

Sistemas Operativos

Trabajo Práctico

Profesor: Lic. Horacio Pendentí
Ayudante: Lic. Luis Rojas
Ushuaia - Tierra del Fuego

Índice de Contenidos

Cronograma	1
Bibliografía	2
Trabajo Práctico N° 1	5
Trabajo Práctico N° 2	7
Trabajo Practico N° 3	10
Trabajo Práctico N° 4	14
Trabajo Práctico N° 5	19
Trabajo Práctico N° 6	22
Trabajo Práctico N° 7	25
Trabajo Práctico Integrador	27

Cronograma

Ficha de la Asignatura			
Asignatura	(IF037) Sistemas Operativos	Carrera/s	Aanlista Univrsitario de Sistemas Licenciatura en Sistemas
Titular	Guillermo Feierherd	Plan de Estudios	2010
JTP	Horacio Pendenti	Carga Horaria	4T y 4P/ Semana. Total 120hs
Asistente	Luis Rojas	Periodo	3er Año - 2do Cuatrimestre
		Correlativas Anteriores	(IF038) Introducción a la Concurrencia
		Correlativas Posteriores	(IF019) Redes y Transmisión de datos (IF060) Sistemas de Tiempo Real

Bibliografía

Bibliografía específica recomendada para la práctica.

TP	Título	Autor/es	Lectura	Cap's.
TP01- GSO	Operating systems: concepts and design. 2nd. ed. McGraw Hill	Milan Milenković	Básica	c1
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Básica	c1
	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Complem.	c1 c2
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c1c2 c3
TP02 - GPR	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c2
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c4
	Operating systems: concepts and design. 2nd. ed. McGraw Hill	Milan Milenković	Complem.	c2.1 c2.2 c2.3 c2.4
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c2.1
TP03- PPR	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c8
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c5
	Operating systems: concepts and design. 2nd. ed. McGraw Hill	Milan Milenković	Complem.	c2.5 c2.6 c2.7
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c2.4

TP	Título	Autor/es	Lectura	Cap's.
TP04 - MAC	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c6
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c8
	Operating systems: concepts and design. 2nd. ed.McGraw Hill	Milan Milenković	Complem.	c5
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c3.1 c3.2
TP05 - MNC	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c7
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c9
	Operating systems: concepts and design. 2nd. ed.McGraw Hill	Milan Milenković	Complem.	c6
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c3.3 c3.4 c3.5 c3.6 c3.7
TP06 - ARC	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c11
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c10 c11
	Operating systems: concepts and design. 2nd. ed.McGraw Hill	Milan Milenković	Complem.	c7
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c4

TP	Título	Autor/es	Lectura	Cap's.
TP07 - ESD	Operating Systems 2nd.ed. PrenticeHall	William Stallings	Básica	c10
	Operating Systems concepts. 5th. ed. Addison-Wesley	Silberschatz - Galvin	Complem.	c12 c13
	Operating systems: concepts and design. 2nd. ed. McGraw Hill	Milan Milenković	Complem.	c8
	Modern Operating Systems 1992. Prentice Hall	Andrew S. Tanenbaum	Complem.	c5
TPI - PP	Toda la bibliografía y teoría de la Asignatura.			
TPI - AM	Toda la bibliografía y teoría de la Asignatura.			
TPI - SD	Toda la bibliografía y teoría de la Asignatura.			

Trabajo Práctico N° 1

GSO – Generalidades de Sistemas Operativos

1. ¿Cuáles son las dos funciones principales de un Sistema Operativo? Descríbalas y mencione al menos dos ejemplos de cada una en algún Sistema Operativo que conozca.
2. Explique qué es una interrupción. Cómo se manejan múltiples interrupciones?. Qué significa deshabilitación de interrupciones.
3. Explique conceptualmente la diferencia entre una “interrupción”, un “trap” y un “system call”.
4. ¿Qué es multiprogramación? ¿Qué es multiprocesamiento? ¿Es posible la multiprogramación sin multiprocesamiento?
5. Mencione algunos recursos de hardware imprescindibles para implementar un Sistema Operativo multiprogramado. Justifique su respuesta.
6. En qué consiste la memoria caché (explique su funcionamiento) ?. Cuál es su utilidad?. Qué problemas causa su uso?.
7. Anote y comente las diferencias entre los significados que distintos autores asignan a los siguientes términos:
 - multiproceso
 - multiusuario
 - multiacceso
 - multitarea
8. En las computadoras primitivas, toda lectura de datos desde un dispositivo de entrada o grabación en un dispositivo de salida eran manejadas directamente por la CPU. Es decir, no había acceso directo a memoria (DMA). ¿Qué implicaciones tiene esta organización para la multiprogramación?
9. ¿Cuáles de las siguientes instrucciones deben admitirse únicamente en modo privilegiado?
 - a) Desactivación de todas las interrupciones.
 - b) Lectura del reloj del sistema que mantiene la hora del día.
 - c) Fijar la hora en el reloj del sistema que mantiene la hora del día.
 - d) Administrar registros de memoria.
 - e) Iniciar una operación de Entrada / Salida.
10. ¿Qué es el núcleo de un sistema operativo? ¿Porqué se suele mantener el núcleo en memoria principal? ¿Qué tareas realiza normalmente el núcleo?

- 11.** Cite algunas razones que justifiquen la eficacia de pasar partes importantes del núcleo a microcódigo.

Trabajo Práctico N° 2

GPR – Generalidades sobre Procesos.

1. Describa las distintas clases de interrupciones que conoce.
2. En algunas computadoras una interrupción provoca que el contenido de algunos registros de la UCP sea automáticamente guardado en posiciones fijas de memoria asociadas con el tipo de interrupción y reemplazado por un nuevo contenido extraído también de posiciones fijas de memoria con el mismo criterio. ¿Bajo qué circunstancias es práctico este enfoque? Explique porqué en general es inconveniente.
3. En la mayoría de los dispositivos, cuando las interrupciones son deshabilitadas (enmascaradas), quedan pendientes hasta que sean nuevamente habilitadas (desenmascaradas) y puedan ser procesadas. En el ínterin no son permitidas nuevas interrupciones y el funcionamiento de los dispositivos se detiene. No obstante, en los sistemas de tiempo real, el ambiente que genera las interrupciones está normalmente separado del sistema de cómputo por lo que, aunque las interrupciones sean enmascaradas, el ambiente continúa produciéndolas de cualquier manera. A menudo estas interrupciones son perdidas.
 - a) Discuta las consecuencias de la pérdida de interrupciones.
 - b) En un sistema de tiempo real, ¿es mejor perder ocasionalmente interrupciones o dejar el ambiente hasta que las interrupciones sean nuevamente desenmascaradas?
4. Proporcione algunas definiciones de proceso.
5. Defina los siguientes términos:
 - a) Programa (program)
 - b) Procedimiento (procedure)
 - c) Procesador (processor)
 - d) Usuarios (users)
 - e) Tarea (tasks) – sistemas operativos – sinónimo de proceso
 - f) Trabajo (job) – trabajo en batch -
6. Enumere la información que debería contener a su criterio un PCB (Process Control Block) o registro de la tabla de procesos.
7. ¿Por qué se necesita la Tabla de Control de Procesos (PCT) en un sistema de tiempo compartido? ¿Se necesita también en sistemas de computación personales en los que solo existe un proceso y ese proceso hace uso de todos los recursos del sistema hasta que termina?

8. Pinkert and Wear definen los siguientes estados para procesos: Ejecución (corriendo), Activo (listo), Bloqueado y Suspendido. Un proceso está bloqueado si está esperando el permiso para utilizar un recurso y está suspendido si está esperando que se complete una operación sobre un recurso que ya adquirió. En otros sistemas operativos estos dos últimos estados se agrupan en uno solo (bloqueado) y el estado suspendido es definido como se hizo en la teoría. Compare las ventajas relativas de ambos conjuntos de definiciones.
9. Un proceso completa su trabajo y sale del sistema. ¿Qué operaciones básicas es necesario realizar antes de continuar con otro proceso?
10. Es importante la capacidad que puede tener un proceso de crear nuevos procesos, pero tiene sus riesgos. Considere las consecuencias de permitir que un usuario ejecute el siguiente programa:

```
1 creador:  
2 process;  
3 while true do  
4     crear otro proceso como éste  
5 end;  
6
```

11. En los sistemas dedicados a un solo usuario, generalmente es obvio cuándo un programa entra en un ciclo infinito. Pero en sistemas multiusuario, en los que se ejecutan decenas o cientos de procesos, no se puede determinar con facilidad cuándo un proceso no está avanzado.
 - a) ¿Puede el Sistema Operativo determinar cuándo un proceso está dentro de un ciclo infinito? (en la práctica no, teóricamente sí)
 - b) ¿Qué métodos de protección se pueden incluir en un Sistema Operativo para evitar que los procesos que están en un ciclo infinito se ejecuten indefinidamente?
12. En algunos sistemas, un proceso creado se destruye automáticamente cuando su padre es destruido; en otros sistemas, los procesos creados son independientes de sus padres y la destrucción de un proceso padre no afecta a sus hijos. Exponga las ventajas y desventajas de cada uno de estos enfoques. Proporcione un ejemplo de una situación en la cual sea contraproducente que la destrucción de un parente lleve la destrucción de sus hijos.
13. Los procesos en espera consumen varios recursos del sistema. ¿Podría alguien sabotear un sistema mediante la creación repetida de procesos que esperan la ocurrencia de eventos que nunca suceden? ¿Qué precauciones podrían tomarse?
14. La administración de la espera es una parte esencial de cualquier sistema operativo. Hemos visto varios estados de espera, denominados ready (listo), blocked (bloqueado),

suspend / ready (suspendido / listo), suspend / blocked (suspendido / bloqueado). Para cada uno de ellos discuta:

- a) Como el proceso puede llegar a ese estado
- b) Que evento está esperando para salir del mismo
- c) La probabilidad de que el proceso pueda “perderse” esperando en ese estado indefinidamente.

Trabajo Practico N° 3

PPR – Planificación de Procesos.

1. Enumere algunas diferencias entre planificación de largo, mediano y corto plazo.
2. ¿A qué nivel de planificador le corresponde tomar la decisión en cada uno de los siguientes casos?
 - a) Asignar la CPU a un proceso listo cuando se encuentre disponible
 - b) Iniciar uno de una serie de procesos por lotes que han sido enviados por el spooler al disco y se encuentran en espera.
 - c) Suspender temporalmente un proceso para aliviar una sobrecarga a corto plazo de la CPU
3. Habitualmente no tiene sentido mantener la Lista de Procesos Bloqueados en orden de prioridad. ¿Porqué? ¿En qué circunstancias tendría sentido ordenarla de esta forma?
4. Los planificadores de torneo (ronda, round-robin) conservan normalmente una lista de todos los procesos ejecutables en la que cada proceso ocurre exactamente una sola vez en la lista. ¿Qué sucedería si un proceso ocurriera dos veces en la lista? ¿Puede pensar en alguna razón para permitir esta situación?
5. Las mediciones de cierto sistema han mostrado que el proceso promedio se ejecuta por un tiempo T antes de bloquearse por entrada/salida. Un cambio de proceso requiere un tiempo S . Para la planificación de un torneo con quantum Q , dé una fórmula para calcular la eficiencia de la CPU para cada uno de los casos siguientes:
 - a) $Q = \infty$
 - b) $Q > T$
 - c) $S < Q < T$
 - d) $Q = S$
 - e) $Q \rightarrow 0$ (Q tiende a 0)
6. Cinco trabajos en lote, identificados como A B C D y E, llegan a un centro de computación casi al mismo tiempo. Estos trabajos tienen tiempos de ejecución estimados de 10, 6, 2, 4 y 8 minutos. Sus prioridades (determinadas externamente) son 3, 5, 2, 1 y 4 respectivamente, donde 5 es la más alta. Determine el tiempo medio de retorno para cada uno de los siguientes algoritmos de planificación:
 - a) Planificación por prioridad
 - b) El primero en llegar es atendido (secuencia A B C D y E) (FCFS)
 - c) El trabajo mas corto primero (SJF)

Nota: asuma que el tiempo de conmutación entre procesos es cero y que los trabajos se ejecutan uno a la vez hasta que terminan

7. Cinco trabajos esperan ser ejecutados. Sus tiempos de ejecución estimados son 9, 6, 3, 5 y X. ¿En qué orden se deben ejecutar para minimizar el Tiempo Medio de Retorno?
8. El algoritmo de envejecimiento con $a=1/2$ se utiliza para predecir tiempos de ejecución. Las cuatro ejecuciones anteriores, de la más antigua a la más reciente, han sido de 40, 20, 40 y 15 mseg. ¿Cuál es la predicción de tiempo que sigue?
9. ¿Qué medida debe tomar el planificador a corto plazo cuando se lo invoca pero no hay ningún proceso preparado? ¿Es posible esta situación? En caso afirmativo sugiera una serie de medidas tales que ni el procesamiento adicional de eventos ni las transiciones de estado de proceso resulten menoscabadas.
10. Suponga que para mejorar el rendimiento de un sistema de computadoras se están considerando las tres opciones enumeradas a continuación:
 - a) Dividir la carga de trabajo en n colas separadas y hacer que a cada una le dé servicio una CPU separada cuya velocidad es x MIPS.
 - b) Mantener toda la carga de trabajo en una sola cola, pero hacer que la atiendan n CPU's separadas, cada una de las cuales tiene una velocidad de x MIPS, de manera que la unidad de trabajo siguiente se asigne a la primera CPU disponible.
 - c) Mantener toda la carga de trabajo en una sola cola, pero hacer que se le dé servicio por una sola CPU n veces mas rápida.

Discuta intuitivamente estas opciones y clasifíquelas por los tiempos medios de espera que sea probable que encuentren los programas individuales, cuando se emplee la planificación FCFS en todos los casos, sin que se permita ninguna sustitución por derecho preferente.

11. Un S.O. de tiempo compartido usa la planificación basada en la prioridad para los procesos críticos en cuanto a tiempo, y planificación de circuito cíclico para los procesos usuarios interactivos. En un momento determinado se aumenta la potencia del hardware sustituyendo la CPU por un modelo funcionalmente equivalente, pero dos veces más rápido. Exponga los cambios que experimentarán las distintas clases de usuarios. ¿Se deben cambiar algunos parámetros del S.O.? En caso afirmativo, ¿cuáles y cómo?. Explique el comportamiento que se espera del sistema como resultado de dichos cambios.
12. Proporcione un ejemplo que muestre por qué FCFS (First Come First Serve) no es un esquema apropiado de planificación de la CPU para usuarios interactivos.
13. Indique por qué son incorrectos los siguientes enunciados:
 - a) SRT (Shortest Remaining Time) siempre tiene un tiempo medio de respuesta menor que SJF (Shortest Job First).

- b) SJF es justo.
 - c) Cuanto más corto sea un trabajo, mejor servicio deberá recibir.
 - d) SJF resulta útil en los sistemas de tiempo compartido, pues da preferencia a los trabajos más cortos.
- 14.** Dos objetivos comunes de las políticas de planificación son reducir al mínimo los tiempos de respuesta y lograr un aprovechamiento máximo de los recursos. Indique por qué son contradictorios entre sí dichos objetivos. Analice las distintas políticas de planificación conocidas desde las siguientes perspectivas:
- a) Políticas dirigidas a lograr tiempos mínimos de respuesta y
 - b) Políticas dirigidas a lograr un aprovechamiento máximo de los recursos.

Luego:

- a) Clasifique las políticas según estos dos criterios.
 - b) Desarrolle una nueva política de planificación que permita a un sistema lograr un equilibrio aceptable entre estos dos objetivos en conflicto.
- 15.** Se tiene un sistema con una carga de trabajo detallada en la siguiente tabla. Un proceso está en estado de Listo cuando ha arribado:

Proceso	Burst time previsto	Remaining Time	Arrival time	Service time (t.total ejec.)
P_1	10	-	0	20
P_2	20	2	4	30
P_3	5	-	8	15
P_4	15	5	10	30

Después de cada ráfaga de ejecución, los procesos se bloquean por 25 unidades de tiempo haciendo Entrada Salida. Donde todos los tiempos están dados en milisegundos; considere los algoritmos de planificación, **FCFS**, **SJF**, **SRT**, y **Round Robin** (con $q = 10\text{ms}$).

Para las 4 estrategias anteriores, dibuje el diagrama de Gantt correspondiente, considerando que los trabajos deben ejecutar hasta completar su service time y calcule:

- a) Tiempo de Retorno de cada trabajo.
- b) Tiempo de Retorno normalizado para cada trabajo.
- c) Tiempo de Retorno de la Tanda.
- d) Tiempo Medio de Retorno de la Tanda.
- e) Tiempo de Espera de cada trabajo.

- f) Tiempo Medio de Espera (considerando tiempo de espera al que cada proceso espera exclusivamente por el procesador).
- * En los casos que así lo requieran, utilice el mismo burst de CPU previsto para cada trabajo, todas las veces que sea necesario hasta terminar de servir. Especifique los supuestos que asuma.

Trabajo Práctico N° 4

MAC – Administración de Memoria. Alocación Contigua.

1. ¿Cuál es la diferencia entre una dirección de memoria lógica y una dirección de memoria física?.
2. Describa cómo puede un Sistema Operativo determinar la capacidad de memoria física instalada en un Sistema.
3. Explique el concepto de Asignación de Memoria.
 - a) ¿Debe estar asignada toda la memoria de un Sistema en todo momento?.
 - b) ¿Es posible que haya fragmentos de memoria asignada y no ocupada? En caso afirmativo indique bajo qué circunstancias.
4. Explique en qué consiste cada uno de los siguientes términos:
 - a) Alocación contigua simple
 - b) Alocación contigua de memoria con especificación estática de particiones
 - c) Alocación contigua de memoria con especificación dinámica de particiones
- Ejemplifique gráficamente.
5. Explique en qué consiste cada uno de los siguientes métodos de selección de particiones:
 - a) **First Fit** (primer ajuste)
 - b) **Next Fit** (próximo ajuste)
 - c) **Best Fit** (mejor ajuste)
 - d) **Worst Fit** (peor ajuste)
6. ¿Puede haber fragmentación interna en un sistema con Especificación Dinámica de Particiones? Justifique.
7. Considere un sistema batch multiprogramado con un esquema de administración de memoria con Especificación Estática de Particiones descripto por la siguiente Tabla de Descripción de Particiones:

IDENT PARTICION	TAMAÑO	DIRECCION COMIENZO	ESTADO INICIAL (Libre/Ocupado)
P1	30 K	40.000	L
P2	40 K	70.000	L
P3	60 K	110.000	L
P4	30 K	170.000	L

Nota: Las primeras 40.000 posiciones de memoria (0 ... 39.999) están ocupadas por el S.O..

- a) ¿Son los datos coleccionados en la tabla anterior suficientes para ejercer la administración? Si no es así, agregue los que Ud. considere necesarios.
- b) ¿Qué ordenamiento le daría a la tabla? ¿Por qué?.
- c) ¿Mantendría una única tabla o 2 separadas? (una para particiones ocupadas y otra para particiones libres) ¿Por qué?.
- d) ¿Se obtienen resultados distintos si se usan las estrategias de alocación de particiones **First-Fit** , **Next-Fit** y **Best-Fit** para una misma tanda de trabajos? Piense en términos de Tiempo de Retorno para la tanda completa ($TR_1 + TR_2 \dots + TR_n$).
- e) Simule este Sistema para distintas tandas de trabajos (más de 8 por cada tanda), donde por cada trabajo se conoce la cantidad de memoria requerida y el tiempo que permanecerá en la memoria. La primera tanda de trabajos a utilizar es la siguiente:

NOMBRE TAREA	MEMORIA REQUERIDA (en K unidades)	TIEMPO DE RETORNO (en Unidades de Tiempo)
T_1	20	7
T_2	30	15
T_3	40	5
T_4	40	5
T_5	30	5
T_6	30	4
T_7	20	5
T_8	30	3

- f) Calcule para cada tanda de trabajos y cada una de las estrategias de alocación de particiones (First-Fit, Next-Fit y Best-Fit) los siguientes indicadores:
 - i. Tiempo de Retorno para la tanda completa.
 - ii. Indice de Fragmentación = (Indice de Fragmentación Interna + Indice de Fragmentación Externa).

Donde **I.F.I.** = \sum (memoria asignada y no ocupada * tiempo que permanece en esa condición) e **I.F.E.** = \sum (memoria no asignada * tiempo que permanece en esa condición). "Mientras haya trabajos esperando para su ejecución".

Asuma que:

- Todos los trabajos están presentes al momento de iniciar este proceso.
- Los trabajos se procesan en estricto orden de llegada.

- Todos los trabajos son ejecutables, es decir que no hay ninguno que requiera más memoria que la mayor de las particiones.
- El Overhead es 0 en todo caso, con la única salvedad que los trabajos se cargan en memoria uno a la vez.

8. Considere un Sistema batch multiprogramado que utiliza un esquema de Administración de Memoria con Especificación Dinámica de Particiones. El sistema cuenta con un total de 130K de memoria para procesar la tanda de trabajos que se enuncia a continuación:

Tarea	Tamaño	Tiempo en Memoria
T_1	30	6
T_2	20	15
T_3	20	4
T_4	20	10
T_5	30	2
T_6	20	8
T_7	30	10
T_8	10	3
T_9	10	5
T_{10}	20	8

Ud. debe:

- Indicar cuál sería una estructura apropiada para poder efectuar la administración de memoria en este sistema.
- Efectúe la simulación de carga de los programas en este sistema bajo las estrategias de selección de particiones First-Fit, Best-Fit, Next-Fit y Worst-Fit e indique los siguientes resultados: Tiempo de Retorno de la Tanda e Indice de Fragmentación externa.
- Compare los resultados obtenidos y proporcione una conclusión.

Para resolver, asuma:

- Todos los trabajos están presentes al momento de iniciar este proceso.
- Los trabajos se procesan en estricto orden de llegada – batch -.
- No hay compactación de memoria.
- La memoria se puede asignar byte a byte.
- El Overhead es 0 en todo caso, con la única salvedad que los trabajos se cargan en memoria uno a la vez.

9. Compare los resultados obtenidos en los ejercicios 7 y 8 al procesar las distintas tandas con las estrategias de alocación de particiones comunes a ambos.
10. Para un sistema que usa especificación dinámica de particiones encuentre una tanda para cada estrategia (First-Fit, Best-Fit y Worst-Fit) para la cual esa estrategia produzca el menor tiempo de retorno para la tanda completa.
11. Explique el mapeo de direcciones lógicas de memoria a direcciones físicas bajo segmentación y paginación.
12. Considere un sistema de paginación simple con los siguientes parámetros:
 - **Memoria física:** 2^{32} bytes.
 - **Tamaño de página:** 2^{10} bytes.
 - **Espacio de direcciones lógico:** 2^{16} páginas.

Ud. debe responder:

- a) Cuántos bits hay en una dirección lógica?
 - b) Cuántos bytes hay en un frame?
 - c) Cuántos frames hay en la memoria física?.
 - d) Cuántos bits en una dirección física o absoluta especifican el frame?
 - e) Cuántas entradas tiene una tabla de páginas?.
 - f) Cuántos bits hay en cada entrada de la tabla de páginas?. Asuma que la tabla de páginas tiene un bit de válida.
 - g) Cuál es el tamaño máximo de un programa en bytes?
13. Realice la traducción de direcciones lógica a física en binario de: **0001010010111010** bajo los siguientes esquemas hipotéticos de administración de memoria, explicando su respuesta y los supuestos que asuma:
 - a) Un sistema paginado con un tamaño de página de 256 direcciones, usando una tabla de páginas en la que el número de frame es cuatro veces más pequeño (en binario) que el número de página.
 - b) Un sistema que usa segmentación simple, con un tamaño máximo de segmento de 1K direcciones, donde el segmento en cuestión tiene su dirección base: **1001110000000000**.
 14. Considere un sistema con administración de memoria consistente en segmentación simple que tiene la siguiente tabla de segmentos:

Dirección comienzo	Longitud
660	248
1752	422
222	198
996	604

Para cada una de las siguientes direcciones lógicas, expresadas en decimal en la forma (s#,offset), determine la dirección física correspondiente o indique si ocurre un fallo de segmento.

- a) 0,198
- b) 2,156
- c) 1,530
- d) 3,444
- e) 0,222

Trabajo Práctico N° 5

MAC – Administración de Memoria. Alocación Contigua.

- 1.** Suponga que tiene la siguiente tabla para los procesos que se están ejecutando en un momento dado en un procesador.

Nro. de Página	Válida	Bit R	Bit M	Nro. de Bloque
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

- a) Describa como se traslada una dirección virtual en una dirección física.
- b) Qué dirección física, si la tiene; le corresponde a cada una de las siguientes direcciones virtuales:
 - i. 1052
 - ii. 2221
 - iii. 5499

Nota: Todos los números que se indican son números decimales, cualquier cosa que se numera se lo hace comenzando por cero, y todas las direcciones de memoria son direcciones de byte. El tamaño de la página es de 1024 bytes.

- 2.** Una computadora tiene cuatro bloques de memoria. El momento de carga, último acceso y los bits **R** y **M** (bit de acceso y modificación respectivamente) para cada página se muestran en la siguiente tabla:

PÁGINA	CARGADA	ULT. REF.	BIT R	BIT M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

Los tiempos indicados en la misma están en pulsos de reloj. Al producirse un page-fault:

- a) ¿Qué página se reemplazará al utilizar un algoritmo **NRU**?

- b) ¿Qué página se reemplazará al utilizar un algoritmo **FIFO**?.
- c) ¿Qué página se reemplazará al utilizar un algoritmo **LRU**?.
3. Un proceso hace referencia a cinco páginas, A, B, C, D y E en el siguiente orden: A; B; C; D; A; B; E; A; B; C; D; E. Asuma que el algoritmo de reemplazo es **FIFO** y encuentre el número de transferencias de páginas durante esta secuencia de referencias comenzando con una memoria vacía para:
- a) 3 bloques de memoria.
- b) 4 bloques de memoria.
4. Compare el método FIFO y LRU para 4 bloques de memoria y la misma secuencia de requerimientos de página del ejercicio anterior. ¿Cuál produce más page-fault?
5. Suponga que la siguiente sentencia de un programa:
- ```
1 for i in 1 ... n do
2 A [i] := B [i] + C [i]
3 endfor;
```
- Es ejecutada en una memoria con un tamaño de página de 1000. Usando una máquina que tiene un conjunto completo de instrucciones registro a registro y emplea registros índice para el direccionamiento, escriba un hipotético programa en lenguaje de máquina para implementar la sentencia anterior. Luego muestre la secuencia de referencias a página durante la ejecución cuando  $n = 1000$ .
6. Cuando se comenzaron a implementar las memorias virtuales, las memorias principales eran mucho más pequeñas que las actuales y a menudo los programas eran más grandes que la memoria disponible. Desde entonces la tecnología de memoria ha cambiado y las memorias de múltiples megabytes son comunes en las máquinas actuales. Con memorias en el rango de los gigabytes para los próximos años y con discos cuyo tiempo de acceso no ha mejorado al mismo ritmo ¿será la memoria virtual un mecanismo conveniente de administración de memoria? Explique su respuesta.
7. A continuación se especifica una secuencia de referencias a páginas efectuadas por un proceso (cadena de referencia o reference string): 1,2,3,4,6,1,7,6,8,1,3,4,3,8,2. Se desea saber cuántos fallos de página ocurrirán al utilizar los siguientes algoritmos de reemplazo de páginas en caso de asignarle al proceso tres y cuatro bloques (frames).
- a) LRU
- b) Óptimo
8. Explique el concepto de “**thrashing**”.
9. a) ¿Por qué es generalmente más deseable reemplazar una página no modificada que una modificada?

- b) ¿Bajo qué circunstancias podría ser deseable reemplazar una página modificada?.
- 10.** Describa la estructura de una Tabla Invertida y su funcionamiento, así como sus ventajas y desventajas frente a una tabla convencional.
- 11.** Una máquina tiene direcciones virtuales de 48 bits y direcciones físicas de 32 bits. Si las páginas son de 8 Kbytes, a) cuántas entradas son necesarias para una tabla de páginas convencional?, b) cuántas para una tabla invertida?.
- 12.** ¿En qué consiste el concepto de “**Prepaginación**”?.
- 13.** Una computadora cuyos procesos tienen 1.024 páginas en sus espacios de direcciones mantiene sus tablas de páginas en memoria. La sobrecarga (overhead) requerida para leer una palabra de la tabla de páginas es de 5ns. Para reducir esta sobrecarga, la computadora tiene un TLB que contiene 32 pares (página virtual, marco de página) y puede realizar una búsqueda en 1 ns. Qué proporción de aciertos se necesita para reducir la sobrecarga promedio a 2 ns?.

## Trabajo Práctico N° 6

ARC - Archivos

1. ¿Porqué algunos sistemas registran el tipo de un archivo mientras otros dejan esta tarea al usuario o simplemente no implementan una facilidad que permita distinguir entre distintos tipos de archivos?
2. ¿Cuáles son las ventajas y desventajas de recordar, conjuntamente con otros atributos de un archivo, el nombre del programa que lo creó?
3. Explique el propósito de las operaciones **OPEN** y **CLOSE**.
4. Algunos sistemas dejan que los archivos sean compartidos permitiendo el acceso a una única copia del mismo, en tanto que otros mantienen varias copias (una por cada usuario que está utilizando el archivo). Discuta los méritos relativos de cada una de las variantes mencionadas.
5. Algunos investigadores han sugerido que, en lugar de tener una lista de acceso asociada con cada archivo (en la que se especifica que usuarios pueden accederlo y qué acciones pueden realizar), deberíamos tener una lista de control de usuarios (en la que se especifique qué archivos puede acceder cada uno y qué acciones puede realizar). Discuta los méritos relativos de cada una de las variantes mencionadas.
6. El concepto de directorio de trabajo en un sistema de archivos jerárquico permite que los usuarios puedan referirse a los archivos que están contenidos en el directorio de trabajo utilizando nombres parcialmente calificados. Esboce un método para extender esta facilidad y permitir que los usuarios hagan referencia a utilitarios del sistema almacenados en otros directorios y que se utilizan frecuentemente suministrando únicamente su calificación parcial y sin abandonar el directorio de trabajo. Su propuesta no debe requerir la creación y mantenimiento de múltiples copias del archivo. Si los hubiera, suministre una especificación de los comandos que deberá agregar al sistema y de los servicios necesarios para soportar su solución.
7. Los archivos secuenciales indexados (index sequential files), son bastante populares pese a que la experiencia ha demostrado que el acceso directo a los mismos puede ser bastante lento. ¿Porqué ocurre esto? ¿En qué circunstancias es conveniente acceder a tales archivos secuencialmente? ¿En qué circunstancias debería el diseñador de la aplicación utilizar archivos directos en lugar de archivos secuenciales indexados?
8. El acceso secuencial a los archivos secuenciales indexados es mucho más lento que a los archivos secuenciales. No obstante, muchos diseñadores implementan sus aplicaciones utilizando el primer tipo de archivos. ¿Porqué lo hacen?.
9. Calcule el número de accesos a disco necesarios para leer 20 bloques lógicos de un archivo con:

- a) Alocación contigua.
- b) Alocación encadenada.
- c) Alocación indexada de espacio.

- 10.** Cuando un archivo es abierto concurrentemente por varios procesos, ¿cada uno de ellos debería construir un bloque de control de archivo (file control block) propio para su conexión con el archivo compartido o deberían todos los procesos involucrados compartir un solo file-control-block?. Proponga una estrategia que Ud. considere como la mas apropiada para el manejo de los archivos compartidos.
- 11.** Considere un file system que implementa alocación indexada de espacio. Se sabe que cada sector del disco consta de 512 bytes, que el nodo índice primario de cada archivo sólo soporta 8 punteros, mientras que los nodos índices de nivel inferior (que contienen únicamente punteros), soportan 128 punteros. Ud. Deberá calcular el tamaño máximo de un archivo en Bytes para el caso en que éste:
- a) - Solo tenga un nodo índice (el primario).
    - Cuente con nodos índices de segundo nivel.
  - b) Para éste FS, cuántos sectores en total ocupa un archivo de 5121 bytes en el disco?.
- Especifique los supuestos que asuma.
- 12.** Complete la siguiente tabla con una “X” en las intersecciones que a su criterio considere apropiadas para la Organización (puede haber más de 1 X por cada fila), y sugiera uno o más métodos de alocación para ese archivo en la última columna.

| Uso del archivo                                                                                           | Organización sugerida |           |                     |          |         | Método/s de Alocación sugerido |
|-----------------------------------------------------------------------------------------------------------|-----------------------|-----------|---------------------|----------|---------|--------------------------------|
|                                                                                                           | Pila                  | Secuencia | Secuencial Indexado | Indexado | Directo |                                |
| Actualización masiva de registros poco frecuente.                                                         |                       |           |                     |          |         |                                |
| Archivo generado por una impresión.                                                                       |                       |           |                     |          |         |                                |
| Poco frecuente se accede a un registro y el archivo no tiene variaciones importantes en tamaño.           |                       |           |                     |          |         |                                |
| Muy frecuentemente se accede a un registro y el archivo no tiene variaciones importantes en tamaño.       |                       |           |                     |          |         |                                |
| Muy frecuentemente se accede a grupos de registros en distinto orden y poco frecuentemente a un registro. |                       |           |                     |          |         |                                |

## Trabajo Práctico N° 7

ESD – Entrada / Salida - Discos

- 1.** Explique los siguientes conceptos:
  - a) Entrada/Salida programada (Programmed I/O).
  - b) Entrada/Salida manejada por interrupciones (Interrupt driven I/O).
  - c) Acceso Directo a Memoria (Direct Memory Access [D.M.A]).
- 2.** Defina y grafique su interpretación de:
  - a) Canal de Entrada/Salida (I/O channel).
  - b) Procesador de Entrada/Salida (I/O processor).
- 3.** ¿Porqué es importante que un Sistema Operativo haga administración de disco (disk scheduling)?.
- 4.** Defina:
  - a) Tiempo de posicionamiento (seek time).
  - b) Retardo rotacional (rotational delay).
  - c) Tiempo de acceso total (acces time según Stallings).
  - d) Tiempo de transferencia (transfer time según Stallings).
- 5.** Un disco tiene 305 cilindros, 4 cabezas y 17 sectores de 512 bytes cada uno por pista. Gira a una velocidad de 3000 rpm., y tiene un conjunto de cabezas móviles con un tiempo de posicionamiento promedio de 30 ms. El dispositivo tiene un coeficiente de transferencia (data transfer rate) de 4 Mbps (Megabits por segundo). Calcule el tiempo necesario para transferir:
  - a) 20 sectores distribuidos en forma contigua en el disco.
  - b) 20 sectores distribuidos en forma randómica.
- Identifique los factores dominantes en la determinación de los tiempos de transferencia. Especifique los parámetros que Ud. asuma en cada uno de los casos.
- 6.** ¿En qué sentido es SCAN mas justo que SSTF?. En qué sentido es más justo C-SCAN que SCAN?.
- 7.** Se cuenta con una unidad de disco que tiene 4.001 cilindros, numerados de 0 a 4.000; se sabe que gira a una velocidad de 6.000 rpm, que tiene 32 sectores por pista y que ha terminado de servir un requerimiento ubicado en el cilindro número 225, mientras que el requerimiento inmediatamente anterior servido fue en el cilindro 189. Además se conoce el tiempo que le toma al brazo recorrer los 4.000 cilindros y es de 40 ms (milisegundos).

El seek time para una distancia cualquiera se acepta como una proporción de los 40 ms. expresados anteriormente. Finalmente, se conoce una lista de requerimientos pendientes de ser servidos, por # de cilindro: 48, 987, 2.605, 3.005, 640, 693, 1.028, 693, 2.240. Para los siguientes algoritmos de scheduling de disco:

- 1) F.C.F.S.
- 2) S.S.T.F.
- 3) SCAN
- 4) LOOK
- 5) C-SCAN
- 6) C-LOOK
- 7) N-STEP-SCAN
- 8) N-STEP-LOOK
- 9) F-SCAN
- 10) F-LOOK

Determine y calcule:

- a) Distancia total.
- b) Tiempo de acceso total (asumiendo que cada requerimiento es por un único bloque).

Asuma para: 7 y 8 N = 4. Para 9 y 10 que los últimos 3 requerimientos llegan al sistema mientras se esta sirviendo la lista presente de los primeros 6 requerimientos

## Trabajo Práctico Integrador

### TRABAJO PRÁCTICO DE IMPLEMENTACION Nº 1 TPI-01-PP – PLANIFICACION DEL PROCESADOR

- **Objetivo:** Se trata de programar un sistema que simule distintas estrategias de planificación del procesador (dispatcher), y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada estrategia.
- **Características del sistema a simular:** Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe leer un archivo en el que cada registro tiene los siguientes datos:

- Nombre del proceso.
- Tiempo de arribo.
- Cantidad de ráfagas de CPU a emplear para terminar.
- Duración de la ráfaga de CPU.
- Duración de la ráfaga de entrada-salida entre ráfagas de CPU.
- Prioridad externa.

Completada la lectura del archivo aceptará una entrada por teclado que indicará la política de planificación a aplicar a la tanda. Como mínimo se deben permitir las siguientes opciones:

- a) FCFS (First Come First Served).
- b) Prioridad Externa.
- c) Round-Robin.
- d) SPN (Shortest Process Next).
- e) SRTN (Shortest Remaining Time Next).

Finalmente permitirá introducir los siguientes datos:

- a) Tiempo que utiliza el sistema operativo para aceptar los nuevos procesos (TIP).
- b) Tiempo que utiliza el sistema operativo para terminar los procesos (TFP).
- c) Tiempo de conmutación entre procesos (TCP).
- d) Quantum (si fuera necesario).

El simulador ejecutará la tanda hasta que se hayan completado la totalidad de los trabajos produciendo las siguientes salidas:

- Un archivo en el que se indiquen todos los eventos que se producen en el sistema a lo largo de la simulación y el tiempo en el que ocurren los mismos. Ejemplos de eventos: arriba un trabajo, se incorpora un trabajo al sistema, se completa la ráfaga del proceso que se está ejecutando, se agota el quantum, termina una operación de entrada-salida, se atiende una interrupción de entrada-salida, termina un proceso.
- Al finalizar la simulación imprimirá y mostrará por pantalla –como mínimo– los siguientes indicadores:
  - a) Para cada proceso: Tiempo de Retorno, Tiempo de Retorno Normalizado, Tiempo en Estado de Listo.
  - b) Para la tanda de procesos: Tiempo de Retorno y Tiempo Medio de Retorno.
  - c) Para el uso de la CPU: Tiempos de CPU desocupada, CPU utilizada por el SO, CPU utilizada por los procesos (en tiempos absolutos y porcentuales).

- **Otras condiciones:**

- Deberá probarlo con al menos cuatro tandas de trabajos que tengan características distintas cada una y comentar los resultados obtenidos con cada estrategia de planificación en función de las características de las tandas.
- Para resolverlo, utilice java, o puede elegir cualquier lenguaje de programación que conozca.
- El trabajo es unipersonal.
- Además de probar el simulador en clase, deberá presentar el ejecutable o ambiente de ejecución y el código fuente en soporte digital o repositorio.
- El simulador deberá ejecutarse de manera intuitiva en cualquier sistema operativo y sin la necesidad de tener que instalar librerías, programas, etc.
- El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los punto/s del mismo que tengan que ver con planificación de procesos, otorgándosele en el examen el máximo puntaje previsto para esos puntos.
- Se fijara una fecha límite para la entrega y muestra del trabajo sin excepción.
- Se acordara un archivo en formato JSON, para todos los trabajos, de modo que los resultados sean aproximados a los trabajos presentados.
- Se deberá presentar diagramas de Gantt, diagramas de clase, de flujo, etc. Que permita su rápida comprensión e interpretación del trabajo entregado. En el caso del diagrama de Gantt, deberá coincidir con los resultados en pantalla del simulador.

- **Acuerdos para su realización:**

- a. Orden de procesamiento de eventos:
  1. Corriendo a Terminado.
  2. Corriendo a Bloqueado.

3. Corriendo a Listo.
  4. Bloqueado a Listo.
  5. Nuevo a Listo.
  6. Finalmente el despacho de Listo a Corriendo.
- b. En Round Robin si tenemos un único proceso y su q termina, lo pasamos a listo y luego le volvemos a asignar la cpu (usamos un TCP). Para despachar el primer proceso también usamos un TCP.
- c. Un proceso pasa de bloqueado a listo instantáneamente (aunque se esté ejecutando otro) y consume 0 unidades de tiempo (este tiempo lo consideramos dentro del TCP posterior).
- d. En RR al producirse el cambio de bloqueado a listo de un proceso mientras otro se estaba ejecutando no nos afecta y debemos terminar el tiempo de quantum.
- e. Las prioridades las definimos de 1 a 100 siendo los valores mas grandes de mayor prioridad.
- f. En Prioridades, y SRT debo expropiarle la CPU a un proceso si, apareció uno con mayor prioridad o con menor tiempo restante y por lo tanto guardo lo que me resta de la ráfaga del proceso que se estaba ejecutando para terminarla cuando le vuelva a tocar.
- g. La tanda de trabajos a procesar se cargará en un archivo que el simulador debe leer y será un txt donde cada línea (registro) define un proceso, y cada uno de los campos a saber, se separan por comas:
- Campos:
1. Nombre del proceso.
  2. Tiempo de arribo.
  3. Ráfagas de CPU para completarse.
  4. Duración de ráfagas de cpu.
  5. Duración de rafagas de I/O.
  6. Prioridad.
- h. Un proceso no computará estado de listo hasta que no haya cumplido su TIP (inicialmente no computa tiempo de listo).
- i. Recordar:
- i. Tiempo de Retorno de un proceso (**TRp**): es desde que arriba el proceso hasta que termina (después de su **TFP**, incluyendo éste).
  - ii. Tiempo de retorno normalizado (**TRn**)= Es el tiempo de Retorno del proceso dividido el tiempo efectivo de CPU que utilizó.
  - iii. Tiempo de retorno de la tanda (**TRt**)= desde que arriba el primer proceso hasta que se realiza el último **TFP** (incluyendo el tiempo de éste).
  - iv. Tiempo Medio de retorno de la tanda ( $TMR_t$ )= la suma de los tiempos de retorno de los procesos, dividido la cantidad de procesos.

**TRABAJO PRÁCTICO DE IMPLEMENTACION Nº 2**  
**TPI-02-AM – ADMINISTRACION DE MEMORIA:**  
**ALOCACION CONTIGUA DE MEMORIA – PARTICIONES DINAMICAS**

**- Objetivo:**

Se trata de programar un sistema que simule distintas estrategias de asignación de particiones dinámicas de memoria a una tanda de trabajos y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada estrategia.

**- Características del sistema a simular::**

Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe leer un archivo de texto que define una tanda de trabajos. Cada registro describe uno de los trabajos de la tanda mediante los siguientes datos:

- Nombre del proceso.
- instante de arribo.
- Duración total del trabajo (tiempo que debe permanecer en memoria principal).
- Cantidad de memoria requerida.

Completada la lectura del archivo aceptará el ingreso por teclado de los siguientes datos:

- Tamaño de la memoria física disponible para usuarios (excluye la utilizada por el sistema operativo).
- Estrategia de asignación de particiones. Contemplará al menos las siguientes: first-fit, best-fit, next-fit y worst-fit.
- Tiempo de selección de partición (incluye el recálculo de la tabla de particiones en caso de corresponder).
- Tiempo de carga promedio (media del tiempo que toma cargar de memoria secundaria a principal un programa).
- Tiempo de liberación de partición.

El simulador simulará la tanda hasta que se hayan completado la totalidad de los trabajos produciendo las siguientes salidas:

- Un archivo en el que se indiquen todos los eventos que se producen en el sistema a lo largo de la simulación y el tiempo en el que ocurren los mismos. Ejemplos de eventos: se selecciona una partición para el trabajo x, se carga el trabajo y, termina el trabajo z, etc. En el mismo archivo (o en uno asociado a éste por el tiempo en el que ocurre un evento), se guardará el estado de la tabla de particiones cada vez que se modifique la misma (al momento de generar nuevas particiones para cargar un trabajo o unificar particiones por terminar otro). Se deberá conservar, como mínimo, identificación de la partición, dirección de comienzo, tamaño y estado (libre/ocupada).

Al finalizar la simulación imprimirá y mostrará por pantalla –como mínimo– los siguientes indicadores:

- a) Para cada proceso: Tiempo de Retorno.
- b) Para la tanda de procesos: Tiempo de Retorno, Tiempo Medio de Retorno e Indice de Fragmentación Externa.

**- Otras condiciones:**

- a) Deberá probarlo con al menos cuatro tandas de trabajos que tengan características distintas cada una y comentar los resultados obtenidos con cada estrategia de selección en función de las características de las tandas.
- b) Resuelva utilizando el lenguaje de programación que resulte apropiado y que conozca.
- c) El trabajo es unipersonal.
- d) Además de probar el simulador en clase, deberá presentar el ejecutable o ambiente de ejecución y el código fuente en soporte digital o repositorio.
- e) El simulador deberá ejecutarse de manera intuitiva en cualquier sistema operativo y sin la necesidad de tener que instalar librerías, programas, etc.
- f) El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los puntos del mismo que tengan que ver con Administración de Memoria con especificación dinámica de particiones, otorgándose en el examen el máximo puntaje previsto para esos ítems.
- g) Se fijara una fecha límite para la entrega y muestra del trabajo sin excepción.
- h) Se acordara un archivo en formato JSON, para todos los trabajos, de modo que los resultados sean aproximados a los trabajos presentados.
- i) Se deberá presentar diagramas de Gantt, diagramas de clase, de flujo, etc. Que permita su rápida comprensión e interpretación del trabajo entregado. En el caso del diagrama de Gantt, deberá coincidir con los resultados en pantalla del simulador.

## TRABAJO PRÁCTICO DE IMPLEMENTACION Nº 3

### TPI-03-SD – SCHEDULING DE DISCO (MAGNETICO)

#### - Objetivos:

Se trata de programar un sistema que simule distintos algoritmos de scheduling de peticiones a un disco magnético y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada uno de ellos.

#### - Características del sistema a simular:

Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe permitir armar y mantener hasta dos cadenas de peticiones a disco determinadas por un **nro. de petición** más el **número de pista** donde se encuentra el bloque requerido. Cada cadena de peticiones a disco debe aceptar por lo menos 15 requerimientos de **bloques** de disco y la simulación ocurre solamente para una de las cadenas, la que se seleccione antes de correr la simulación.

Además, el programa debe requerir el ingreso de los siguientes parámetros del disco rígido:

- a) **ST<sub>M</sub>**: Seek time medio.
- b) **VR**: Velocidad de rotación del disco dada en rpm.
- c) **TT<sub>1S</sub>**: Tiempo de transferencia de 1 sector.
- d) **TB**: Tamaño de bloque dado en sectores.
- e) **TP**: Total de pistas del disco.
- f) **PC**: Posición de las cabezas de lectura/escritura dada en nro. de pista (desde 0 a PT).
- g) **SC**: Sentido en el que está avanzando las cabezas (C = creciente, D = Decreciente), ésto sólo en el caso que PC <> 0, PC <> TP y la estrategia lo requiera.

Algoritmo a utilizar:

- **FIFO** (First In First Out)
- **SSTF** (Shortest Seek Time First)
- **SCAN**
- **C-SCAN**
- **LOOK**
- **C-LOOK**
- **F-SCAN**: Al elegir esta estrategia, el simulador debe requerir además, que se indique cómo se divide la lista.
- **N-STEP-SCAN**: Al elegir esta estrategia, el simulador debe requerir además, que se indique el parámetro adicional de longitud de lista **N**.

Una vez ingresados todos los datos y parámetros, un botón debe permitir iniciar la simulación. El simulador ejecutará la estrategia de scheduling sobre la cadena de peticiones de bloques hasta que se haya completado la totalidad de los requerimientos, produciendo como salida el siguiente detalle:

- La propia cadena de peticiones que se simuló.
- La estrategia utilizada por el scheduler de disco para simular.
- Si la estrategia elegida para simular fuera F-SCAN o N-STEP-SCAN se debe detallar cómo quedaron armadas las sublistas.
- Detalle del orden en el que se atendieron las peticiones, en el que se especifique el orden, nro. de petición, pista o cilindro requerido y distancia parcial recorrida para satisfacer ese requerimiento.
- La  $d_t$  (distancia total) que arroja el algoritmo.
- El  $rr$  (retardo rotacional) calculado en base a **VR**.
- El  $tt_{1B}$  (tiempo de transferencia de 1 bloque) calculado en base al **TT<sub>1S</sub>** y el **TB**.
- El tiempo de acceso total dado por la fórmula general:  $t_{aT} = st + rr + tt$ . No solo se debe indicar el resultado de  $t_{aT}$ , sino que se debe poder ver cuántos STM y rr se han considerado en este cálculo.

Con el fin de estudiar, comparar estrategias y validar el simulador, Ud. debe proveer una lista de peticiones de bloques de disco para cada estrategia, tal que ésta sea mejor que las otras en términos de  $t_{aT}$ , elaborando una conclusión sobre el tema.

- **Asuma que:**

- El disco magnético en cuestión posee un mecanismo de cabezas solidarias (todas se mueven juntas) o bien hay una única cara de un único plato. Esto además implica que no hay tiempo de selección de cabezas a considerar.
- El dispositivo está siempre listo; no hay que considerar este delay antes de procesar una lista de requerimientos.

- **Otras consideraciones:**

- a) Resuelva utilizando el lenguaje que sea apropiado y conozca.
- b) El trabajo es unipersonal.
- c) Además de probar el simulador en clase, deberá presentar el ejecutable o ambiente de ejecución y el código fuente en soporte digital o repositorio.
- d) El simulador deberá ejecutarse de manera intuitiva en cualquier sistema operativo y sin la necesidad de tener que instalar librerías, programas, etc.

- e) El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los puntos del mismo que tengan que ver con Scheduling de Disco, otorgándose en el examen el máximo puntaje previsto para ese/os ítems.
- f) Se fijara una fecha limite para la entrega y muestra del trabajo sin excepción.
- g) Se acordara un archivo o una cadena de peticiones común a todos los trabajos, de modo que todos tengas similitudes en sus resultados.
- h) Se deberá presentar diagramas de clase, de flujo, etc. Que permita su rápida comprensión e interpretación del trabajo entregado.