

CS 212: Final Review Worksheet 1

Consider the following program which reads up to 128 bytes from the file `input.txt` and writes all of the bytes that were read to the file `output.txt`. Note that the program does not make any system calls directly, since it uses the C standard library's file functions for opening, reading, and writing, but behind the scenes those functions will be making system calls. Also note that for simplicity the program lacks any sort of error handling, but in general it is good practice to check return values for errors.

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    char *buffer = malloc(128);

    FILE *f_input = fopen("input.txt", "r");
    FILE *f_output = fopen("output.txt", "w");

    size_t chars_read = fread(buffer, 1, 128, f_input);
    fwrite(buffer, 1, chars_read, f_output);

    return 0;
}
```

Assume you have compiled this program to the executable `read_and_write` and you then type this command into a shell in a POSIX operating system:

```
$ ./read_and_write
```

What happens between the time you press enter and the time the process running the program exits? Be as detailed as possible based on what we have discussed this semester. Discuss the roles of the shell, the process running this program, the kernel, the CPU, main memory, and the disk on which the files are stored.