

## EXAMEN SISTEMAS OPERATIVOS AVANZADOS (TEORIA)

10 ENERO 2020

NOMBRE:

- 1) ¿Cuándo y por qué se debe ejecutar la siguiente función en el driver IDE de xv6?

```
static int ide_wait_ready() {  
    while (((int) r = inb(0x1f7)) & IDE_BSY) ||  
        !(r & IDE_DRDY))  
        ; // loop until drive isn't busy  
}
```

- 2) Se dispone de un disco con interfaz SATA3 ((**600 MB/s**). La velocidad de rotación es 7200RPM, tiene 20 *tracks*. El tiempo requerido para iniciar el movimiento de la cabeza (hasta alcanzar *coasting*) es **0.15 ms**, para moverla entre dos tracks adyacentes es **0.1 ms** y para depositarla en la pista destino **0.5 ms**.
- ¿Cuál el tiempo de búsqueda promedio en el disco?
  - ¿Cuál es la tasa de I/O que se obtiene en acceso aleatorio o *random*, si en acceso secuencial con **100 MB** de datos es de **150 MB/s**?
  - ¿Cuál sería el peor y mejor tiempo de acceso si tuviésemos que escribir y leer 2 bloques de **4KB**, siendo los sectores en este disco de **512bytes**? (suponiendo que la cabeza lectora está en la pista más exterior)
- 3) Suponiendo que el disco anterior está usando un VSFS
- ¿Cuántos accesos serían necesarios para crear un fichero con **2** bloques de datos en el directorio `"/`?
  - ¿Cuántos accesos serían necesarios para crear un fichero con **2** bloques de datos en el directorio `"/foo/bar/"`?
  - ¿Cuánto sería el tiempo de peor y mejor caso para realizar una operación del caso a), suponiendo el disco descrito en 2)?
- 4) Suponiendo que estamos usando un FFS, con 5 grupos de cilindros (i.e. **4 cilindros** por grupo) en el disco de 2):
- ¿En qué tamaño hemos de partir los ficheros grandes para obtener un 90% de la tasa de acceso obtenido en VSFS en ese tipo de ficheros?
  - ¿Cómo mejoraría la operación en el peor caso de la operación (3) (suponer que los datos y metadatos del fichero están dentro del mismo grupo de cilindros)?
- 5) Suponiendo que al sistema anterior le agregamos *Journaling*, ¿Cuál es el impacto en el tiempo requerido para hacer en el mejor caso de la operación (3.a) si pasamos a emplear *Journaling* de **metadatos** en lugar de *Journaling* de **datos**? ¿Si empleamos metadatos desordenados?

## EXAMEN SISTEMAS OPERATIVOS AVANZADOS LABORATORIO

JUEVES 10 ENERO 2020

Sobre la base de tú implementación de las prácticas 4 y 5 de la asignatura, crear un nuevo proyecto llamado Examen3 en gitlab.com que incluya como *developer* a [vpuente@gmail.com](mailto:vpuente@gmail.com) y en dos *commits* independientes (sobre el trabajo inicial ya entregado) que extiendan su implementación, incorporando las siguientes funcionalidades:

1. (6pt) Los *locks* de la librería deben evitar el *spinning*. Es decir, si el *lock* se encuentra tomado, el proceso deberá suspenderse hasta que no este libre. **[pista]** Ten en cuenta que es posible que esta funcionalidad puede no ser posible en el espacio de. *Git commit hash* \_\_\_\_\_
2. (4pt) Como ocurre en el *sub-blocking* de FFS, cuando el volumen de información que se pretende escribir en un fichero pequeño, supera el valor límite, su contenido debe migrarse automáticamente a un fichero normal. **[pista]** Ten en cuenta que es posible que esta funcionalidad requiera cerrar/eliminar el fichero antiguo y abrir/crear uno nuevo, reusando el mismo descriptor. *Git commit hash*: \_\_\_\_\_

*En ambos casos es necesario escribir un programa de prueba que verifique la funcionalidad.*