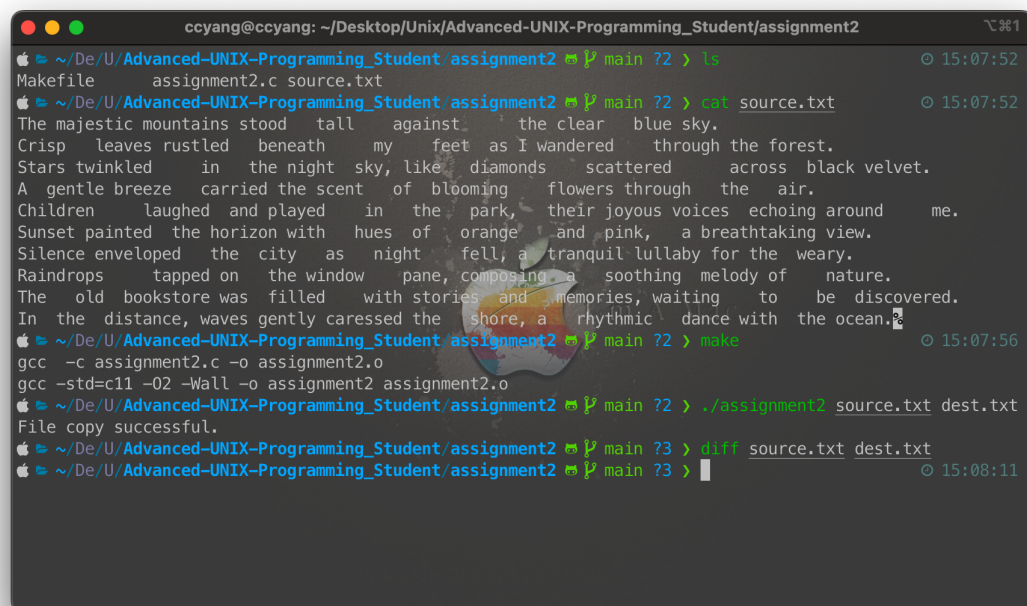


Unix Assignment2

Team19 108072244 邱煒甯 108032053 陳凱揚 112065525 簡佩如

Implementation

(1) (3 pt) Implement your own cp(1) which can copy the file correctly.



```
ccyang@ccyang: ~/Desktop/Unix/Advanced-UNIX-Programming_Student/assignment2
#1
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 72 > ls
Makefile      assignment2.c source.txt
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 72 > cat source.txt
The majestic mountains stood tall against the clear blue sky.
Crisp leaves rustled beneath my feet as I wandered through the forest.
Stars twinkled in the night sky, like diamonds scattered across black velvet.
A gentle breeze carried the scent of blooming flowers through the air.
Children laughed and played in the park, their joyous voices echoing around me.
Sunset painted the horizon with hues of orange and pink, a breathtaking view.
Silence enveloped the city as night fell, a tranquil lullaby for the weary.
Raindrops tapped on the window pane, composing a soothing melody of nature.
The old bookstore was filled with stories and memories, waiting to be discovered.
In the distance, waves gently caressed the shore, a rhythmic dance with the ocean.
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 72 > make
gcc -c assignment2.c -o assignment2.o
gcc -std=c11 -O2 -Wall -o assignment2 assignment2.o
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 72 > ./assignment2 source.txt dest.txt
File copy successful.
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 73 > diff source.txt dest.txt
🍏 ~ /De/U/Advanced-UNIX-Programming_Student/assignment2 🍏 main 73 >
#1
```

Result

(2) (1 pt) Describe your implementation in your report.

1. Our program, `assignment2.c`, first checks if the number of input parameters provided is correct. If it is correct, the program proceeds with the subsequent copying operation.

```

32  int main(int argc, char *argv[]) {
33      // Check the number of parameter
34      if (argc != 3) {
35          printf("Error: This program requires exactly 2 parameters.\n");
36          return 1;
37      }
38
39      // Copy the file from argv[1] to argv[2]
40      copyFile(argv[1], argv[2]);
41
42      return 0;
43  }

```

2. Next, we use `fopen` to perform the file copying. We open the source file in binary read mode using `"rb"`, and then open the destination file in binary write mode using `"wb"`. This part is from line 5 to line 17.

Then, we copy the content character by character from the source file to the destination file. The specific method involves using the `fgetc` and `fputc` functions. This part is from line 19 to line 23.

Finally, we use `fclose` to complete the task of copying the file. This part is from line 26 to line 27."

```

3 void copyFile(const char *source_filename, const char *dest_filename) {
4     // Open the source file in binary read mode
5     FILE *source_file = fopen(source_filename, "rb");
6     if (source_file == NULL) {
7         printf("Error opening source file.\n");
8         return;
9     }
10
11     // Open the dest file in binary write mode
12     FILE *dest_file = fopen(dest_filename, "wb");
13     if (dest_file == NULL) {
14         printf("Error opening dest file.\n");
15         fclose(source_file);
16         return;
17     }
18
19     // Copy content character by character
20     char ch;
21     while ((ch = fgetc(source_file)) != EOF) {
22         fputc(ch, dest_file);
23     }
24
25     // Close the files
26     fclose(source_file);
27     fclose(dest_file);
28
29     printf("File copy successful.\n");
30 }

```