mismo tiempo de CPU

Planificación en sistemas de tiempo real

- Se pueden categorizar en tiempo real duro que significa que tiene limites absolutos y de tiempo real suave en el cual no es conveniente fallar en un tiempo limite pero es tolerable.

- Para llevar a cabo el comportamiento en tiempo real el programa se divide en múltiples procesos predecibles. Por lo general son procesos con tiempos de vida cortos, es responsabilidad del planificador que estos se ejecuten de manera que se cumplan los tiempos límites

COMUNICACIÓN ENTRE PROCESOS

Condiciones de carrera

Dos o más procesos están leyendo o escribiendo datos compartidos y el resultado final depende de quién se ejecuta y exactamente cuándo lo hace.

¿Cómo solucionarlo?

Condiciones

1. No puede haber dos procesos de manera simultánea dentro de sus regiones críticas.
2. No pueden hacerse suposiciones acerca de las velocidades o el número de CPUs.
3. Ningún proceso que se ejecute fuera de su región crítica puede bloquear otros procesos.
4. Ningún proceso tiene que esperar para siempre para entrar a su región crítica.

Exclusión mutua

Ningún proceso puede entrar a la región crítica de otro proceso mientras la está actualizando.

Propuestas de exclusión mutua con espera ocupada

- Deshabilitando interrupciones
 - En sistemas de un procesador, cada proceso deshabilite todas las interrupciones antes de entrar a su región crítica y las rehabilite antes de salir.
 - Si hay más procesadores, sólo se ve afectada la CPU que se deshabilitó.
 - Desventajas
 - No es conveniente dar a los procesos de usuario el poder para desactivar interrupciones.

Conclusión: Deshabilitar interrupciones es útil dentro del mismo OS, pero no es apropiada como mecanismo para excluir mutuamente procesos de usuario.

- Variables de candado
 - Variable compartida que inicia en 0.
 - Se evalúa la variable candado, si es 0, el proceso lo fija en 1 y se debe esperar que sea 0 para entrar.
 - Posee el mismo problema que el spooler (dos procesos fijan la variable al mismo tiempo).
- Alternancia estricta
 - Se maneja a través de turnos para entrar a la **región crítica**, desperdiciando tiempo de CPU.
 - No se recomienda tomar turnos cuando un proceso es más lento que otro.

Viola la condición 3: El proceso 0 es bloqueado por un proceso que no está en la región crítica.
- Solución de Peterson
 - Utiliza el algoritmo de Peterson.
 - Es una mejora de la alternancia estricta.
 - Utiliza llamadas a procedimientos.
 - No viola la condición 3.
 - Utiliza variables de señal y turno.

Region crítica

Parte del programa en la que se accede a la memoria compartida.

Dormir y despertar

- Sleep
 - Llamada al sistema que hace que el proceso que llama se bloquee o desactive.
- WakeUp
 - Llamada al sistema que despierta o activa al proceso pasado como parámetro.

Problema de inversión de prioridades

Un proceso tiene mayor prioridad que otro, A (prioridad alta) - B (prioridad baja).

A se ejecuta siempre que está en estado = listo

B está en su región crítica, A cambia a estado = listo

A empieza a esperar, B nunca sale, A itera de forma indefinida.

Problema del productor-consumidor

Dos procesos comparten un búfer común, de tamaño fijo.

Uno de ellos (productor), coloca información en el buffer.

El otro (consumidor), saca información del buffer.

Si el búffer está lleno, el productor duerme.

Si el búffer está vacío, el consumidor duerme.

Si el consumidor se quiere dormir pero antes consume, el productor inserta, no puede despertar al consumidor porque está despierto. El consumidor se duerme y se llena el búffer.

Se soluciona agregando un bit de espera de "wakeups" para que el consumidor no se vaya a dormir.

Semáforos

Un semaforo es una variable que tiene tres valores disponibles la cual dependiendo sus valores se realiza una operación u otra.

Un semáforo podría tener el valor 0, indicando que no se guardaron señales de despertar o algún valor positivo si estuvieran pendientes una o más señales de despertar.

Operación down

Comprueba si el valor es mayor que 0.

De ser así, disminuye el valor (es decir, utiliza una señal de despertar almacenada) y sólo continúa.

Si el valor es 0, el proceso se pone a dormir sin completar la operación down por el momento.

Operación up

Incrementa el valor del semáforo direccionado.

Si uno o más procesos estaban inactivos en ese semáforo, sin poder completar una operación down anterior, el sistema selecciona uno de ellos (al azar) y permite que complete su *operación down*. Así, después de una operación up en un semáforo que contenga procesos dormidos, el semáforo seguirá en 0 pero habrá un proceso menos dormido en él. La operación de incrementar el semáforo y despertar a un proceso también es indivisible.

Acciones atómicas

Las acciones de comprobar el valor, modificarlo y posiblemente pasar a dormir, se realizan en conjunto como una sola **acción atómica indivisible**.

Se garantiza que, una vez que empieza una operación de semáforo, ningún otro proceso podrá acceder al semáforo sino hasta que la operación se haya completado o bloqueado.

Esta atomicidad es absolutamente esencial para resolver problemas de sincronización y evitar condiciones de carrera.

Mutexes

Un **mutex** es una variable que puede estar en uno de dos estados: abierto (desbloqueado) o cerrado (bloqueado). En consecuencia, se requiere sólo 1 bit para representarla, pero en la práctica se utiliza con frecuencia un entero, en donde 0 indica que está abierto y todos los demás valores indican que está cerrado.

Se utilizan dos procedimientos con los mutexes.

- **mutex_lock**
 - Se llama cuando un hilo (o proceso) necesita acceso a una región crítica.
 - Si el mutex está actualmente abierto (lo que significa que la región crítica está disponible), la llamada tiene éxito y entonces el hilo llamador puede entrar a la región crítica.
- **mutex_unlock**
 - Por otro lado, si el mutex ya se encuentra cerrado, el hilo que hizo la llamada se bloquea hasta que el hilo que está en la región crítica termine y llame a `mutex_unlock`. Si se bloquean varios hilos por el mutex, se selecciona uno de ellos al azar y se permite que adquiera el mutex.

Monitores

Un **monitor** es una colección de procedimientos, variables y estructuras de datos que se agrupan en un tipo especial de módulo o paquete.

Los procesos pueden llamar a los procedimientos en un monitor cada vez que lo desean, pero no pueden acceder de manera directa a las estructuras de datos internas del monitor desde procedimientos declarados fuera de éste.

- Son una construcción del lenguaje

- El compilador sabe que son especiales y puede manejar las llamadas a los procedimientos del monitor en forma distinta a las llamadas a otros procedimientos.
- Al convertir todas las regiones críticas en procedimientos de monitor, nunca habrá dos procesos que ejecuten sus regiones críticas al mismo tiempo.

Pasaje o transmisión de mensajes

Este método de comunicación entre procesos utiliza dos primitivas (send y receive) que, al igual que los semáforos y a diferencia de los monitores, son llamadas al sistema en vez de construcciones del lenguaje.

- send
 - Envía un mensaje a un destino especificado.
- receive
 - Recibe un mensaje de un origen especificado.

Si no hay un mensaje disponible ~ El receptor se puede bloquear hasta que llegue uno.

Barreras

Ningún proceso puede continuar a la siguiente fase sino hasta que todos los procesos estén listos para hacerlo.

HILOS

Son miniprocesos pseudo-paralelos que comparten un espacio de direcciones y todos los datos dentro del mismo.

Cada hilo tiene su propia pila.

Ventajas

- Muchas aplicaciones desarrollan varias actividades a la vez (se puede descomponer en hilos)
- Son más ligeros que los procesos, más fáciles (rápidos) de crear y destruir.
- Agilizan la velocidad de la aplicación cuando hay una cantidad considerable de cálculos y operaciones E/S.
- Útiles en sistemas con varias CPUs.

INTERBLOQUEOS

DEFINICION

Cada proceso en el conjunto está esperando un evento que sólo puede ser ocasionado por otro proceso en el conjunto.

Interbloqueo de recursos

Son los interbloqueos basados en la petición insatisfactoria de recursos, ya que los recursos están en tenencia de otros procesos **interbloqueados**.

Condiciones para los interbloqueos de recursos

Se deben cumplir **todas** las condiciones para que haya un interbloqueo.

1. **Condición de exclusión mutua.** Cada recurso se asigna en un momento dado a sólo un proceso, o está disponible.
2. **Condición de contención y espera.** Los procesos que actualmente contienen recursos que se les otorgaron antes pueden solicitar nuevos recursos.
3. **Condición no apropiativa.** Los recursos otorgados previamente no se pueden quitar a un proceso por la fuerza. Deben ser liberados de manera explícita por el proceso que los contiene.
4. **Condición de espera circular.** Debe haber una cadena circular de dos o más procesos, cada uno de los cuales espera un recurso contenido por el siguiente miembro de la cadena.

¿Cómo atacarlas?

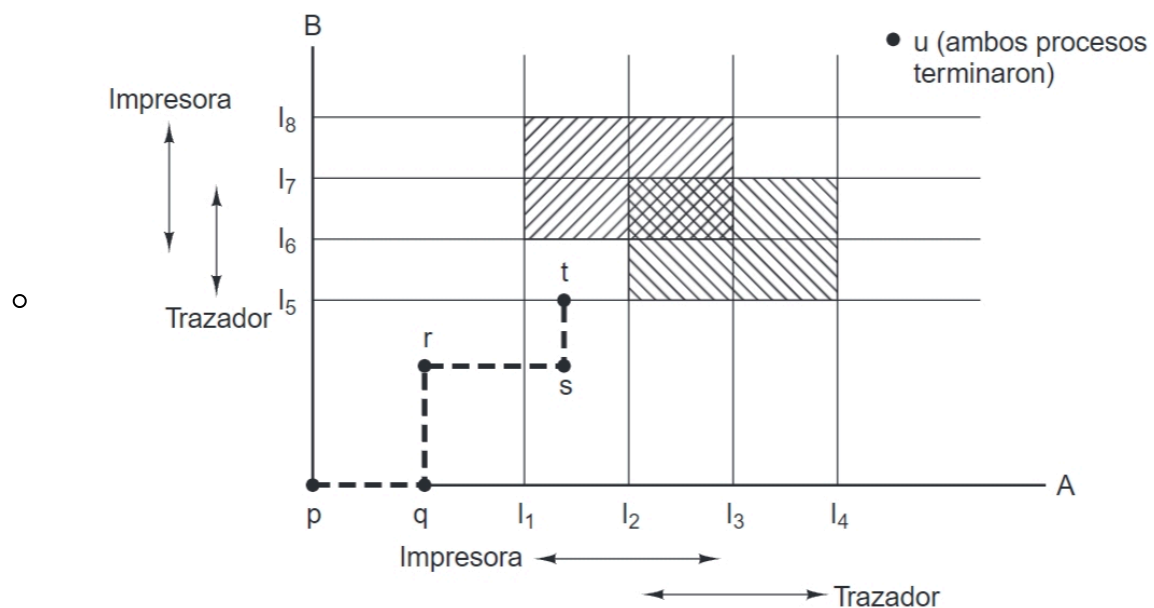
- **Condición de exclusión mutua**
 - Evitar asignar un recurso cuando no es absolutamente necesario.
 - Asegurar la menor cantidad de procesos reclamando ese recurso.
- **Condición de contención y espera**
 - Solicitar todos los recursos al principio nunca se sabe cuántos recursos necesitarán
 - Si un proceso solicita un recurso, que libere temporalmente todos los recursos que tiene y los vaya recuperando de a poco.
- **Condición no apropiativa**
 - Se apropian los recursos y se virtualizan bloqueando el acceso dándole solo poder solo a un demonio

Cabe aclarar que no es aplicable en muchos casos como en los registros de base de datos

- **Condición de espera circular**
 - Ordenar los recursos en forma numérica.

MODELADO

- **Grafos.**
 - □ → Recurso
 - ○ → Proceso
- **Trayectoria de los recursos**



- **Estados seguros e inseguros**

◦

A	3	9
B	2	4
C	2	7

Libres: 3

(a)

Tiene Máx.

A	3	9
B	4	4
C	2	7

Libres: 1

(b)

Tiene Máx.

A	3	9
B	0	–
C	2	7

Libres: 5

(c)

Tiene Máx.

A	3	9
B	0	–
C	7	7

Libres: 0

(d)

Tiene Máx.

A	3	9
B	0	–
C	0	–

Libres: 7

(e)

¿Cómo lidiar con un interbloqueo?

- **EL ALGORITMO DE LA AVESTRUZ**

Consiste en no hacer nada

- **DETECCIÓN Y RECUPERACIÓN**

Dejar que ocurran los interbloqueos, detectarlos y tomar acción.

- **Evitarlos mediante la asignación cuidadosa de los recursos.**

- **PREVENCIÓN**

Evitar estructuralmente una de las cuatro condiciones requeridas.