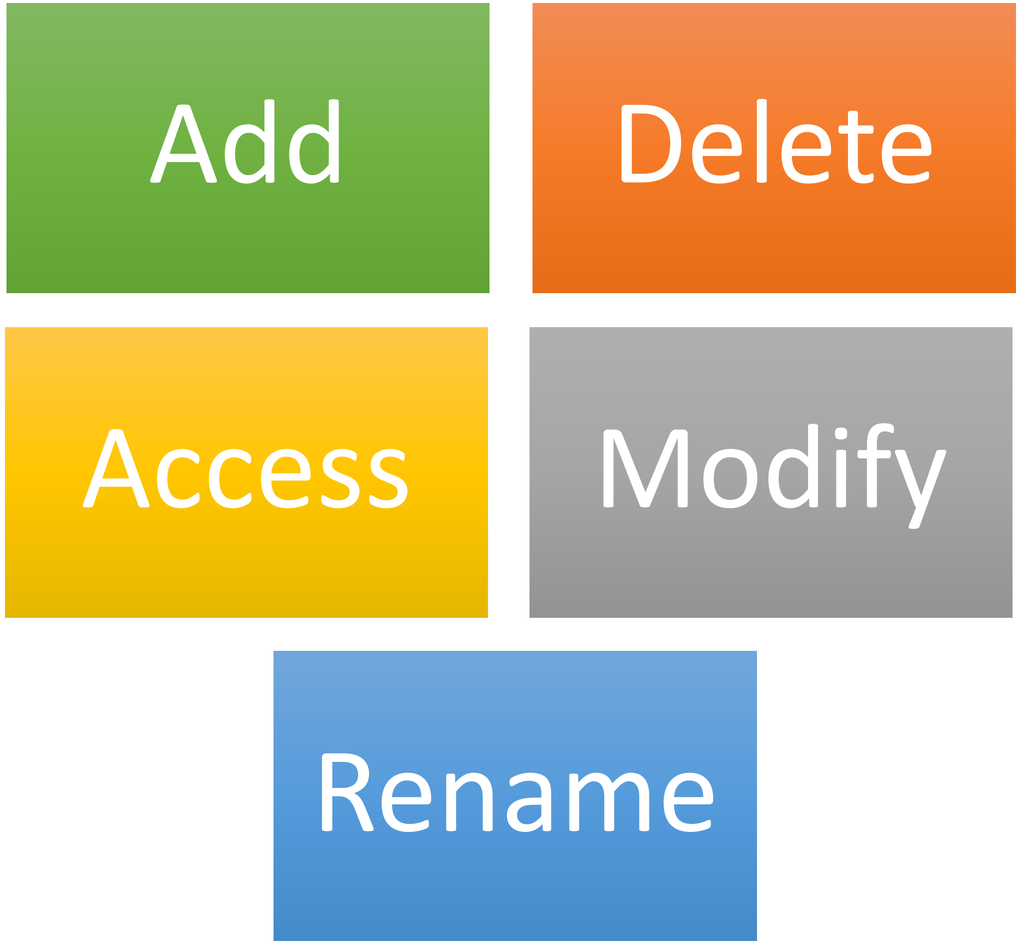
Tracing ext3 file system operations in the emulator QEMU

Stepanov Vlad

vladislav.stepanov@ispras.ru

Tracing file system operations



Add

Delete

Access

Modify

Rename

File operations monitoring is needed for:

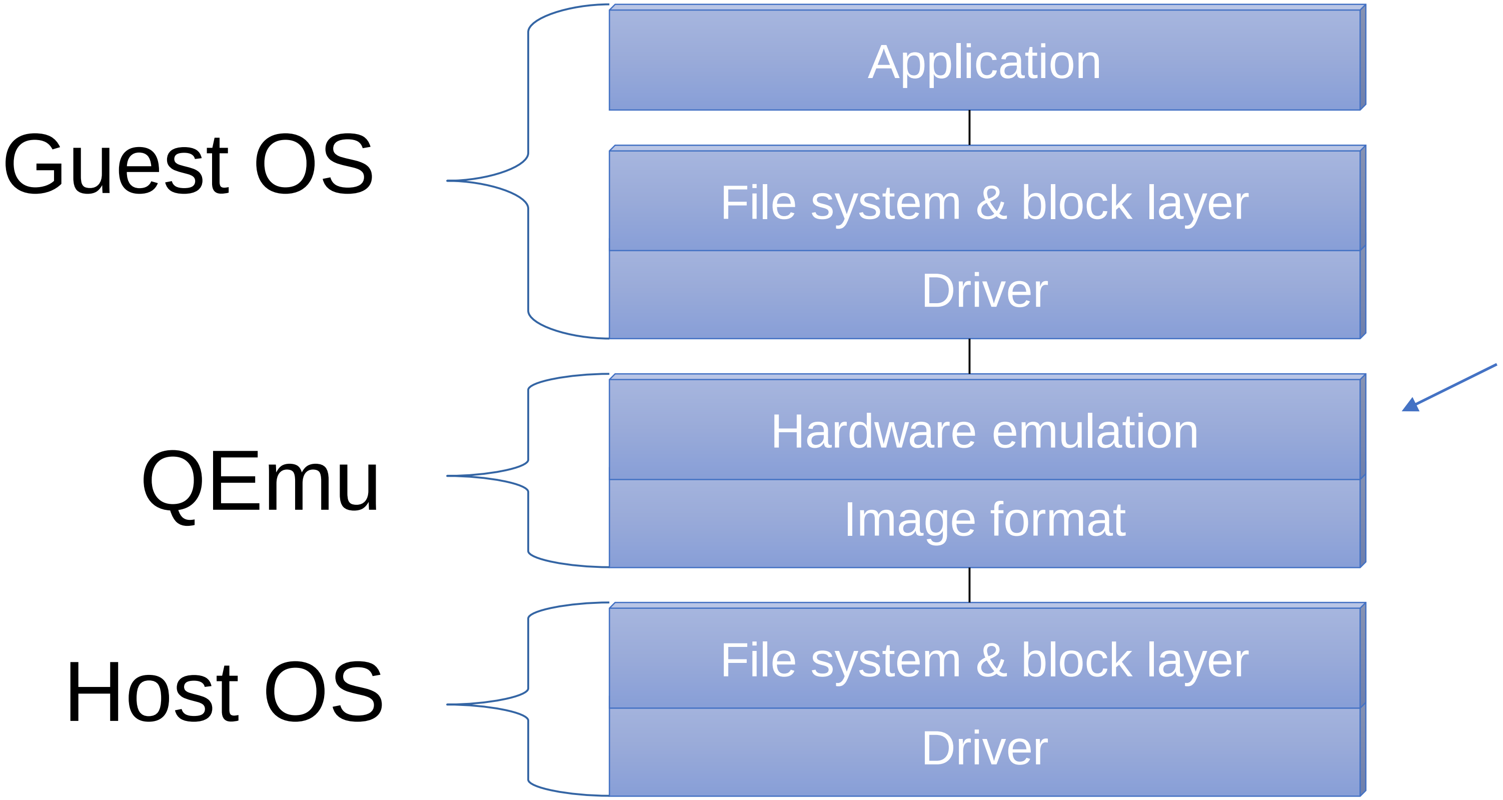
- Debug of OS and FS drivers
- Monitor top critical change events
- Investigate systems with unknown internal organization

Review existing solutions

- Based on OS kernel subsystems
  1. Inotify (Linux)
  2. FileSystemWatcher (Windows)
  3. Kqueue (Mac Os X и FreeBSD)
- Based on the interception of system calls

  QEMU-based framework for non-intrusive virtual machine instrumentation and introspection. P. Dovgalyuk, N. Fursova, I. Vasiliev, V. Makarov. 2017.

Solution



The module intercepts requests to the virtual disk at this level.

Advantages:

- non-intrusive
- no OS dependent parts

Disadvantages:

- Each file system requires a separate implementation.

Intercepting a request to a virtual disk

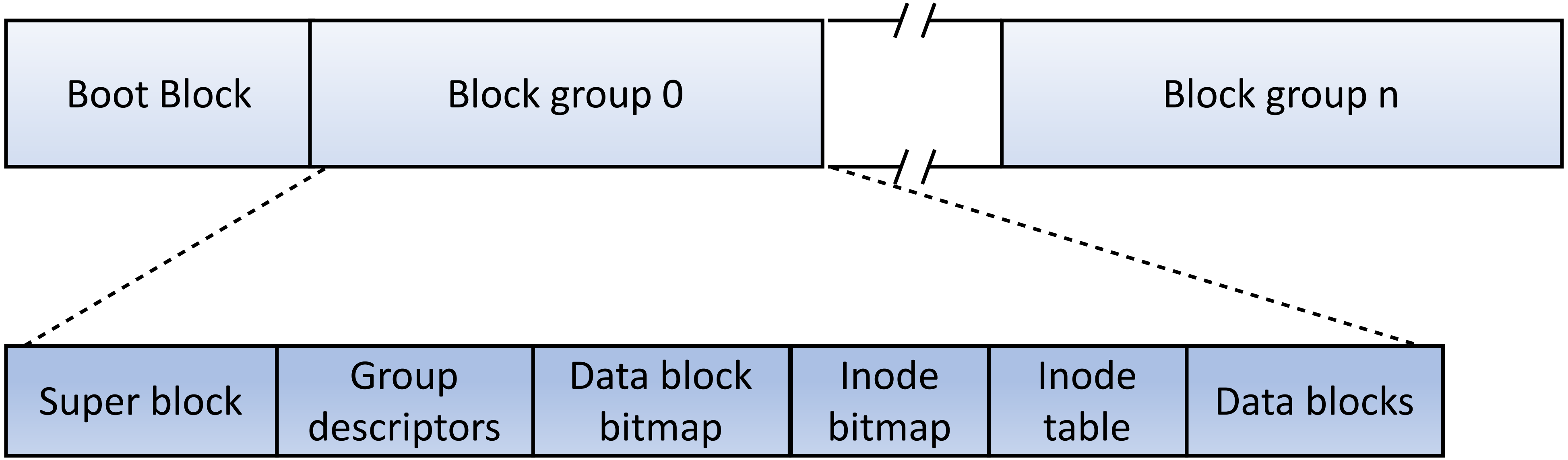
A virtual disk request consists of:

- disk sector number
- number of bytes read or written

We find the file name based on:

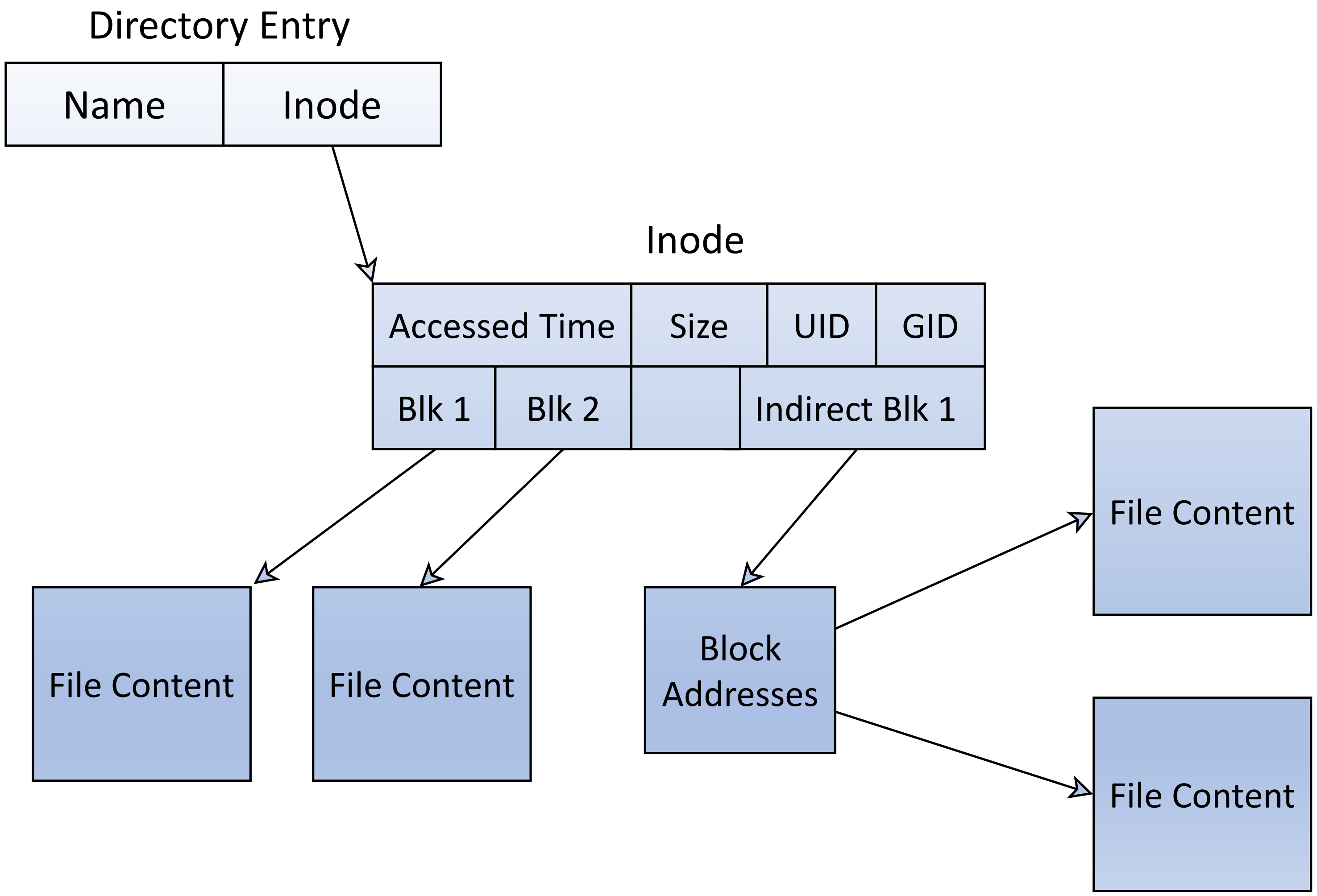
- sector
- disk contents
- file system knowledge

Ext3



Boot Block
Block group 0
Block group n
Super block
Group descriptors
Data block bitmap
Inode bitmap
Inode table
Data blocks

Ext3



Directory Entry

Name

Inode

Inode

Accessed Time

Size

UID

GID

Blk 1

Blk 2

Indirect Blk 1

File Content

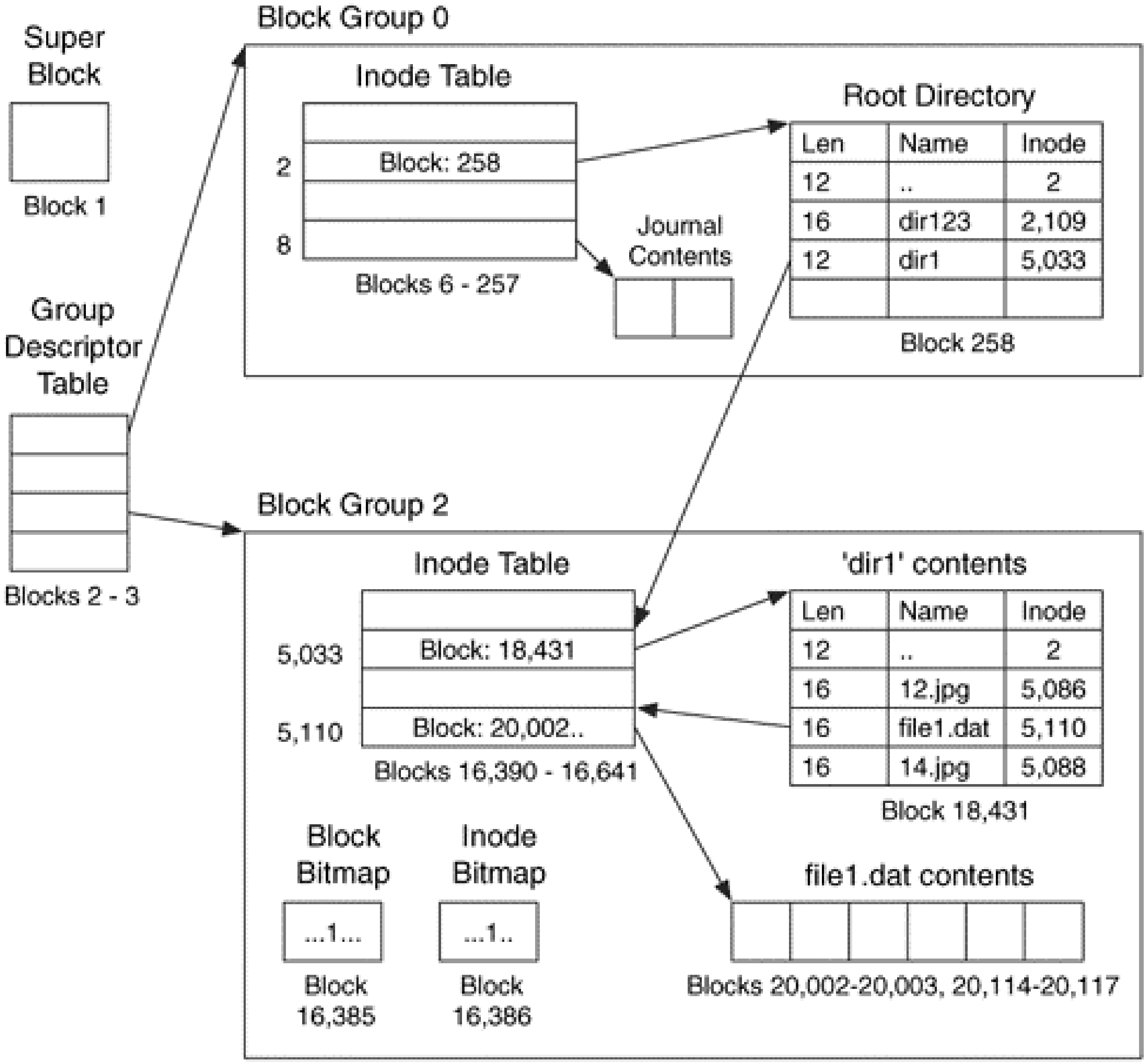
File Content

Block Addresses

File Content

File Content

File name search



Root Directory

Len	Name	Inode
12	..	2
16	dir123	2,109
12	dir1	5,033

Block 258

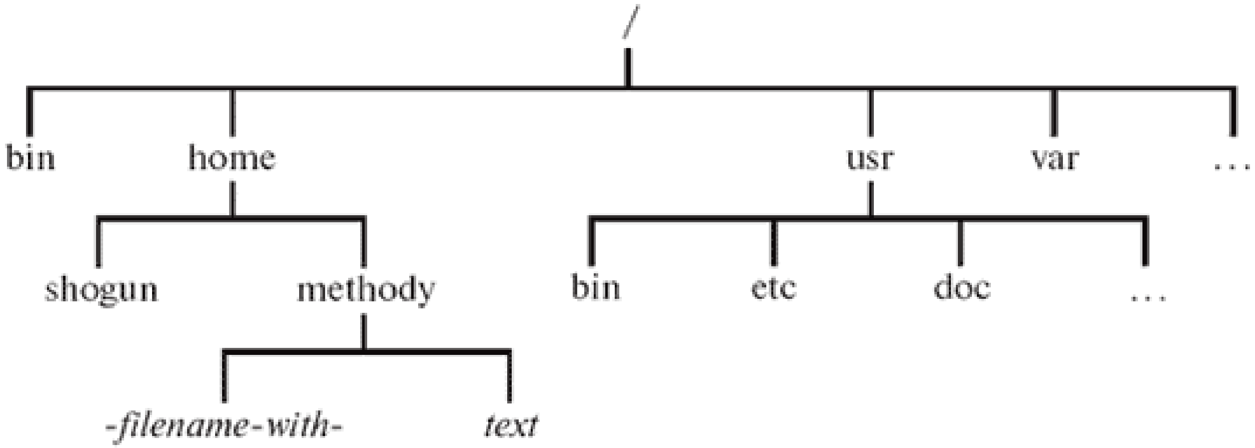
'dir1' contents

Len	Name	Inode
12	..	2
16	12.jpg	5,086
16	file1.dat	5,110
16	14.jpg	5,088

Block 18,431

Data structures for quick file name lookup

Directory tree



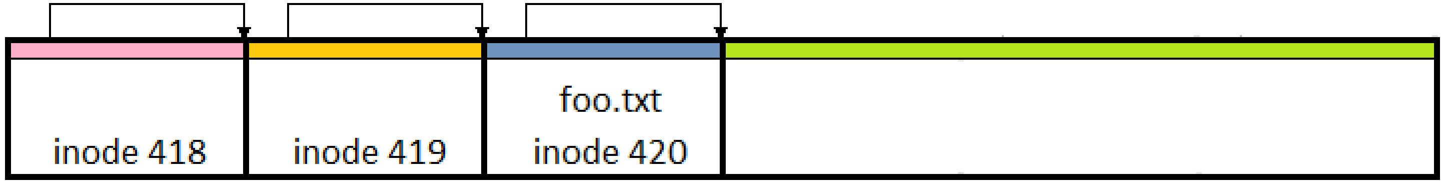
Associative array

Block number	File name
645345	/home
658345	/usr/bin

Changing data structures

Adding Inode Entry

link("bar.txt")



inode 418

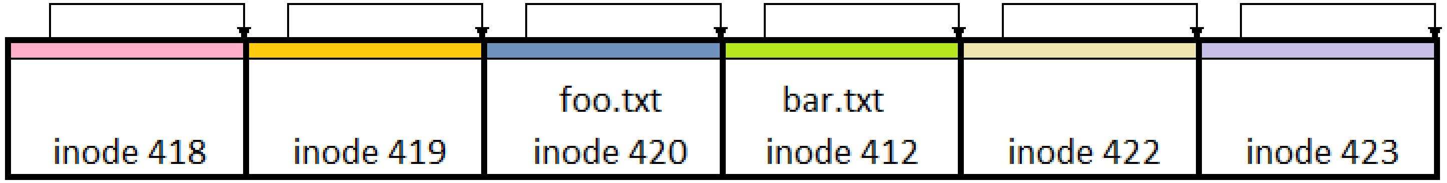
inode 419

foo.txt

inode 420

Deleting Inode Entry

unlink("bar.txt")



inode 418

inode 419

foo.txt

inode 420

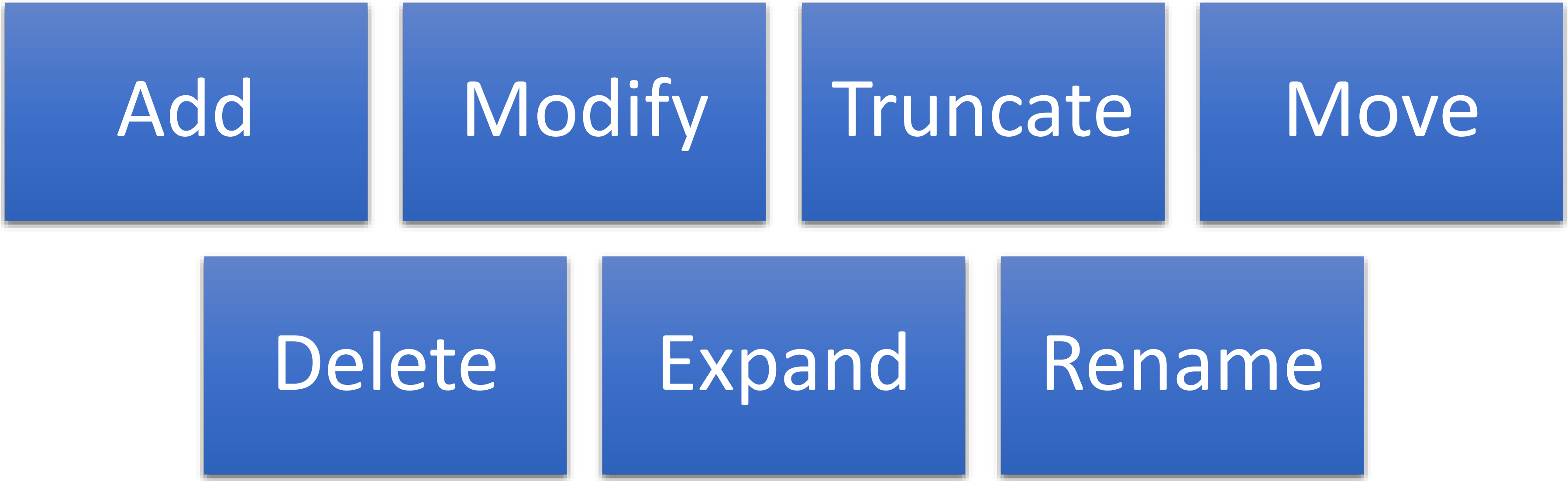
bar.txt

inode 412

inode 422

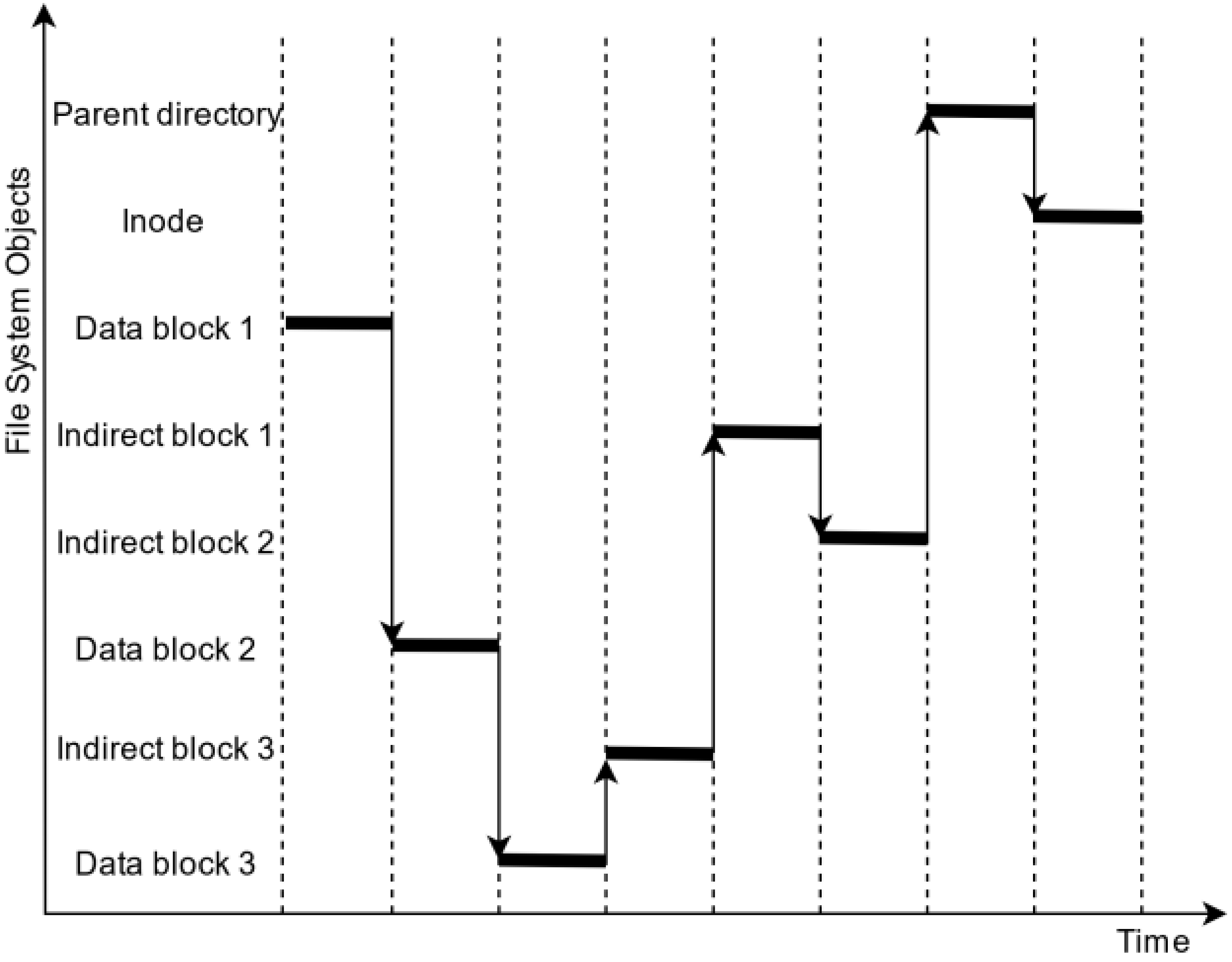
inode 423

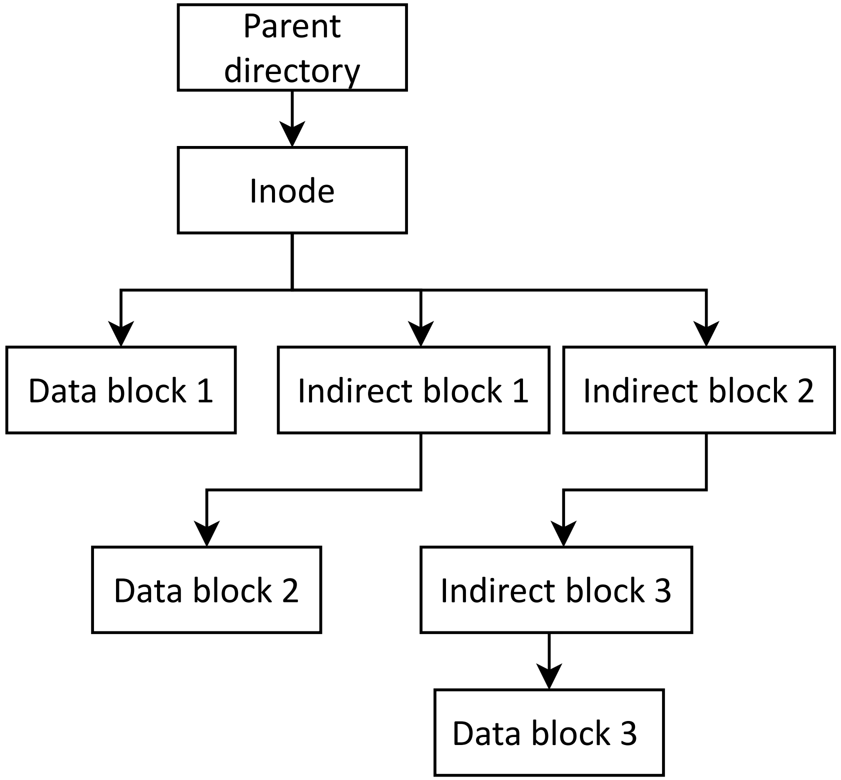
Recognition of file operations



- Add
- Modify
- Truncate
- Move
- Delete
- Expand
- Rename

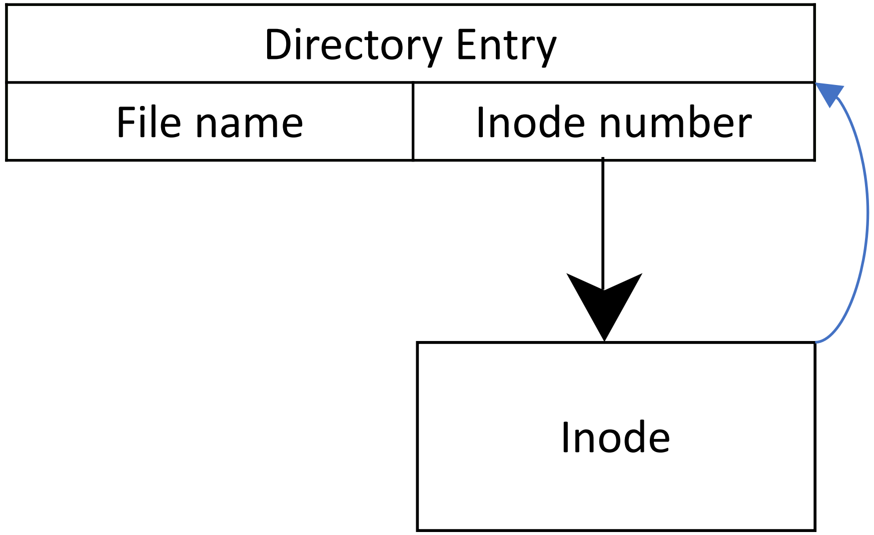
The problem of disk queries indefinite order

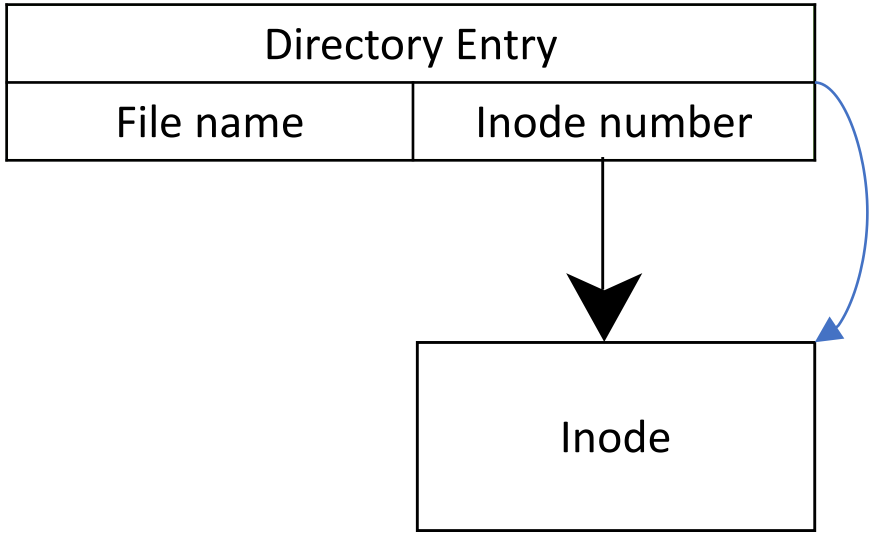




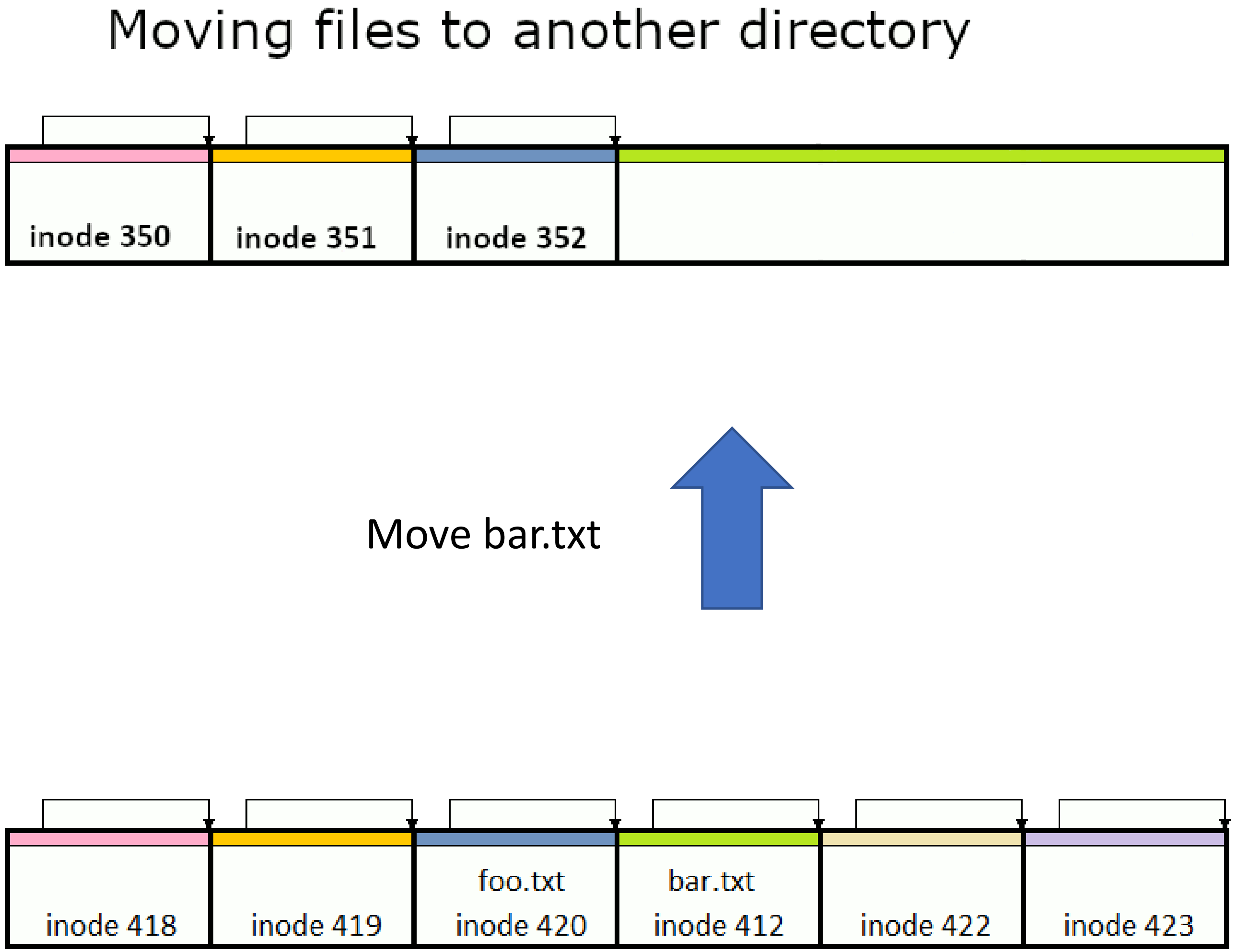
The problem of disk queries indefinite order

Adding file





The problem of disk queries indefinite order



Example

```
$ dd if=test of=test1
```

```
read 6926160 16384 /bin/dd
read 6926192 32768 /bin/dd
read 6926256 12288 /bin/dd
read 10263952 4096 /home/debian/test
...
create /home/debian/test1
write 2640360 4096 /home/debian
write 10490400 4096 /home/debian/test1
```

Testing

OS:

- Linux
- Windows 10
- Free BSD
- KolibriOS

Disk capacity:

- 500 MiB
- 6 GiB

Conclusion

- The QEMU module for tracing ext3 file system operations was created.
- This module gets information about file operations by intercepting disk requests.
- Disk requests are processed without delay.
- The order of writing structures to disk is not important. All file operations will be processed.

Stepanov Vlad

vladislav.stepanov@ispras.ru