POSIX File I/O Extensions Cheat Sheet

Comprehensive Guide to Extended File I/O System Calls

Basic Calls:

open(path, flags, mode)	Open or create a file, returning a file descriptor.
close(fd)	Close an open file descriptor.
read(fd, buf, count)	Read data from file at current offset.
write(fd, buf, count)	Write data to file at current offset.
lseek(fd, offset, whence)	Move file offset (SEEK_SET, SEEK_CUR, SEEK_END).

Extended / Random Access:

pread(fd, buf, count, offset)	Read at given offset, without changing file offset.
pwrite(fd, buf, count, offset)	Write at given offset, without changing file offset.
truncate(path, length)	Resize file to specified length.
ftruncate(fd, length)	Resize file (by descriptor).

Vector I/O:

readv(fd, iov, iovcnt)	Read into multiple buffers (scatter read).
writev(fd, iov, iovcnt)	Write from multiple buffers (gather write).
preadv(fd, iov, iovcnt, offset)	Vector read at offset.
pwritev(fd, iov, iovcnt, offset)	Vector write at offset.

Asynchronous / Advanced:

aio_read()	Asynchronous read request (POSIX AIO).
aio_write()	Asynchronous write request.
aio_suspend()	Wait for one or more async I/O operations.
aio_cancel()	Cancel async I/O request.
sync()/fsync(fd)	Flush file buffers to disk.
fdatasync(fd)	Flush only data (not metadata) to disk.

Other Useful Extensions:

dup(fd)	Duplicate file descriptor (lowest available number).
dup2(fd, newfd)	Duplicate to specific descriptor number.
pipe(fds[2])	Create unidirectional data channel (read/write ends).
mkfifo(path, mode)	Create a named pipe (FIFO).
sendfile(out_fd, in_fd, *offset, count)	Efficiently copy data between fds (Linux extension).