FIT3142 Optional Laboratory #1 + #2: Raw File I/O Operations

Dr Carlo Kopp, SMIEEE, AFAIAA, FLSS, PEng
Faculty of IT, Clayton.
Email: Carlo.Kopp@monash.edu

July 29, 2016

Revision Status:

$Id: FIT3142-Laboratory-1+2-Bonus.tex,v 1.1 2016/07/29 09:58:59 carlo Exp carlo $

Contents

1 Introduction

As many students will have completed introductory courses on `*nix/Linux` and C-language programming, and will be able to complete Laboratory Sessions 1 and 2 quickly, we offer students the optional choice of also doing this Laboratory exercise for bonus marks.

Please note that students are still required to complete and submit Laboratory Sessions 1 and 2 that will be assessed.

This Laboratory task is assessed if students opt to complete it, and bonus marks will be awarded, with the caveat that the total number of regular and bonus marks for laboratory work cannot exceed the total of 100% for laboratory work in Semester.

DO NOT ATTEMPT THIS LAB UNLESS YOU HAVE PRIOR EXPERIENCE IN C-LANGUAGE and *NIX!

In previous C-programming exercises, students will have learned the use of the `stdio` library (i.e. `fopen()`, `fclose()`, `fseek()`, `fread()`, `fwrite()`, `fgetc()`, `fputc()`) for manipulating files on a `*nix` operating system. This I/O library was designed for ease of use, and to hide the specifics of the actual *system calls* that access these services on an underlying operating system.

The aim of this lab will be to familiarise you with the behaviour of raw file I/O system calls (i.e. `open()`, `close()`, `lseek()`, `read()`, `write()`), using the Linux system and the C language.

The free textbook by *David A. Curry* entitled *"Using C on the UNIX System"* provides a good introduction to this subject matter. It is available via a link on Moodle in PDF and Kindle formats. Details of the system calls specific to the Linux version in laboratory use can be found in the `man` pages on the system.

It is important that you do this work independently. Collaboration, copying and plagiarism will attract zero marks in this subject, without exception.

2 Lab Session 1 Bonus (100%)

1. (25 %) Write a simple program in C using only `stdio` library calls that allows the user to interactively open a file, seek to a position in the file, and read or write some user entered ASCII text string of up to 48 ASCII characters into the file.
2. Test the program and demonstrate how it works to your demonstrator.
3. (25 %) Modify this program so that it can seek to a position in a file, read a specified number of bytes from the file, reverse the order of bytes in this array, and write them back into the file in the reversed order.
4. Test the program and demonstrate how it works to your demonstrator.
5. (50 %) Modify your program to use only raw `Linux` file I/O system calls instead of `stdio` library calls, i.e. `open()`, `close()`, `read()`, `write()`, `lseek()`.
6. Test the program and demonstrate how it works to your demonstrator. Explain in at least 100 words the respective advantages and disadvantages of using `stdio` library versus system calls for I/O operations on files.

3 Lab Session 2 Bonus (100%)

1. (100 %) Write a utility using the `stat()` system call that dumps in human readable form (ASCII to display, refer Linux `man` page) all of the attributes of the file held in the file's `i-node` structure.
2. Test the program and demonstrate how it works to your demonstrator.

4 Notes

1. If your program does not compile you will be awarded zero marks for this prac.
2. You must demonstrate your program to the demonstrator before the end of each lab session. If your program does not work at all you will be awarded zero marks for this lab. If it has bugs or is incomplete, marks will be deducted accordingly.
3. Cheating, collaborating and plagiarism will attract zero marks. You are not permitted to use verbatim any examples in the Curry textbook.