
Project 3:

An Introduction to File Systems

COP 4610 / CGS 5765

Principles of Operating Systems

Overview

- Email me your new team information
- Project write-up
 - List of required operations
- FAT32 specification
 - Locations of information to traverse filesystem

Overview

- Similar interface as shell program
- Do not have to support absolute or relative paths
 - `cd <dir>` only needs to search the current working directory

Extracting Integers from Image

- Required information
 - Location of data
 - Size of data
 - Type of data (e.g., signed vs unsigned)
 - Endianness
 - All fat32 integers are little-endian (same as x86)

Coding

■ Datatypes

- ❑ `#include <stdint>`
- ❑ `uint8_t, uint16_t, ...`

■ **mmap**

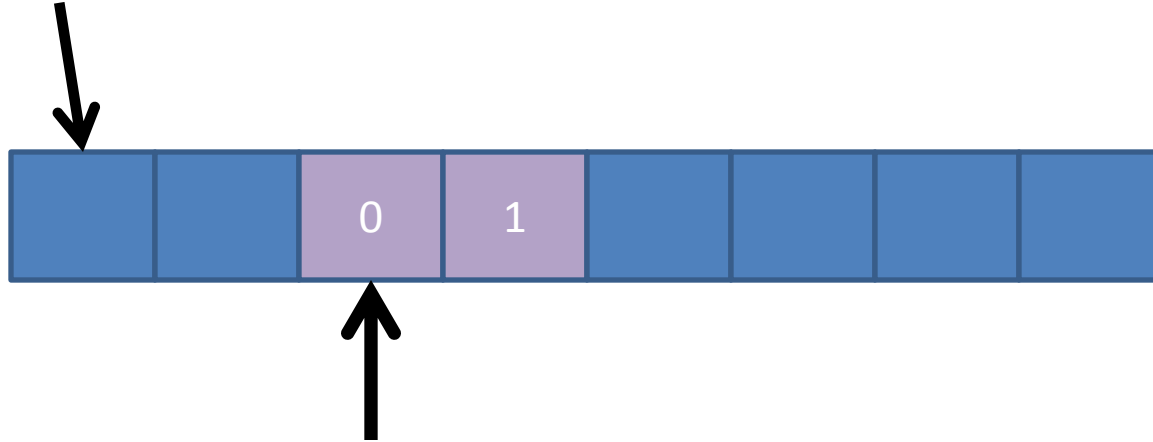
- ❑ http://pubs.opengroup.org/onlinepubs/9699919799/functions/mmap.html#tag_16_332

■ Endianness

- ❑ Swap bytes necessary?

Little Endian Source

`mmap(img)`



$$\text{uint16_t} = 0$$

$$\text{uint16_t} | = 1 \ll 8$$

Defined Behavior

- Make sure that your code uses defined behavior
 - Understand the language
- Previous example
 - Appropriate conversions (e.g., casts)
 - Shift operations on appropriate types
 - <https://www.securecoding.cert.org/confluence/display/cplusplus/VOID+INT34-CPP.+Do+not+shift+a+negative+number+of+bits+or+more+bits+than+exist+in+the+operand>

Parse Boot Sector

- Bytes per sector
- Sector per cluster
- Number of FATs
- ...