Lab 3: Reading Material

Lab 3 is dedicated to dynamic data structures in C, program debugging via the use of `valgrind(1)` and patching of binary files.

Dynamic data structures in C

Dynamic data structures in C are usually built using pointers and C structures. Please read chapter Complex types of the wikibook. Memory for dynamic data is allocated and released dynamically, on the heap. C library functions for memory management are described in chapter Memory management.

Patching

A convenient utility for inspecting binary files, called `hexedit(1)`, is installed on the lab computers. Please read the **man** for the utility and familiarize yourself with `hexedit` by trying to view and edit a few different files.

Debugging

Please read the **man** page for `valgrind(1)`, you can focus on the `--leak-check`, `--show-reachable` and `-v` parameters. `valgrind` can help you detect memory leaks and other types of errors (e.g. illegal access to memory address). For a complete list of error messages and their meaning - http://valgrind.org/docs/manual/mc-manual.html.

Input from `stdin` or other files

You already know `fgets(3)` for getting "strings" from files (like `stdin`). Now you must also learn to use fread( ) (see man `fread(3)`) to read a pre-specified number of bytes from a file. In order to format printouts, you should be more familiar with `printf(3)`. In order to parse strings efficiently and extract values of different types from them, you should use `sscanf(3)`.

Other functions to learn

Look up fseek( ) and fwrite( ).