Group_7 Assignment_1 report

Write a program to verify if you can open a file with the append flag to:
(1) Read from the specific place in the file using `lseek`.
(2) Write data at the specific place in the file using `lseek`.
(1 pts) Answer the above questions in your report.

- Since the O_APPEND flag specifies that any data written to the file will be appended to the end of the file, rather than overwriting existing data, therefore :
  1. Yes, it is possible to read the file with the append flag by using lseek.
  2. No, it is impossible to write data at the specific place in the file using lseek with the append flag.

---

(3) (1 pt) If the append flag cannot support you to do so, please **replace it with the right one and explain your implementation in your report**.

Since the append flag cannot support the task, we replace it with the O_RDWR flag, which allows that the file can be read from and write to, in any order.

- Below is the explanation of the code we implement

1. First, we `open` the “sample.txt” with read and write permissions(`O_RDWR`), and the file descriptor is stored in `fd`
   - If open returns `-1` , an error has occurred. The program will prints an error message and returns `1` to indicate failure.

- Also, in the commented line is the record that we had tried to use the `O_APPEND` flag to test the code.

```
int fd = open("sample.txt", O_RDWR); // Open file with read-write permissions
// int fd = open("sample.txt", O_APPEND); // Open file with append permissions
if (fd == -1) {
  perror("Failed to open file");
  return 1;
}
```

2. Second, we use `lseek` to move the file offset to the 15th position and reads 8 characters into the `buffer`, then use `printf` to print out the “student.”

```
char buffer[9];
lseek(fd, 14, SEEK_SET);
read(fd, buffer, 8);
printf("%s\n", buffer);
```

3. Third, we again move the file offset to the 15th position using `lseek` . Then use `write` function to writes “NTHU student.” into the file from the position.

```
// Write "NTHU student."
lseek(fd, 14, SEEK_SET);
write(fd, "NTHU student.", 13); // Adding NTHU student.
```

4. Last, we set the file offset back to the beginning and reads the whole file into `finalBuffer` , then prints out the buffer.

```
// Read and print the whole file
char finalBuffer[30];
lseek(fd, 0, SEEK_SET);
read(fd, finalBuffer, 30);
printf("%s\n", finalBuffer);
```

5. below is the final outcome