```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <string.h>
#include <time.h>

int main(int argc, char *argv[]) {
    printf("Check output.log\n");
    pid_t pid;
    int status;

    if (argc != 3) {
        fprintf(stderr, "Usage: %s <input_file> <output_file>\n", argv[0]);
        return 1;
    }


    FILE *logFile = freopen(argv[2], "w", stdout);
    FILE *filePtr = fopen(argv[1], "r");


    if (filePtr == NULL) {
        printf("Error opening file.\n");
        return 1;
    }

    char buffer[256];
    while (fgets(buffer, sizeof(buffer), filePtr)) {
        buffer[strcspn(buffer, "\r\n")] = '\0';

        time_t start_time = time(NULL);
        char *start_str = ctime(&start_time);


        printf("Start Time: %s", start_str);
        printf("Commands: %s\n", buffer);
        printf("Output: \n");
        printf("-----\n");

        fflush(stdout);

        pid_t pid = fork();

        if (pid < 0) {
        perror("fork");
        exit(EXIT_FAILURE);
        }else if (pid == 0) {
        // Child process
        int status = system(buffer);
        exit(WEXITSTATUS(status));
        }else{

        int status;
        wait(&status);

        time_t end_time = time(NULL);
        char *end_str = ctime(&end_time);

        printf("-----\n");

        if (WIFEXITED(status)) {
            printf("Exit Status: %d\n", WEXITSTATUS(status));

        } else {
            printf("Command did not terminate normally.\n");
        }
```

```c
            printf("TimeEnd: %s", end_str);
            printf("----------------------------------------\n");
            fflush(stdout);
        }
    }

    printf("\n");
    fclose(filePtr);
    fclose(logFile);
    return 0;
}


/*
pid = fork(); Don not uncomment or could will break and run forever
Uncomment at your own risk
*/


// Leave no lines blank will run infnitley
// Only works with Two commands somehow leave no blank space after line 1,2
// Only have two commands
// gcc -o jlplayerlab7 jlplayerlab7.c
// ./jlplayerlab7 command.txt output.log
```