

Project #3: FUSE

Istanbul Technical University,
Faculty of Computer and Informatics
BLG413E- System Programming
2020-2021 Fall

Implement a file system using FUSE for displaying the contents of a JSON file. An example JSON file is going to look like this:

```
{
  "class": {
    "operatingsystem": {
      "students": {
        "student1": "111111111",
        "student2": "232323232",
        "student3": "323232323"
      }
    },
    "systemprogramming": {
      "students": {
        "student1": "111111111",
        "student3": "323232323",
        "student4": "444444444"
      },
      "teachers": {
        "gokhansecinti": "123456789",
        "turgutuyar": "2345678"
      }
    }
  },
  "club": {
    "basketball": {
      "students": {
        "student1": "111111111"
      }
    }
  }
}
```

Do not assume student1 in systemprogramming and student1 in basketball is the same, each entry should be considered individually while reading the JSON file.

The JSON file will be mounted as a directory. If a key contains a value only (example: "student1": "111111111"), it will be displayed as a file with filename as the key and contents as the value. If a key contains child object(s), it will be displayed as a directory. Your filesystem should make mounted directory look like:

```

mount_directory
/class
  /operatingsystem
    /students
      student1
      student2
      student3
    /systemprogramming
      /students
        student1
        student3
        student4
      /teachers
        gokhansecinti
        turgutuyar
  /club
    /basketball
      /students
        student1

```

Example file contents:

```

cat mount_directory/class/operatingsystem/student1 , prints 111111111
cat mount_directory/class/systemprogramming/turgutuyar , prints 2345678
cat mount_directory/club/basketball/students/student1 , prints 111111111

```

Implementation Details:

If a file is deleted from the virtual file system, the corresponding entry has to be deleted from the JSON file.

A new file cannot be manually added to the virtual file system. The new entry has to be made in the JSON file.

The contents of the files in the virtual file system cannot be modified. Any modification should be done on the JSON file.

Your filesystem should display the attributes correctly (whether it is a directory or not, size of a file should match according to its content...).

BONUS: If a new file is added to the virtual file system, the contents of the file will be added as a new entry to the JSON file in the appropriate location.

Parsing the JSON File

For parsing the JSON file, you can use cJSON library (<https://github.com/DaveGamble/cJSON>), or parse the file yourself. If you choose to parse file yourself, file given to you will be in the same format as the example JSON file and you can assume below:

- The file given to you will always be valid, so you do not need to check for errors and make corrections in the file.
- File will always start with a { and end with a }.
- Each key will be in separate lines.
- If a key is followed by a { character in the same line, it is a directory.
- If a key is followed by a value right after it, it is a file.

Submission Details

Deadline: Jan 18th, 2021

- Every group member is required to submit source code file(s) through the Ninova system as a zip file.
- Any form of cheating or plagiarism will not be tolerated. This includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.