Project 3: Utilities

COP 4610 / CGS 5765

Principles of Operating Systems

FAT

- Fat entries are 32-bits
  - Next cluster number uses the low 28-bits
    - Free cluster
      - `0x0000000`
    - Reserved cluster
      - `0x0000001`
    - Used cluster (value points to next cluster)
      - `0x0000002 – 0xFFFFFEF`
    - Reserved
      - `0xFFFFFF0 – 0xFFFFFF6`
    - “BAD CLUSTER”
      - `0xFFFFFF7`
    - End of Clusterchain (EOC) mark
      - `>= 0xFFFFFF8`
  - High 4-bits of entry are reserved for other purposes and should not be changed (i.e., preserved when updating FAT)

Multiple Clusters

- Does the root directory span more than one cluster?
  - Locate next cluster number in the FAT
  - Attempt to find ***`end of directory`*** record

File With No Data

- Zero-length file
  - cluster number in record set to zero
  - cluster[0] and cluster[1] do not exist

Starting FAT32 Utility

```
$>./fmod fat32.img
[fat32.img]>
```

Handling Open Files

- Read/write require file to be open
- Maintain a table of files that open files

Opening Files

```
[fat32.img]> open fatinfo.txt rw
```

- If “fatinfo.txt” is found in the current directory, open “fatinfo.txt”
  - Find fatinfo.txt by parsing the current directory and determine whether fatinfo.txt is part of that directory
- Once you open fatinfo.txt, record it as open
  - E.g., Store its name, permissions, …, in an open file table

`read` command

```
[fat32.img]> read fatinfo.txt 0 100
```

- Only allow the read if fatinfo.txt is open
- To read:
  - Find file in open file table
  - Find data location
  - Read enough of the file's data cluster(s) to satisfy the read request
    - **Note** file may be smaller than size of read request

`write` command

```
[fat32.img]> write fatinfo.txt 0 “Hello”
```

- If write stays within the cluster
  - Just write data
- If write goes beyond cluster
  - Find a free cluster, remember as next_cluster_number
  - Change FAT[current_cluster] from EoC to next_cluster_number
  - Change FAT[next_cluster_number] to EoC
  - Write the data in the cluster next_cluster_number
  - If there's still more to write, repeat from 1.

`close` command

```
[fat32.img]> close fatinfo.txt
```

- Remove file's entry from the open file table

Coding

- Parse the boot directory
  - Cannot go anywhere without this code
- Read directories
- Open and close files
- Read files
- Once you can read directories, implementing **`ls`** and **`size`** should be easy
- Writing to files
- Create directories

Writing Considerations

- Update ***ALL*** FATS
  - Likely will have 2, but may have more or less