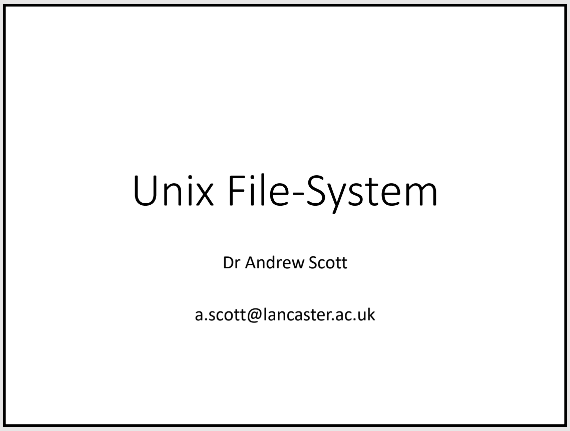
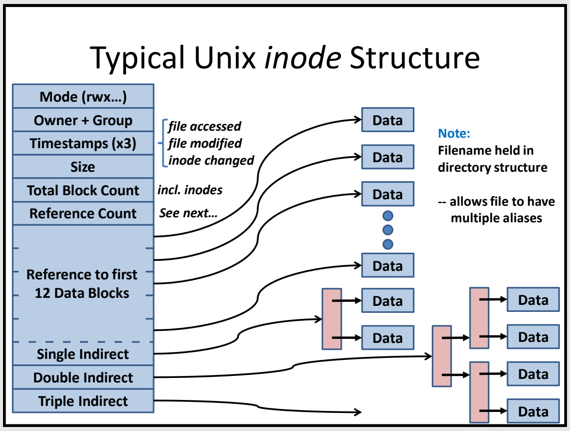


Unix File-System

Dr Andrew Scott

a.scott@lancaster.ac.uk

1



Typical Unix *inode* Structure

Mode (rwx…)

Owner + Group

Timestamps (x3) — *file accessed*, *file modified*, *inode changed*

Size

Total Block Count — *incl. inodes*

Reference Count — *See next…*

Reference to first 12 Data Blocks

Single Indirect

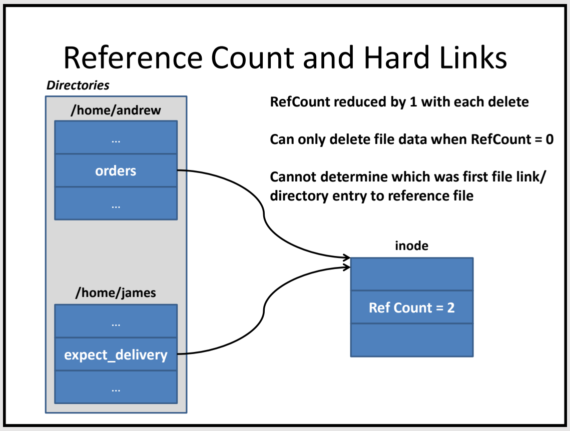
Double Indirect

Triple Indirect

Note:
Filename held in directory structure

-- allows file to have multiple aliases

2



Reference Count and Hard Links

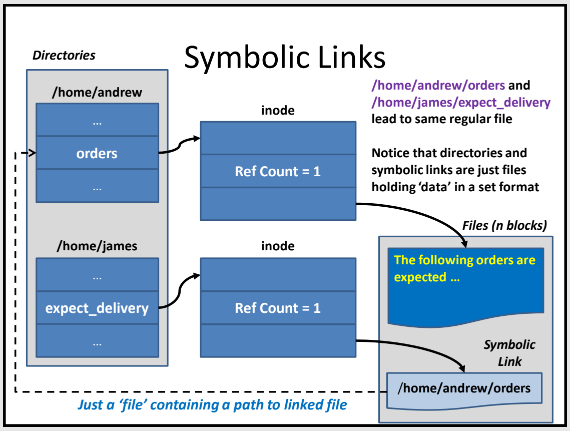
Directories

RefCount reduced by 1 with each delete

Can only delete file data when RefCount = 0

Cannot determine which was first file link/ directory entry to reference file

3

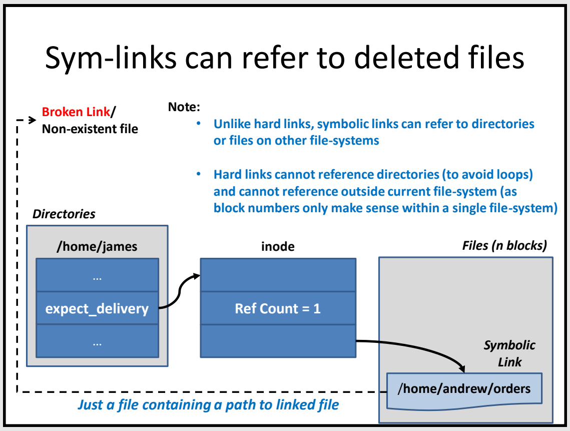


Symbolic Links

/home/andrew/orders and /home/james/expect_delivery lead to same regular file

Notice that directories and symbolic links are just files holding 'data' in a set format

4

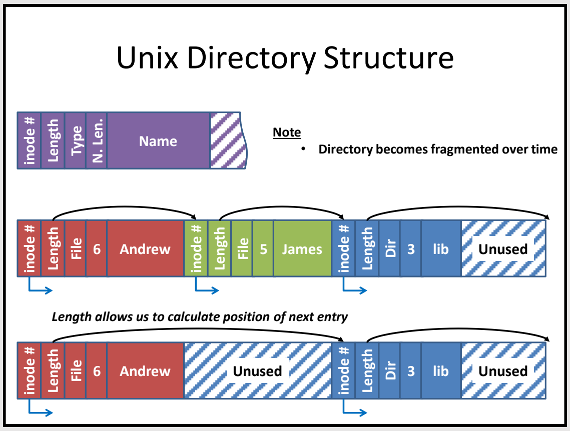


Sym-links can refer to deleted files

Note:

- Unlike hard links, symbolic links can refer to directories or files on other file-systems
- Hard links cannot reference directories (to avoid loops) and cannot reference outside current file-system (as block numbers only make sense within a single file-system)

5



Unix Directory Structure

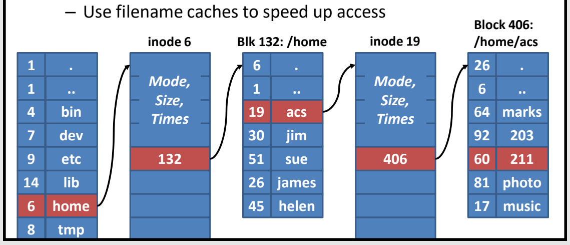
Note

- Directory becomes fragmented over time

6

File Lookup: */home/acs/211*

- Often
  - Search from last directory entry read
    - Processes typically scan directories sequentially
  - Use filename caches to speed up access



1	.
1	..
4	bin
7	dev
9	etc
14	lib
6	home
8	tmp

inode 6

Mode, Size, Times

132

Blk 132: /home

6	.
1	..
19	acs
30	jim
51	sue
26	james
45	helen

inode 19

Mode, Size, Times

406

Block 406: /home/acs

26	.
6	..
64	marks
92	203
60	211
81	photo
17	music

7

Block Size and Efficiency/ Waste

- Values taken off 1.2GB 4.3BSD Unix file-system
  - Wastage as percentage of disk space
  - Larger block sizes allow for larger files
    - More links per index block
    - More data in each referenced data block

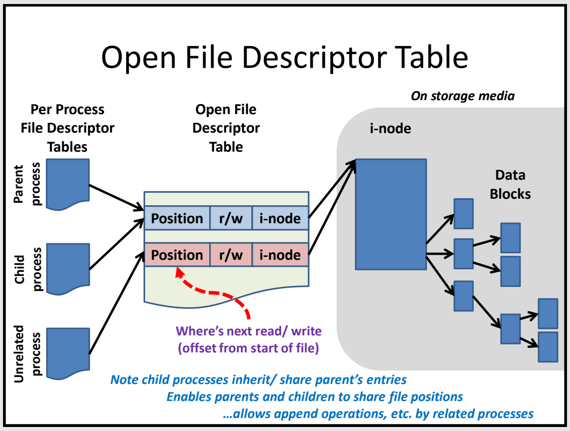
Space Used (Mbyte)	%age Waste	Structure
775.2	0.0	(control value) Just the raw data
828.7	6.9	Data & inodes with **512** byte blocks
866.5	11.8	Data & inodes with **1024** byte blocks
948.5	22.4	Data & inodes with **2048** byte blocks
1128.3	45.6	Data & inodes with **4096** byte blocks

8

Block Fragments

- Efficiency best when block and file sizes matched, but
  - Small blocks tend to suit small files
  - Large blocks tend to suit large files
- Tended to move to larger blocks as file sizes increased
  - Media files, etc.
- Some file-systems can allocate fragments of blocks
  - i.e. 4 x 1K fragments from a single 4K disk block
- Blocks must be multiple of disk sector size

9



Open File Descriptor Table

On storage media

Per Process File Descriptor Tables

Open File Descriptor Table

i-node

Data Blocks

Parent process

Child process

Unrelated process

Position	r/w	i-node
Position	r/w	i-node

Where's next read/ write (offset from start of file)

Note child processes inherit/ share parent's entries

Enables parents and children to share file positions

…allows append operations, etc. by related processes

10