- Main
- Course info: Arch&SPlab, Splab, Arch only.
- Announcements
- Assignments
- Class material
- Practical sessions
- Lab sessions
- Lab Completions
- Labs Schedule
- FAQ
- Exams
- Previous exams
- Grades
- Useful links
- Online Quiz
- Forum
- Labs Forum
- Submission system

- recent changes
- logout

zoharith logged in
student mode

• Lab3 » Reading printable version

Contents (hide)
1 Memory in C
2 Function Dispatching
2.1 Function Pointers
2.2 Handlers and dispatchers

Lab 3: Reading Material

Memory in C

Lab 2 demonstrates a few aspects of the memory layout of a C program. You should read this useful document to get a better understanding of the memory layout of C programs. So far, most of our work used the stack to store all of the data our programs needed, in future labs we will work with dynamically allocated memory (the heap) using APIs like malloc(3) and free(3).

Function Dispatching

Function Pointers

In C, just about every thing is a number. There are integers, floats, chars and pointers. Pointers can be used to reference almost anything in the program's memory - for instance, functions. A pointer points to a function, a function pointer, allows us to turn parts of the program into data and then back again. This allows the program to control the execution flow by calling functions without using (or event knowing) their name based on runtime data (e.g. user's input or calculations). Function pointers allow the programmer to create functions that accept other functions, called callbacks, and invoke those callbacks when required, thus creating a mechanism for asynchronous programming (and a basis for multi-threaded programming). IT also allows the programmer to implement the control flow of the program using handlers and dispatchers.
See this page for more details an examples on function pointers.

Handlers and dispatchers

A handler is simply a function that handles a certain part of program's flow. Any function can be a handler, however, handlers are usually grouped based on their signatures and usage, like a set of functions designed to handle user's or a set of functions designed to handle parsing of some text.

A dispatcher is a piece of code designed to select and invoke a handler from a given set based on some program state or input. Dispatchers are usually used to create modular code since they decouple the handlers' logic from the program's control flow. In lab 3 we'll use a dispatcher and a set (or sets) of handlers to improve our toy_printf.