

Carrera: Analista Programador

Arquitectura de Sistemas Operativos

Módulo I

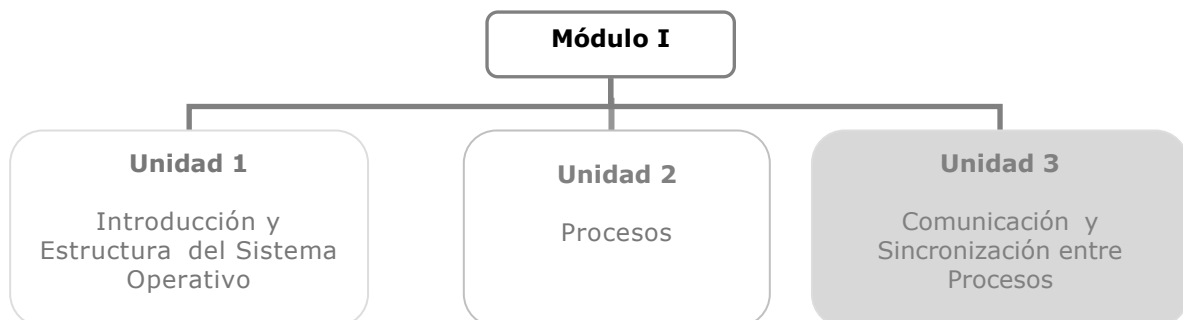
Aprender el objetivo y las funciones del sistema operativo, la planificación del procesador, la comunicación y sincronización entre procesos.

Unidad 3

Comunicación y sincronización entre procesos.

Profesor Titular: Martín Mario.

Autor de contenidos: lic. Juan Carlos Romero.



Presentación

Sería difícil imaginar un mundo sin comunicación y sin sincronización.

Las “señales de humo”, “los tambores”, “un simple grito” y en un futuro, no muy lejano, quizá “la telepatía”; fueron y serán formas de comunicación que nos permiten intercambiar información.

Cuando viajamos al trabajo, los semáforos, las señales, los obstáculos, las barreras ferroviarias, los carriles de desvío, son elementos que ayudan a sincronizar los vehículos en nuestro viaje.

En muchas situaciones, en la vida diaria, utilizamos la comunicación y la sincronización simultáneamente ¿cuántas veces nos cruzamos con personas que hablan simultáneamente con su interlocutor? De esas personas solemos decir, que no escuchan porque hablan sobre las palabras de su interlocutor.

Si esa misma conversación se realiza por vía telefónica es aún más confusa, entonces, pueden aparecer frases como: “escuchame un minuto”, “ahora escúchame a mí”, “cállate y déjame hablar”, “no hables cuando estoy hablando”, “deja de hablar cinco minutos”. Estas frases ayudan a ordenar, si se puede decir así, la conversación. Es decir, en el mejor de los casos, ambas personas comienzan a sincronizarse para no perder la oportunidad de comunicarse y en el peor de los casos no logran hacerlo.

Traslademos este simple ejemplo a una organización donde trabajan miles de personas para brindar un servicio o para poner un producto en el mercado, ¿cuánto tiempo se dedica a la comunicación?, ¿cuánto tiempo se dedica a la sincronización?, ¿qué estrategias de comunicación y sincronización se utilizan?

En la primera unidad presentamos al Sistema Operativo y su estructura, en la Unidad 2, los procesos como unidades productivas que realizan entre otras cosas las tareas de administración. En esta oportunidad daremos a conocer los elementos de comunicación y sincronización que necesitan los procesos para llevar adelante las tareas de administración en forma cooperativa y concurrente.

Por todo lo expresado hasta aquí es que esperamos que usted, a través del estudio de esta unidad, adquiera capacidad para:

- Comprender el concepto de concurrencia.
- Entender el problema de la sección crítica.
- Estudiar los mecanismos de comunicación y sincronización entre procesos.
- Crear procesos productores y consumidores para poner en práctica la problemática de la comunicación y sincronización.



A continuación, le presentamos un detalle de los contenidos y actividades que integran esta unidad. Usted deberá ir avanzando en el estudio y profundización de los diferentes temas, realizando las lecturas requeridas y elaborando las actividades propuestas, algunas de desarrollo individual y otras para resolver en colaboración con otros estudiantes y con su profesor tutor.

Contenidos y Actividades

1. Procesos concurrentes.



Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 4. Procesos.. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p. 100.



Lectura Sugerida

- Stallings W.; Capítulo 5 Concurrencia. Exclusión mutua y sincronización.. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño–**; 5ta Edición; España Prentice Hall, 2007. p.201.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 3 Procesos.; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p.73
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5 Comunicación y sincronización entre procesos. **En su: Sistemas Operativos –Una visión aplicada–**; España Mc Graw Hill, 2001. p. 223.



Trabajo Práctico Sugerido

- Trabajo Práctico Nº 5: **Proceso Productor y Proceso Consumidor**

2. Sección crítica.



Lectura requerida

- Silberschatz A. y Galvin P.; Capítulo 6 Sincronización de Procesos. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p. 155.



Lectura Sugerida

- Stallings W.; Capítulo 5 Concurrencia. Exclusión mutua y sincronización. **En su: Sistemas Operativos –Aspectos Internos**



y principios de diseño-; 5ta Edición; España Prentice Hall, 2007. p.201.

- Silberschatz A. Galvin P. Gagne G.; Capítulo 6 Sincronización de Procesos.; En su: **Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p.171.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5 Comunicación y sincronización entre procesos.. En su: **Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. p.223.



Trabajo Práctico Sugerido

- Trabajo Práctico Nº 6 **Sección Crítica**

3. Mecanismos de comunicación y sincronización.



Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 6 Sincronización de Procesos. En su: **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p. 155



Lectura Sugerida

- Stallings W.; Capítulo 5 Concurrencia. Exclusión mutua y sincronización.. En su: **Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007.p 201.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 6 Sincronización de procesos.; En su: **Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p.171.



Trabajo Práctico Sugerido

- Trabajo Práctico Nº 7 **Semáforos**

3.1 Servicios que brinda el SO para la comunicación y sincronización.



Lectura requerida

- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5 Comunicación y sincronización entre procesos. En su: **Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. P.223.



Lectura Sugerida

- Stallings W.; Capítulo 5 Concurrencia. Exclusión mutua y sincronización. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. P. 201.



Trabajo Práctico Sugerido

- Trabajo Práctico N° 8 **Semáforos en el sistema operativo LINUX.**

3.2 Problemas clásicos de sincronización



Lectura requerida

- Silberschatz A. y Galvin P.; Capitulo 6 Sincronización de Procesos. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p.155.



Lectura Sugerida

- Stallings W.; Capítulo 5 Concurrencia. Exclusión mutua y sincronización. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. p.201.
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5 Comunicación y sincronización entre procesos. **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. p. 223.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 6 Sincronización de procesos.; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p. 171.



Trabajo Práctico Requerido

Trabajo Práctico N°9 **El problema de los lectores y escritores**

Cierre de la unidad



EVALUACIÓN PARCIAL

Propuesta para la Integración del Módulo I

ANEXO

Para el estudio de estos contenidos usted deberá consultar la bibliografía que aquí se menciona:

BIBLIOGRAFÍA OBLIGATORIA

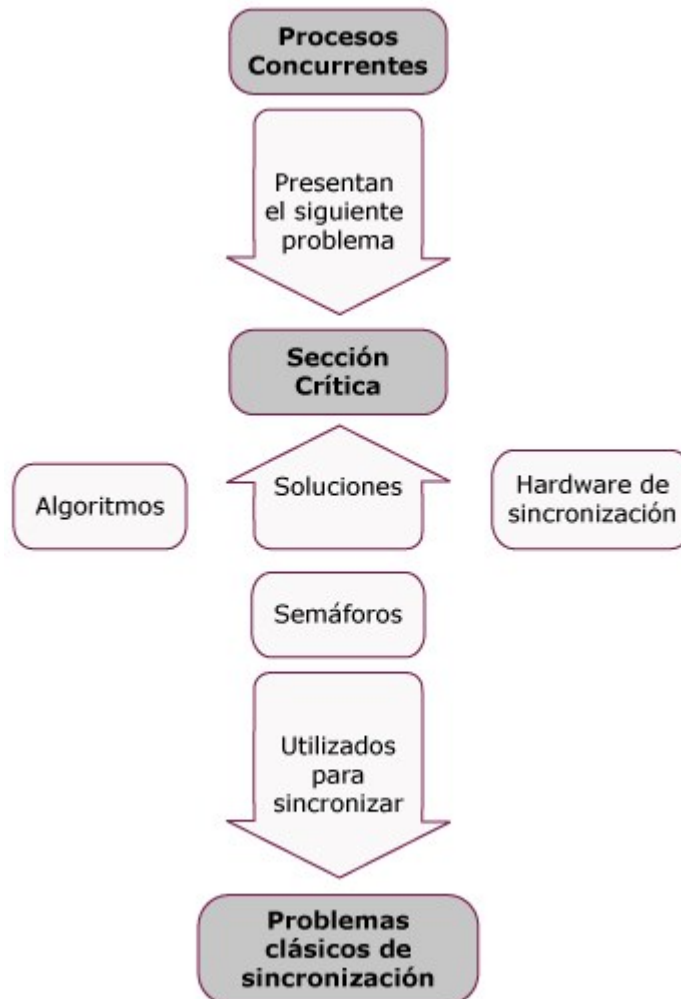
- **Silberschatz A. y Galvin P.; Sistemas Operativos; 5ta Edición; México Addison Wesley; 1999.**
- **Stallings W.; Sistemas Operativos –Aspectos Internos y principios de diseño–; 5ta Edición; España Prentice Hall, 2007.**
- **Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Sistemas Operativos –Una visión aplicada–; España Mc Graw Hill, 2001.**

BIBLIOGRAFIA AMPLIATORIA

- **Tanembaum A. Woodhull A.; Sistemas Operativos – Diseño e implementación–; 2da Edición; México Prentice Hall 1997.**
- **Silberschatz A. Galvin P. Gagne G.; Fundamentos de Sistemas Operativos; 7ma Edición; España Mc Graw Hill 2007.**

Organizador Gráfico

El siguiente esquema le permitirá visualizar la interrelación entre los conceptos que a continuación abordaremos. Le sugerimos que vuelva a este organizador una vez completado el estudio de la unidad, le ayudará a ordenar sus ideas.



Lo/a invitamos ahora a comenzar con el estudio de los contenidos que conforman esta unidad.



1. Procesos concurrentes

Hasta el momento realizamos una breve presentación del tema concurrencia en la Unidad 2 en el ítem "Proceso: multitarea, estados e información del proceso". Profundizaremos el tema en esta unidad como eje central de la problemática de la comunicación y sincronización.

Imagine que usted está felizmente casado y tiene hermosos trillizos - por no decir, cuatrillizos o quintillizos- de aproximadamente 4 años y que son particularmente muy inquietos y activos, tienen mucha energía; una energía que se duplica o triplica después de un buen desayuno con mucho cereal y leche de soja. Los trillizos están listos para el juego y el entretenimiento, dispuestos a usar sus juguetes en su sala de juegos.

Felices padres, prepararse! Porque no sé si sus trillizos son igual que mis dos hijos que, siempre en el mismo instante de tiempo, quieren jugar con el mismo juguete y cuando digo mismo juguete, digo exactamente igual, en color, peso, forma, tamaño, desperfecto, etc.

Algunas veces pienso: *"Porque no compre dos juguetes exactamente iguales"* y me respondo: *"No, porque debería gastar el doble de dinero, y no puedo hacer eso, y además porque en realidad una de mis funciones como padre es enseñar a compartir"*, entonces, se me ocurre que tengo que explicarles que deberían jugar un rato cada uno con el mismo juguete, ambos niños en mi caso y en su caso los trillizos estarían satisfechos.

Sus trillizos juegan en forma concurrente, todos al mismo tiempo en la misma sala de juegos. Hasta aquí ningún problema, éste surge cuando los tres pequeños quieren hacer uso del mismo juguete en el mismo instante de tiempo.

También, puede encontrarse con otro problema, similar al anterior, que a un niño se le ocurra jugar con un autito a fricción y a otro con el tren eléctrico y al tercero con las acuarelas; y de repente cuando todo esta transcurriendo perfectamente, al tercer niño se le ocurre tener el autito a fricción, el tren eléctrico y las acuarelas, dudo que cada uno quiera ceder lo suyo.

Podríamos pensar en algunas soluciones para evitar este inconveniente:

- Pedir que jueguen en el mismo horario pero en forma individual, en distintas salas de juego con los mismos juguetes.
- Pedir que jueguen en distintos horarios en forma individual en la misma sala de juego con los mismos juguetes.
- Pedir que jueguen en el mismo horario en la misma sala de juego y que compartan los mismos juguetes.

Esta problemática representa una forma de trabajo, o una forma en la cual las tareas se pueden realizar entre distintos procesos. Cada trillizo representa un proceso, cada juego una tarea; también aparecen recursos (los juguetes), un



espacio en el cual los trillizos realizan su juego, la sala de juegos (la memoria de trabajo) y algo muy importante: aprender a compartir los juguetes (compartir recursos entre los procesos). Para poder resolver estos inconvenientes necesitamos el apoyo de las estrategias de comunicación y sincronización. Para seguir profundizando sobre esta temática lo invitamos a realizar la siguiente actividad.



Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 4. Procesos. En su: **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p 100.

Guía para la lectura

1. ¿Qué entiende por proceso cooperativo?
2. ¿Cuáles son las razones para crear un entorno que permita la cooperación entre procesos?
3. ¿Cuándo un proceso es independiente y cuando es dependiente?
4. ¿Cuáles son los peligros de la compartición de recursos globales?
5. ¿Qué es la condición de carrera?
6. ¿Qué es la interacción de procesos?



Lectura Sugerida

- Stallings W.; Capítulo 5. Concurrencia. Exclusión mutua y sincronización.. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. p201
- Silberschatz A. Galvin P. Gagne G.; Capítulo 3. Procesos.; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p. 81
- Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5. Comunicación y

sincronización entre procesos.. **En su: Sistemas Operativos**
-Una visión aplicada-; España Mc Graw Hill, 2001. p.223.

Usted está en condiciones de transferir lo aprendido a través de la resolución del siguiente trabajo práctico sugerido.



Trabajo Práctico Sugerido

- Trabajo Práctico Nº 5 : Proceso Productor y Proceso Consumidor

Usted encontrará las consignas de este Trabajo Práctico en el Anexo que incluimos al final de este Orientador.

Comparta sus dudas e inquietudes con sus pares y con su tutor a través de los medios de comunicación disponibles en el Campus.



Preste atención a las dudas, preguntas y dificultades que surjan durante el estudio. Es muy importante que tome nota de ellas y que procure resolverlas. Compártalas con el grupo y/o con su tutor/a.

2. Sección crítica

Imagine la siguiente situación: usted está trabajando en una empresa y ésta le deposita su sueldo en una cuenta bancaria. Usted puede acceder a esa cuenta desde cualquier cajero bancario con una tarjeta de plástico. Su esposa posee otra tarjeta a su nombre para poder operar de la misma forma que usted. Simplifiquemos la temática y consideremos que sólo pueden hacer tres operaciones con la tarjeta: depósito de efectivo, extracción de efectivo y consulta de saldo.

Ahora lo/a invito a resolver el siguiente problema de álgebra:



Usted está ahorrando dinero para irse de vacaciones con su familia, y tuvo la muy buena suerte de acertar un número a la quiniela, como apostó \$1, cobro \$ 70. Inmediatamente, muy contento, fue a un cajero bancario dispuesto a depositar su ganancia. Antes de hacerlo, consulto su saldo, tenía \$100 y luego, deposito los \$70.

Mientras, que usted realizaba las transacciones, su señora esposa necesitaba hacer unas compras de librería para el comienzo de las clases de sus niños, para ello extrajo del cajero bancario \$ 50, pero antes consulto su saldo que en ese momento era de \$170.

Pregunta: ¿Cuál es su saldo luego de la extracción de su señora?

Respuesta: \$120.

¿Qué saldo tendría, si su señora hubiera realizado la extracción antes que usted realice el depósito?

Respuesta: \$ 100.

Ahora consideremos que por una de esas casualidades que tiene el destino que usted y su señora realizan la consulta de saldo en el mismo instante de tiempo y también realizan la operación de depósito y extracción en el mismo instante de tiempo y luego para verificar la operación, ambos consultan nuevamente el saldo.

Pregunta: ¿Qué saldo vería su señora?

Respuesta: \$50.

Pregunta: ¿Qué saldo vería usted?

Respuesta: \$170.

El banco se encarga de que esto no ocurra en la cuenta bancaria y seguramente, al fin de la jornada su saldo será de \$120. Sin embargo, en el momento de confirmar su saldo ni usted ni su señora están viendo el fiel reflejo de su realidad financiera.



Evidentemente, encontramos elementos críticos que se deben tener en cuenta para resolver determinadas problemáticas.

Las operaciones de depósito y extracción afectan la misma cuenta, entonces no se deben realizar en el mismo instante de tiempo.

Entonces, *¿qué es la sección crítica?*

La sección crítica es el conjunto de las sub operaciones que están dentro de la operación depósito y extracción que afectan a la misma cuenta bancaria y esos conjuntos de sub operaciones (sección crítica) son las que se tienen que sincronizar para que puedan acceder a la misma cuenta bancaria.

A continuación, lo/a invitamos a realizar una actividad de lectura que le permitirá conocer las secciones críticas y cómo anticiparse a ellas.



Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 6. Sincronización de Procesos. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p.155.

Guía para la lectura

1. ¿Qué es la sección crítica?
2. ¿Cuáles son los requisitos que se deben satisfacer para dar una solución al problema de la sección crítica?
3. Describa los algoritmos que dan la solución de la sección crítica para dos procesos.
4. Explique el hardware de sincronización.



Lectura Sugerida

Stallings W.; Capítulo 5. Concurrencia. Exclusión mutua y sincronización. Página 201. **En su: Sistemas Operativos – Aspectos Internos y principios de diseño**; 5ta Edición; España Prentice Hall, 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5. Comunicación y sincronización

entre procesos. Página 223. **En su: Sistemas Operativos – Una visión aplicada-**; España Mc Graw Hill, 2001.

Llegados a este momento del estudio le solicitamos la realización del Trabajo Práctico Sugerido.



Trabajo Práctico Sugerido

Trabajo Práctico Nº 6 Sección Crítica

Usted encontrará las consignas de este Trabajo Práctico en el Anexo que incluimos al final de este Orientador.

Comparta sus dudas e inquietudes con sus pares y con su tutor a través de los medios de comunicación disponibles en el Campus.



Considere de relevancia las dudas, preguntas y dificultades que surjan durante el estudio. Es muy importante que tome nota de ellas y que procure resolverlas. Compártalas con el grupo y/o con su tutor/a.

No acumule dudas!

3. Mecanismos de comunicación y sincronización

En este punto de la unidad estudiaremos a nivel teórico los mecanismos de comunicación y sincronización que utilizan los procesos para comunicarse y sincronizarse para poder cumplir con sus tareas cooperativas.

Entre ellos estudiaremos:

- **La memoria compartida** como mecanismos de comunicación.
- **La cola de mensajes** como mecanismo de comunicación y sincronización.
- **Los semáforos, las regiones críticas y los monitores** como mecanismos de sincronización.



Estos mecanismos lo ayudarán a entender la problemática de la comunicación y sincronización entre procesos en el entorno operativo multiusuario, multiprogramación o multitarea y monoprocesamiento (un único procesador).

Tenemos que destacar que no son los únicos mecanismos que se utilizan, los SO implementan otros. Entre ellos podemos destacar:

- los pipe y los fifo's para la comunicación y sincronización.
- las señales para la sincronización
- los archivos como medio de comunicación

Estos últimos no son los más óptimos por su modo de trabajo.

Le proponemos realizar la siguiente lectura en ella encontrará explicaciones y descripciones de cada uno de los mecanismos mencionados.



Lectura requerida

Silberschatz A. y Galvin P.; Capítulo 6. Sincronización de Procesos.. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p.155.

Guía para la Lectura

1. ¿Qué es un semáforo?
2. ¿Qué diferencias existe entre un semáforo binario y un semáforo contador?
3. ¿Cómo se implementan los semáforos?
4. ¿Qué es el bloqueo mutuo?
5. Describa la solución al problema de la sincronización, utilizando regiones críticas.
6. Describa la solución al problema de la sincronización, utilizando monitores.



Lectura Sugerida

Stallings W.; Capítulo 5. Concurrencia. Exclusión mutua y sincronización. Página 201. **En su: Sistemas Operativos** –



Aspectos Internos y principios de diseño-; 5ta Edición;
España Prentice Hall, 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F.
Pérez Costoya F.; Capítulo 5. Comunicación y sincronización
entre procesos. Página 223. **En su: Sistemas Operativos –
Una visión aplicada-**; España Mc Graw Hill, 2001.

Con el fin de poner en práctica lo aprendido, lo invitamos a realizar el Trabajo
Práctico que presentamos a continuación.



Trabajo Práctico Sugerido

Trabajo Práctico N°7 Semáforos

*Usted encontrará las consignas de este Trabajo Práctico en el
Anexo que incluimos al final de este Orientador.*

*Comparta sus dudas e inquietudes con sus pares y con su tutor a
través de los medios de comunicación disponibles en el Campus.*

3.1 Servicios que brinda el SO para la comunicación y sincronización.

Luego de estudiar los conceptos teóricos de los mecanismos de comunicación y sincronización entre procesos en entornos operativos multitarea, monoprocesador, intentaremos emprender la difícil tarea de llevar a la práctica estos conceptos, introduciéndonos dentro del SO, más específicamente en la parte del SO que brinda los servicios para que estos conceptos se hagan realidad, una tarea compleja pero muy interesante.

Nos adentraremos en el código mismo de los procesos, desde dos áreas, una de ellas es la de entender el código ya escrito y la otra es escribir el código de los procesos utilizando mecanismos de comunicación y sincronización y los servicios del SO.

Además de ayudarlo/a a internalizar los conceptos de una manera activa, este punto de la unidad lo/a fortalecerá en la capacidad de programar en lenguaje C y le dará una visión real del SO, ya que: creará, ejecutará y monitoreará procesos dentro del SO Linux. Para lograr nuestro propósito lo invitamos a realizar la siguiente actividad de lectura.



Lectura requerida

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5. Comunicación y sincronización entre procesos.. **En su: Sistemas Operativos –Una visión aplicada-**; España Mc Graw Hill, 2001. p. 223.

Guía para la Lectura

1. ¿Qué es el estándar System V?
2. ¿Cuáles son los servicios que brinda el SO Linux para la comunicación entre procesos?
3. ¿Cuáles son los servicios que brinda el SO Linux para la sincronización entre procesos?
4. ¿Cuáles son las principales características de la memoria compartida en Linux?
5. ¿Cuáles son las principales características de los semáforos en Linux?
 - i.¿Cómo se implementan las colas de mensajes en Linux?



Lectura Sugerida

Stallings W.; Capítulo 5. Concurrencia. Exclusión mutua y sincronización. **En su: Sistemas Operativos –Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007. p. 201.

Luego de realizar la lectura requerida lo/a invitamos a implementar los ejemplos de sincronización que se presentan en la propuesta del siguiente trabajo práctico.



Trabajo Práctico Sugerido

Trabajo Práctico N°8 Semáforos en el sistema operativo LINUX.



Usted encontrará las consignas de este Trabajo Práctico en el Anexo que incluimos al final de este Orientador.

Comparta sus dudas e inquietudes con sus pares y con su tutor a través de los medios de comunicación disponibles en el Campus.



Preste atención a las dudas, preguntas y dificultades que surjan durante el estudio. Es muy importante que tome nota de ellas y que procure resolverlas. Compártalas con el grupo y/o con su tutor/a.

3.2 Problemas clásicos de sincronización

En nuestra vida personal y profesional nos enfrentamos a una serie de problemas y, muchos de ellos, resultan tener características similares pondríamos decir que es posible aplicar la misma solución.

Recuerdo cuando empecé como ayudante docente en SO hace 16 años, el profesor estaba tomando un parcial y en la hoja del examen figuraba repetida una pregunta, entonces un alumno dijo: "*Profesor en el examen hay dos preguntas iguales*" y el profesor le responde: "Entonces tienen la misma respuesta".

Qué bueno sería si todos los problemas tuvieran la misma solución, si fuera de esa manera siempre nos estaríamos enfrentando al mismo problema, quizá podríamos decir, en definitiva, que el problema no tiene solución porque siempre ocurre.

Los problemas clásicos de sincronización son:

- El problema del buffer limitado.
- El problema de los lectores y escritores.
- El problema de los filósofos comensales.

Silverschatz en su libro *Sistemas Operativos* 5ta edición dice "*Estos problemas sirven para probar casi cualquier esquema de sincronización nuevo que se proponga*".

Por tal motivo concluimos, que es importante conocer estos problemas y plantear una solución. Para ello lo invito a realizar la siguiente lectura y realizar el trabajo práctico que proponemos a continuación.



Lectura requerida

Silberschatz A. y Galvin P.; Capitulo 6. Sincronización de Procesos.. En su: **Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. p.155.

Guía para la lectura

1. ¿Qué es el buffer limitado?
2. Describa un proceso productor y un procesos consumidor que acceden a un buffer limitado
3. ¿Cuál es el problema que se genera cuando procesos lectores y escritores acceden al mismo archivo?
4. Describa el problema de los filósofos comensales



Lectura Sugerida

Stallings W.; Capítulo 5. Concurrencia. Exclusión mutua y sincronización. Página 201. **En su: Sistemas Operativos – Aspectos Internos y principios de diseño-**; 5ta Edición; España Prentice Hall, 2007.

Carretero Pérez J. De Miguel Anasagasti P. García Carballeira F. Pérez Costoya F.; Capítulo 5. Comunicación y sincronización entre procesos. Página 223. **En su: Sistemas Operativos – Una visión aplicada-**; España Mc Graw Hill, 2001.

Llegando al final de esta unidad, la/o invitamos a realizar el siguiente trabajo práctico.



Trabajo Práctico Requerido

Trabajo Práctico N°9: El problema de los lectores y escritores

Cierre de la unidad

Los procesos que estudió y creó en la unidad anterior cooperan para alcanzar un objetivo común, compiten para tener acceso a recursos compartidos, necesitan intercambiar información y tienen restricciones en el orden en que se ejecutan algunas de sus acciones.

Dejar que los procesos se ejecuten sin controlar el acceso a recursos compartidos produciría resultados incorrectos y un mal funcionamiento del sistema.

Esta unidad presentamos los mecanismos necesarios para que los procesos puedan cooperar y llegar a cumplir un objetivo común, tener acceso a recursos compartidos e intercambiar información sin producir resultados incorrectos y lograr un buen funcionamiento del sistema indispensable para lograr los objetivos de los procesos.



EVALUACIÓN PARCIAL

Propuesta para la Integración del Módulo I

En esta primera Evaluación Parcial de la asignatura usted deberá desarrollar consignas de trabajo que apuntan a la comprensión las funciones del sistema operativo, la planificación del procesador y la comunicación y sincronización entre procesos.

Durante el desarrollo de la propuesta ponga en juego sus conocimientos, revise el material y no olvide revisar los criterios que se tomarán en cuenta para su evaluación, tratando de ajustar su producción a ellos.

Este trabajo tiene fecha de comienzo y finalización. Consulte el Cronograma de actividades de la Asignatura para realizar su entrega en tiempo y forma, ya que es de carácter obligatorio.

Fin del Módulo I



Anexo



Trabajo práctico sugerido

Trabajo práctico Nº 5: **Proceso productor y proceso consumidor**

Presentación

Este trabajo tiene el propósito de orientarlo/a para la comprensión de los temas desarrollados en la tercer unidad de esta asignatura.

Nuestro objetivo es comenzar a percibir el problema que se presenta en los sistemas cuando los procesos tienen que trabajar en forma cooperativa para realizar sus tareas, recurriendo a elementos de comunicación para transmitirse información.

Este trabajo intenta ser el principio de nuestro estudio de la comunicación y sincronización entre procesos, por ello lo primero que tenemos que hacer es conocer el problema, para luego pensar en una solución, así los sistemas operativos tuvieron que brindar herramientas para que los procesos pudieran llevar adelante su trabajo en forma cooperativo y sincronizada.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

- Comenzar a entender el problema de los procesos productores y consumidores.
- Comenzar a pensar en las posibles soluciones a este problema.
- Prepararnos para entender las herramientas utilizadas en la comunicación y sincronización entre procesos.

Le presentamos a continuación, las consignas de trabajo:

Consignas

1. Realizar una breve descripción del proceso productor y del proceso consumidor.
2. Identificar los elementos del productor y del consumidor que son compartidos por ambos procesos.
3. Describir el funcionamiento del buffer.
4. Describir la función de cada una de las variables de memoria.



5. Explicar porque esta solución permite que haya $n-1$ elementos en el buffer al mismo tiempo.
6. Proponer una solución que permita utilizar los n elementos en el buffer al mismo tiempo.

Al finalizar, compare su producción con la grilla que incluimos a continuación.



Grilla de Autocorrección Nº 5: Proceso productor y proceso consumidor

Orientaciones para la corrección

La solución de este trabajo práctico está basada en la bibliografía requerida de esta materia y presenta el pseudo código de un proceso productor y un proceso consumidor que trabajan en forma concurrente y cooperativa.

Las consignas presentadas están estrechamente relacionadas con este tema. Es por ello que lo invitamos a realizar la siguiente actividad de lectura.

- Silberschatz A. y Galvin P.; Capitulo 4. Procesos. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. página. 102.
- Silberschatz A. y Galvin P.; Capitulo 6. Sincronización de Procesos.. **En su: Sistemas Operativos**; 5ta Edición; México Addison Wesley; 1999. páginas. 154-157.
- Silberschatz A. Galvin P. Gagne G.; Capítulo 3 Procesos.; **En su: Fundamentos de Sistemas Operativos**; 7ma Edición; España Mc Graw Hill 2007. p.88.



Trabajo práctico sugerido

Trabajo práctico Nº 6: **Sección Crítica**

Presentación

El objetivo de esta práctica es fortalecer los conceptos teóricos aprendidos de la sección crítica, tema muy importante para poder llevar adelante una buena sincronización entre procesos.

La práctica se basa en el análisis teórico de la problemática de la sección crítica de los procesos y los recursos utilizados para resolver dicha problemática.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

- Entender el problema de la sección crítica, para poder en las soluciones que hacen que la sección crítica quede protegida.
- Estudiar las soluciones al problema de la sección crítica.
- Comprender la necesidad de un recurso que les permita a los diseñadores de sistemas productor consumidor solucionar el problema del acceso concurrente a la sección crítica.

Le presentamos a continuación, las consignas de trabajo:

Consignas

- 1.- ¿Cuáles son las soluciones basadas en el software para el problema de la sección crítica para dos procesos y para n procesos?
- 2.- ¿Cuál es la solución basada en el hardware para el problema de la sección crítica, y cuál es la ventaja de esta solución?
- 3.- Encuentre un ejemplo que muestre una sección crítica de un proceso y explique porque es una sección crítica.
- 4.- Si necesitamos que dos procesos se comuniquen entre si, podemos proponer como medio de comunicación un simple archivo (aunque ese no

sea el medio de comunicación mas indicado), de forma que un proceso escribe en el archivo y el otro proceso lee del archivo. Proponga un esquema del código para el proceso escritor y para el proceso lector.

5.- El siguiente fragmento de código intenta resolver el problema de la sección crítica para dos procesos P0 y P1.

```
While (turno != i);  
    SECCIÓN CRÍTICA;  
    Turno = j;
```

La variable turno tiene valor inicial 0. La variable i, vale 0 en el proceso P0 y 1 en el proceso P1. La variable j vale 1 en el proceso P0 y 0 en el proceso P1. ¿Resuelve este código el problema de la sección crítica?



Si surgen dudas u obstáculos que dificultan el aprendizaje o la comprensión de los contenidos durante la resolución de estas consignas, por favor, comuníquese con su tutor. Preséntele con claridad sus consultas para que él pueda brindarle las orientaciones que le permitirán resolverlas.



Trabajo práctico sugerido

Trabajo práctico N° 7: **Semáforos**

Presentación

Este trabajo consiste en una serie de ejercicios que esta orientada a la aplicación de los conceptos adquiridos en el tema semáforos sobre ejercicios de razonamiento práctico. Su finalidad es reforzar la habilidad en la programación concurrente utilizando conceptos de sincronización.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

- Entender a través de la resolución de ejercicios el funcionamiento y el uso de los semáforos como recurso de sincronización entre procesos.
- Comprender la necesidad de la sincronización entre procesos para que puedan realizar tareas de forma cooperativa.

- Interpretar la resolución de la sincronización utilizando el recurso semáforo.

Le presentamos a continuación, las consignas de trabajo:

Consignas

Indicar los semáforos necesarios, su correspondiente inicialización, la ubicación de las operaciones wait y signal dentro de cada proceso para lograr las siguientes secuencias de ejecución de procesos y el estado de cada semáforo en los distintos instantes de ejecución.

Ejercicio 1

Secuencia de ejecución ABCABC...

- 1) Semáforos:.....
- 2) Inicialización:.....
- 3) Ubicación de las operaciones wait y signal dentro de cada proceso.

Proceso A	Proceso B	Proceso C

- 4) Estado de cada semáforo en los distintos instantes de ejecución.



Semáforos		Sem 1			
Valor Inicial		1			
Instante	Proceso				
T0	A				
T1	A				
T2	B				
T3	B				
T4	C				
T5	C				

Ejercicio 2

Secuencia de ejecución A(BoC)A(BoC)...

Ejercicio 3

Secuencia de ejecución (AoB)C(AoB)C...

Ejercicio 4

Secuencia de ejecución ABACABAC...

Ejercicio 5

Secuencia de ejecución A(BoC)D(EyF)...

Ejercicio 6

Secuencia de ejecución A B C B A B C B A B C B...



Si surgen dudas u obstáculos que dificultan el aprendizaje o la comprensión de los contenidos durante la resolución de estas consignas, por favor, comuníquese con su tutor. Preséntele con claridad sus consultas para que él pueda brindarle las orientaciones que le permitirán resolverlas.



Trabajo práctico sugerido

Trabajo práctico Nº 8: **Semáforos en el sistema operativo LINUX**

Presentación

Este trabajo consiste en la implementación en el sistema operativo LINUX, utilizando el compilador gcc o cc y el editor de código ASCII, vi o vim para poder compilar y ejecutar los ejercicios propuestos en este trabajo práctico, relacionados con la sincronización entre procesos.

Este trabajo intenta favorecerle el acceso a las siguientes metas de aprendizaje:

- Aprender el uso de los semáforos en el sistema operativo LINUX.
- Desarrollar una práctica de programación de semáforos para tener el verdadero conocimiento de la implementación de semáforos para la sincronización entre procesos.
- Interpretar el uso real de los semáforos.

Le presentamos a continuación, las consignas de trabajo:

Consignas

1. Editar y compilar los códigos ejemplo, usando el editor vi o vim y el compilador gcc o cc en su sistema operativo LINUX.
2. Una vez editados y compilados los programas ejemplos, proceder a ejecutar los programas.
3. Interpretar y hacer un informe de la experiencia realizada.



Usted ya sabe, si surgen dudas u obstáculos que dificultan el aprendizaje o la comprensión de los contenidos durante la resolución de estas consignas, por favor, comuníquese con su tutor. Preséntele con claridad sus consultas para que él pueda brindarle las orientaciones que le permitirán resolverlas.

Definición de las funciones a utilizar.

1) Crear Semáforo

Para **crear semáforos** utilizamos la función **int crea_sem (key_t key, int nsems)** esta función definida en la cabecera **operIPC.h** llama a la función **int semget (key_t key, int nsems, int semflg)**.

crea_sem retorna el identificador del array de semáforos que se creó. Recibe como argumentos la clave del array de semáforos y la cantidad de semáforos que se van a crear en el array de semáforos.

Ejemplo para crear un array de semáforos:

```
#include <operIPC.h>
#include <string.h>

main(int argc, char **argv)
{
    int semid ;
    int nsem ;
    key_t key;
    key = atoi(argv[1]);
    nsem = atoi(argv[2]);
    semid = crea_sem(key, nsem);
    if (semid == -1) perror("error semget");
    else perror("Exito");
}
```

2) Inicializar semáforos

Para inicializar semáforos utilizamos la función **int init_sem(int semid, int nsem, int valor)** esta función definida en la cabecera **operIPC.h** llama a la función **int semctl(int semid, int semnun, int cmd, union semun arg)**.

init_sem retorna 0 o -1. Recibe como parámetros el identificador del array de semáforos, el número del semáforo y el valor de inicialización del semáforo.

Ejemplo para inicializar semáforos

```
#include <operIPC.h>

main(int argc, char **argv)
{
    int semid;
    int nsem;
    int valor;
    int status;
    semid = atoi(argv[1]);
    nsem = atoi(argv[2]);
    valor = atoi(argv[3]);
    status = init_sem(semid,nsem,valor);
    if (status == -1) perror("error semctl\n");
    else perror("");
}
```

3) Operación Wait

Para realizar la operación wait utilizamos la función **void P(int semid, int nsem)** esta función definida en la cabecera **operIPC.h** llama a la función **int semop(int semid, struct sembuf *sops, unsigned nsops)**.

P recibe como parámetros el identificador del array de semáforos, el número del semáforo sobre el cual se realiza la operación **P**.

4) Operación Signal

Para realizar la operación signal utilizamos la función **void V(int semid, int nsem)** esta función definida en la cabecera **operIPC.h** llama a la función **int semop (int semid, struct sembuf *sops, unsigned nsops)**.

V recibe como parámetros el identificador del array de semáforos, el número del semáforo sobre el cual se realiza la operación **V**.



Ejemplo de Sincronización entre Procesos. Secuencia de ejecución ABABABABAB...

1. Creamos un array de dos semáforos (semáforo 0 y semáforo 1)
2. Inicializamos el semáforo 0 en 1 y el semáforo 1 en 0.
3. Los procesos reciben desde línea de comando el identificador del array de semáforos.

Proceso A

```
#include <operIPC.h>
#include <stdio.h>

main(int argc, char **argv)
{
    int semid;
    semid = atoi(argv[1]);
    while(1)
    {
        P(semid,0);
        printf("A\n"); // Seccion critica
        sleep(1);      //
        V(semid,1);
    }
}
```

Proceso B

```
#include <operIPC.h>
#include <stdio.h>

main(int argc, char **argv)
{
    int semid;
    semid = atoi(argv[1]);
    while(1)
    {
        P(semid,1);
        printf("B\n"); // Seccion critica
        sleep(1);      //
        V(semid,0);
    }
}
```

Los procesos A y B ejecutan la sincronización en un ciclo infinito, modifique los procesos para que la sincronización termine luego de ejecutar la sección crítica n veces, el n se ingresa desde línea de comando cuando se lanzan a ejecución los procesos.

Es interesante ejecutar el proceso A en una consola y el proceso B en otra.
Sincronizar la siguiente secuencia de ejecución de procesos ABACABACABAC...

Emisor A	Receptor B	Receptor C
<pre>#include<operIPC.h> #include <stdio.h> main(int argc, char **argv) { int semid; semid = atoi(argv[1]); while(1) { P(semid,0); printf("A\n"); sleep(1); V(semid,3); } }</pre>	<pre>#include <operIPC.h> #include <stdio.h> main(int argc, char **argv) { int semid; semid = atoi(argv[1]); while(1) { P(semid,1); P(semid,3); printf("B\n"); sleep(1); V(semid,2); V(semid,0); } }</pre>	<pre>#include <operIPC.h> #include <stdio.h> main(int argc, char **argv) { int semid; semid = atoi(argv[1]); while(1) { P(semid,2); P(semid,3); printf("C\n"); sleep(1); V(semid,0); V(semid,1); } }</pre>

Contenido de la cabecera operIPC.h

Conjunto de funciones que permiten realizar la operación P o wait, V o signal sobre semáforos además de crear e inicializar semáforos.

```
/* operación wait */
/* Parámetros
    Identificador del array del semáforo
    Numero del semáforo */
```

```
#include <sys/ipc.h>
#include <sys/sem.h>
#include <stdio.h>

void P(int semid, int nsem)
{
    struct sembuf buf;
    buf.sem_num = nsem;
    buf.sem_op = -1;
    buf.sem_flg = 0;
    semop(semid,&buf,1);
}
```

```
/* operación signal */
```



/* Parámetros

identificador del array del semáforo
Numero del semáforo

```
*/  
void V(int semid, int nsem)  
{  
    struct sembuf buf;  
    buf.sem_num = nsem;  
    buf.sem_op = 1;  
    buf.sem_flg = 0;  
    semop(semid,&buf,1);  
}
```

/* crea un array de semáforos */

/* Parámetros

Clave del semáforo
Cantidad de semáforos del array

```
*/  
int crea_sem(key_t key, int cantsem)  
{  
    int semid ;  
    semid = semget(key,cantsem,IPC_CREAT | 0666);  
    printf("Anote en su cuaderno el Identificador %d de su sem\n",semid);  
    return semid;  
}
```

/* inicializa semáforos */

/* Parámetros

Identificador del array de semáforos
Numero del semáforo
Valor inicial del semáforo (0 o 1)

```
*/  
int init_sem(int semid,int nsem,int valor)  
{  
    int status;  
    status = semctl(semid,nsem,SETVAL,valor);  
    return status;  
}
```