

Son 10 preguntas en total, todas tienen la misma puntuación.

✓ En un filesystem tipo Unix, el tamaño de los bloques viene dado exclusivamente por el hardware donde se almacenarán los datos. *10/10

☐ Verdadero

☒ Falso



✓ Los inodos en un filesystem tipo Unix *

10/10

- ☐ Contienen el nombre del archivo/directorio
- ☒ Contienen referencia a los bloques de datos del archivo/directorio ✓
- ☒ Contienen metadata del archivo/directorio ✓
- ☐ Mantienen una relación 1:1 entre archivo/directorio e i-nodo

Respuesta correcta

- ☒ Contienen referencia a los bloques de datos del archivo/directorio
- ☒ Contienen metadata del archivo/directorio
- ☒ Mantienen una relación 1:1 entre archivo/directorio e i-nodo

✓ ¿Dónde se encuentra, en un filesystem de tipo Unix, la referencia al inodo raíz?

*10/10

- ☐ Está definido por el hardware, dependiendo entonces del modelo de disco
- ☒ Lo define el sistema de archivos, dentro del superbloque ✓
- ☐ Lo define el sistema de archivos, está siempre dentro del primer bloque de datos
- ☐ Ninguna de las anteriores



✓ En un filesystem tipo Unix, ¿qué consecuencia tiene incrementar la cantidad de bloques dedicados a inodos? *10/10

Solamente se aumenta la cantidad de bloques dedicados a inodos, **sin modificar** la cantidad **total** de bloques del sistema de archivos.

- ☐ Al haber más inodos, el límite de archivos por directorio se incrementa de forma proporcional
- ☐ Permite soportar más archivos, que a su vez pueden ser más grandes en promedio
- ☒ Permite soportar más archivos, aunque el tamaño promedio de los archivos podría ser menor ✓
- ☐ Siempre representa una pérdida de espacio en disco: cuanto menos inodos, mejor

✓ La cache de bloque de disco tiene como finalidad la de reducir la cantidad de operaciones de I/O para la lectura de archivos *10/10

- ☒ Verdadero ✓
- ☐ Falso

✓ Los bloques indirectos en un inodo: ¿Qué beneficio/s traen? * 10/10

- ☒ Cada nivel de indirección hace crecer el tamaño final del archivo exponencialmente ✓
- ☐ Las referencias hacen al filesystem independiente de la arquitectura
- ☒ Mantiene el tamaño del inodo constante, sin importar el tamaño de los archivos ✓
- ☐ Hacen que los accesos a disco sean más rápidos



✗ Dado un filesystem de tipo FUSE que soporta lecto-escritura de archivos/directorios ...

*0/10

Similar a lo que podría ser "tmpfs"

- ☒ El kernel no interviene de ninguna forma, dado que el filesystem se implementa exclusivamente en modo usuario ✗
- ☐ Es necesario implementar la totalidad de las operaciones disponibles
- ☐ Es estrictamente necesario un disco físico donde persistir los datos
- ☒ El programa de usuario del filesystem debe compilarse con librerías especiales ✓

Respuesta correcta

- ☒ El programa de usuario del filesystem debe compilarse con librerías especiales

✗ Sobre Linux Virtual File System (VFS): *

0/10

- ☒ Provee una interfaz genérica para todo tipo de filesystem ✓
- ☒ Define los parámetros de cuántos inodos y bloques de datos puede tener un filesystem tipo Unix ✗
- ☐ Sólo soporta filesystems en discos físicos
- ☐ Se maneja en modo usuario

Respuesta correcta

- ☒ Provee una interfaz genérica para todo tipo de filesystem



✓ En un filesystem de tipo Unix, los directorios... *

10/10

- ☐ Solo pueden tener un único bloque de datos
- ☒ Sus bloques de datos son una lista de entradas compuestas por: nombre de archivo - número de inodo ✓
- ☐ Sus bloques de datos son una lista de entradas compuestas por: nombre de archivo - metadata del archivo - número de inodo
- ☐ Sus bloques de datos son una lista de entradas compuestas por: número de inodo

✗ El mecanismo de interacción entre el Kernel y FUSE es: *

0/10

Nota: el proceso que implementa el filesystem en FUSE se conoce también como daemon de FUSE

- ☐ El kernel siempre recibe todas las peticiones de syscalls, y las reenvía al daemon de FUSE para que las maneje acordemente
- ☐ El daemon de FUSE recibe directamente las peticiones del usuario del filesystem y las maneja acordemente sin necesidad del kernel
- ☐ El daemon de FUSE intercepta las peticiones y se las reenvía al kernel para que éste las maneje
- ☒ Ninguna de las anteriores ✗

Respuesta correcta

- ☒ El kernel siempre recibe todas las peticiones de syscalls, y las reenvía al daemon de FUSE para que las maneje acordemente

Este formulario se creó en Facultad de Ingeniería - Universidad de Buenos Aires. [Denunciar abuso](#)

Google Formularios



