Session 21

File Locking in Linux (User-level Overview)

File locking is used in Linux to prevent conflicts when multiple processes access the same file simultaneously — ensuring data consistency and integrity.

Types of File Locks in Linux

Lock Type	Description	Shared Between Processes?
Advisory	Voluntary; processes must cooperate	✅ Yes
Mandatory	Enforced by the OS kernel	⚠ Rare; must be enabled manually

Linux default is advisory locking (most commonly used and safer for portability).

File Locking via fcntl()

The fcntl() system call provides a way to implement record-level file locking.

Key Structure: struct flock

```
struct flock {
   short l_type;    // F_RDLCK, F_WRLCK, F_UNLCK
   short l_whence;  // SEEK_SET, SEEK_CUR, SEEK_END
   off_t l_start;   // Byte offset, relative to l_whence
   off_t l_len;     // # bytes (0 = to EOF)
   pid_t l_pid;     // Process ID (for F_GETLK only)
};
```

Lock Types:

Flag	Meaning
F_RDLCK	Shared (read) lock
F_WRLCK	Exclusive (write) lock
F_UNLCK	Unlock

fcntl() Commands for Locking:

Command	Description
F_GETLK	Check for existing locks
F_SETLK	Set lock (non-blocking)
F_SETLKW	Set lock (blocking — wait if needed)

Example: File Locking using fcntl()

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
int main() {
   int fd = open("file.txt", O_WRONLY);
   if (fd < 0) {
      perror("open");
      return 1;
   }
   struct flock lock;
   lock.l_type = F_WRLCK;      // Exclusive write lock
   lock.l_whence = SEEK_SET;
   lock.l_start = 0;          // From start of file
   lock.l_len = 0;            // Lock the whole file
   printf("Trying to acquire lock...\n");
   if (fcntl(fd, F_SETLKW, &lock) == -1) {
      perror("fcntl - lock");
      close(fd);
      return 1;
   }
```

```
    printf("Lock acquired. Press Enter to release...\n");
    getchar();  // Hold the lock until Enter is pressed
    lock.l_type = F_UNLCK;
    fcntl(fd, F_SETLK, &lock);   // Release the lock
    printf("Lock released.\n");
    close(fd);
    return 0;
}
```

Important Notes:

File locks apply per file description (not per descriptor), so dup(), fork() share the lock.

Locking only works if all cooperating processes use it (advisory).

Use blocking mode (F_SETLKW) when you want the process to wait until the lock is available.

VS flock() vs fcntl()

Feature	**flock()**	**fcntl()**
Simpler	✅	❌ More complex
File-level	✅ Locks entire file	✅ Record-level possible
Portable	❌ BSD/Linux only	✅ POSIX compliant