UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

# 7.5 File systems

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

**KEY WORDS** inode • journaling

## Objective

Follow the pointers, price the reads, and order the crash writes.

## 1 Key knowledge and worked examples

### ■ The layout

The inode holds 4 direct pointers and one indirect (256 pointers, 1 KiB blocks): max file $(4 + 256) \times 1$ KiB $= 260$ KiB ✓. The directory is a file of name → inode entries ✓.

### ■ Worked example: block 6

Beyond the four direct pointers: read the indirect block (pointer 2), then the data – two disk reads for one logical block ✓. A 100-block file reads sequentially in $100 + 1$ reads ✓.

## 2 Practice

**2.1** With 2 KiB blocks and 512 pointers per indirect block, compute the max file with 4 direct pointers. [3]

**2.2** Reading block 9 (4 direct pointers): how many disk reads, and why? [3]

**2.3** Does deletion erase the data? Answer and explain in one line. [3]

## 3 Exam-style questions

**3.1** A file system uses 4 KiB blocks, 8 direct pointers, and one indirect block holding 1024 pointers.

Compute the maximum file size. Compute the reads for block 12 with a cached inode (name the pointer index). Compute the total reads for a sequentially read 3000-block file (indirect block read once). State what changes if the indirect block is evicted mid-read. [8]

**3.2** An append writes three things: the data block, then the free-list bit, then the inode size.

State what is lost if the crash lands between data and free-list, and between free-list and inode. Name the write order rule that never loses user data (only space). Explain how journaling removes the ordering constraint, and its per-write cost in one line. [7]

## Solutions

**2.1**

- With 4 direct pointers: $(4 + 512) \times 2$ KiB $= 1032$ KiB ✓.

**2.2**

- Two: the indirect block (pointer $9 - 4 = 5$), then the data block ✓.

**2.3**

- No – deletion clears the pointers; the bytes survive until the blocks are reused ✓.

**3.1**

- Max file: $(8 + 1024) \times 4$ KiB $= 4128$ KiB $\approx 4.03$ MiB ✓.
- Block 12: two reads – the indirect block (pointer $12 - 8 = 4$), then the data ✓.
- 3000-block file: $3000 + 1 = 3001$ reads ✓.
- Mid-read eviction: the indirect block is re-read on the next beyond-direct access – occasional extra reads, not incorrect ones ✓.

**3.2**

- Between data and free-list: nothing user-visible – an orphan block the free list no longer offers (lost space) ✓.
- Between free-list and inode: the size lies – reads past the old end return garbage ✓.
- Rule: data before metadata (inode last) – a crash then loses only space, never user bytes ✓.
- Journaling writes the intent first and replays it after the crash, so any order recovers; cost: one extra journal write per update ✓.

exercises.lol exercises.com.cn