

Assignment 2 - Answers

Name: Aiden Burgess

Login: abur970

Question 1

The `a2fuse1.py` program mounts the *source* directory onto the *mount* directory. FUSE makes this appear like the files *one*, *two*, *three* are actually in the *mount*.

FUSE has created a new file system in *mount*.

The request from the userspace is passed through the kernel, through FUSE, to the handler program. The response is passed back through FUSE and the virtual file system back to the user.

```
root@LAPTOP-M7BM40VA:../2/code# ls -l source
total 0
-rwxrwxrwx 1 root root 2 Oct  3 09:01 one
-rwxrwxrwx 1 root root 2 Oct  2 22:17 three
-rwxrwxrwx 1 root root 2 Oct  2 22:17 two
root@LAPTOP-M7BM40VA:../2/code# ls -l mount
total 0
-rwxrwxrwx 1 root root 2 Oct  3 09:01 one
-rwxrwxrwx 1 root root 2 Oct  2 22:17 three
-rwxrwxrwx 1 root root 2 Oct  2 22:17 two
```

Question 2

cd mount

```
DEBUG:fuse.log-mixin:-> getattr / (None,)
DEBUG:fuse.log-mixin:<- getattr {'st_atime': 1601774637.9675114, 'st_ctime':
1601774637.9675114, 'st_gid': 0, 'st_mode': 16895, 'st_mtime':
1601774637.9675114, 'st_nlink': 1, 'st_size': 4096, 'st_uid': 0}
```

Get attributes for root directory

```
DEBUG:fuse.log-mixin:-> access / (1,)
DEBUG:fuse.log-mixin:<- access None
```

Check if program can access "1"

cat > newfile

```
DEBUG:fuse.log-mixin:-> getattr /newfile (None,)
DEBUG:fuse.log-mixin:<- getattr "[Errno 2] No such file or directory:
'source/newfile'
```

Try and get attributes for "newfile"

```
DEBUG:fuse.log-mixin:-> create /newfile (33188,)
DEBUG:fuse.log-mixin:<- create 4
```

Create "newfile" with id 4

```
DEBUG:fuse.log-mixin:-> getattr /newfile (4,)
DEBUG:fuse.log-mixin:<- getattr {'st_atime': 1601774755.2196066, 'st_ctime':
1601774755.2196066, 'st_gid': 0, 'st_mode': 33279, 'st_mtime':
1601774755.2196066, 'st_nlink': 1, 'st_size': 0, 'st_uid': 0}
```

Get attributes for "newfile"

```
DEBUG:fuse.log-mixin:-> flush /newfile (4,)
DEBUG:fuse.log-mixin:<- flush None
```

Flush buffer for "newfile"

hello world

```
DEBUG:fuse.log-mixin:-> write /newfile (b'hello world\n', 0, 4)
DEBUG:fuse.log-mixin:<- write 12
```

Write "hello world" into newfile (12 bytes)

^D

```
DEBUG:fuse.log-mixin:-> flush /newfile (4,)
DEBUG:fuse.log-mixin:<- flush None
```

Flush buffer for "newfile"

```
DEBUG:fuse.log-mixin:-> release /newfile (4,)
DEBUG:fuse.log-mixin:<- release None
```

Release access for "newfile"

cd ../

```
DEBUG:fuse.log-mixin:-> getattr / (None,)
DEBUG:fuse.log-mixin:<- getattr {'st_atime': 1601774778.9632428, 'st_ctime':
1601774755.2196066, 'st_gid': 0, 'st_mode': 16895, 'st_mtime':
1601774755.2196066, 'st_nlink': 1, 'st_size': 4096, 'st_uid': 0}
```

Get attributes for root directory

fusermount -u mount

```
DEBUG:fuse.log-mixin:-> destroy / ()
DEBUG:fuse.log-mixin:<- destroy None
```

Destroy mount directory

Question 3

__init__

```
def __init__(self):
    self.files = {}
    self.data = defaultdict(bytes)
    self.fid = 0
    now = time()
    self.files['/' ] = dict(st_mode=(S_IFDIR | 0o755), st_ctime=now,
                           st_mtime=now, st_atime=now, st_nlink=2)
```

Creates an empty dictionary `self.files` for the files. This will use the path names as the keys. Each value in the dictionary will be another dictionary.

`self.data` is a dictionary for the files' data. The path names are the keys. The values are the data of that file.

Sets the starting value for the file descriptors, these are going to be used as unique file identifiers.

Grabs the current time and stores it in "now"

Sets the file attributes for the root of this file system. It is a directory, with creation, modified and accessed times set to now. It has two links.

getattr

Checks if the path is not in the current file system variable.

If it isn't then throw a `FUSEOSError` with no such file or directory code

Otherwise, return the file attributes by indexing the "files" field.

readdir

Return a list of files and directories in the file system. Automatically include ".", ".." which are the current directory and parent directory respectively. Remove the first character from each file, which is presumably "/". Exclude "/" directory from output as this is covered by "." already.

open

Increment the current file descriptor "fd" by one

Return the current file descriptor

create

Create a new entry in the file system, with current time for time parameters, one link, as a file with permissions specified as an input.

Increment file descriptor "fd" by one

Return the current file descriptor

unlink

Remove the specified file from the file system "files"

write

Append input data to the dictionary value in "data" corresponding to the input file. The data is added after the input "offset" size variable.

Update the size of the file in the file system "files" by using the length of updated data.

Return the length of added data.

read

Return the data stored in "data" for the file "path". Retrieve data at a certain offset and with certain size