8.- La diferencia entre un gusano y un virus es:

a) *que el gusano es un programa en si mismo y el virus es parte del código de un programa.*
b) que el gusano es parte del código de un programa, y el virus es un programa en si mismo
c) que el gusano realiza acciones dentro de un programa y el virus es externo al programa
d) que el gusano es parte del sistema operativo y el virus es parte de los programas de usuario.

2.- (3puntos). Deducir las expresiones y calcular el tiempo que es necesario para leer 5 bloques consecutivos de un archivo en un sistema con:

a) Asignación contigua.
b) Asignación mediante listas enlazadas.
c) Asignación mediante indexación.

Considerar que el tiempo de búsqueda es tb=15 ms, el retardo rotacional tr=10 ms, y el tiempo de transferencia de los datos de un bloque tt=1.2 ms

Solución:

a) asignación contigua: t=tb+tr+N*tt = 15 + 10 + 5*1.2
b) asignación mediante listas enlazadas: t=N*(tb+tr+tt) = 5*(15+10+1.2)
c) asignación mediante indexación: t= (N+1)*(tb+tr+tt) = 6*(15+10+1.2)

ver también la explicación del problema 5-8

8.- Un caballo de Troya es

a) *un programa o procedimiento útil o de apariencia útil que contiene un código que realiza una función dañina o no deseada.*
b) es parte del código de un programa que realiza una operación dañina.
c) es un caso especial de virus informático que se transmite al copiarlo en dispositivos de almacenamiento.
d) es un programa utilizado para detectar falsos ataques informáticos.

10.- indique si las siguientes afirmaciones son correctas:

I. Los directorios son, básicamente, tablas simbólicas de archivos.
II. Los archivos son estructuras de datos que contienen la información de los directorios.

a) I: sí, II: sí *b) I: sí, II: no* c) I: no, II: sí d) I: no, II: no

7.- En un sistema de archivos tipo UNIX, un nodo-i es:

a) Un nodo de un esquema en la representación de datos del sistema.
b) Un tipo determinado de archivo del sistema operativo, para la gestión de los procesos en ejecución.
c) *Una estructura de control que contiene información clave de un archivo necesaria para el sistema operativo.*
d) Un registro de memoria principal para el acceso al sistema de almacenamiento de información.

9.- Suponiendo que el directorio actual es /usr/soi, indicar cual de los siguientes nombres de ruta para el archivo /usr/soi/docum no es correcto

a) docum *b) ../docum* c)../soi/docum d) ./docum

7.- Si encontrándonos en el directorio /usr/man queremos copiar el archivo examen desde este directorio al directorio /usr/ara, indicar cuál de las siguientes opciones no es correcta:

a) cp examen /usr/ara/examen *b) cp /usr/man/examen examen*
c) cp examen ../ara/examen d) cp /usr/man/examen /usr/ara/examen

7.- Una lista de acceso es:

a) Una lista ordenada que enlaza a cada uno de los archivos del sistema.
b) Una lista ordenada con todos los dominios que pueden tener acceso a un determinado objeto (programa, archivo, etc) y la forma de acceso.
c) Una lista con los punteros que dan acceso a los archivos almacenados en el disco.
d) Una lista con los accesos y los dominios de los sistemas que están conectados.

9.- Un programa que oculta parte de su funcionalidad destinada a obtener datos o derechos de acceso del usuario es:

a) Un virus. b) Un gusano. **c) Un caballo de Troya**. d) El talón de Aquiles.

2.- (3 puntos)

Se ha diseñado un sistema operativo parecido a UNIX con una estructura formada por bloques de 4 Kb, como el mostrado en la figura:

1 bloque de arranque	N bloques de mapa de bloques	M bloques de mapa de nodos-i	R bloques de nodos-i	S bloques de archivos

Un nodo-i ocupa 32 bytes, el tamaño máximo de un fichero es de 65536 bloques:

a) Calcular el número máximo de archivos que puede haber almacenados en este sistema si M=1.
b) Calcular el valor mínimo de las constantes N, R y S para que puedan existir 128 archivos de tamaño máximo.

Solución:

a) El número de ficheros está restringido por el número de nodos-i disponibles. Este número viene determinado por los M=1 bloques de mapa de nodos-i, cuyos bits son utilizados para indicar que nodos-i están libres y cuales ocupados. Así, el número posible de nodos-i y, por tanto, de ficheros está determinado por el número de bits de este mapa:

   4 Kb/bloque * 1024 bytes/Kbytes * 8 bits/byte = 32768 bits
   por lo que se pueden tener 32768 ficheros.

b) Los valores de las constantes serán:

   a. R bloques para los nodos-i

      El sistema está preparado para alojar los 32768 ficheros calculados. Por lo que tiene espacio para 32768 nodos-i, aunque solo se tengan 128 archivos. El número de nodos-i por bloque será:
      (4*1024bytes)/32 = 128 nodos-i/bloque
      El número de bloques R será:
      R = 32768 / 128 = 256 bloques

   b. S bloques para ficheros

      Ignorando los bloques de indirección, todos los bloques serán de datos:

      S = 65536 * 128 = 8388608 bloques

   c. N bloques de mapa de bloques

      Para manejar esta cantidad de bloques, hacen falta otros tantos bits. En cada bloque caben 32768 bits, por tanto se necesitan:

      N = 8388608/32768 = 256 bloques.

7.- El tiempo necesario para leer N bloques consecutivos de un archivo en un sistema con asignación mediante listas enlazadas es (t_b tiempo de búsqueda, t_r tiempo rotacional, t_t tiempo de transferencia):

a) $t_{E/S} = t_b + t_r + N * t_t$
b) $t_{E/S} = N * (t_b + t_r + t_t)$
c) $t_{E/S} = (N+1) * (t_b + t_r + t_t)$
d) $t_{E/S} = t_b + t_r + t_t$

2.- (3 puntos) En un sistema operativo se utiliza una estructura de nodos-i similar a la de Unix. Los bloques son de 2048 bytes. Las entradas en los nodos-i dedican 64 bits al tamaño del archivo y 32 bits a los punteros de los bloques.
El nodo-i tiene 8 entradas de direccionamiento directo, una de direccionamiento indirecto simple, y otra de direccionamiento indirecto doble. La entrada de archivos abiertos tiene una entrada para cada archivo con un campo de 64 bits que indica el desplazamiento.
Calcular el tamaño máximo de un archivo que utiliza todo el disco.

Solución:

Hay que considerar todos los parámetros que pueden limitar dicho tamaño y buscar el más restrictivo.

1) Considerar la estructura del sistema de archivos, el número máximo de bloques asignado a un archivo en su nodo-i (bloques).

Como el tamaño de un bloque es de 2.048 bytes, y los punteros son de 32 bits=4 bytes, entonces, el número de punteros a bloques que caben en un bloque es de 2048/4 = 512 punteros. Por lo tanto tendremos:

Direccionamiento directo: 8 bloques
Indirecto simple: 512 bloques
Indirecto doble: 512*512 bloques = 262144

Total= 262664 bloques

2) Considerando el tamaño de un puntero. Como el tamaño de un puntero es de 32 bits, el máximo número de bloques que se puede referenciar con un puntero es de 2^{32}

3) Teniendo en cuenta el tamaño del archivo en el nodo-i, que es de 64 bits. El tamaño máximo considerando sólo esta limitación sería de 2^{64} bytes, en bloques sería $2^{64}/2048 = 2^{64}/2^{11} = 2^{53}$

4) Teniendo en cuenta el campo desplazamiento en la tabla de archivos abiertos, que es de 64 bits.
Luego sería también de 2^{64} bytes

La solución será por tanto la mas restrictiva que corresponde a la opción 1, por la estructura de archivos.

3.- (2 puntos) Suponer un sistema de archivos, parecido al de Unix, cuyos bloques son de tamaño de 1 Kb y los punteros a bloques son de 4 bytes. Se tienen 10 punteros a bloques directos de datos, un puntero a un bloque indirecto simple y uno a bloque indirecto doble. Se quiere incrementar el tamaño máximo del fichero. Cuál de las siguientes acciones permitiría un mayor aumento: Añadir un bloque de triple indirección o incrementar el tamaño del bloque a 4 Kb.

Solución:

El tamaño máximo de un fichero viene determinado por el número de bloques de datos posibles que permita el sistema. Entonces, tal y como es el sistema de archivos actual, el número de bloques máximo que puede tener un archivo es:
Directo----------10
Indirecto simple----------2^8
Indirecto doble----------2^{16}
Total en bytes $(2^{10} \times (10 + 2^8 + 2^{16}))$
Se ha tenido en cuenta para el computo de los bloques que en el uso del direccionamiento indirecto el número de punteros que caben en un bloque es $2^{10} / 4 = 2^8$ punteros.

a) Si se añade un bloque de triple indirección, el máximo tamaño es:

Directo----------10
Indirecto simple----------2^8
Indirecto doble----------2^{16}
Indirecto triple----------2^{24}
Total en bytes $(2^{10} \times (10 + 2^8 + 2^{16} + 2^{24})) = 2^{34} + 2^{26} + 2^{18} + ...$

b) Si se incrementa el tamaño del bloque a 4 Kb, se debe tener en cuenta que varía en número de punteros en los bloques de indirección, siendo ahora, $2^{12} / 4 = 2^{10}$ punteros y, por lo tanto, el tamaño máximo quedará:

Directo----------10
Indirecto simple----------2^{10}
Indirecto doble----------2^{20}
Total en bytes $(2^{12} \times (10 + 2^{10} + 2^{20})) = 2^{32} + 2^{22} + ...$
Es mayor en el primer caso.

2.- (3 puntos) En la figura se representan los 15 primeros bloques de un dispositivo de un disco que en total dispone de 30 Mb de capacidad. La asignación de espacio en disco se realiza mediante listas enlazadas. Los bloques tienen un tamaño de 512 bytes.

a) Calcular para cada bloque, cuántos bytes se podrán asignar a datos y cuántos a punteros a otros bloques.

b) Calcular el tamaño máximo, en bytes, de los datos almacenados en el archivo *Notas*, que también se representa en la figura.

c) Indicar el problema que presenta el acceso directo a un bloque en los archivos de este sistema.

d) Si se utilizara un método mediante indexación, ¿se resolvería el problema indicado anteriormente para las listas enlazadas?

e) Para el método mediante indexación, calcular el tamaño máximo de los datos que se pueden ahora almacenar en el archivo notas. ¿Hay alguna variación con respecto al caso anterior?

f) ¿Existe pérdida de espacio?¿Cuánto?

0	1	2	3 14	4
5 3	6	7 12	8	9
10	11	12 5	13	14 -1

Directorio
Notas
Bloque de comienzo: 7
Bloque Final: 14

a) El disco contiene un total de 30*1024*1024 bytes= 31.457.280 bytes
Como los bloques son de 512 bytes, el número total de bloques en el disco es de: 31.457.280 /512 = 61440, para poder direccional todos estos bloques se necesitan por lo menos 16 bits (2^{16} = 65.536), es decir 2 bytes. Luego habrá en cada bloque 2 bytes para punteros, y 510 para datos.

b) El archivo *Notas* ocupa los bloques 7, 12, 5, 3 y 14, cinco bloques. Como cada bloque usa 510 bytes para datos, en total se tendrá 5*510 = 2550 bytes.

c) El problema que presenta es que hay que leer todos los bloques para llegar a uno determinado. Así, para leer el bloque k deben leerse previamente los k-1 para ir accediendo a los punteros que apuntan al siguiente bloque.

d) Sí. En este método, el directorio contiene la dirección del bloque donde están los índices a los bloques de datos del archivo.

e) En este tipo de acceso, todos los bytes de un bloque se utilizan para datos, con lo cuál para el archivo notas se tendrán 5*512 = 2560 bytes. Con respecto al anterior método se utilizan dos bytes mas por cada bloque para datos.

f) La asignación mediante indexación presenta sin embargo pérdida de espacio. Si en la tabla de índices se le asigna un bloque entero, como los índices son de 2 bytes, el bloque está ocupado por 5 índices x 2 bytes = 10 bytes. Por lo que en el bloque está desaprovechado : 512 – 10 = 502 bytes para este fichero en concreto.

9.- La asignación del espacio del disco mediante el método de *asignación continua* tiene como inconveniente que

a) No es realizable salvo que se conozca el tamaño máximo del archivo en el momento de su creación.
b) Tiene un rendimiento pobre si se quiere leer un archivo completo.
c) Las dos anteriores son verdaderas.
d) Ninguna de las anteriores

Solución: pp.314 del libro base de la asignatura
La asignación del espacio del disco mediante el *método de asignación continua* consiste en asignar a cada archivo un conjunto de direcciones contiguas en el disco. Aunque este método es fácil de entender y tiene un buen rendimiento cuando se quiere leer un archivo completo, presenta también grandes inconvenientes. Uno de ellos es que no es realizable salvo que se conozca el tamaño máximo del archivo en el momento de su creación. Este problema es grave en muchas aplicaciones en las que los archivos pueden crecer dinámicamente y por lo tanto no se sabe el tamaño máximo en el momento de la creación.

Respuesta: **A) No es realizable salvo que se conozca el tamaño máximo del archivo en el momento de su creación.**

2.- (3 puntos) Considérese un sistema de ficheros cuyos bloques son de 1 Kbytes y cada puntero a un bloque de disco requiere 2 bytes. Un nodo-i de este sistema contiene 9 punteros directos a bloques de datos, un puntero a un bloque de indirección simple, y otro a uno doble. Se pide:

a) **(1.5 puntos)** Determinar los números de los bloques de datos a los que se puede acceder con los 11 punteros contenidos en un nodo-i.

b) **(1.5 puntos)** Supuesto que el sistema operativo ha leido ya el nodo-i de un fichero en memoria principal, desterminar el número de lecturas adicionales a disco que se requerirán para leer el bloque de datos número 325 y el número 605.

Solución:

a) De acuerdo con el enunciado un nodo-i de este sistema de ficheros contiene entre otras informaciones 11 punteros:

- 9 *punteros directos a bloques de datos*, que apuntarán a los bloques de datos nº 1 al nº 9 del fichero.
- 1 *puntero a un bloque de indirección simple*, es decir, cuyo contenido son punteros a bloques. Puesto que un bloque tiene una capacidad de S_B =1 Kbyte=2^{10} bytes y un puntero a un bloque de disco ocupa S_P= 2 bytes, entonces el número de punteros N_P que pueden almacenarse en un bloque sería:

$$N_P = \frac{S_B}{S_P} = \frac{2^{10}}{2} = 2^9 = 512 \text{ punteros.}$$

  En consecuencia un bloque de indirección simple apuntará a 512 bloques de datos del nº 10 al nº 521.
- 1 *puntero a un bloque de indirección doble*, es decir, cuyo contenido son punteros a bloques de indirección simple que a su vez contienen punteros a bloques de datos. Como el número de punteros que puede contener un bloque son N_P=512. Un bloque de indirección doble apuntará a N_P^2=512^2=262144 bloques de datos del nº 522 al nº 262665.

b) Supuesto que el nodo-i ya se encuentra en memoria principal, para leer un bloque referenciado con un puntero directo se requiere una lectura adicional a disco. Para leer un bloque referenciado con un puntero indirecto simple se requieren dos lecturas adicionales a disco. Para leer un bloque referenciado con un puntero indirecto doble se requieren tres lecturas adicionales a disco.

Así, al bloque de datos nº 325 de acuerdo con el apartado anterior se accede a través del puntero de indirección simple luego se requerirán dos lecturas adicionales a disco. Mientras que el bloque de datos nº 605 se accede a través del puntero de indirección doble luego se requerirán tres lecturas adicionales a disco.

3.- ¿Se puede producir un ataque mediante "caballos de Troya" en un sistema protegido mediante listas de capacidades?

a) Si.
b) No.
c) En algunos casos.
d) No es posible saberlo sin tener más datos.

Solución: pp. 351 del libro base de la asignatura.

Un "caballo de Troya" es un programa útil o de apariencia útil que contiene un código oculto que, cuando se invoca, lleva a cabo una función dañina o no deseada.

En las listas de capacidades a cada dominio se le asocia una lista de objetos a los cuales puede terner acceso, junto con una indicación de las operaciones permitidas sobre cada objeto. La ventaja de la lista de capacidades es que es un objeto protegido, mantenido por el sistema operativo y de forma que nunca se permite que una capacidad se mueva al espacio de direcciones accesibles por un proceso de usuario. Manteniendo las capacidades seguras, los objetos a los que protegen también están seguros frente accesos no autorizados. Por lo tanto, al ser el "caballo de Troya" un programa de usuario **no** puede tener acceso a la lista de capacidades y modificar los accesos.

Respuesta: **B) No.**