Mandatory Assignment 4

Linux and C Programming (62558)

Mads Richardt (s224948)

October 20, 2022

Exercise 8.2

```
Stack
-----
int x = 1
int y = 2
int temp = *y;
*y = *x;
*x = temp;
```

Exercise 8.3

When the StackAllocation function is called we get the following output.

```
***************
StackAllocation
***************
Value of x = 7
Address of x = 0x7ffc7811b05c
Value of y = 7
Address of y = 0x7ffc7811b05c
```

When foo1 is called, the variable x gets declared, meaning that x is allocated the address at the top of the stack. Then, x gets initialized to 7, meaning that 7 gets stored at the address allocated to x. When foo1 exits, the address assigned to x is freed up. Subsequently, when foo2 is called the address, which was formerly allocated to x, is now allocated to y. However, y is not initialized to anything, so the address still stores the value assigned to x in foo1. Accordingly, the addresses and values of x and y are identical.