

Esercitazione: File Mapping Concorrente con `mmap` e Thread

Tutorato di Sistemi Operativi

Giugno 2025

Esercizio 3a — Contare righe e parole con `mmap`

Scrivere un programma in C denominato `count-mmap.c` che utilizza `mmap` per contare il numero di righe e parole in un file. Il programma deve:

1. aprire il file in modalità lettura;
2. mappare il contenuto in memoria con `mmap`;
3. contare righe e parole;
4. stampare i risultati a video;
5. gestire correttamente eventuali errori e liberare le risorse con `munmap`.

Esercizio 3b — Concatenazione di due file con `mmap`

Scrivere un programma in C denominato `concat-mmap.c` che utilizza `mmap` per concatenare due file in un terzo file. Il programma deve:

1. aprire i due file sorgente in lettura e il file destinazione in scrittura;
2. mappare i file sorgente e il file destinazione;
3. scrivere i contenuti dei due file sorgente nel file destinazione usando `memcpy`;
4. gestire errori, chiudere i file e liberare le risorse.

Esercizio 3c — Ricerca carattere con `mmap`

Scrivere un programma in C denominato `search-mmap.c` che utilizza `mmap` per cercare tutte le occorrenze di un carattere specificato in un file. Il programma deve:

1. aprire il file in lettura;
2. mappare il file in memoria;
3. contare e stampare tutte le posizioni in cui appare il carattere cercato;
4. gestire errori, chiudere il file e liberare le risorse.

Esercizio 3d — Inversione concorrente con `thread`

Scrivere un programma in C denominato `mmap-reverse-parallel.c` che mappa un file in memoria e ne inverte il contenuto in parallelo usando più thread. Ogni thread inverte una parte distinta del file, utilizzando `pthread.h` per la gestione concorrente. Il programma deve gestire la mappatura con permessi di lettura e scrittura condivisi (`MAP_SHARED`), smappare correttamente il file e chiuderlo.