

# Memory-mapped files

Vantaggi:

**trasparenza** gestire dati su disco come se fossero in memoria;

**zero copy I/O** non c'è bisogno di copiare i dati in un buffer kernel e poi copiarli nel buffer dell'utente, ma vengono letti direttamente nella pagina appropriata;

**IPC** condivisione istantanea delle modifiche fatte ai file;

**file grandi** il programma non deve preoccuparsi di limitare la quantità di dati memorizzati in ogni momento, viene gestito in automatico dal SO.

La chiamata a `mmap` inizializza una serie di pagine logiche, una per ogni blocco da 4 KB del file, impostando il bit di presenza a 0. Quando il processo accede ad una pagina mappata non ancora caricata,

- miss TLB → ricerca in hardware nella tabella delle pagine;
- l'hardware trova la pagina con  $P = 0$ , e genera un'eccezione (*page fault*);
- il kernel gestisce l'eccezione scegliendo una pagina fisica (libera o vittima) e scrivendo il contenuto del file (interrupt al termine della lettura da disco e copia manuale o direttamente tramite DMA);
- il kernel aggiorna la tabella delle pagine con l'indirizzo della pagina fisica allocata e  $P = 1$ ;
- riprende l'esecuzione del processo dall'istruzione che ha causato il fault, nuovo TLB miss ma stavolta si carica la nuova pagina nel TLB.

In alcuni casi si può ridurre il numero di page fault con prepagina.