Systems Programming: Practical 4 — Pointers

This practical will provide a few simple exercises to help you understand pointers better.

A Pointer Practice

Look at the file, `pointer_example.c`. Without running it, work out what this code will print out. Next, try running it and see if it prints out what you expected.

B Swapping Pointers

The file, `swap_pointers.c`, contains a function to swap two integers. It also contains a function to swap two pointers to strings, but this doesn't work correctly. Fix the code.

C Reversing Strings

Write a function that takes a pointer to a C string and prints it out with the characters in reverse order. Do this only using pointer arithmetic, without using array (`a[1]` etc.) notation. Do not use `strlen()`.

D Point-wise Multiplication

A vector is an ordered collection of values. Vectors can be multiplied entry-by-entry e.g. `(1,2,3)*(4,5,6)=(4,10,18)` Write a function `mult_vec(int *v1, int *v2, int *v3, int n)` that multiplies the vectors `v1` and `v2`, and stores the result in `v3`. `n` is the length of the vectors.

E Extension Exercise: Pointer Arithmetic

Look at the file `complex_pointers.c`. Can you figure out what this code does and how?