

## SISTEMAS OPERATIVOS I

### Práctica 6: Proyección de archivos en memoria

#### Objetivo

Uso de la llamada al sistema que permite proyectar un archivo en memoria y la utilización de dicha proyección **como área de memoria compartida entre procesos**.

#### Información útil para la práctica

La llamada al sistema *mmap* permite proyectar un archivo en memoria de forma que se puede acceder al archivo desde un programa como si fuese un array, y por lo tanto sin necesidad de utilizar operaciones de entrada/salida. De esta manera, el contenido del fichero se asocia a una zona de la memoria virtual del proceso. Nótese que el contenido del fichero y el de la zona de la memoria no tienen por qué ser coherentes en todo momento.

Para evitar errores en el código, ten en cuenta que los punteros que no se inicien en la declaración de variables deben tener un valor inicial NULL. En los códigos añade chequeos para comprobar que las llamadas al sistema tienen éxito.

La parte central de la práctica es la proyección de archivos en memoria y manipulación de ésta proyección por parte de los procesos como un espacio compartido de sus memorias virtuales.

Conviene que leas detenidamente el manual de las funciones que se deben utilizar: *open*, *close*, *fstat*, *mmap*, *munmap*, *msync*, *truncate* o *ftruncate*.

#### Enunciado

1. Haz un programa que realice las siguientes acciones:

- Abre un archivo para la lectura con la función *open*, averigua su longitud con *fstat* e imprime su contenido por pantalla carácter a carácter utilizando un lazo con tantas iteraciones como la longitud del archivo. El nombre del archivo debes pasarlo como argumento del programa principal.
- Proyecta el archivo en memoria con *mmap*. Deja que el sistema operativo elija la dirección de memoria. Establece accesos de sólo lectura, área de memoria privada y un desplazamiento desde el inicio del archivo (*offset*) de 0.
- Cierra el archivo e imprime el contenido de la proyección en memoria carácter a carácter.
- Cierra la proyección con la función *munmap*.

2. Haz un programa que proyecte un archivo en memoria con permisos de lectura y escritura y área de memoria compartida. Modifica el valor de alguno de los caracteres de la proyección del archivo. Comprueba si las modificaciones que se hacen en memoria se reflejan en el archivo. No olvides cerrar la proyección al final.

3. Compara los mapas de memoria del proceso antes y después de hacer la proyección del archivo en memoria para determinar la zona del mapa de memoria donde se proyecta el archivo. Estudia las diferencias en el mapa de memoria del proceso entre establecer permisos de solo lectura o de lectura y escritura para la proyección, y entre definir el área de memoria como privada o compartida.
4. Las modificaciones puede que no se escriban en el archivo hasta que se ejecute la función *munmap*. Para conseguir una actualización inmediata del archivo debes utilizar *msync*. Comprueba su funcionamiento.
5. **ENTREGABLE.** Escribe un programa en C que modifique cualquier carácter de un archivo que se corresponda con una letra minúscula (de la *a* a la *z*) por la correspondiente letra mayúscula (de la *A* a la *Z*), y que sustituya cualquier carácter correspondiente a un número por el número equivalente de asteriscos (por ejemplo, un 4 por *\*\*\*\**). Finalmente, se escribirá en una nueva línea al final del archivo el recuento de asteriscos del archivo. El programa debe funcionar de la siguiente manera:
  - El programa recibe como argumento un nombre de archivo de entrada en texto plano y un nombre **diferente** de archivo de salida.
  - No modifiques la proyección de memoria del archivo directamente. Construye un string temporal con los valores adecuados usando un buffer *char\**. Una vez que el string esté completo cópialo en la proyección de memoria usando *memcpy*.
  - Observa que la longitud final del archivo puede ser mayor que la longitud inicial, por lo que tienes que modificar los valores correctos de tamaño en las llamadas a *mmap*. Así mismo, debes modificar el tamaño del fichero de salida usando *truncate* o *ftruncate*.
  - Para cambiar los tamaños iniciales de las proyecciones, utiliza *mremap* o crea nuevas proyecciones con el tamaño adecuado.
  - El proceso principal creará un proceso hijo para trabajar de forma concurrente. El padre pasará las letras a mayúsculas y el proceso hijo cambiará los números por asteriscos. El padre irá dejando los huecos necesarios al encontrar un número para que el hijo pueda hacer su trabajo.
  - El proceso hijo debe esperar a que el padre haya procesado la mitad del archivo antes de empezar, y debe volver a esperar antes de continuar con la segunda mitad. No podrá continuar hasta que el padre termine de procesar todo el archivo.
  - Cuando padre e hijo hayan terminado la segunda mitad, el padre contará el número de asteriscos presentes en la proyección y añadirá una línea nueva al final indicando el recuento (por ejemplo: "Total asteriscos: 27").
  - Debes buscar un mecanismo de sincronización de modo que las operaciones se realicen en los momentos oportunos según el orden indicado. La sincronización debe ser mediante señales o chequeando los valores de la zona de memoria compartida.