CSCI-3753: Operating Systems Fall 2019

Abigail Fernandes

Department of Computer Science

University of Colorado Boulder

Announcements

- PA4 Interview Grading Scheduler
- Come prepared for the interview

Week 13
> Virtual File System
> PA4 Workday

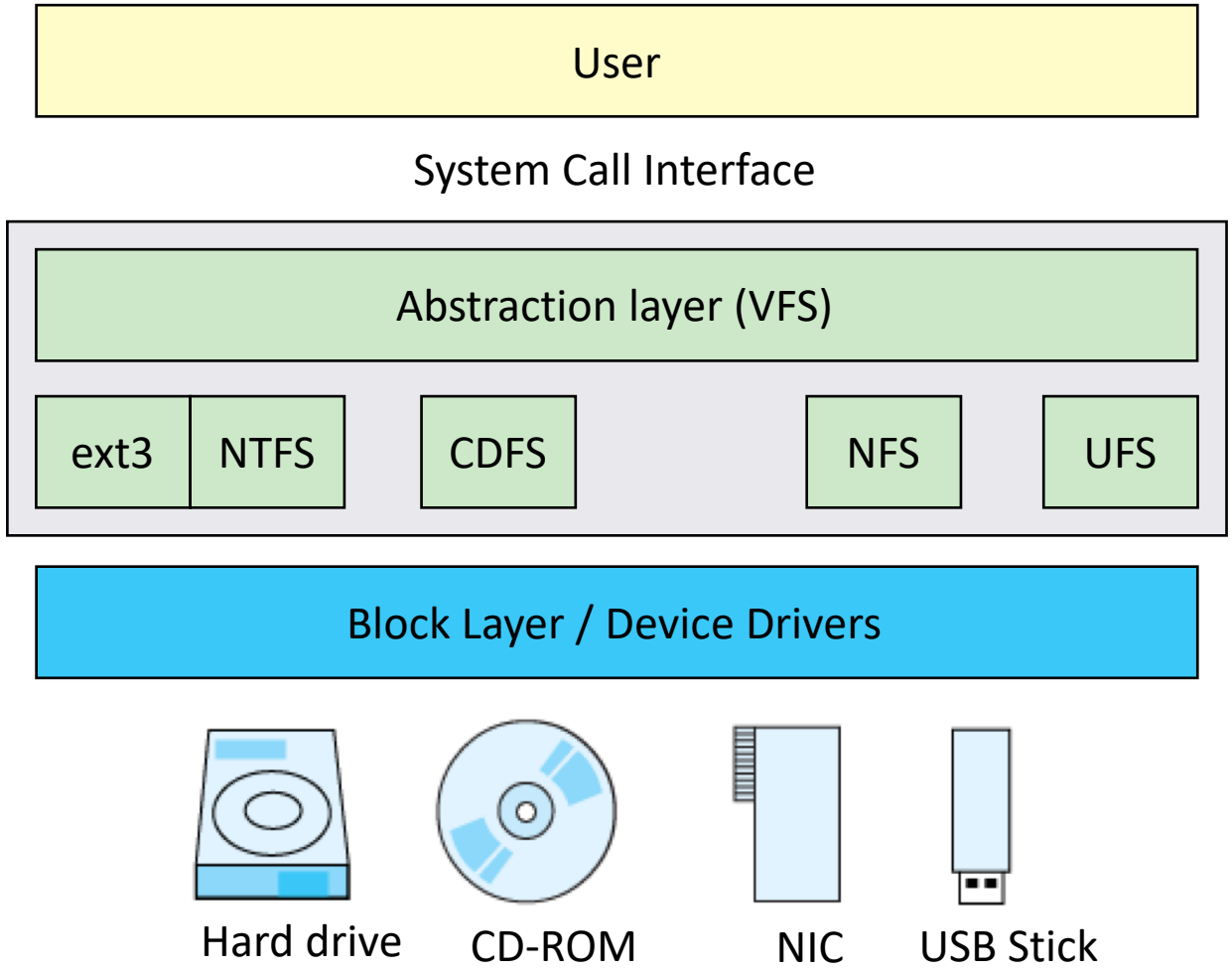
File System

- Control how data is stored and retrieved
- Types of file system
- **Local data storage**
  - FAT16, ext2, ext3, ext4
  - ISO 9660
  - CDFS
  - Removable USB flash drive (UFD)
- **Network data storage**
  - NFS
  - Plan 9

Check **what filesystems** are on your machine?
→ `df -T`

Check out the **partitioning table** on your machine?
→ `sudo parted -l`

File System


User
System Call Interface
Abstraction layer (VFS)
ext3
NTFS
CDFS
NFS
UFS
Block Layer / Device Drivers
Hard drive
CD-ROM
NIC
USB Stick

Virtual File System

- An abstraction layer on top of a more concrete file system
- **Purpose**
  - To allow client applications to access different types of concrete file systems in a uniform way
  - To manage all of the different file systems that are mounted at any given time
- **Method**
  - Provide a set of standard interfaces for upper-layer applications to perform file I/O over a diverse set of file systems
  - Describe the system's files in terms of superblocks and inodes

VFS Objects

- Four primary object types
  1. **Superblock**
     - Represents a specific mounted file system
  2. **Inode**
     - Represents a specific file
  3. **Dentry**
     - Represents a directory entry, single component of a path name
  4. **File**
     - Represents an open file as associated with a process

VFS Auxiliary Objects

- **Filesystem types**
  - Used to connect the name of the filesystem to the routines for setting it up (at mount time) or tearing it down (at umount time).
- **A *struct vfsmount***
  - Represents a subtree in the big file hierarchy - basically a pair (device, mountpoint).
- **A *struct nameidata***
  - Represents the result of a lookup.
- **A *struct address_space***
  - Gives the mapping between the blocks in a file and blocks on disk. It is needed for I/O.

Superblock object

- A container for essentially high-level metadata about a file system
- A critical structure that
  - Exists on disk and also in memory
  - Is stored in multiple redundant copies for each file system
- The **VFS Superblock** contains the following information:
  - Device
  - Inode pointers
  - Blocksize
  - Superblock operations
  - File System type
  - File System specific

Inode Object

- Represents all the information needed to manipulate a file or directory
- Constructed in memory, regardless of how file system stores metadata information
- Contains information:
  - File type - regular file, directory, character device, etc
  - Owner
  - Group
  - Access permissions
  - Timestamps - mtime (time of last file modification), ctime (time of last attribute change), atime (time of last access)
  - Number of hardlinks to the file
  - Size of the file
  - Number of blocks allocated to the file
  - Pointers to the data blocks of the file - most important!

Inode Object

- When are inodes created?
  - When a filesystem is created, the space for inodes is allocated as well.
  - Determining how much inode space needed depends on the volume of the disk and more.
  - Rare but possible: Errors for out of inodes !!!

  → Unable to create more files

Check how many inodes are left on your machine?

→`df -i`

Inode Object

- Inode Information
  - Inodes are identified by numbers
  - On file creation, it is assigned an inode number

View inode number
→ `ls -li`

```
pete@icebox:~$ ls -li
140 drwxr-xr-x 2 pete pete 6 Jan 20 20:13 Desktop
141 drwxr-xr-x 2 pete pete 6 Jan 20 20:01 Documents
```

Inode

- You can get detailed information of a file with stat which gives you information about the inode

```
pete@icebox:~$ stat ~/Desktop/
  File: '/home/pete/Desktop/'
  Size: 6               Blocks: 0          IO Block: 4096   directory
Device: 806h/2054d      Inode: 140         Links: 2
Access: (0755/drwxr-xr-x)  Uid: ( 1000/   pete)   Gid: ( 1000/   pete
Access: 2016-01-20 20:13:50.647435982 -0800
Modify: 2016-01-20 20:13:06.191675843 -0800
Change: 2016-01-20 20:13:06.191675843 -0800
 Birth: -
```

Inodes

- How do inodes locate files?

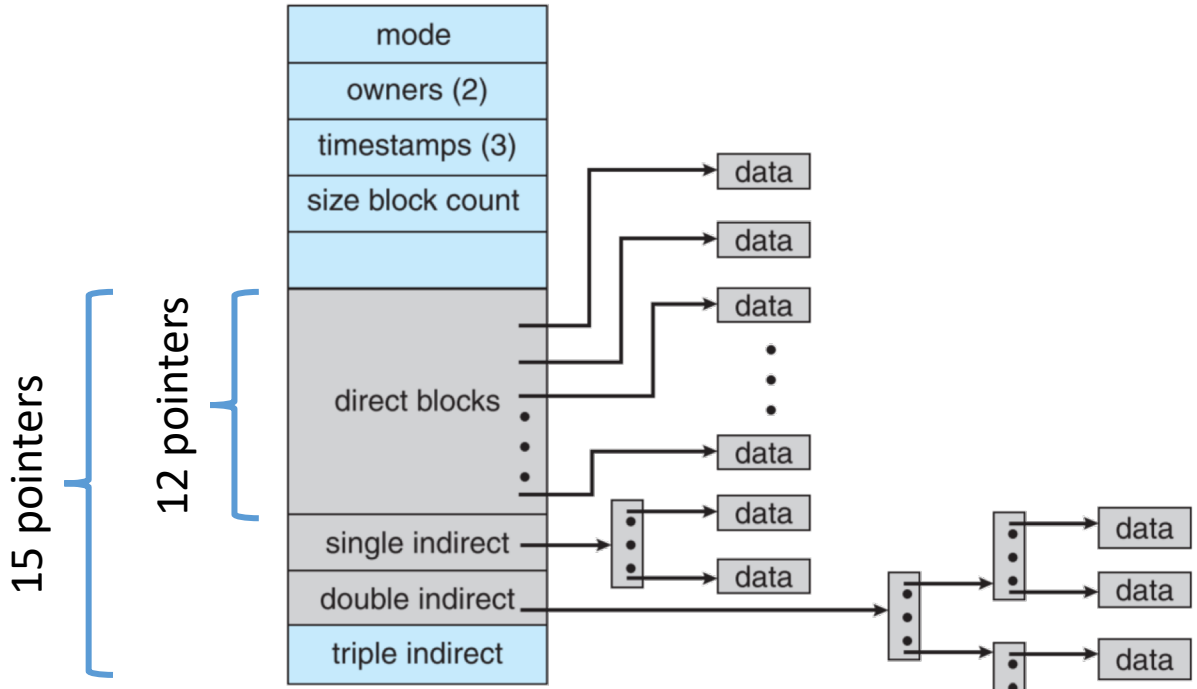


Illustration of a file system

Dentry

- A glue that holds inodes and files together by relating inode numbers to file names
- Also play a role in directory caching which, ideally, keeps the most frequently used files on-hand for faster access.
- Maintain a relationship between directories and their files for file system traversal
- The dentry objects exist only in file system memory and are not stored on disk as they are used to improve performance only.

Dentry

- A file system will have one root dentry
  - → Superblock
  - → The only dentry without a parent.
- All other dentries have parents, and some have children.

How many dentry objects are created if the following file is opened?

`/home/user/name`

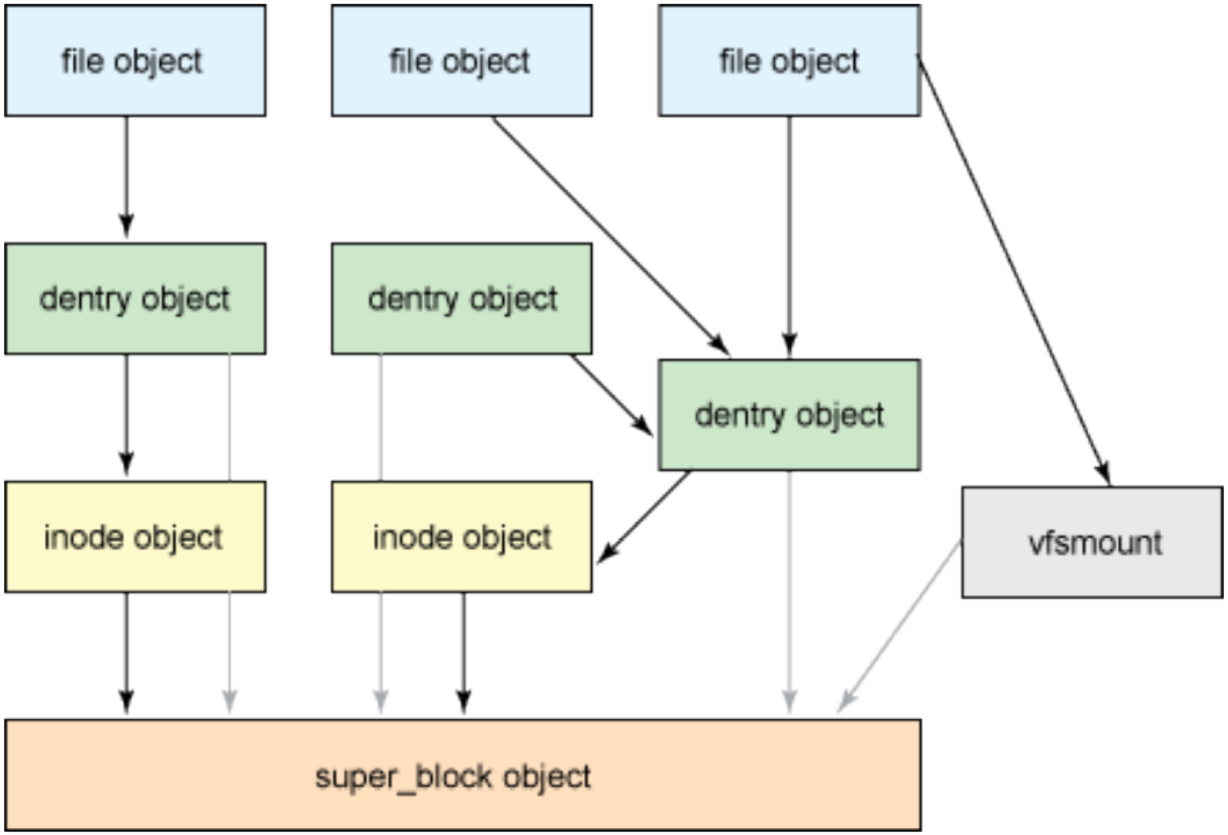
File Object

- For each opened file in a Linux system, a file object exists.
- Contains information specific to the open instance for a given user
  - Where the file is stored
  - What processes are using it
- Thrown away when the file is closed

struct file

```
struct path f_path;
struct dentry (f_path.dentry);
const struct file_operations *f_op;
unsigned int f_flags;
fmode_t f_mode;
lodd_t f_pos;
...
```

Object Relationship

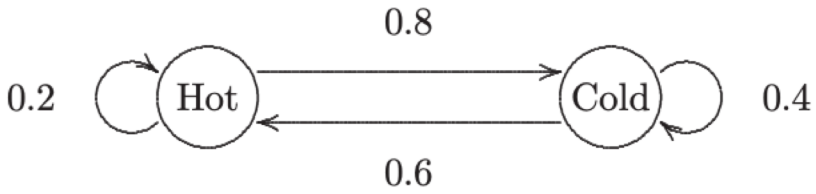


Week 13
> Virtual File System
> **PA4 Workday**

Trouble with Pager-Predict?

Familiar with Transitions?

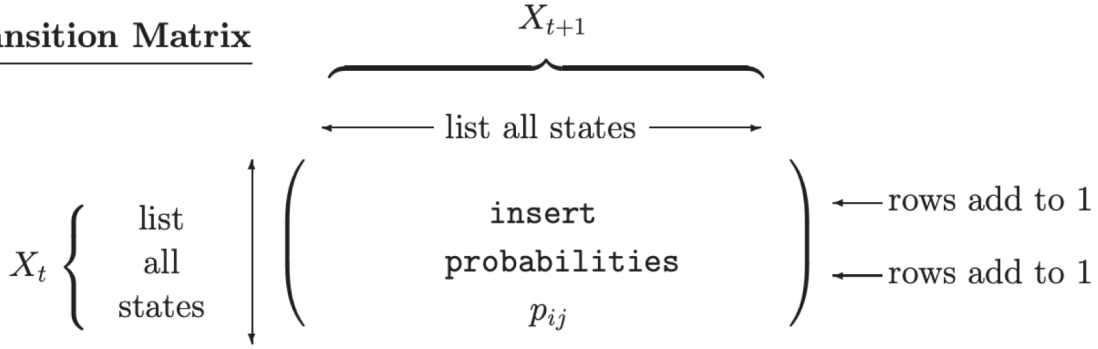
- State Transition Matrix or Transition Probability Matrix
- Example of a transition diagram to describe Markov Chains.
- The transition diagram is so called because it shows transitions between different states.
- The matrix describing the Markov chain is called the transition matrix.



We can also summarize the probabilities in a **matrix:**

$$X_t \begin{Bmatrix} \text{Hot} \\ \text{Cold} \end{Bmatrix} \overbrace{\begin{pmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \end{pmatrix}}^{X_{t+1}} \quad \begin{matrix} \text{Hot} & \text{Cold} \end{matrix}$$

Transition Matrix



The transition matrix is usually given the symbol $P = (p_{ij})$.

In the transition matrix P:

- *the ROWS represent NOW, or FROM (X_t);*
- *the COLUMNS represent NEXT, or TO (X_{t+1});*
- *entry (i, j) is the CONDITIONAL probability that NEXT $= j$, given that NOW $= i$: the probability of going FROM state i TO state j.*

$$p_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i).$$