

Virtual File System (VFS) in Linux Kernel

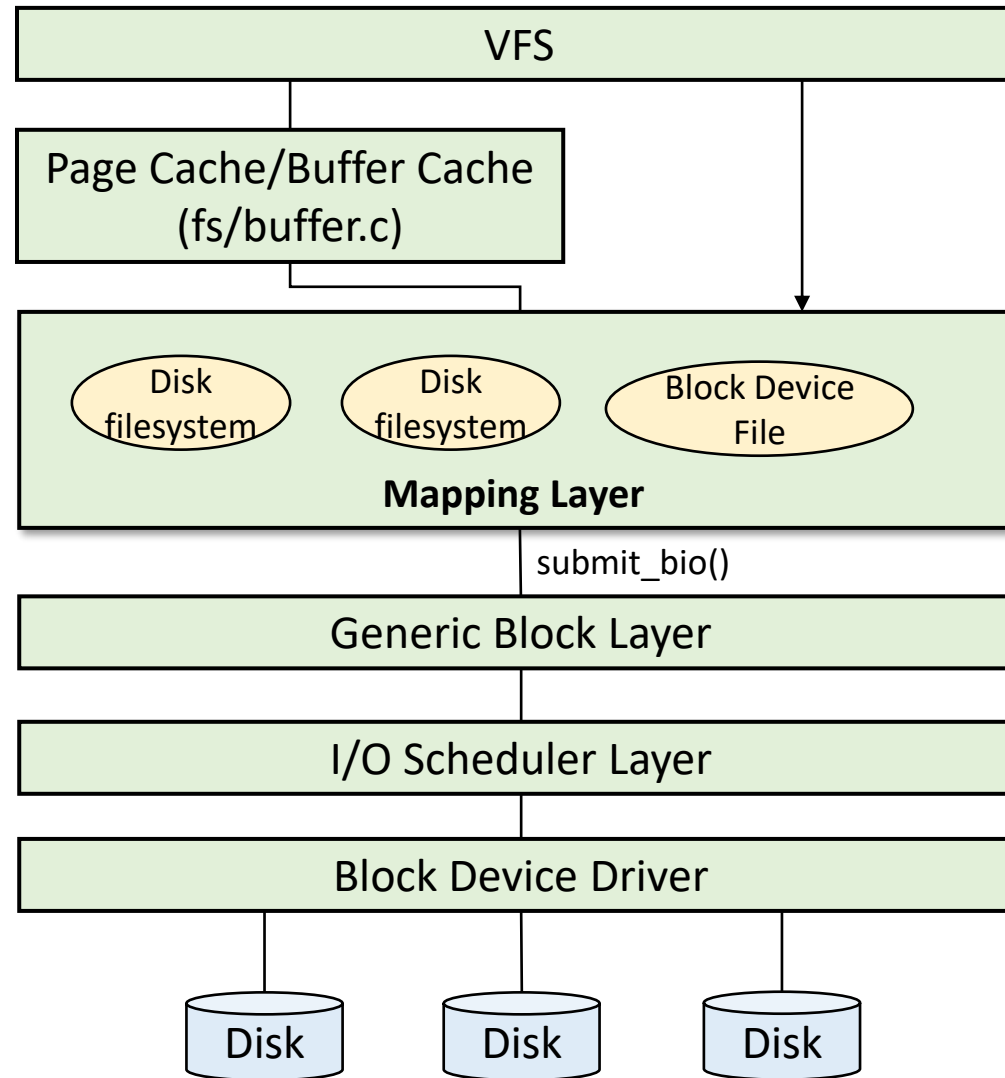
Adrian Huang | Nov, 2021

- * Based on kernel 5.11 (x86_64) – QEMU
- * SMP (4 CPUs) and 8GB memory
- * Kernel parameter: nokaslr norandmaps
- * Userspace: ASLR is disabled
- * Legacy BIOS

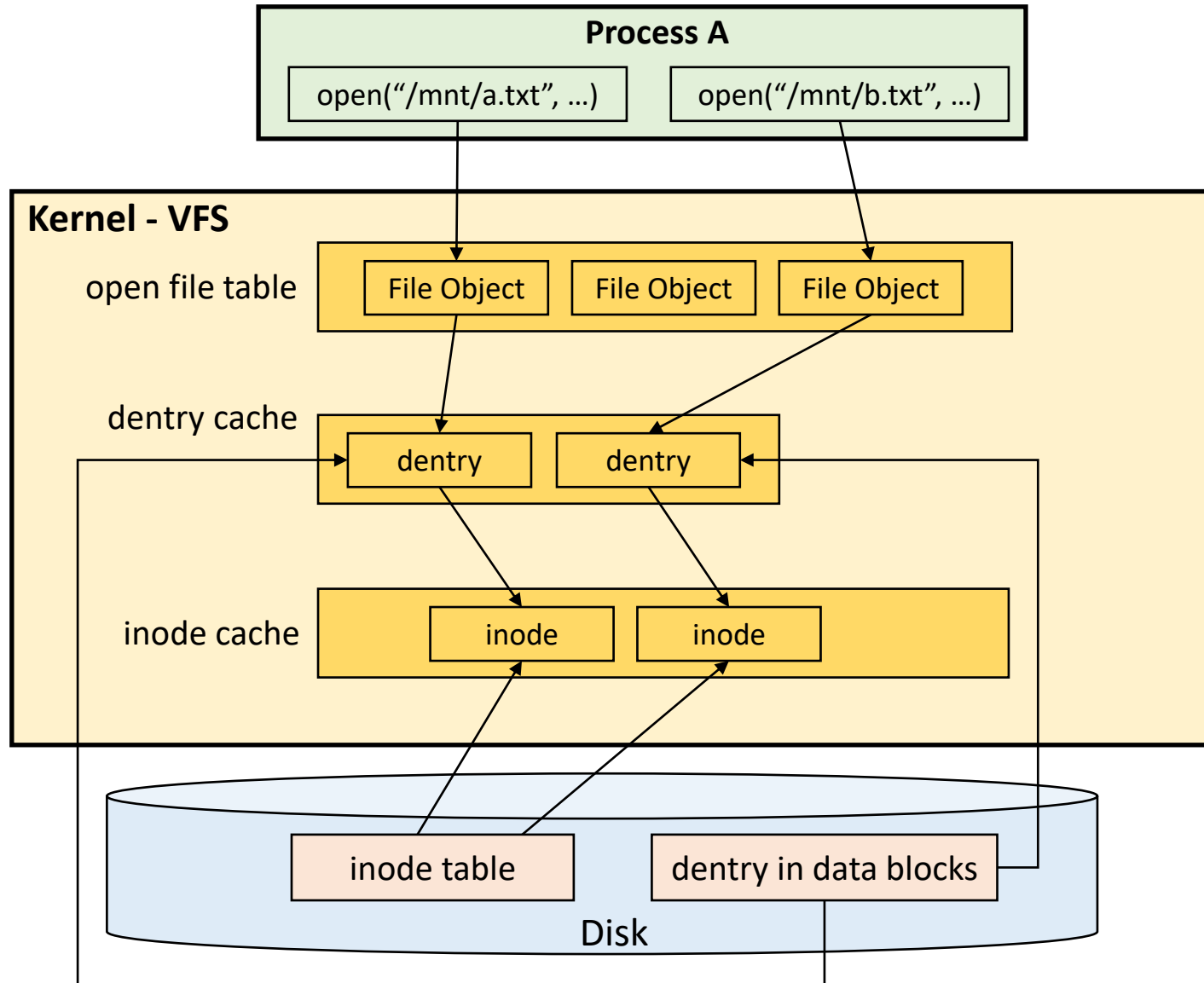
Agenda

- Kernel components affected by an IO request
- Object relationship in VFS
- Ext4 file system format
- dentry (Directory Entry) and inode (Index Node)
 - Example: ext4
- Anatomy of `mount` system call
- Anatomy of `ls` system call

Kernel components affected by an IO request



Object relationship in VFS



Note

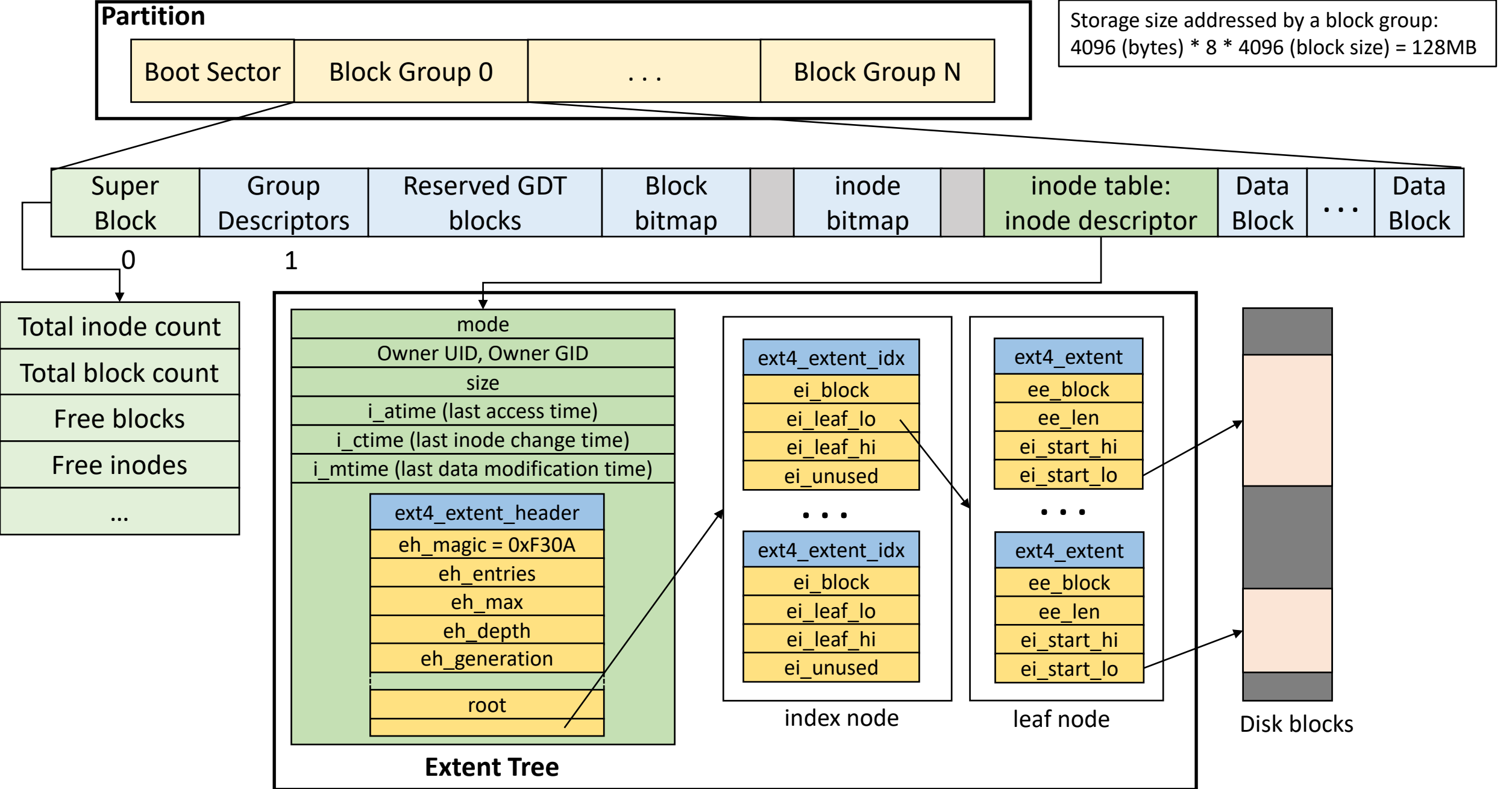
dentry cache list

- Linked via `hlist_bl_{head,node}` (hash table with bit lock)

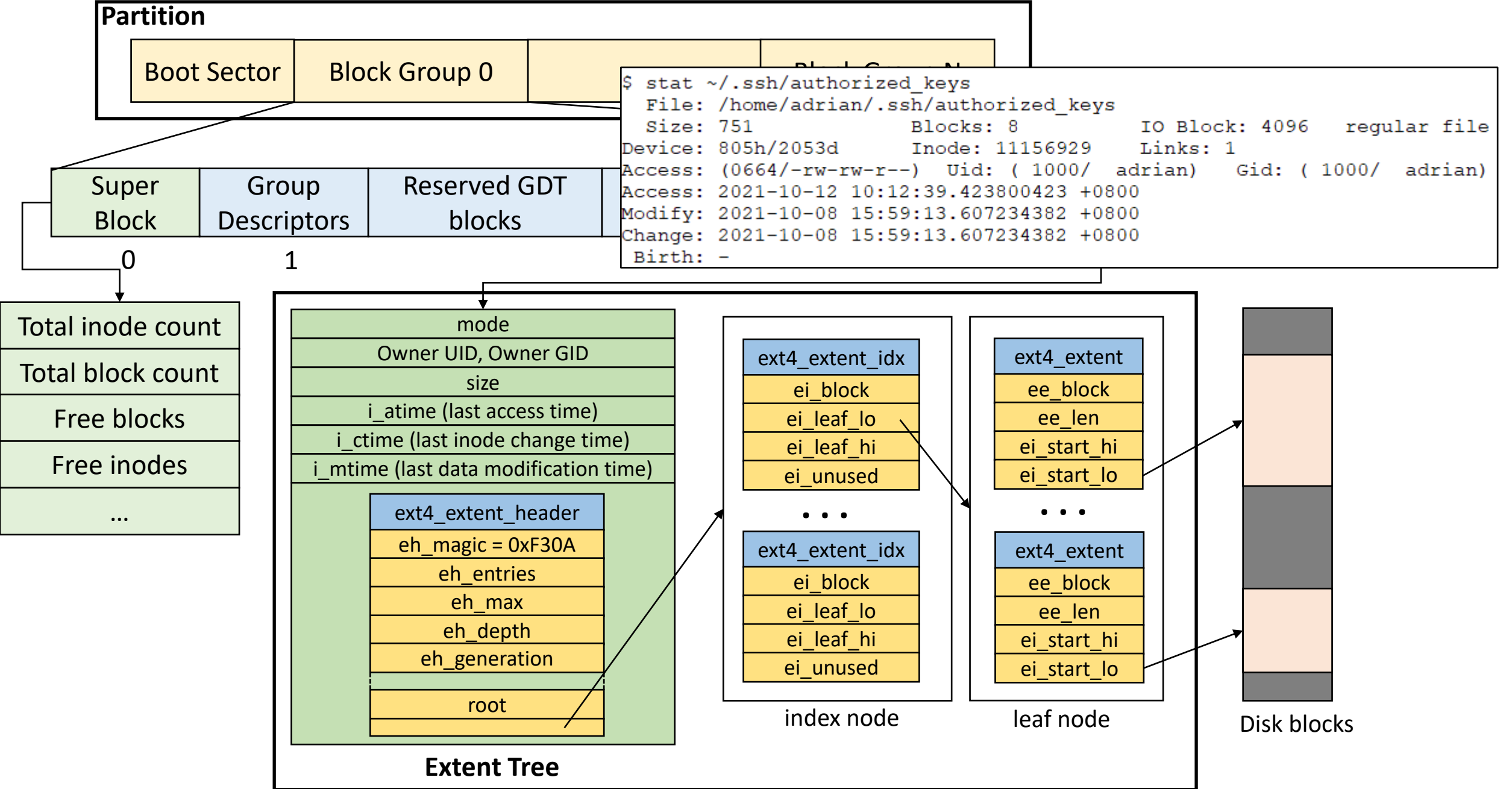
inode cache list

- Linked via `hlist_{head,node}` (hash table)

Ext4 file system format



Ext4 file system format



Relationship between inode and directory entry (dentry)

```
root@adrian-ubuntu:/home/adrian/archive/img/small-lab# mount 16MiB.img mnt/
```

```
root@adrian-ubuntu:/home/adrian/archive/img/small-lab# ls -liaR mnt/
mnt/:
total 32
 2 drwxr-xr-x 5 root root 4096 Oct 27 17:13 .
2646185 drwxrwxr-x 3 adrian adrian 4096 Oct 27 17:14 ..
 12 drwxr-xr-x 3 root root 4096 Oct 27 17:11 a123
 17 drwxr-xr-x 2 root root 4096 Oct 27 17:13 a789
 11 drwx----- 2 root root 16384 Oct 27 17:09 lost+found
inode number      link count
mnt/a123:
total 20
12 drwxr-xr-x 3 root root 4096 Oct 27 17:11 .
 2 drwxr-xr-x 5 root root 4096 Oct 27 17:13 ..
13 drwxr-xr-x 2 root root 4096 Oct 27 17:13 b456
15 -rw-r--r-- 2 root root  78 Oct 27 17:11 hello.c
15 -rw-r--r-- 2 root root  78 Oct 27 17:11 hello-hard-link.c

mnt/a123/b456:
total 16
13 drwxr-xr-x 2 root root 4096 Oct 27 17:13 .
12 drwxr-xr-x 3 root root 4096 Oct 27 17:11 ..
16 -rw-r--r-- 2 root root  100 Oct 27 17:12 test.c
16 -rw-r--r-- 2 root root  100 Oct 27 17:12 test-hard-link.c
14 lrwxrwxrwx 1 root root    6 Oct 27 17:13 test-soft-link.c -> test.c

mnt/a789:
total 8
17 drwxr-xr-x 2 root root 4096 Oct 27 17:13 .
 2 drwxr-xr-x 5 root root 4096 Oct 27 17:13 ..

mnt/lost+found:
total 20
11 drwx----- 2 root root 16384 Oct 27 17:09 .
 2 drwxr-xr-x 5 root root  4096 Oct 27 17:13 ..
```

inode#	Purpose
0	Doesn't exist; there is no inode 0
1	List of defective blocks
2	Root directory
3	User quota
4	Group quota
5	Boot loader
6	Undelete directory
7	Reserved group descriptors inode. ("resize inode")
8	Journal inode
9	The "exclude" inode
10	Replica inode
11	First non-reserved inode. Usually this is the lost+found directory. See s_first_ino in the superblock.

Relationship between inode and directory entry (dentry)

`dumpe2fs 16MiB.img`

```
...
Block size:          4096
...
Group 0: (Blocks 0-4095) csum 0x46c2 [ITABLE_ZEROED]
  Primary superblock at 0, Group descriptors at 1-1
  Reserved GDT blocks at 2-2
  Block bitmap at 3 (+3), csum 0xca84d2c1
  Inode bitmap at 19 (+19), csum 0x40f23db5
  Inode table at 35-162 (+35)
  2928 free blocks, 4079 free inodes, 5 directories, 4079 unused inodes
  Free blocks: 1167-1679, 1681-4095
  Free inodes: 18-4096
```

inode bitmap: $19 * 4096 = 77824$ (0x13000)

```
adrian@adrian-ubuntu:small-lab$ xxd -l 16 -seek 0x13000 16MiB.img
00013000: ffff 0100 0000 0000 0000 0000 0000 0000  ....
```

inode 1-16 inode 17

```
root@adrian-ubuntu:/home/adrian/archive/img/small-lab# ls -liaR mnt/
mnt/:
total 32
  2 drwxr-xr-x 5 root  root  4096 Oct 27 17:13 .
2646185 drwxrwxr-x 3 adrian adrian 4096 Oct 27 17:14 ..
 12 drwxr-xr-x 3 root  root  4096 Oct 27 17:11 a123
 17 drwxr-xr-x 2 root  root  4096 Oct 27 17:13 a789
 11 drwx----- 2 root  root 16384 Oct 27 17:09 lost+found

mnt/a123:
total 20
12 drwxr-xr-x 3 root  root 4096 Oct 27 17:11 .
 2 drwxr-xr-x 5 root  root 4096 Oct 27 17:13 ..
13 drwxr-xr-x 2 root  root 4096 Oct 27 17:13 b456
15 -rw-r--r-- 2 root  root   78 Oct 27 17:11 hello.c
15 -rw-r--r-- 2 root  root   78 Oct 27 17:11 hello-hard-link.c

mnt/a123/b456:
total 16
13 drwxr-xr-x 2 root  root 4096 Oct 27 17:13 .
12 drwxr-xr-x 3 root  root 4096 Oct 27 17:11 ..
16 -rw-r--r-- 2 root  root  100 Oct 27 17:12 test.c
16 -rw-r--r-- 2 root  root  100 Oct 27 17:12 test-hard-link.c
14 lrwxrwxrwx 1 root  root    6 Oct 27 17:13 test-soft-link.c -> test.c

mnt/a789:
total 8
17 drwxr-xr-x 2 root  root 4096 Oct 27 17:13 .
 2 drwxr-xr-x 5 root  root 4096 Oct 27 17:13 ..

mnt/lost+found:
total 20
11 drwx----- 2 root  root 16384 Oct 27 17:09 .
 2 drwxr-xr-x 5 root  root  4096 Oct 27 17:13 ..
```


Relationship between inode and directory entry (dentry)

`dumpe2fs 16MiB.img`

```
...
Inode size:          128
....

Group 0: (Blocks 0-4095) csum 0x46c2 [ITABLE_ZEROED]
  Primary superblock at 0, Group descriptors at 1-1
  Reserved GDT blocks at 2-2
  Block bitmap at 3 (+3), csum 0xca84d2c1
  Inode bitmap at 19 (+19), csum 0x40f23db5
  Inode table at 35-162 (+35)
  2928 free blocks, 4079 free inodes, 5 directories, 4079 unused inodes
  Free blocks: 1167-1679, 1681-4095
  Free inodes: 18-4096
```

inode entry: $35 * 4096 + 128 * (2 - 1)$ [inode #2] = 0x23080

inode structure data of inode #2

```
adrian@adrian-ubuntu:small-lab$ xxd -l 128 -seek 0x23080 16MiB.img
00023080: ed41 0000 0010 0000 3c18 7961 3b18 7961  .A.....<.ya;.ya
00023090: 3b18 7961 0000 0000 0000 0500 0800 0000  ;.ya.....
000230a0: 0000 0800 0200 0000 0af3 0100 0400 0000  .....
000230b0: 0000 0000 0000 0000 0100 0000 0400 0000  .....
000230c0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230f0: 0000 0000 0000 0000 0000 0000 8de5 0000  .....
```

```
root@adrian-ubuntu:/home/adrian/archive/img/small-lab# ls -liaR mnt/
mnt/:
total 32
  2 drwxr-xr-x 5 root   root   4096 Oct 27 17:13 .
2646185 drwxrwxr-x 3 adrian adrian 4096 Oct 27 17:14 ..
  12 drwxr-xr-x 3 root   root   4096 Oct 27 17:11 a123
  17 drwxr-xr-x 2 root   root   4096 Oct 27 17:13 a789
  11 drwx----- 2 root   root  16384 Oct 27 17:09 lost+found

mnt/a123:
total 20
 12 drwxr-xr-x 3 root   root   4096 Oct 27 17:11 .
  2 drwxr-xr-x 5 root   root   4096 Oct 27 17:13 ..
 13 drwxr-xr-x 2 root   root   4096 Oct 27 17:13 b456
 15 -rw-r--r-- 2 root   root    78 Oct 27 17:11 hello.c
 15 -rw-r--r-- 2 root   root    78 Oct 27 17:11 hello-hard-link.c

mnt/a123/b456:
total 16
 13 drwxr-xr-x 2 root   root   4096 Oct 27 17:13 .
 12 drwxr-xr-x 3 root   root   4096 Oct 27 17:11 ..
 16 -rw-r--r-- 2 root   root    100 Oct 27 17:12 test.c
 16 -rw-r--r-- 2 root   root    100 Oct 27 17:12 test-hard-link.c
 14 lrwxrwxrwx 1 root   root     6 Oct 27 17:13 test-soft-link.c -> test.c

mnt/a789:
total 8
 17 drwxr-xr-x 2 root   root   4096 Oct 27 17:13 .
  2 drwxr-xr-x 5 root   root   4096 Oct 27 17:13 ..

mnt/lost+found:
total 20
 11 drwx----- 2 root   root  16384 Oct 27 17:09 .
  2 drwxr-xr-x 5 root   root   4096 Oct 27 17:13 ..
```

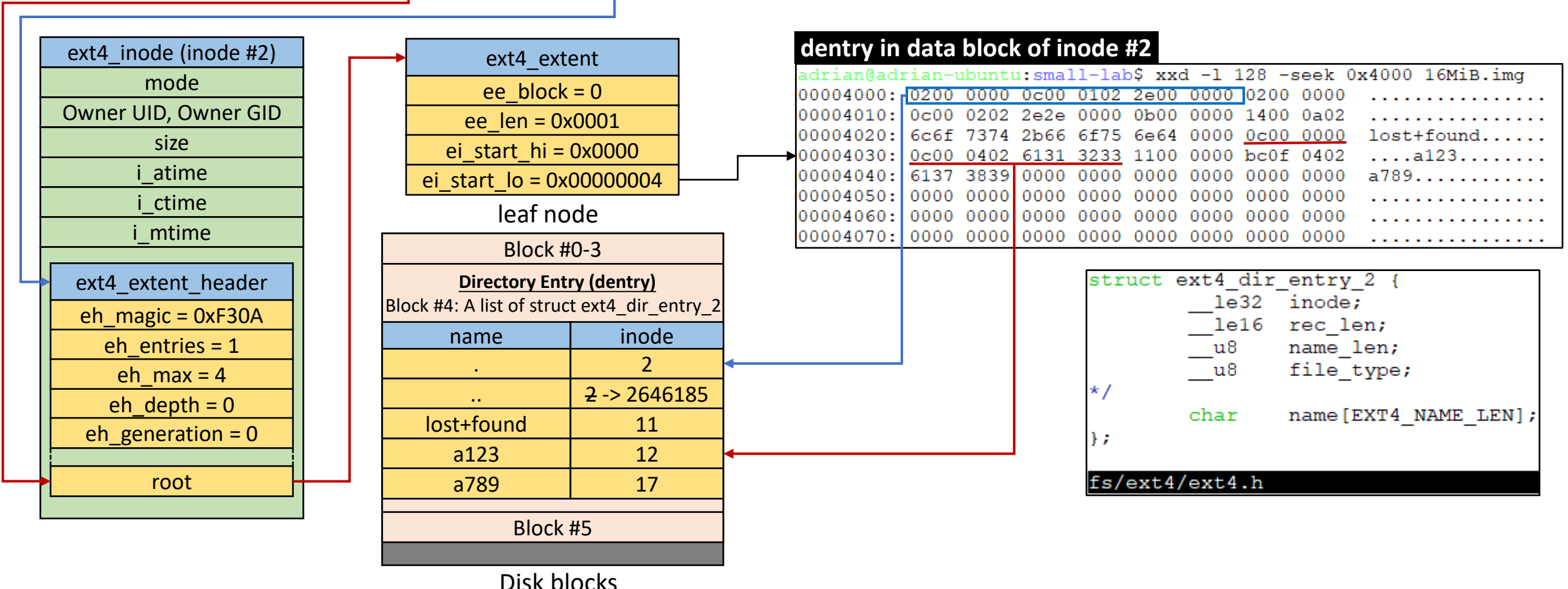
Relationship between inode and dentry – root (inode #2)

inode structure data of inode #2

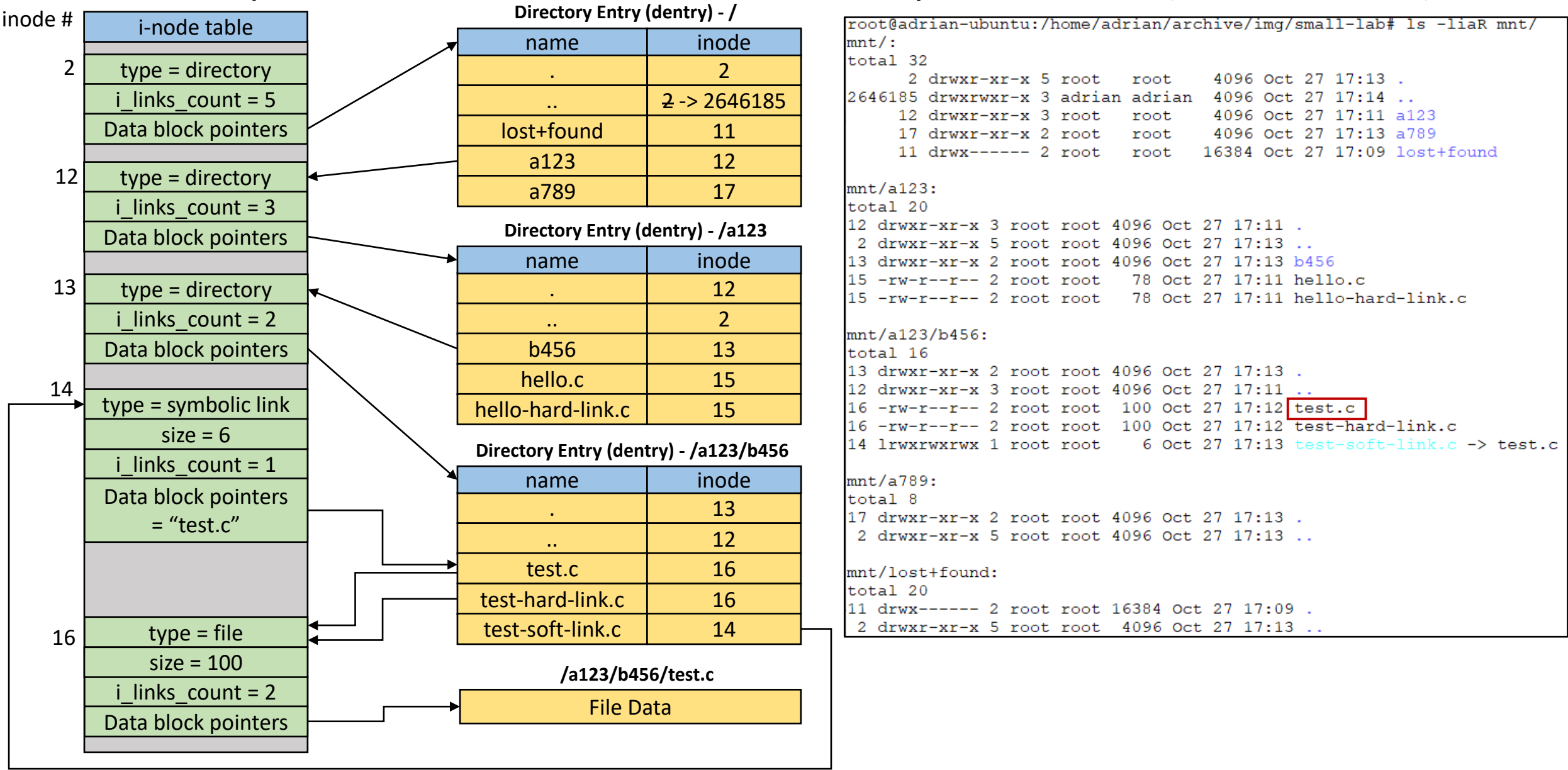
```
adrian@adrian-ubuntu:small-lab$ xxd -l 128 -seek 0x23080 16MiB.img
00023080: ed41 0000 0010 0000 3c18 7961 3b18 7961  .A.....<.ya;.ya
00023090: 3b18 7961 0000 0000 0000 0500 0800 0000  ;.ya.....
000230a0: 0000 0800 0200 0000 0af3 0100 0400 0000  .....
000230b0: 0000 0000 0000 0000 0100 0000 0400 0000  .....
000230c0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000230f0: 0000 0000 0000 0000 0000 8de5 0000 0000  .....
```

Root Directory

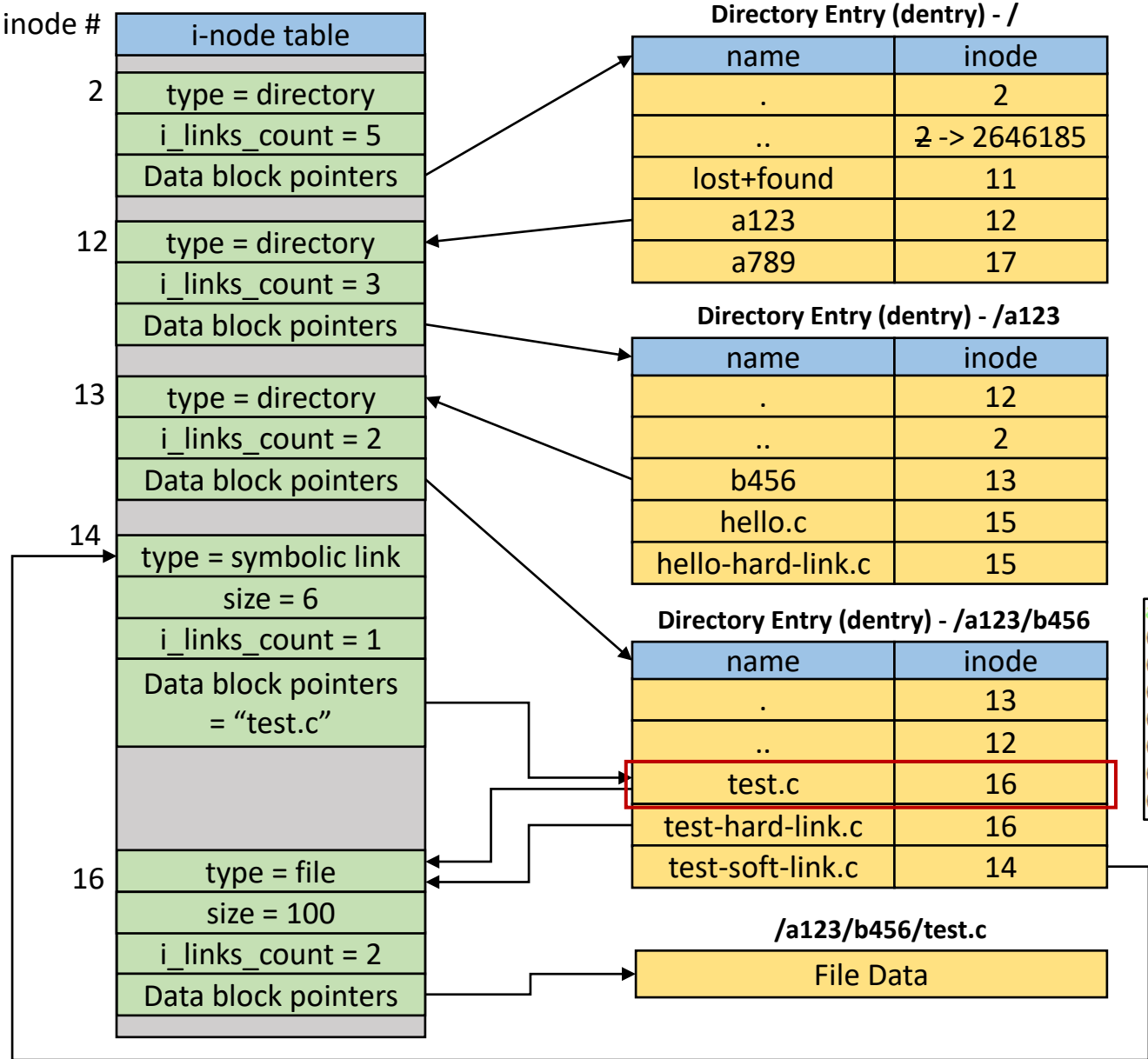
```
root@adrian-ubuntu:/home/adrian/archive/img/small-lab# ls -liaR mnt/
mnt/:
total 32
  2 drwxr-xr-x 5 root   root    4096 Oct 27 17:13 .
2646185 drwxrwxr-x 3 adrian adrian 4096 Oct 27 17:14 ..
 12 drwxr-xr-x 3 root   root    4096 Oct 27 17:11 a123
 17 drwxr-xr-x 2 root   root    4096 Oct 27 17:13 a789
 11 drwx----- 2 root   root   16384 Oct 27 17:09 lost+found
```



Relationship between inode and dentry – ‘b456’ (inode #13)



Relationship between inode and dentry – ‘test.c’ (inode #16)



address inode structure of inode #16

inode entry: $35 * 4096 + 128 * (16 - 1)$ [inode #16] = 0x23780

inode structure data of inode #16

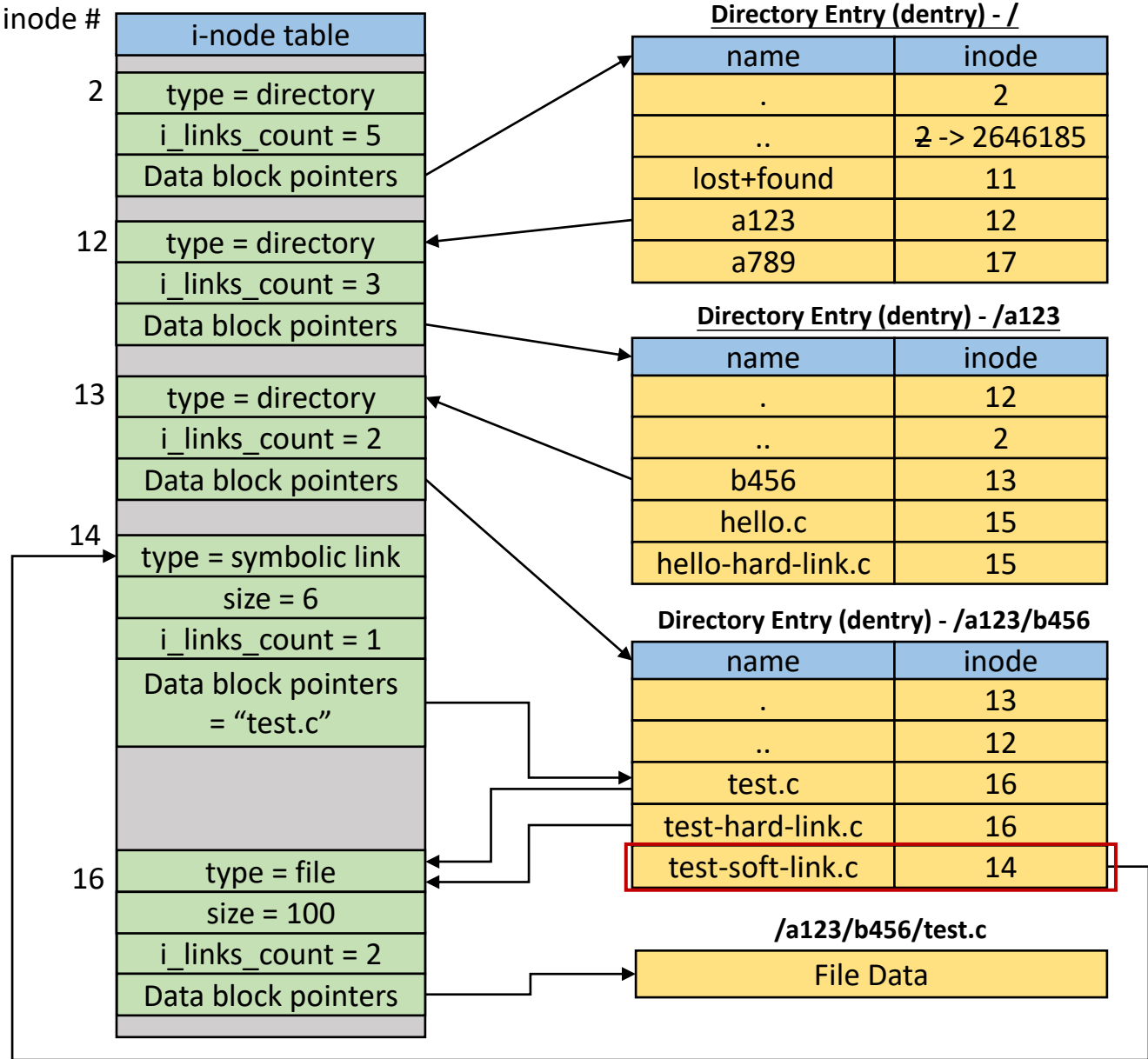
```
adrian@adrian-ubuntu:small-lab$ xxd -l 128 -seek 0x23780 16MiB.img
00023780: a481 0000 6400 0000 a5fb 7961 1518 7961 ....d....ya..ya
00023790: 0c18 7961 0000 0000 0000 0200 0800 0000 ..ya.....
000237a0: 0000 0800 0100 0000 0af3 0100 0400 0000 .....
000237b0: 0000 0000 0000 0000 0100 0000 9006 0000 .....
000237c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000237d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000237e0: 0000 0000 3605 568b 0000 0000 0000 0000 ....6.V.....
000237f0: 0000 0000 0000 0000 0000 0000 899a 0000 .....
```

i_links_count ← 0200

Data block pointer represented by struct ext4_ext ← 9006

```
adrian@adrian-ubuntu:small-lab$ xxd -l 100 -seek 0x690000 16MiB.img
00690000: 2369 6e63 6c75 6465 203c 7374 6469 6f2e #include <stdio.
00690010: 683e 0a0a 696e 7420 7465 7374 2869 6e74 h>..int test(int
00690020: 2061 290a 7b0a 0972 6574 7572 6e20 6120 a){..return a
00690030: 2b20 323b 0a7d 0a0a 696e 7420 6d61 696e + 2;..}..int main
00690040: 2876 6f69 6429 0a7b 0a09 7465 7374 2831 (void){..test(1
00690050: 3029 3b0a 090a 0972 6574 7572 6e20 303b 0);...return 0;
00690060: 0a7d 0a0a
```

Relationship between inode and dentry – ‘test-soft-link.c’ (inode #14)



address inode structure of inode #14

inode entry: $35 * 4096 + 128 * (14 - 1)$ [inode #14] = 0x23680

inode structure data of inode #14

```
adrian@adrian-ubuntu:small-lab$ xxd -l 128 -seek 0x23680 16MiB.img
00023680: ffa1 0000 0600 0000 2218 7961 2118 7961 .....".ya!.ya
00023690: 2118 7961 0000 0000 0000 0100 0000 0000 !.ya.....
000236a0: 0000 0000 0100 0000 7465 7374 2e63 0000 .....test.c..
000236b0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000236c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000236d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000236e0: 0000 0000 de9a fbb3 0000 0000 0000 0000 .....
000236f0: 0000 0000 0000 0000 0000 0000 8737 0000 .....7..
```

Data block pointer ← i_links_count

- Hard Link**
- 1. Refer to an inode number
 - 2. Must reside on the same file system
 - 3. Cannot be made to a directory
- Soft Link**
- 1. Refer to a filename instead of an inode number
 - 2. Can be linked to a file in a different file system
 - 3. Can be made to a directory

mount – mount a new ext4 file system

```
/ # strace /bin/mount /adrian/img/src.img /adrian/mnt/  
execve("/bin/mount", ["/bin/mount", "/adrian/img/src.img", "/adrian/mnt/"], 0x70  
...  
stat("/adrian/img/src.img", {st_mode=S_IFREG|0664, st_size=33554432, ...}) = 0  
...  
openat(AT_FDCWD, "/dev/loop0", O_RDWR) = 4  
...  
mount("/dev/loop0", "/adrian/mnt/", "ext4", MS_SILENT, NULL) = 0  
close(4) = 0
```

- stat() – get file status from inode
 - Name resolution → get dentry → get inode
- openat()
 - Name resolution
 - Update open file table
- mount()
 - Name resolution
 - Add a mount entry to 'mount' hash table
 - Parse file system

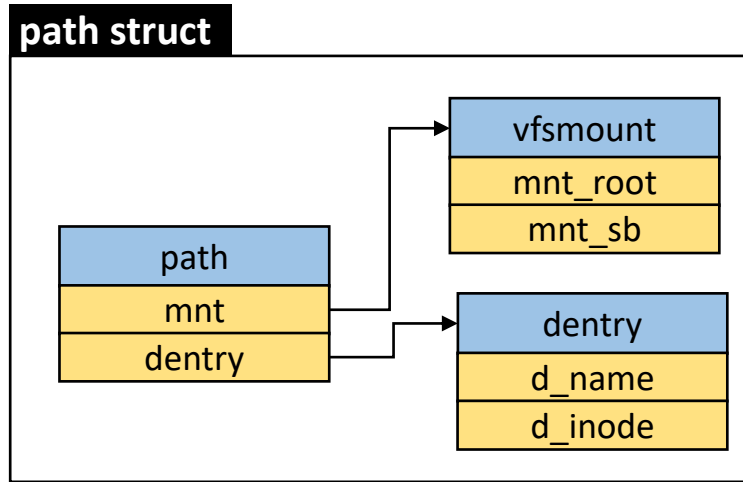
stat("/adrian/img/src.img", ...)

```
sys_stat [fs/stat.c]
|- vfs_stat
  |- vfs_fstatat(AT_FDCWD, filename, stat, 0);
  |- vfs_statx
    struct path path;
    |- user_path_at(dfd, filename, lookup_flags, &path)
    |- vfs_getattr(&path, stat, request_mask, flags);
  |- cp_old_stat

user_path_at [fs/stat.c]
|- user_path_at_empty
  |- struct filename *filename = getname_flags(...)
  |- filename_lookup
    /*
     * Fill a 'path' struct after finding a file.
     * - Prepare a nameidata struct 'nd'
     */
    |- set_nameidata
    |- path_lookupat

path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
  Case 1. Absolute pathname
  |- nd_jump_root
    if (!nd->root.mnt) {
      |- set_root
        nd->root = current->fs->root
    }
    nd->path = nd->root
    d = nd->path.dentry
    nd->inode = d->d_inode
  Case 2. Relative pathname
  nd->path = current->fs->pwd
  nd->inode = nd->path.dentry->d_inode
  return nd->name->name;
while (!(err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
  ;
|- complete_walk
```

- stat() – get file status from inode
 - Name resolution → get dentry → get inode

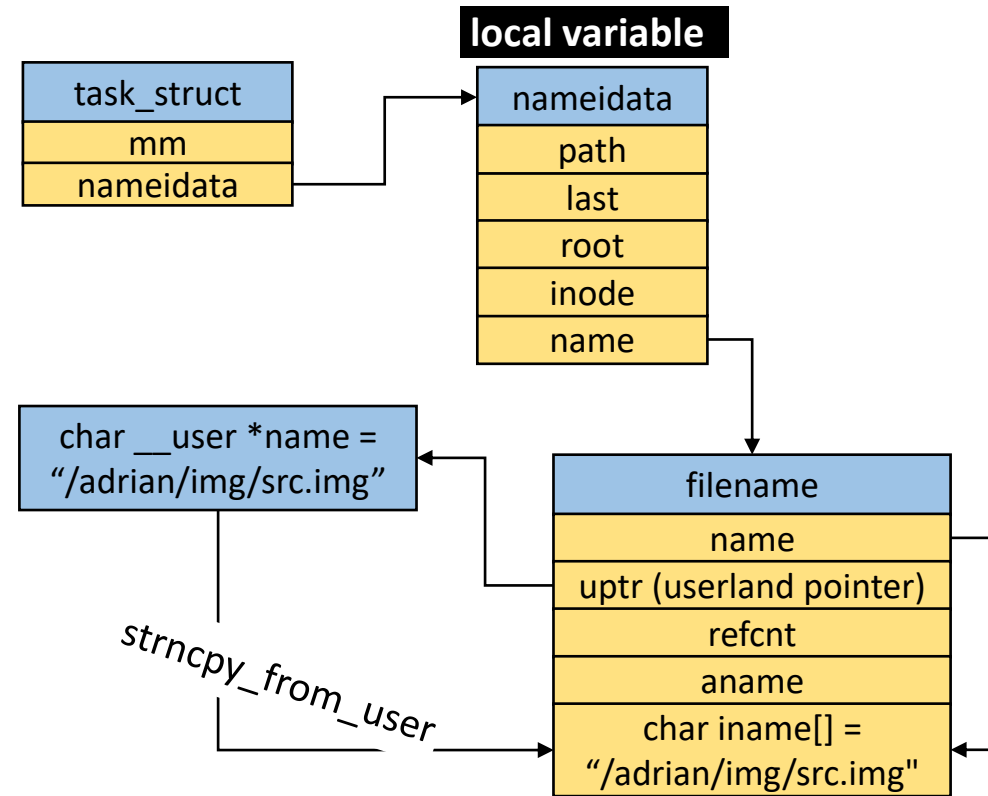


user_path_at(): fill a path struct (dentry and mount info) after resolving a pathname

stat("/adrian/img/src.img", ...)

```
user_path_at [fs/stat.c]
|- user_path_at_empty
|- struct filename *filename = getname_flags(...)
|- filename_lookup
/*
 * Fill a 'path' struct after finding a file.
 * - Prepare a nameidata struct 'nd'
 */
|- set_nameidata
|- path_lookupat

path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
Case 1. Absolute pathname
|- nd_jump_root
if (!nd->root.mnt) {
|- set_root
nd->root = current->fs->root
}
nd->path = nd->root
d = nd->path.dentry
nd->inode = d->d_inode
Case 2. Relative pathname
nd->path = current->fs->pwd
nd->inode = nd->path.dentry->d_inode
return nd->name->name;
while (!(err = link_path_walk(s, nd)) &&
(s = lookup_last(nd)) != NULL)
;
|- complete_walk
```

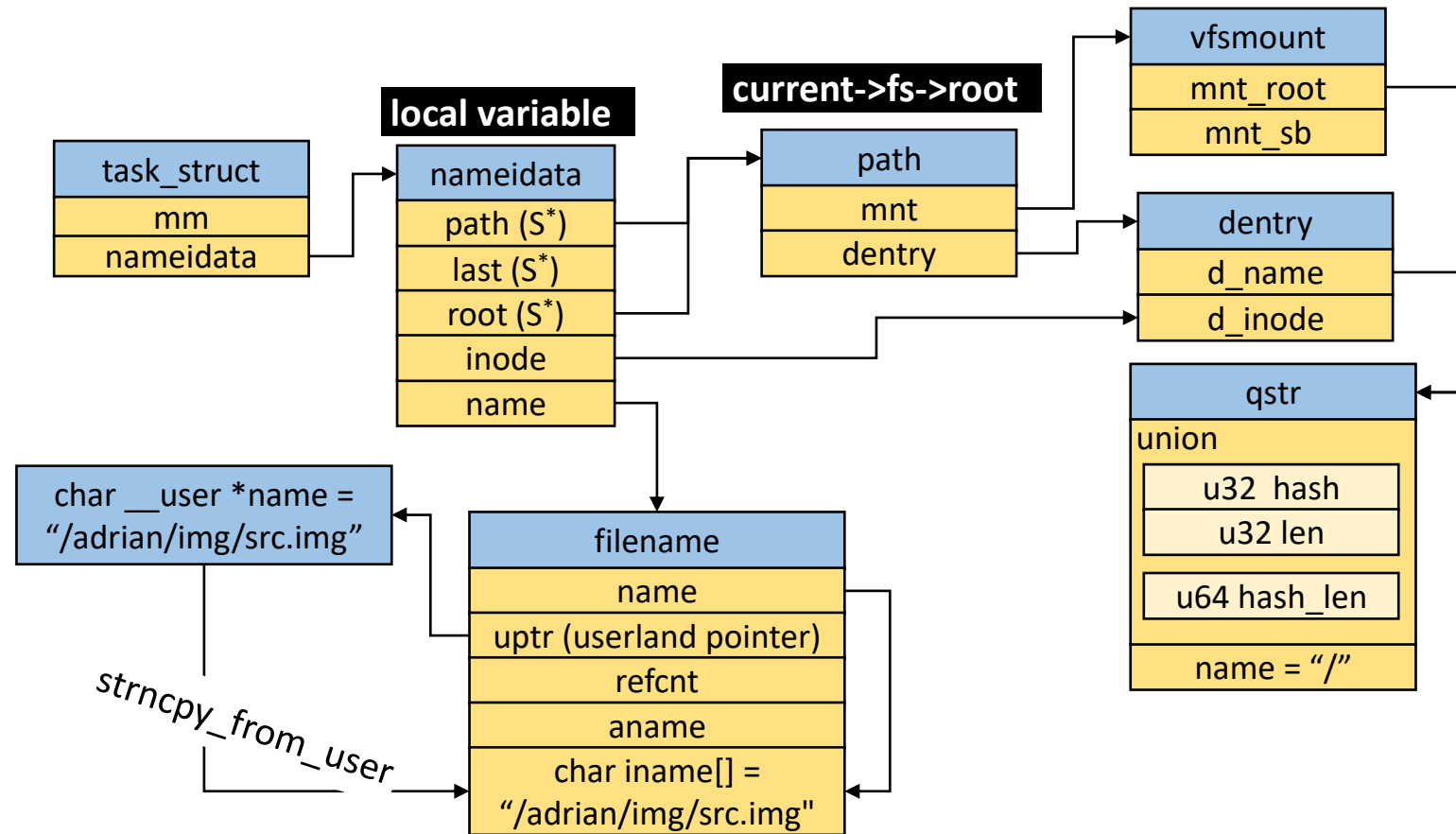


1. namei: convert a "name" to an "inode"
2. nameidata: store the current status when walking a path

stat("/adrian/img/src.img", ...)

```
user_path_at [fs/stat.c]
|- user_path_at_empty
|- struct filename *filename = getname_flags(...)
|- filename_lookup
/*
 * Fill a 'path' struct after finding a file.
 * - Prepare a nameidata struct 'nd'
 */
|- set_nameidata
|- path_lookupat

path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
Case 1. Absolute pathname
|- nd_jump_root
  if (!nd->root.mnt) {
    |- set_root
      nd->root = current->fs->root
  }
  nd->path = nd->root
  d = nd->path.dentry
  nd->inode = d->d_inode
Case 2. Relative pathname
nd->path = current->fs->pwd
nd->inode = nd->path.dentry->d_inode
return nd->name->name;
while (!(err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
  ;
|- complete_walk
```



Note

* Data storage (not a data pointer)

path_lookupat():

1. Configure members of nameidata struct based on absolute pathname or relative pathname
2. Name resolution

link_path_walk(): name resolution – round 1

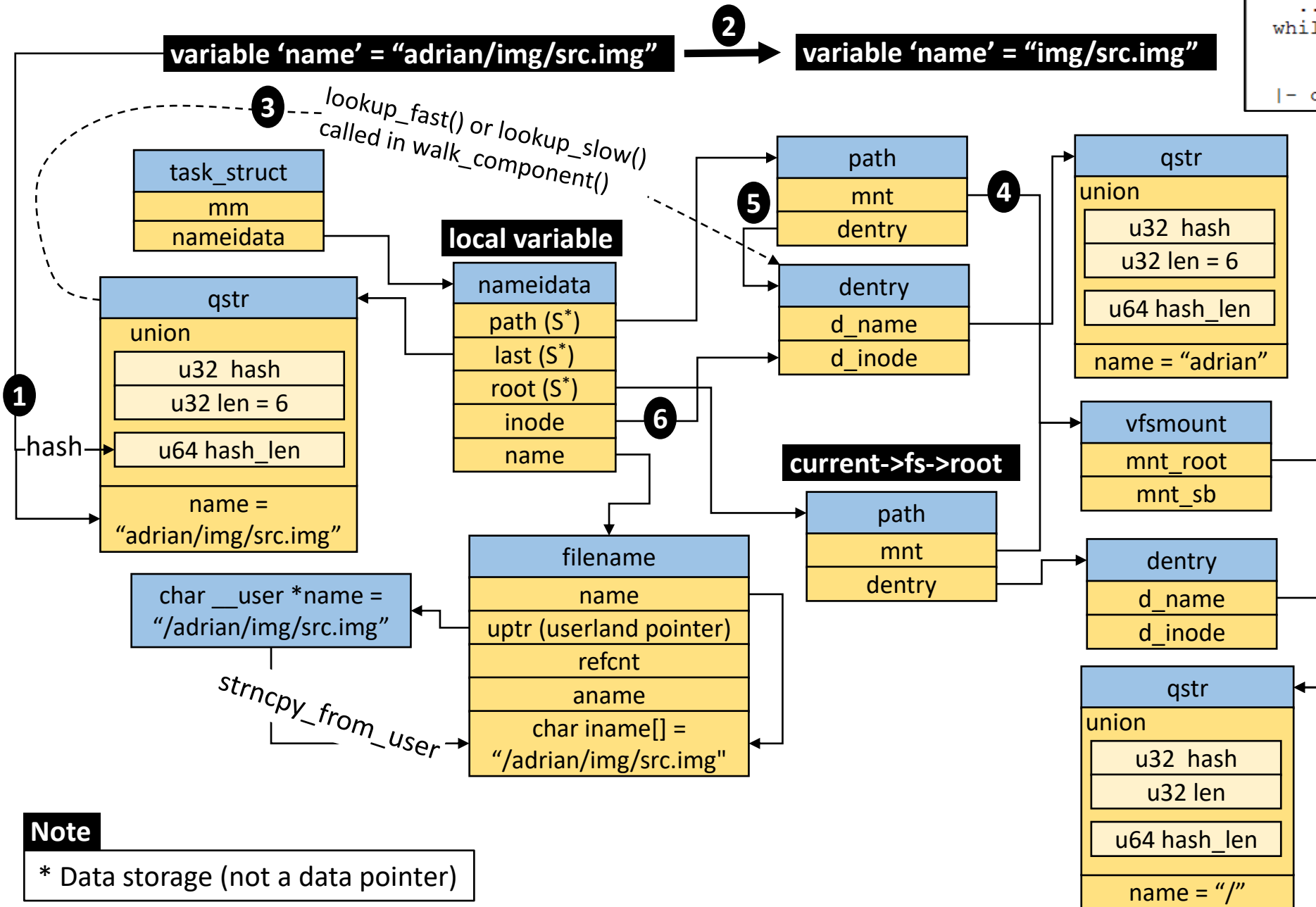
```
path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
...
while (! (err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
;
|- complete_walk
```

```
link_path_walk or lookup_last [fs/namei.c]
for(;;) {
    modify string variable 'name'
    if (!*name)
        return 0;
    |- walk_component
    |- lookup_fast
    return dentry
    |- lookup_slow if lookup_fast fails
    return dentry
    |- step_into
}

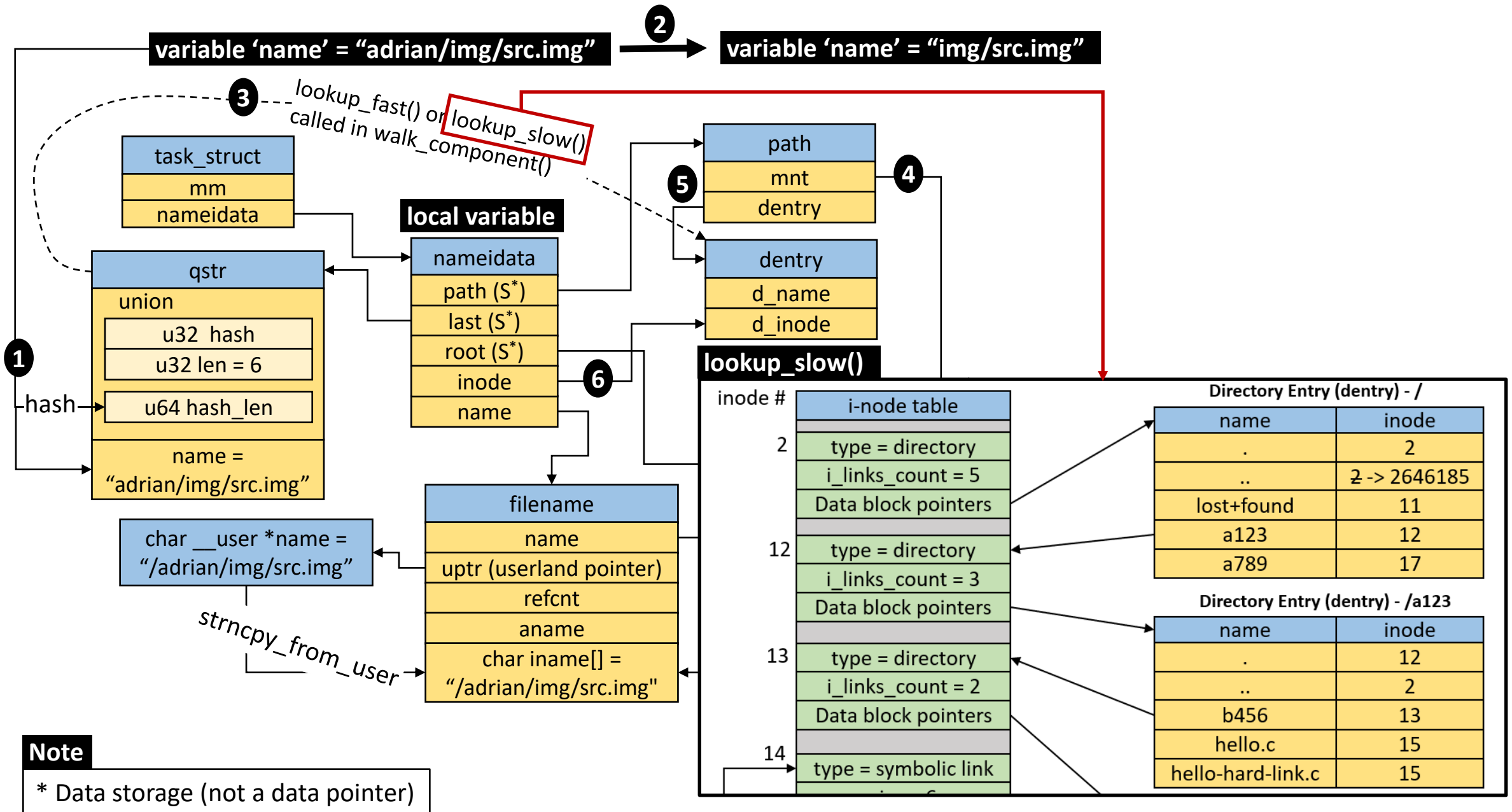
step_into [fs/namei.c]
/* Goal: Update nd->path and nd->inode */
|- handle_mounts
    path->mnt = nd->path.mnt;
    path->dentry = dentry;
    |- traverse_mounts
    if (!d_is_symlink(path.dentry)) {
        nd->path = path;
        nd->inode = inode;
        return NULL;
    }
|- pick_link
```

link_path_walk()

- stop until the last dentry is resolved
- for-loop in this function



link_path_walk(): name resolution – round 1



link_path_walk(): name resolution – round 2

```
path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
...
while (! (err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
;
|- complete_walk
```

```
link_path_walk or lookup_last [fs/namei.c]
for(;;) {
    modify string variable 'name'
    if (!*name)
        return 0;
    |- walk_component
    |- lookup_fast
    return dentry
    |- lookup_slow if lookup_fast fails
    return dentry
    |- step_into
}

step_into [fs/namei.c]
/* Goal: Update nd->path and nd->inode */
|- handle_mounts
    path->mnt = nd->path.mnt;
    path->dentry = dentry;
    |- traverse_mounts
    if (!d_is_symlink(path.dentry)) {
        nd->path = path;
        nd->inode = inode;
        return NULL;
    }
    |- pick_link
```

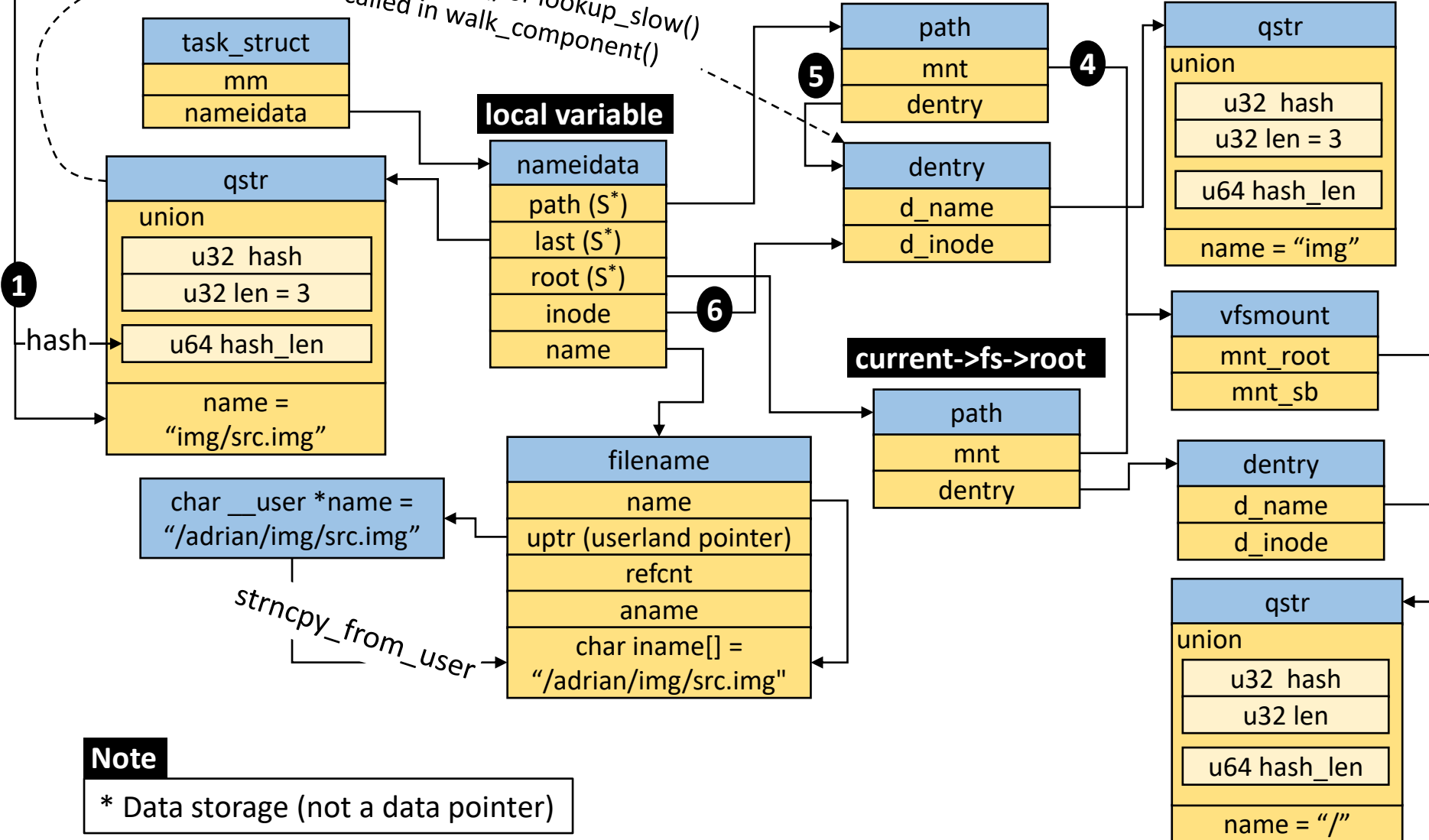
- link_path_walk()
- stop until the last dentry is resolved
 - for-loop in this function

variable 'name' = "img/src.img"

2

variable 'name' = "src.img"

3 lookup_fast() or lookup_slow() called in walk_component()



Note

* Data storage (not a data pointer)

link_path_walk(): name resolution – round 3

```
path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
...
while (! (err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
;
|- complete_walk
```

variable 'name' = "src.img"

2

variable 'name' = ""

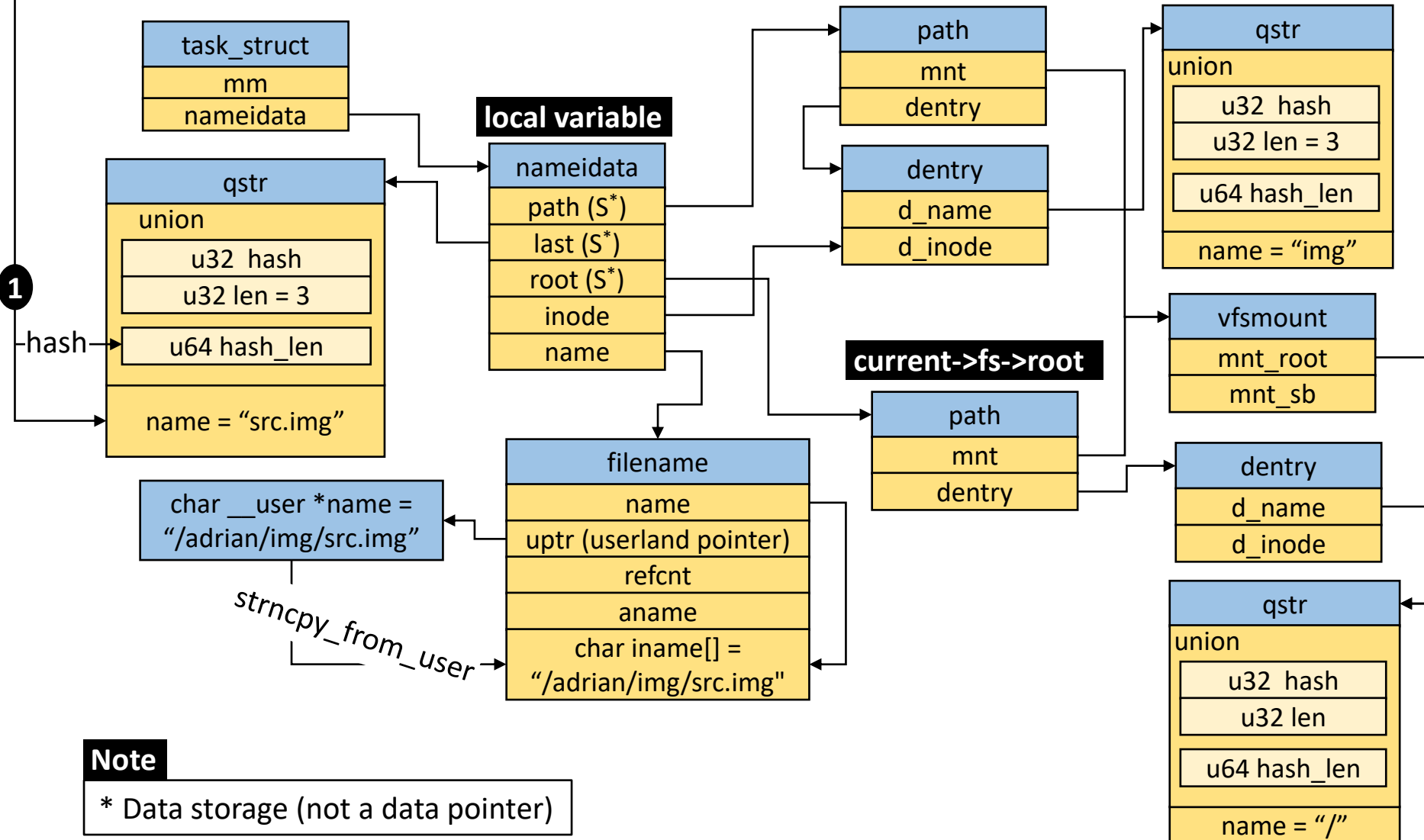
Function return

```
link_path_walk or lookup_last [fs/namei.c]
for(;;) {
    modify string variable 'name'
    if (!*name)
        return 0;
    |- walk_component
    |- lookup_fast
    return dentry
    |- lookup_slow if lookup_fast fails
    return dentry
    |- step_into
}

step_into [fs/namei.c]
/* Goal: Update nd->path and nd->inode */
|- handle_mounts
path->mnt = nd->path.mnt;
path->dentry = dentry;
|- traverse_mounts
if (!d_is_symlink(path.dentry)) {
    nd->path = path;
    nd->inode = inode;
    return NULL;
}
|- pick_link
```

link_path_walk()

- stop until the last dentry is resolved
- for-loop in this function



lookup_last(): name resolution

```
path_lookupat [fs/namei.c]
|- const char *s = path_init(nd, flags)
...
while (!(err = link_path_walk(s, nd)) &&
      (s = lookup_last(nd)) != NULL)
;
|- complete_walk
```

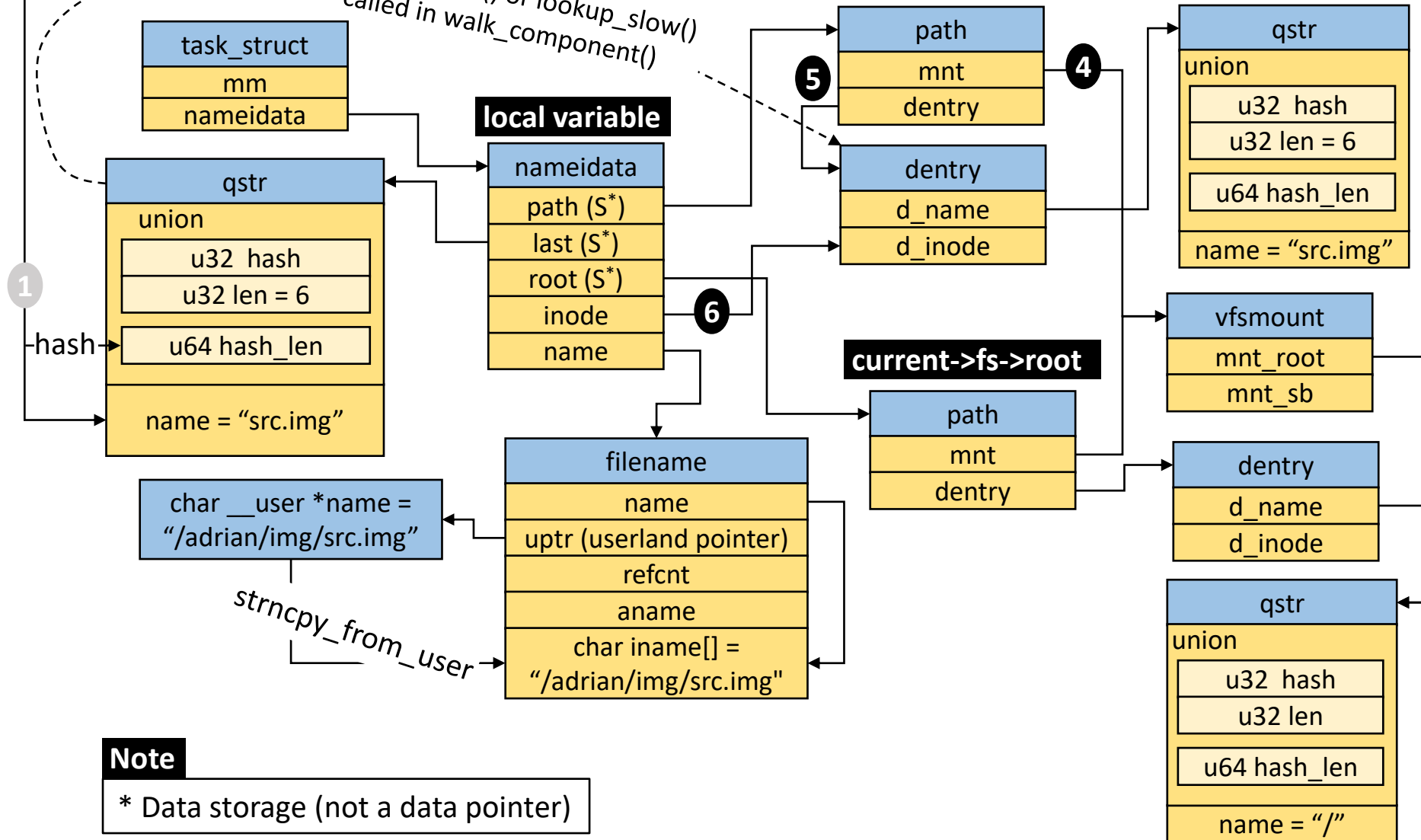
variable 'name' = "src.img"

2

variable 'name' = ""

Function return

3 lookup_fast() or lookup_slow()
called in walk_component()



```
lookup_last [fs/namei.c]
|- walk_component
|- lookup_fast
  return dentry
|- lookup_slow if lookup_fast fails
  return dentry
|- step_into

step_into [fs/namei.c]
/* Goal: Update nd->path and nd->inode */
|- handle_mounts
  path->mnt = nd->path.mnt;
  path->dentry = dentry;
|- traverse_mounts
if (!d_is_symlink(path.dentry)) {
  nd->path = path;
  nd->inode = inode;
  return NULL;
}
|- pick_link
```

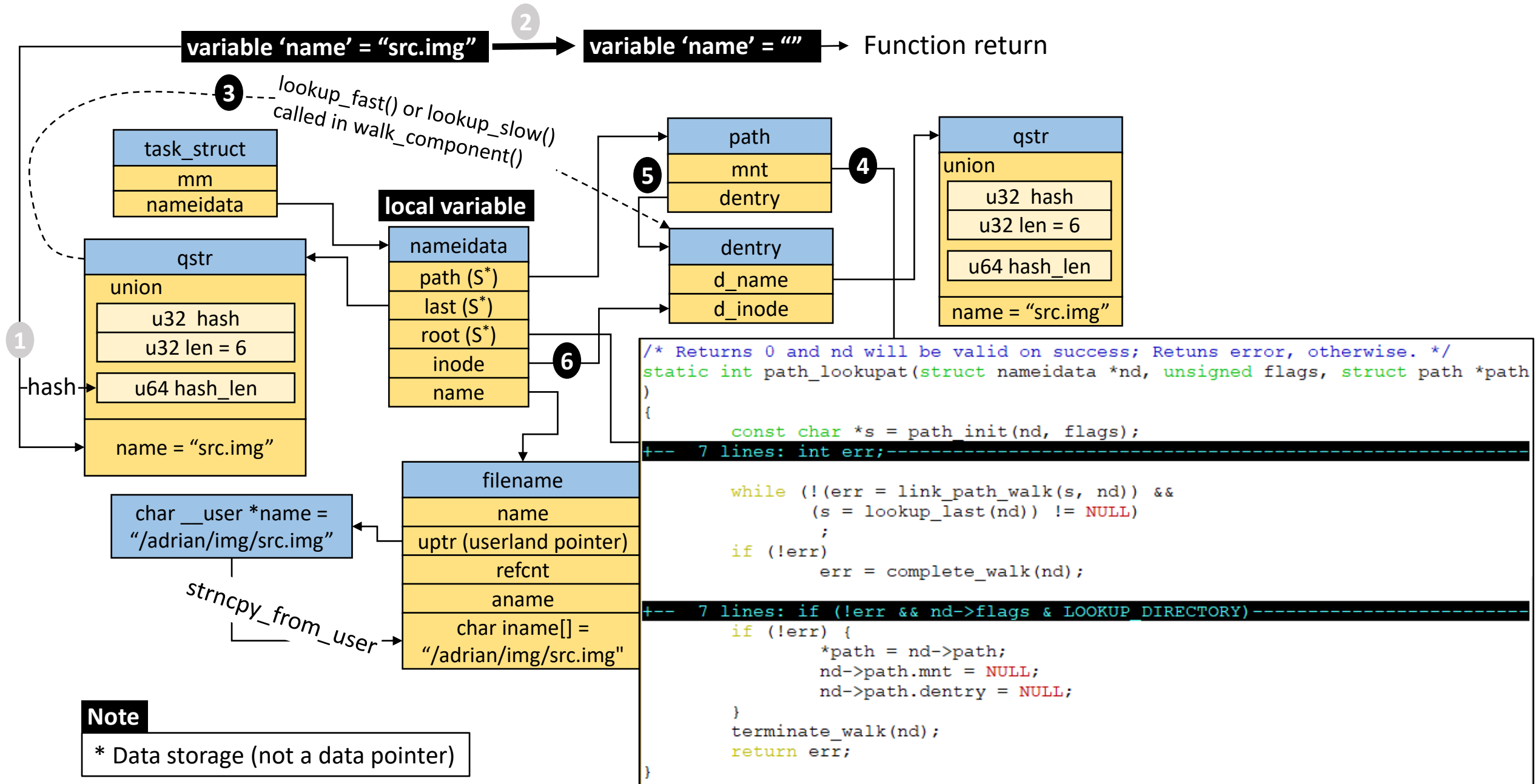
lookup_last()

- get the corresponding dentry based on the filename
- "while loop": Follow the symlink in the final component

Note

* Data storage (not a data pointer)

lookup_last(): name resolution



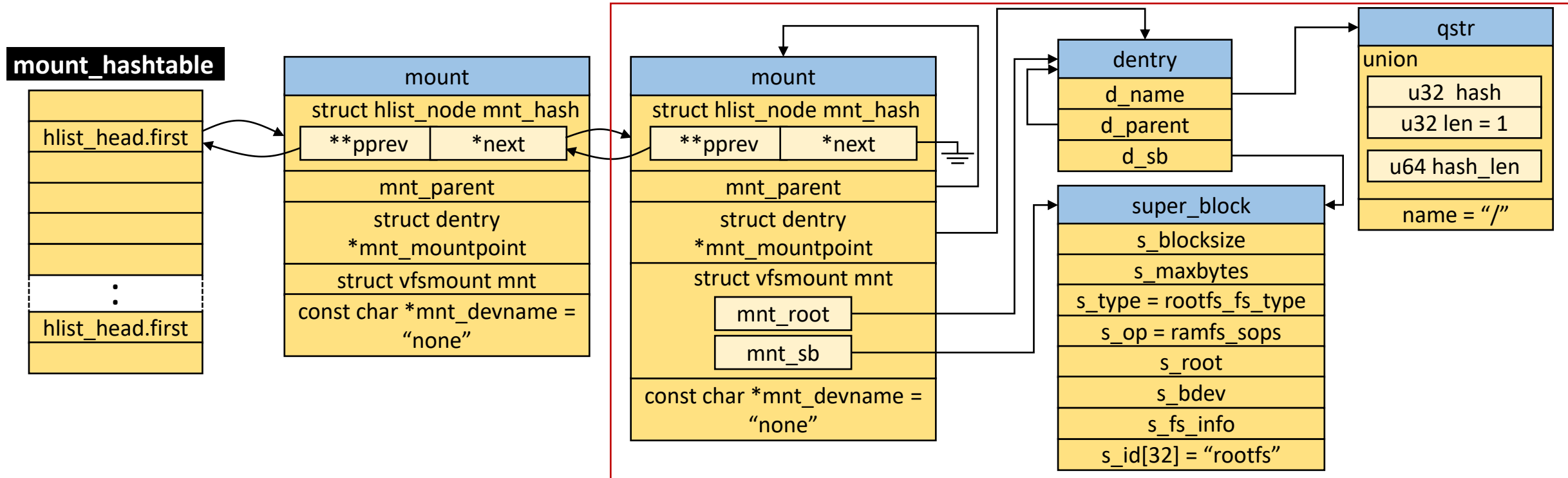
mount – mount a new ext4 file system

```
/ # strace /bin/mount /adrian/img/src.img /adrian/mnt/  
execve("/bin/mount", ["/bin/mount", "/adrian/img/src.img", "/adrian/mnt/"], 0x70  
...  
stat("/adrian/img/src.img", {st_mode=S_IFREG|0664, st_size=33554432, ...}) = 0  
...  
openat(AT_FDCWD, "/dev/loop0", O_RDWR) = 4  
...  
mount("/dev/loop0", "/adrian/mnt/", "ext4", MS_SILENT, NULL) = 0  
close(4) = 0
```

```
sys_mount [fs/namespace.c]  
|- do_mount  
  |- user_path_at(AT_FDCWD, dir_name, LOOKUP_FOLLOW, &path)  
  |- path_mount  
    |- do_new_mount  
      |- get_fs_type  
      |- vfs_get_tree  
        |- fc->ops->get_tree --> legacy_get_tree  
        /* Get the mountable root dentry. */  
        |- fc->fs_type->mount --> ext4_mount  
          |- mount_bdev  
            |- sget  
              /* Allocate/init a super_block struct. */  
            |- ext4_fill_super  
              /* Get the inode of root directory (inode #2) */  
          |- do_new_mount_fc((fc, path, mnt_flags)
```

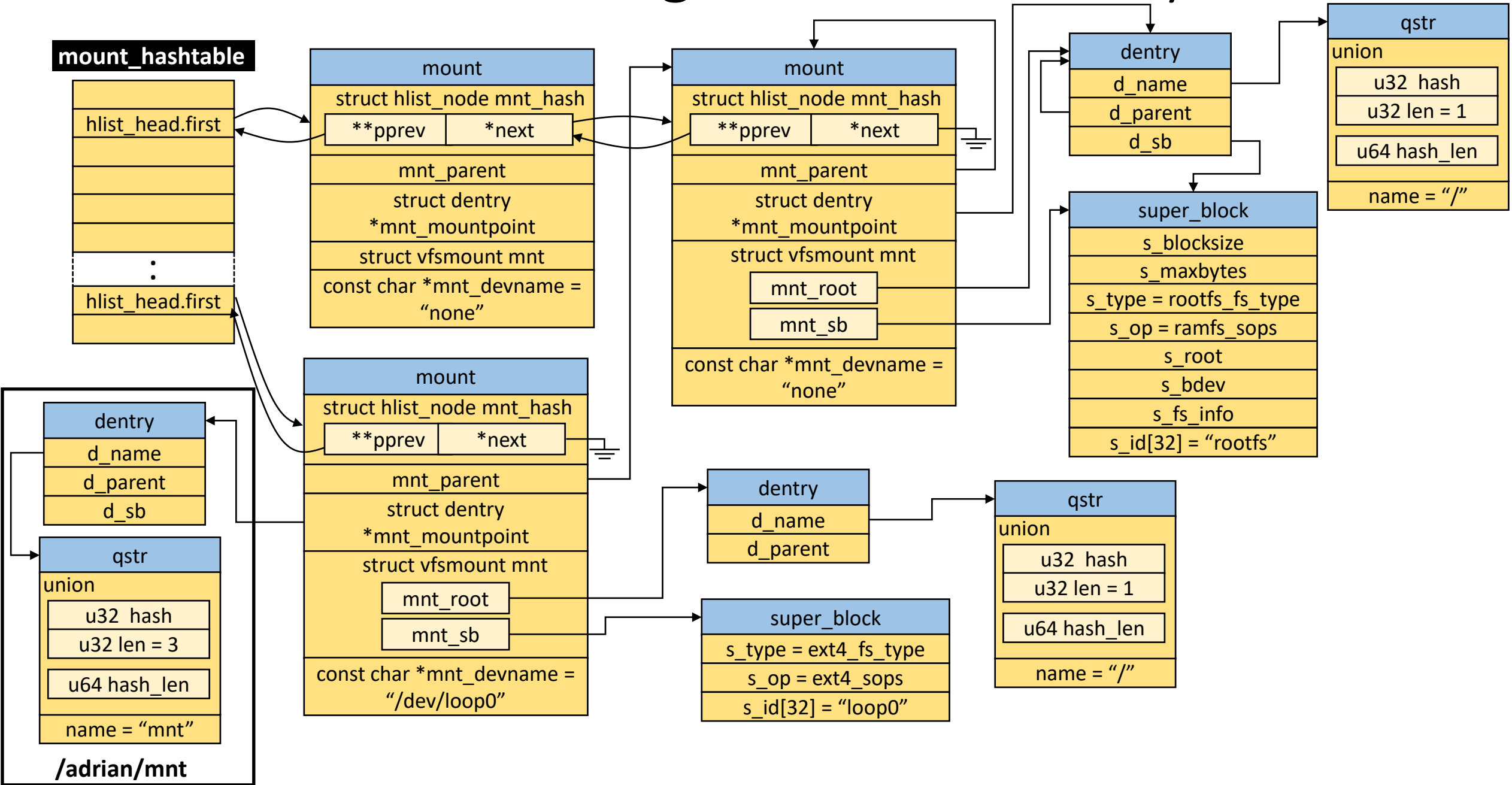

mount – before mounting a new ext4 file system

```
/ # strace /bin/mount /adrian/img/src.img /adrian/mnt/  
execve("/bin/mount", ["/bin/mount", "/adrian/img/src.img", "/adrian/mnt/"], 0x70  
...  
stat("/adrian/img/src.img", {st_mode=S_IFREG|0664, st_size=33554432, ...}) = 0  
...  
openat(AT_FDCWD, "/dev/loop0", O_RDWR) = 4  
...  
mount("/dev/loop0", "/adrian/mnt/", "ext4", MS_SILENT, NULL) = 0  
close(4) = 0
```

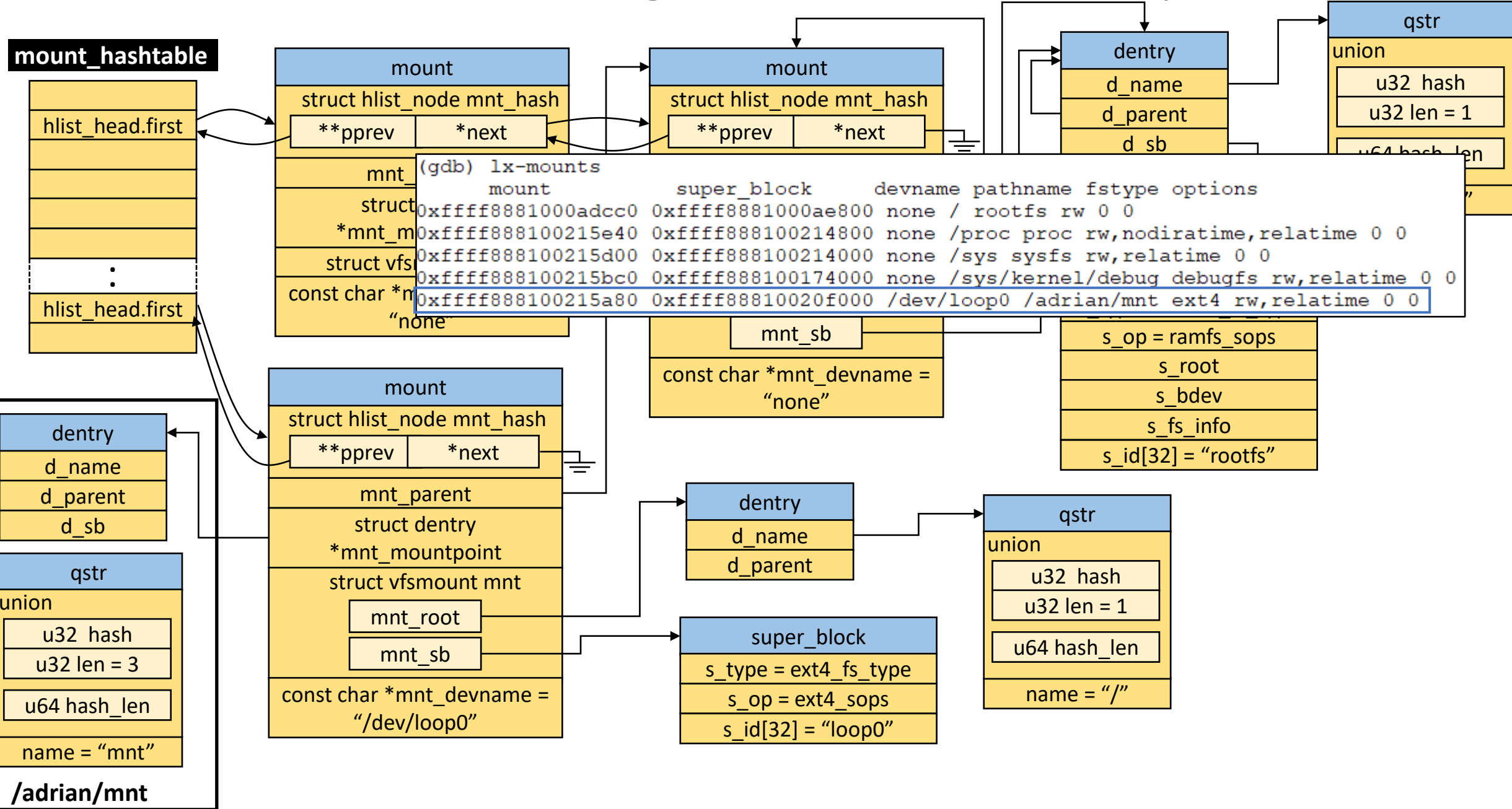


```
(gdb) lx-mounts  
mount super block devname pathname fstype options  
0xffff8881000adcc0 0xffff8881000ae800 none / rootfs rw 0 0  
0xffff888100228e40 0xffff888100227800 none /proc proc rw,nodiratime,relatime 0 0  
0xffff888100228d00 0xffff888100227000 none /sys sysfs rw,relatime 0 0  
0xffff888100228bc0 0xffff888100174000 none /sys/kernel/debug debugfs rw,relatime  
0 0
```

mount – after mounting a new ext4 file system



mount – after mounting a new ext4 file system



`ls --color mnt`

```
/ # strace /bin/ls --color /adrian/mnt
execve("/bin/ls", ["/bin/ls", "--color", "/adrian/mnt"], 0x7fffffffedf0 /* 5 variables */) = 0
...
stat("/adrian/mnt", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
openat(AT_FDCWD, "/adrian/mnt", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getdents64(3, 0x694c90 /* 5 entries */, 32768) = 152
lstat("/adrian/mnt/iperf-3.7", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat("/adrian/mnt/acpica-unix-20200925", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
lstat("/adrian/mnt/lost+found", {st_mode=S_IFDIR|0700, st_size=16384, ...}) = 0
getdents64(3, 0x694c90 /* 0 entries */, 32768) = 0
close(3) = 0
fstat(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(0x5, 0x1), ...}) = 0
ioctl(1, TCGETS, {B9600 opost isig icanon echo ...}) = 0
write(1, "\33[1;34macpica-unix-20200925\33[m " ..., 85acpica-unix-20200925 iperf-3.7
lost+found
) = 85
exit_group(0) = ?
```

Is -color mnt: openat(AT_FDCWD, "/adrian/mnt", ...)

```
sys_open or sys_openat [fs/open.c]
|- do_sys_open
  |- do_sys_openat2
    |- fd = get_unused_fd_flags(how->flags)
    |- struct file *f = do_filp_open(dfd, tmp, &op)
    |- fd_install(fd, f)
    return fd;

do_filp_open [fs/namei.c]
|- set_nameidata
|- filp = path_openat
|- restore_nameidata
return filp;

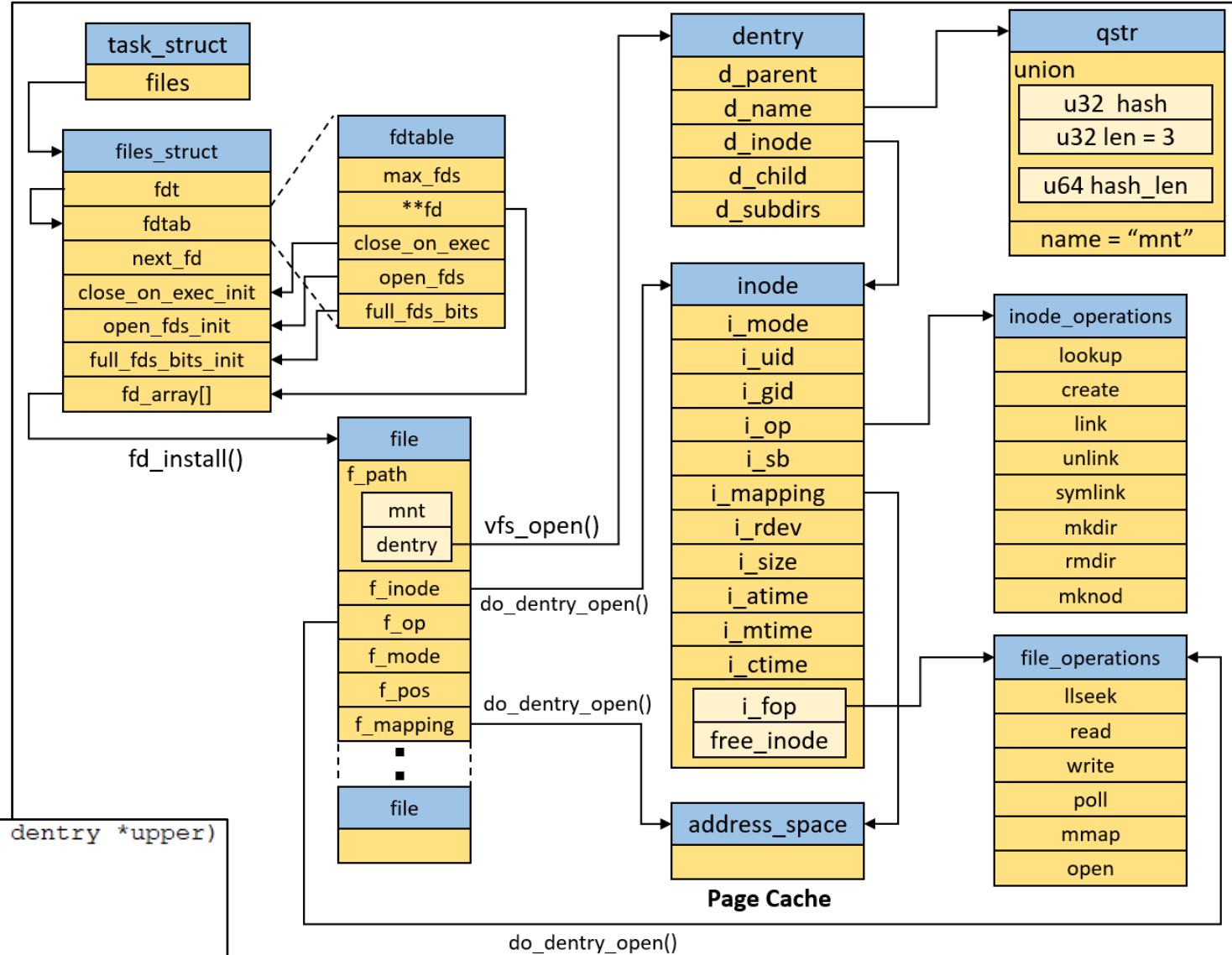
path_openat [fs/namei.c]
|- file = alloc_empty_file(...)
|- s = path_init(nd, flags)
while (!(error = link_path_walk(s, nd)) &&
      (s = open_last_lookups(nd, file, op)) != NULL)
;
|- do_open(nd, file, op)
  |- vfs_open
    struct inode *inode = d_backing_inode(path->dentry);
    file->f_path = *path
    |- do_dentry_open
      f->f_inode = inode;
      f->f_mapping = inode->i_mapping;
      f->f_op = fops_get(inode->i_fop);
      if (f->f_op->open)
        |- f->f_op->open(...)
```

```
static inline struct inode *d_backing_inode(const struct dentry *upper)
{
    struct inode *inode = upper->d_inode;

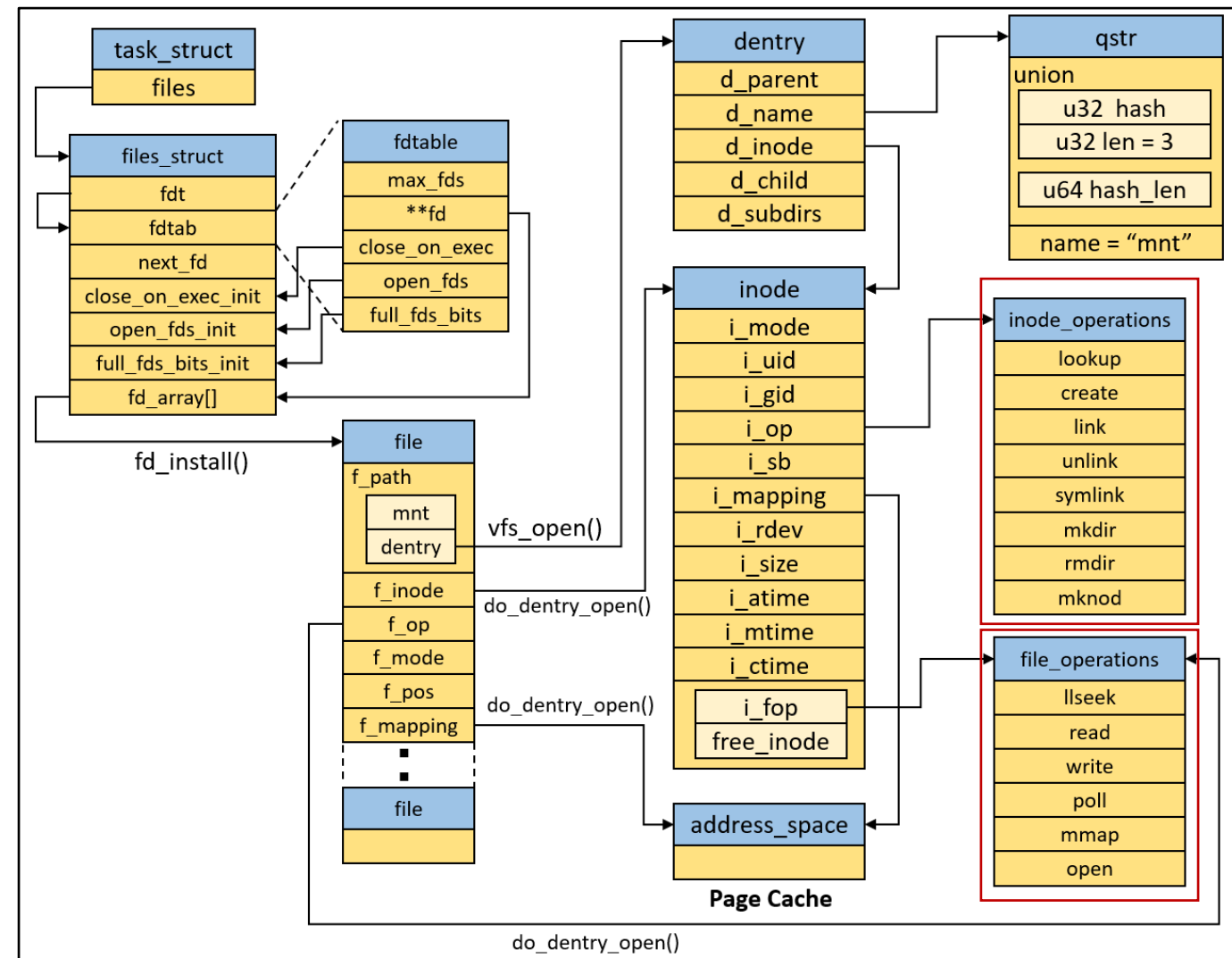
    return inode;
}
```

include/linux/dcache.h

530,1



Is -color mnt: openat(AT_FDCWD, "/adrian/mnt", ...)



```
| - do_dentry_open
  if (f->f_op->open)
    | - f->f_op->open(...)
```

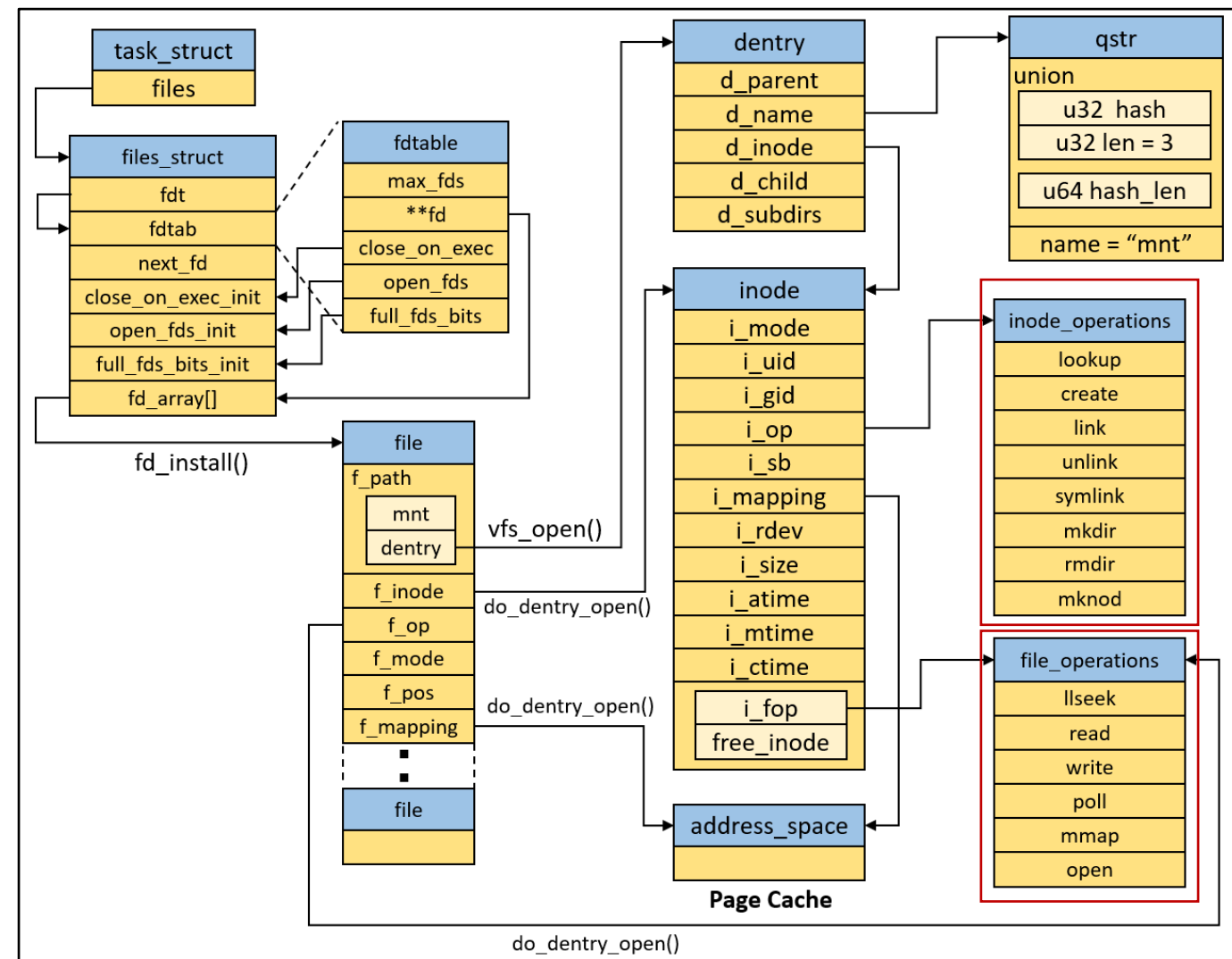
```
const struct inode_operations ext4_dir_inode_operations = {
    .create       = ext4_create,
    .lookup       = ext4_lookup,
    .link         = ext4_link,
    .unlink       = ext4_unlink,
    .symlink      = ext4_symlink,
    .mkdir        = ext4_mkdir,
    .rmdir        = ext4_rmdir,
    .mknod        = ext4_mknod,
    .tmpfile      = ext4_tmpfile,
    .rename       = ext4_rename2,
    .setattr      = ext4_setattr,
    .getattr      = ext4_getattr,
    .listxattr    = ext4_listxattr,
    .get_acl      = ext4_get_acl,
    .set_acl      = ext4_set_acl,
    .fiemap       = ext4_fiemap,
};
```

fs/ext4/namei.c

```
const struct file_operations ext4_dir_operations = {
    .llseek       = ext4_dir_llseek,
    .read         = generic_read_dir,
    .iterate_shared = ext4_readdir,
    .unlocked_ioctl = ext4_ioctl,
#ifdef CONFIG_COMPAT
    .compat_ioctl = ext4_compat_ioctl,
#endif
    .fsync        = ext4_sync_file,
    .release       = ext4_release_dir,
};
```

fs/ext4/dir.c

Is -color mnt: openat(AT_FDCWD, "/adrian/mnt", ...)



```
|- do_dentry_open
   if (f->f_op->open)
       |- f->f_op->open(...)
```

```
const struct inode_operations ext4_file_inode_operations = {
    .setattr      = ext4_setattr,
    .getattr      = ext4_file_getattr,
    .listxattr    = ext4_listxattr,
    .get_acl      = ext4_get_acl,
    .set_acl      = ext4_set_acl,
    .fiemap       = ext4_fiemap,
};
```

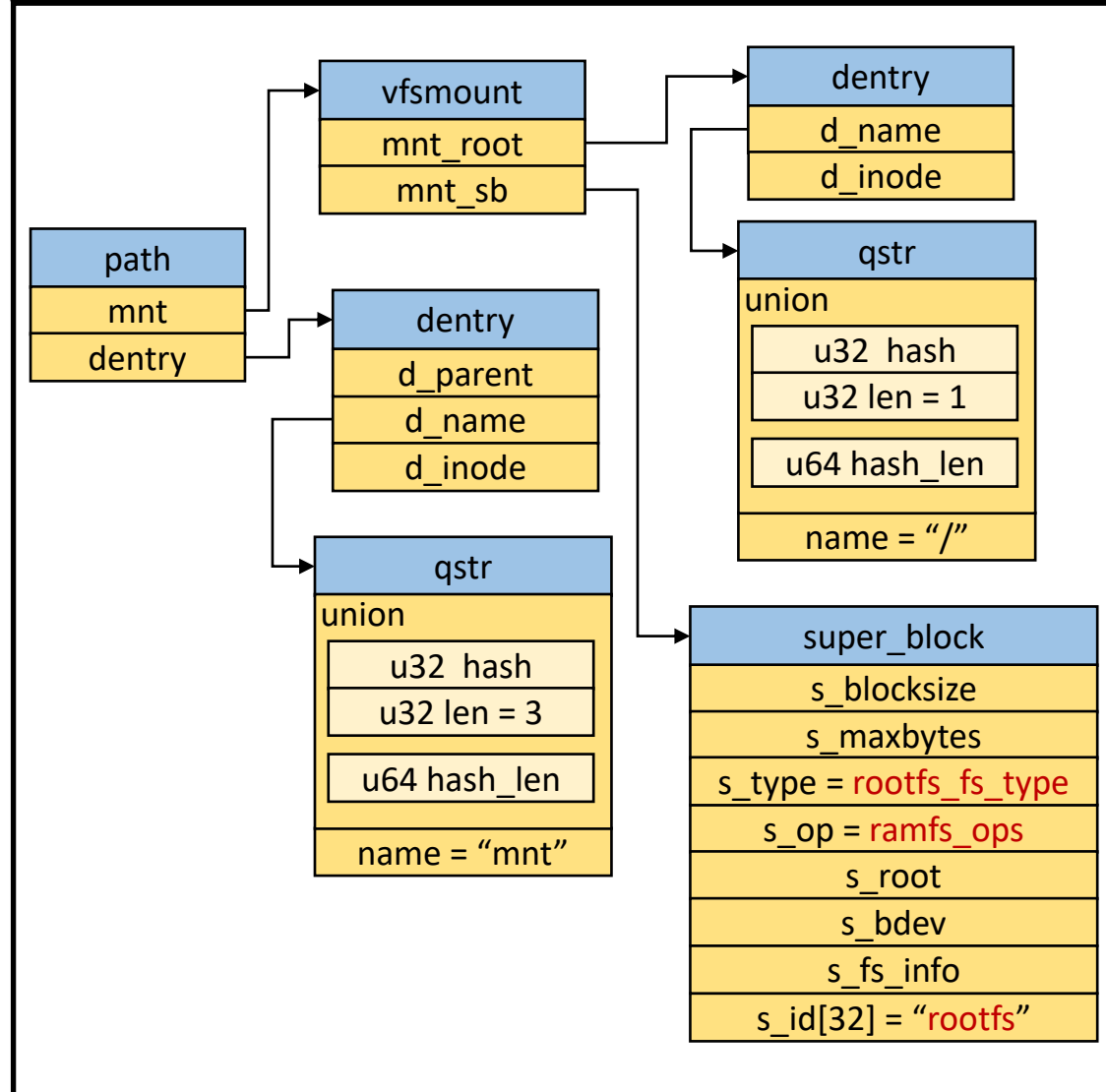
fs/ext4/file.c

```
const struct file_operations ext4_file_operations = {
    .llseek       = ext4_llseek,
    .read_iter    = ext4_file_read_iter,
    .write_iter   = ext4_file_write_iter,
    .iopoll       = iomap_dio_iopoll,
    .unlocked_ioctl = ext4_ioctl,
#ifdef CONFIG_COMPAT
    .compat_ioctl = ext4_compat_ioctl,
#endif
    .mmap         = ext4_file_mmap,
    .mmap_supported_flags = MAP_SYNC,
    .open         = ext4_file_open,
    .release      = ext4_release_file,
    .fsync        = ext4_sync_file,
    .get_unmapped_area = thp_get_unmapped_area,
    .splice_read  = generic_file_splice_read,
    .splice_write = iter_file_splice_write,
    .fallocate    = ext4_fallocate,
};
```

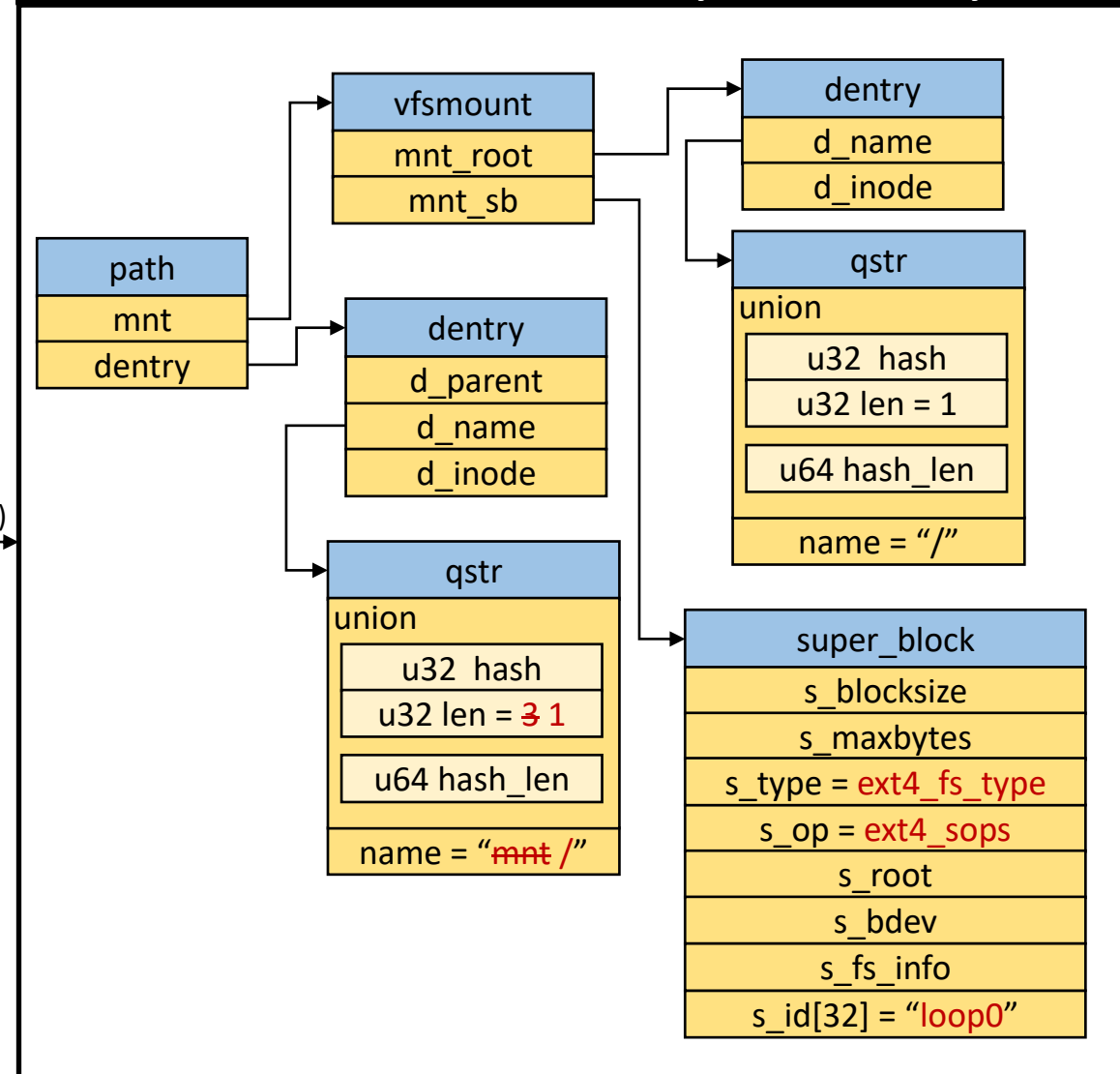
fs/ext4/file.c

`ls -color mnt`: `openat(AT_FDCWD, "/adrian/mnt", ...)`

/adrian/mnt data structures in the original file system



/adrian/mnt data structures in newly mounted file system



traverse_mounts()

The original mount point is changed the newly one when running name resolution

ls -color mnt: openat(AT_FDCWD, "/adrian/mnt", ...)

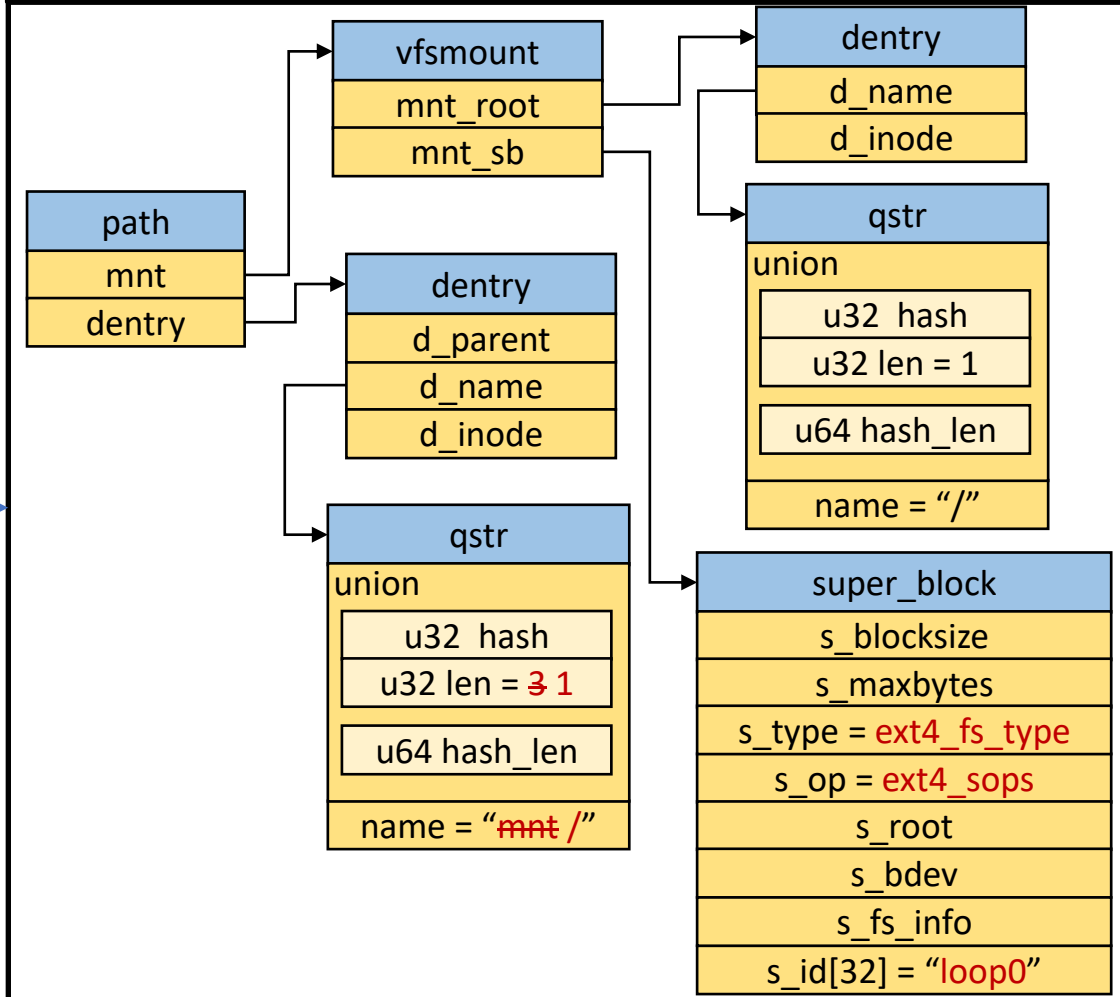
```
link_path_walk or lookup_last [fs/namei.c]
|- walk_component
|- lookup_fast
  return dentry
|- lookup_slow if lookup_fast fails
  return dentry
|- step_into
```

```
step_into [fs/namei.c]
/* Goal: Update nd->path and nd->inode */
|- handle_mounts
  path->mnt = nd->path.mnt;
  path->dentry = dentry;
|- traverse_mounts
  if (!d_is_symlink(path.dentry)) {
    nd->path = path;
    nd->inode = inode;
    return NULL;
  }
|- pick_link
```

```
traverse_mounts [fs/namei.c]
|- __traverse_mounts
  while (flags & DCACHE_MANAGED_DENTRY) {
    if (flags & DCACHE_MOUNTED) {
      struct vfsmount *mounted = lookup_mnt(path);
      if (mounted) {
        path->mnt = mounted;
        path->dentry = dget(mounted->mnt_root);
      }
    }
    flags: path->dentry->d_flags
  }
}
```

REF-walk

/adrian/mnt data structures in newly mounted file system



[Two concurrent mechanisms for name resolution]

1. REF-walk: concurrent management with refcounts and spinlock
2. RCU-walk: faster name resolution lookup