Glosario SOII

Tema 1

- **Condición de carrera**: Cuando dos o más procesos está leyendo o escribiendo algunos datos compartidos y el resultado depende de quien se ejecuta y exactamente cuando lo hace.
- **Región crítica**: Parte del programa de un proceso que accede a memoria compartida.
- **Exclusión mútua**: Asegura, que si un proceso está utilizando datos compartidos, los demás no podrán accedera ellos.
- **Espera ocupada**: Acción de evaluar en forma continua una variable hasta que aparezca cierto valor.
- **Candado**: Variable compartida, que se utilizan varios procesos para poder acceder a una región crítica (mueve la región crítica)
- **Candado de giro**: Candado que utiliza la espera ocupada. (Alternancia estricta)
- **Instrucción TSL**: Test and set Lock
- **Problema de inversión de prioridades**: Proceso High priority y proceso Low priority, L entra en la zona crítica pero al ejecutarse H, no deja que se ejecute L para salir de la zona crítica, provocando que H esté esperando de forma indefinida.
- **Problema productor-consumidor**: Dos procesos comparte un búfer común, de tamaño fijo. El producto coloca información en búfer y el otro la saca. El consumidor debe dormirse cuando no hay productos y el productor cuando se llena el buffer.
- **Bit de espera de despertar**. Cuando se envía una señal de despertar a un proceso que sigue todavía despierto, se fija este bit. Más adelante, cuando el proceso intenta pasar a dormir, si el bit de espera de despertar está encendido, se apagará pero el proceso permanecerá despierto.
- **Semáforo**: Variable entera para contar el número de señales de despertar. Sirve para indicar cuantos recursos hay disponibles.
- **Semáforos binarios**: Semaforos que se inicializan a 1 y son utilizados por dos o más procesos para asegurar que sólo uno de ellos puede entrar a su región crítica en un momento dado.
- **Semáforos de sincronización**: Semáforos que se usan para garantizar que ciertas secuencias de eventos ocurran o no. Como los semáforos vacías y llenas, que se usan para garantizar que ciertas secuencias de enventos ocurran o no.
- **Mutex**: Variable que puede estar en dos estados: abierto (desbloqueado) o cerrado(bloqueado). Se utilizan dos procedimientos con los mutexes. Cuando un hilo necestia acceso a una región crítica, llama a **mutex_lock**. Si el mutex está actualmente abierto, la llamada tiene exito y entra. SI esta cerrado, el hilo que hizo la llamada se bloquea.
- **Variables de condición**: Permite que los hilos se bloqueen debido a que cierta condición no se está cumpliendo. No tiene memoria a diferencia que el mutex. Si se envía una señal a una variable de condición en la que no haya un hilo en espera, se pierde la señal
- **Monitores**: Es una clase, objeto o módulo seguro para hilos (thread-safe) que envuelve un mutex con el fin de permitir el acceso seguro a un método o variable por parte de más de un hilo. La característica definitoria de un monitor es que sus métodos se ejecutan con exclusión mutua: en cada momento, como máximo un hilo puede estar ejecutando cualquiera de sus métodos. Utilizando una o más variables de condición, también puede proporcionar la capacidad de que los hilos esperen hasta que se cumpla cierta condición (usando así la definición anterior de "monitor").
- **Pasaje de mensajes**: Método de comunicación entre procesos que utiliza dos primitivas (send y receive), al igual que los semáforos y a diferencia de los monitores, son llamadas al sistema en vez construcciones del lenguaje. Si no hay un mensaje disponible, el receptor se puede bloquear hasta que llegue uno. De manera alternativa, puede regresar de inmediato con un código de error.

- **Buzón**: Es un lugar para color en el búfer cierto número de mensajes, que por lo genera se especifica a la hora de crear el buzón. Cuando un proceso trata de enviar un buzón que está lleno, se suspende hasta que se remueva un mensaje de ese buzón haciendo espacio para un nuevo mensaje
- **Estrategia de encuentro**: Estrategía entre dos procesos en la que un proceso se bloquea hasta que ambos estén listos para realizar una operación.
- **Barreras**: Es un punto de encuentro en el que varios procesos o hilos deben esperar hasta que todos hayan llegado a ese punto. Una vez que todos han alcanzado la barrera, se les permite continuar su ejecución.

Tema 2

- **Recurso apropiativo**: Un recurso que **se puede quitar** al proceso que lo posee sin **efectos dañinos**. Ejemplo: la memoria, un proceso puede **apropiarse de la memoria** de otro proceso, **evitando un interbloqueo**
- **Recurso no apropiativo**: Un recurso que no se puede quitar a su propietario actual sin hacer que el cómputo falle. Ejemplo: Un proceso que está escribiendo en un CD
- **Interbloqueo**: Un conjunto de procesos se encuentra en un interbloqueo si cada proceso en el conjunto está esperando un evento que sólo puede ser ocasionado por otro proceso en el conjunto.
- **Interbloqueo de recursos**: Interbloqueo en el que el evento que espera cada proceso es un recurso.
- **Grafo de recursos**: Grafo que modela las condiciones de interbloqueo. Nos permite ver si una secuencia de petición/liberación conduce a interbloqueo. Los cuadrados son los recursos y los círculos son los procesos.
- **Algoritmo de la avestruz**: Consiste en ignorar el problema de los interbloqueos
- **Estado seguro**: Si hay cierto orden de programación en el que se puede ejecutar cada proceso hasta completarse, incluso aunque todos ellos solicitaran de manera repentina su número máximo de recursos de inmediato.
- **Estado inseguro**: No se puede garantizar que todos los procesos terminarán
- **Virtualizar un recurso**: significa crear una versión lógica o simulada del recurso físico, de forma que los usuarios o procesos crean que tienen acceso exclusivo a ese recurso, cuando en realidad están compartiéndolo.
- **Bloqueo activo**: es una forma de espera en la que un proceso se mantiene ejecutando y verificando continuamente una condición, consumiendo CPU mientras espera, en vez de ceder el procesador.

Tema 3

- **Tiempo de respuesta**: Tiempo entre el que se debe completar cada cuadro de una ráfaga de la CPU
- **Ráfaga de CPU**: Periodo de tiempo durante el cual un proceso tiene acceso a la CPU para realizar su trabajo.
- **RMS**:Es un algoritmo de programación de **tiempo real estático** para los procesos **periodicos** y **preferentes**.
- **EDF**: Es un algoritmo de programación de tiempo real **dinámico**