# **Lab Sheet 6**

**Exercise 1**

Write a program that searches and replaces whole words in text files, as requested by the end-user, by first transferring the entire file content to memory.

**Exercise 2**

Scale the previous program for large files by avoiding having to copy the entire file into memory. You need to make use of a second temporary file.

**Exercise 3**

Write a function that generates indexes that 'point' to file offsets for the file created by the program in the following listing. These indexes are to be subsequently used by display functions in the same program that fetch data items directly from disk, without having to iterate through the entire data collection in memory. The functions should display all books in descending and ascending order by all fields (i.e., you need to create 3 indexes in total). You may make use of the selection sort introduced earlier in the study unit.

#include <stdio.h>

typedef struct {

char title[40];

char author[40];

float price;

} Book;

int main(void) {

Book book1 = {

"The C Programming Language",

"Kernighan & Ritchie",

29.99

};

Book book2 = {

"Clean Code",

"Robert C. Martin",

24.99

};

FILE \*fp = fopen("book.dat", "wb");

if (fp == NULL) {

perror("Unable to open file for writing");

return 1;

}

fwrite(&book1, sizeof(Book), 1, fp);

fwrite(&book2, sizeof(Book), 1, fp);

fclose(fp);

Book read\_book1, read\_book2;

fp = fopen("book.dat", "rb");

if (fp == NULL) {

perror("Unable to open file for reading");

return 1;

}

fread(&read\_book1, sizeof(Book), 1, fp);

fread(&read\_book2, sizeof(Book), 1, fp);

fclose(fp);

printf("Title: %s\nAuthor: %s\nPrice: $%.2f\n\n",

read\_book1.title, read\_book1.author, read\_book1.price);

printf("Title: %s\nAuthor: %s\nPrice: $%.2f\n",

read\_book2.title, read\_book2.author, read\_book2.price);

return 0;

}

**Exercise 4**

Modify the previous program to load, rather than generate, indexes from disk. For these indexes to be available, the previous program must be enhanced further to create indexes on the fly while new data items are being added. On program exit, these indexes must be persisted to files on disk.