

Per gestire la concorrenza nell'accesso ai dati nel file `dev_ufs` si possono prevedere vari scenari:

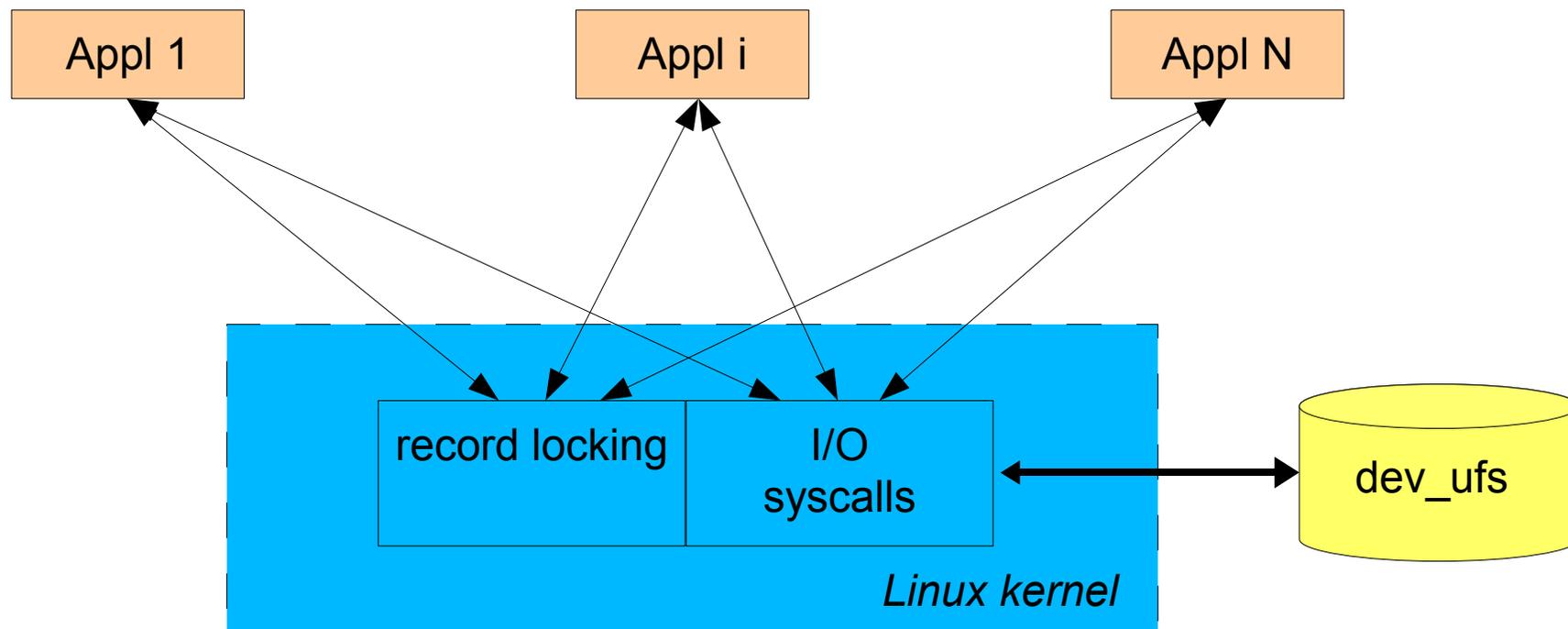
1) NO SERVER `ufsd`: le applicazioni utilizzano direttamente i file lock, implementati nel codice della libreria `libufs.a`, per gestire l'accesso a `dev_ufs`

2) UN SOLO SERVER `ufsd`: le applicazioni inviano mediante IPC (FIFO) le loro richieste all'unico server condiviso che esegue l'accesso a `dev_ufs`, senza la necessità di file lock

3) SERVER `ufsd` MULTIPLI: le applicazioni inviano mediante IPC (FIFO) le loro richieste ai rispettivi server, che utilizzato i file lock per gestire l'accesso a `dev_ufs`

Per il progetto/esonero 2 si richiede l'implementazione dello scenario 3.

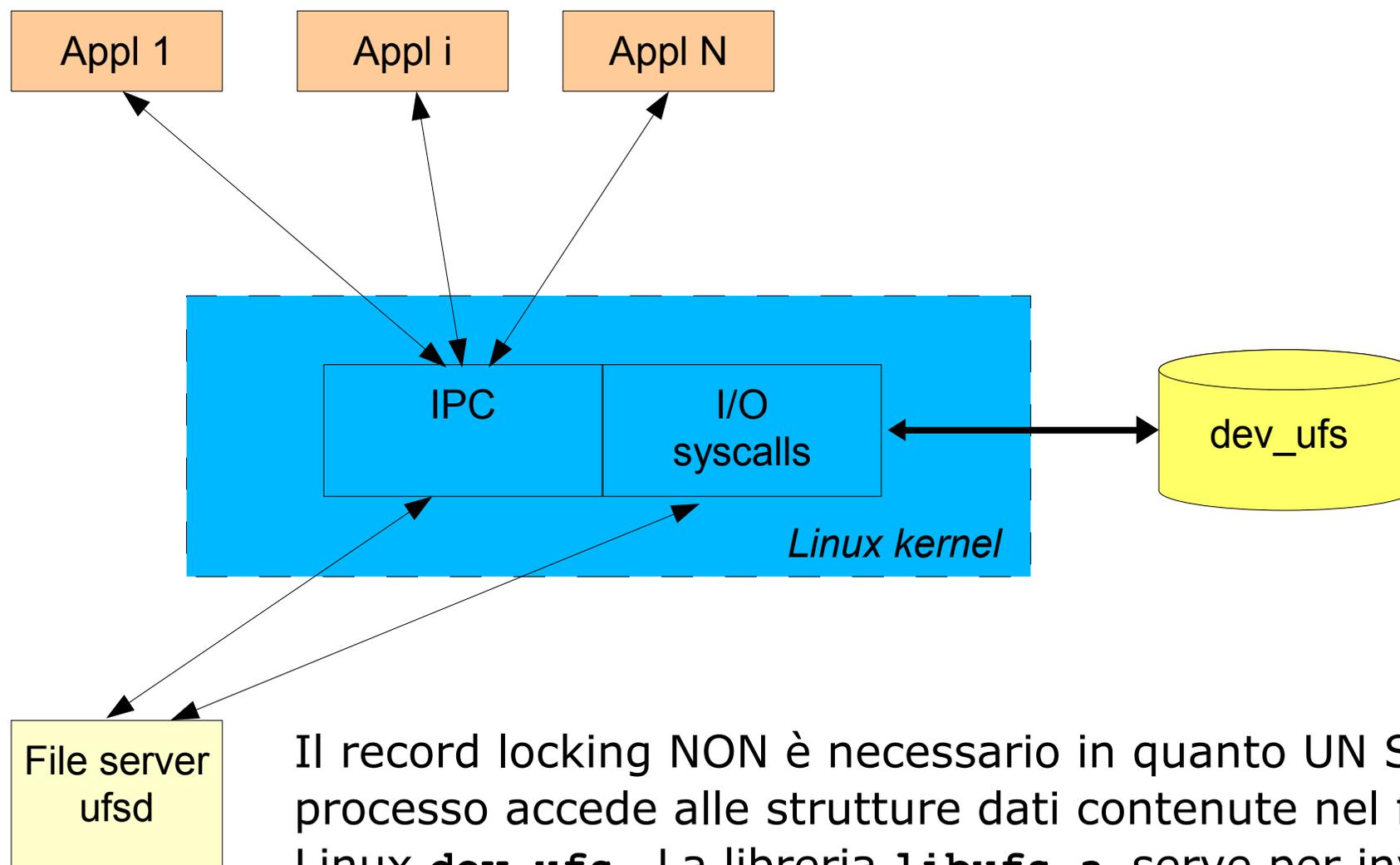
1) Gestione della concorrenza implementata nelle applicazioni stesse



Il record locking è necessario in quanto più processi contemporaneamente accedono alle strutture dati contenute nel file Linux `dev_ufs`.

La libreria `libufs.a` ha il compito di richiedere i lock e di fare l'accesso diretto al file Linux `dev_ufs`.

2) Gestione della concorrenza implementata mediante singolo server



Il record locking NON è necessario in quanto UN SOLO processo accede alle strutture dati contenute nel file Linux `dev_ufs`. La libreria `libufs.a` serve per inviare e ricevere messaggi via FIFO.

3) Gestione della concorrenza implementata mediante server multipli

Typical access pattern:

- (1) Appl sends I/O request via FIFO
- (2) Server receives I/O request via FIFO
- (3) Server obtains lock on dev_ufs data
- (4) Server calls I/O syscall
- (5) Kernel performs I/O access to dev_ufs
- (6) Kernel retrieves data from dev_ufs
- (7) Server gets rc/data of I/O syscall
- (8) Server releases lock on dev_ufs data
- (9) Server sends reply/data via FIFO
- (10) Appl receives reply/data via FIFO

