

1. Code explanation

First, we open the sample.txt with read and write flag (O\_RDWR) and initiate a char buffer called rbuffer [30] to store string read from file. Then, we read the file with read(2) and set the length is the target's length. Then, compare rbuffer read from file with target which is "student.", and use lseek(2) to move the position by one step every time. If we get the same value with target, break the iteration and print rbuffer out. we also initiate wbuffer to store the string that we want to write into file, use the offset we count before to indicate where we need to write the "NTHU student." In the end, lseek(fd, 0, SEEK\_END) can help us to know the total length of the file, move to the beginning of the file and read the whole file into rbuffer and print it out.

2. Question

No, the append flag doesn't support us to do so, because append flag will force the file offset is positioned at the end of the file(described in manual) before calling write(2). That prevent us from revising the content in the file, but we still can append string behind it.

Therefore, we use read and write flag (O\_RDWR) instead of it.

3. Result

```
student.  
Hello, I am a NTHU student.
```

Because our implementation uses relative path, so sample.txt needs placing into the same folder.