

# Módulo 2

## Sistema de ficheros

### Buffering

Ampliación de Sistemas Operativos. Curso 2017-2018

#### Cuestiones

**/dev/zero** . ¿Qué funcionalidad tiene el fichero de dispositivo **/dev/zero**?  
¿Qué ocurre cuando leemos de él? ¿Y si escribimos?

**Copia con dd** . Consulta el manual de la orden **dd**. A continuación, ejecuta la orden

```
BS=128
dd if=/dev/zero of=salida bs=$(BS) count= $(( 16*1024*1024 / BS ))
```

¿Qué hace esa orden?

**Más con dd** . Prueba nuevamente la orden anterior pero dando valores **BS=** 1, 128, 512, 1024, 2048, 2200, 4096, 5000. Anota el tiempo de ejecución de cada prueba y trata de explicar las diferencias en el tiempo de ejecución.

#### E/S síncrona

**copia.c** Considera nuevamente el código de copia (**rw.c**) implementado en la hoja anterior. Realiza la apertura del fichero de salida con el flag **O\_SYNC** y compara los tiempos con los observados anteriormente usando distintos tamaños de lectura/escritura. ¿Y si se usa el flag en el fichero de entrada?

#### Mapeo de ficheros en memoria

**mmap.c** Escribe un programa llamada **mmap.c** que mapee los 100 primeros bytes de un fichero de texto existente (puedes crear uno copiando cualquier código fuente en un nuevo fichero) a memoria. Si el fichero fuese menor de 100 bytes, se mapeará el fichero completo. Posteriormente, se pasará a mayúsculas todo el texto del fichero.

## E/S biblioteca

**fcopia.c** Una vez más, crea una nueva versión de la copia de ficheros, pero usando las llamadas de la librería estándar (**fopen**, **fread**...). Vuelve a realizar diversas pruebas con diferentes tamaños de lectura/escritura midiendo los tiempos. ¿Las diferencias de tiempo son igual de notables cuando se usan tamaños de lectura/escritura de unos pocos bytes? Repite las pruebas incluyendo una llamada a **fflush(NULL)** tras cada escritura. ¿Se producen diferencias de tiempo?

- **OPCIONAL.** Utiliza **strace** con tu aplicación de copia (sin **fflush**). Prueba con tamaños de lectura/escritura de 1 byte y 512 bytes. ¿Cuántas veces se invoca a las llamadas al sistema **read** y **write** en cada caso?. ¿Cómo explicas las diferencias de tiempo de ejecución observadas?

**vbuf.c** Estudia, compila y ejecuta el código del fichero **vbuf.c**. A continuación, usa **setvbuf** para probar el efecto de las 3 estrategias de *buffering* disponibles en la librería estándar: sin *buffer*, *buffer* de línea y *buffer* completo. ¿En qué se diferencia el comportamiento de cada una y por qué? ¿Cuál es la estrategia por defecto cuando escribimos en el terminal?

**copyBuf.c** Estudia, compila y ejecuta el código del fichero **copyBuf.c** que copia un fichero byte a byte usando la librería estándar. Compara el tiempo de ejecución (usando **time**) con el código de **fcopia.c** con tamaño de copia de 1 byte. ¿Hay diferencias de tiempo? ¿Por qué?

## Entrega

La entrega consistirá en un solo fichero comprimido (**.tgz**) que contendrá:

- Todos los ficheros fuentes completos (NO debe incluirse ningún fichero objeto o ELF).
- El fichero *makefile*.
- Un fichero de texto o PDF con las respuestas a las preguntas planteadas en cada ejercicio, así como una copia de las fuentes de cada apartado completado.