

LABWORK 5: Longest Path

Student's name: Nguyen Thai Duong

Student's ID: BI12-117

I. Workflow

a. *Function find_longest_path*

Function find_longest_path:

- This function takes a file pointer as input and returns a pointer to the longest path found in the file.
- It initializes a pointer longest_path to NULL.
- Inside a while loop, it reads each line of the file using fgets.
- It removes the newline character from the path if present.
- If longest_path is NULL or the length of the current path is greater than the length of the longest_path, it dynamically allocates memory for longest_path, copies the current path into it, and updates the length.
- After the loop, it returns the pointer to the longest_path.

```
// Function to find the longest path in a file
char* find_longest_path(FILE* file) {
    char path[MAX_PATH_LENGTH];
    char* longest_path = NULL; // Use a pointer to dynamically allocate memory

    while (fgets(path, MAX_PATH_LENGTH, file) != NULL) {
        int length = strlen(path);
        // Remove newline character if present
        if (path[length - 1] == '\n')
            path[length - 1] = '\0';

        if (!longest_path || length > strlen(longest_path)) {
            // Free previously allocated memory (if any)
            free(longest_path);
            longest_path = malloc(length + 1); // Allocate memory for the path
            if (longest_path == NULL) {
                perror("Memory allocation failed");
                exit(EXIT_FAILURE);
            }
            strcpy(longest_path, path);
        }
    }

    return longest_path; // Return the pointer to the longest path
}
```

b. Main

Function main:

- It opens the file "test.txt" in read mode. If it fails to open the file, it prints an error message and exits.
- It calls the function `find_longest_path` with the file pointer.
- If the returned pointer is not NULL, it prints the longest path and frees the memory allocated for `longest_path`. If the pointer is NULL, it prints a message indicating that no paths were found.
- It closes the file and returns 0 indicating successful execution.

```
int main() {  
    FILE* file = fopen("test.txt", "r");  
    if (file == NULL) {  
        perror("Error opening file");  
        return 1;  
    }  
  
    char* longest_path = find_longest_path(file); // Call the function  
    if (longest_path != NULL) {  
        printf("The longest path found: %s\n", longest_path);  
        free(longest_path); // Free the allocated memory  
    } else {  
        printf("No paths found in the file.\n");  
    }  
  
    fclose(file);  
    return 0;  
}
```

c. Figure

Here's is the figure that illustrated the workflow of Mapper and Reducer:

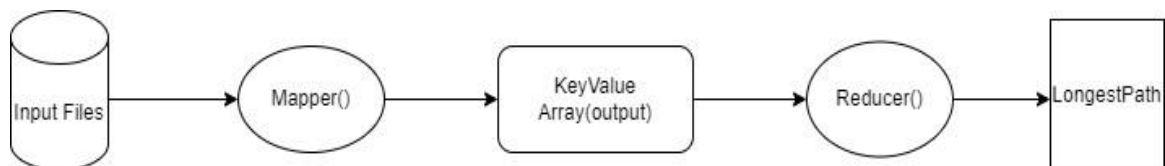
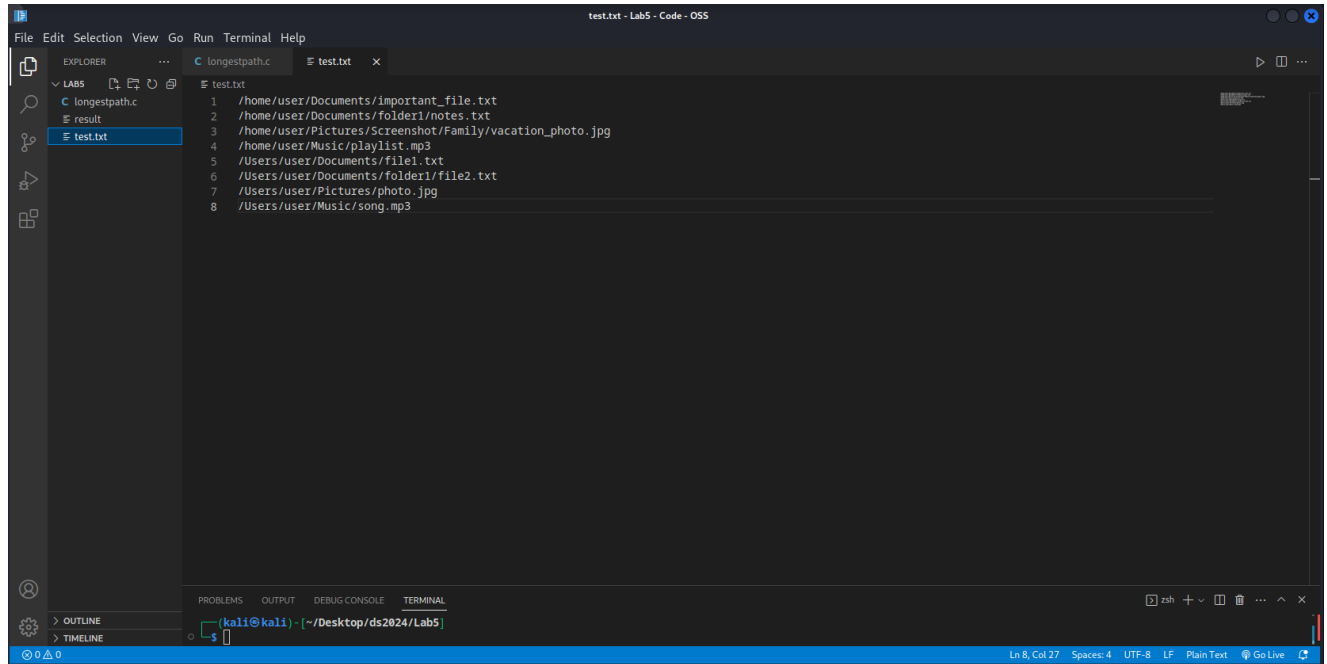


Figure 1. Workflow of MapReduce

II. Implementation

Test file:



```
test.txt - Lab5 - Code - OSS

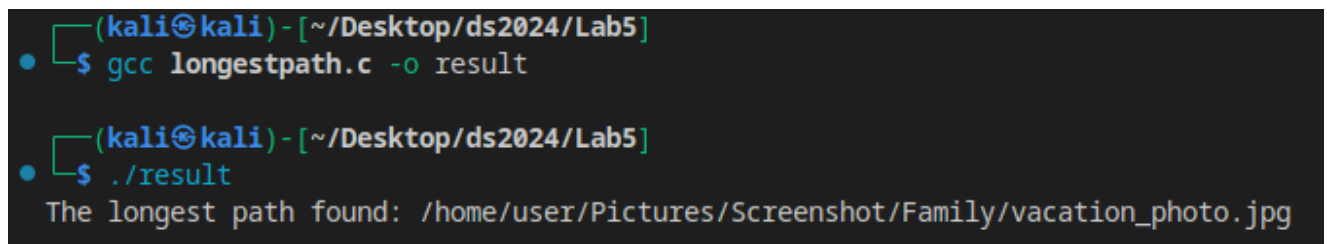
File Edit Selection View Go Run Terminal Help

EXPLORER
└─ LABS
   └─ longestpath.c
      └─ result
         └─ test.txt

test.txt
1 /home/user/Documents/important_file.txt
2 /home/user/Documents/folder1/notes.txt
3 /home/user/Pictures/Screenshot/Family/vacation_photo.jpg
4 /home/user/Music/playlist.mp3
5 /Users/user/Documents/file1.txt
6 /Users/user/Documents/folder1/file2.txt
7 /Users/user/Pictures/photo.jpg
8 /Users/user/Music/song.mp3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(kali@kali) - [~/Desktop/ds2024/Lab5]
$
```

The result:



```
(kali@kali) - [~/Desktop/ds2024/Lab5]
$ gcc longestpath.c -o result

(kali@kali) - [~/Desktop/ds2024/Lab5]
$ ./result
The longest path found: /home/user/Pictures/Screenshot/Family/vacation_photo.jpg
```