

운영체제 실습

[Assignment #5]

Class : 목3
Professor : 최상호 교수님
Student ID : 2021202045
Name : 김예은

Introduction

5차 과제는 총 2개의 과제로 이루어져 있다. 5-1의 경우, proc file system module로, 모듈 적재 시 /proc/proc_2021202045/processInfo 파일을 생성 후 pid, ppid, uid, gid, utime, stime, state에 대한 내용을 출력한 것을 읽도록 한다. 5-2의 경우, FAT filesystem을 간단하게 구현하는 코드로, create, write, read, list, delete를 구현하고, load 및 save를 실행한다. FAT table은 file data를 저장하고, 각 블록은 다음 블록을 가리킨다. 마지막 블록은 -1을 가진다. block size는 32byte로, data가 block size보다 크면, 새 block을 할당하여 write해준다. 읽고, 쓸 때는 다음 블록으로 이동하면서 data에 접근한다.

Result

구현한 함수에 대한 설명은 다음과 같다.

FAT filesystem의 경우, fat table, file entry, data area를 가진다. file entry의 경우 각각 file name과 start_block과 file size를 가진다. save_file_system의 경우, binary로 file을 만들어서 저장하고 myfat 구조체의 크기를 바이트 단위로 읽어 가져오게끔 한다. write의 경우, 주어진 파일에 데이터를 기록하는데 블록 크기(32바이트)를 고려하여 데이터를 할당해준다. read의 경우, 주어진 파일을 읽고, 파일이 차지하는 블록들을 순차적으로 loop를 돌며 읽으면서 내용을 출력해준다. delete의 경우 주어진 파일을 삭제하고 해당 블록들을 0을 통해 비어있는 상태로 만들어준다. execute_cmd를 통해 명령어를 cmd창에서 받아 그 명령어에 따라 함수를 실행한다.

```
os2021202045@ubuntu:~/ASS5/5-2$ ./fat delete B
File 'B' deleted.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat delete A
File 'A' deleted.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat create A
File 'A' created.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat create B
File 'B' created.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat list
Files in the file system:
File: A, Size: 0 bytes
File: B, Size: 0 bytes
os2021202045@ubuntu:~/ASS5/5-2$ ./fat write A "Hello, world"
Data written to 'A'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat list
Files in the file system:
File: A, Size: 12 bytes
File: B, Size: 0 bytes
os2021202045@ubuntu:~/ASS5/5-2$ ./fat write B "Hello, world"
Data written to 'B'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat write B "Hello, world"
Data written to 'B'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat list
Files in the file system:
File: A, Size: 12 bytes
File: B, Size: 24 bytes
os2021202045@ubuntu:~/ASS5/5-2$ ./fat read A
Content of 'A':
Hello, world
os2021202045@ubuntu:~/ASS5/5-2$ ./fat read B
Content of 'B':
Hello, worldHello, world
os2021202045@ubuntu:~/ASS5/5-2$ ./fat delete B
File 'B' deleted.
```

A와 B file을 생성한 뒤, 각각 write를 해준다. ./fat list로 command를 주면, A 파일과 B 파일에 적힌 byte수가 나온다. ./fat read로 각각의 파일들을 읽어준 뒤, file B를 삭제한다.

```

os2021202045@ubuntu:~/ASS5/5-2$ ./fat write A "hello"
Data written to 'A'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat create B
File 'B' created.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat write B "hellohellohellohellohello"
Data written to 'B'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat list
Files in the file system:
File: A, Size: 5 bytes
File: B, Size: 25 bytes
os2021202045@ubuntu:~/ASS5/5-2$ ./fat read B
hellohellohellohellohello
os2021202045@ubuntu:~/ASS5/5-2$ ./fat write B "hello hello hello"
Data written to 'B'.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat list
Files in the file system:
File: A, Size: 5 bytes
File: B, Size: 42 bytes
os2021202045@ubuntu:~/ASS5/5-2$ ./fat read B
hellohellohellohellohellohello hello hello
os2021202045@ubuntu:~/ASS5/5-2$ ./fat delete B
File 'B' deleted.
os2021202045@ubuntu:~/ASS5/5-2$ ./fat delete A
File 'A' deleted.
os2021202045@ubuntu:~/ASS5/5-2$

```

A file을 생성하고, hello를 write해준다. B file을 create로 생성해준 뒤, 위와 같이 2번의 write를 통해 32byte보다 길게 적어준다. list를 통해 A 파일은 5 byte가, B 파일은 47byte가 나오는 것을 확인할 수 있고, read를 통해 write한 것이 제대로 출력되는 것을 확인할 수 있다. delete로 A와 B file을 지워준다.

고찰

delete를 할 때, 첫 번째 파일은 잘 삭제가 되는데 두 번째 파일은 삭제할 때마다 0번째 block에 머물러있어서 무한루프가 돌았다. 이 부분을 while문에 block != -1을 추가해줬는데, 그 뒤로 무한루프가 돌지 않았다. 0xFFFF가 unsigned가 아니라 그냥 int로 읽히면, -1이 아니라 양수형으로 읽히는데 그때문이지 않았나 싶다. 또한 read의 경우 "%/s"로 읽은값을 출력해주니 write할 때 띄어쓰기 없이 값을 넣어줘도 알아서 띄어쓰기를 해준 채로 읽어서 마지막에 값이 잘려서 출력되었다. 이 부분을 putchar로 바꿔서 write한 값 그대로 잘 출력되게끔 수정했다.