Intro to Debug C Programs with GDB

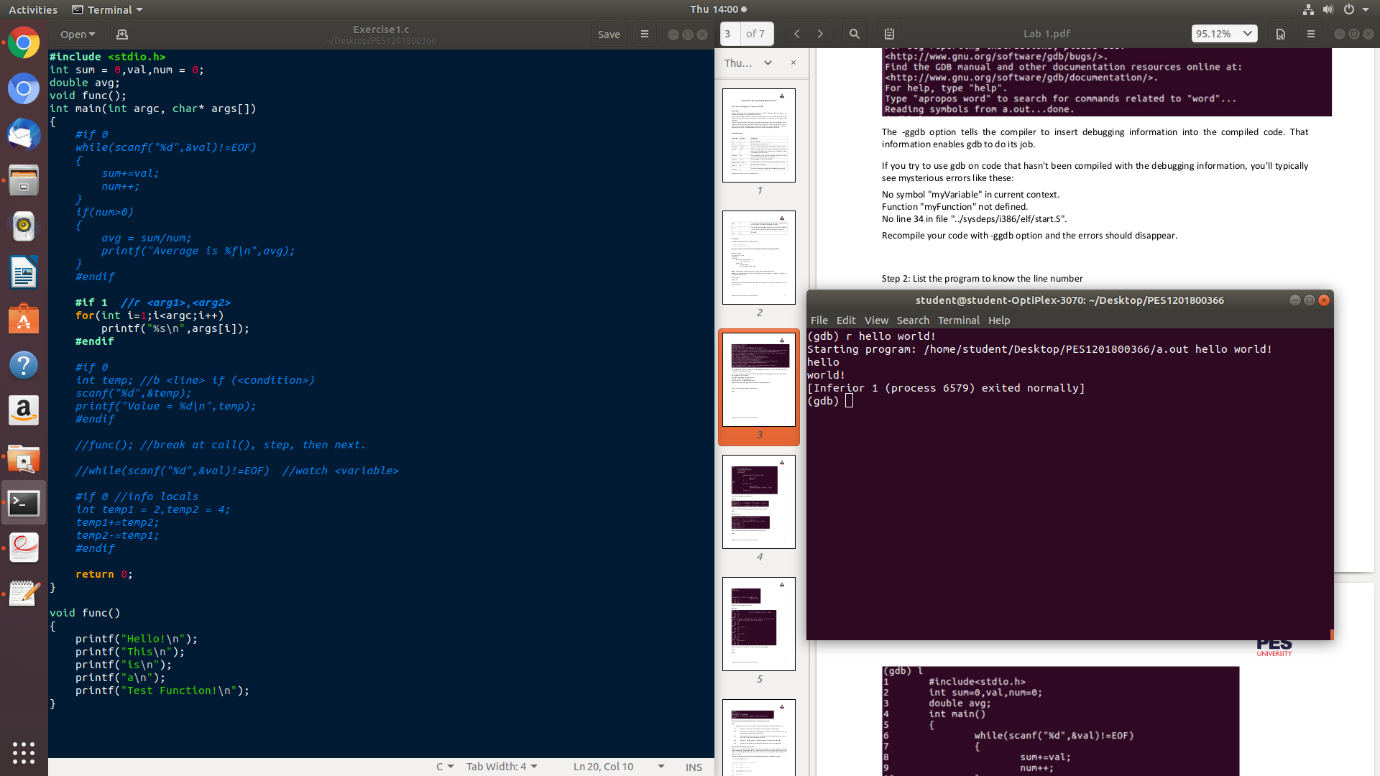
# Exercise 1

1. How do you pass command line arguments to a program when using gdb?

Ans – While using r to run the entire program, include arguments just as how we would pass them while executing the outfile.

Syntax: r <*arg1*> <*arg2*>…

Example: r hello world!

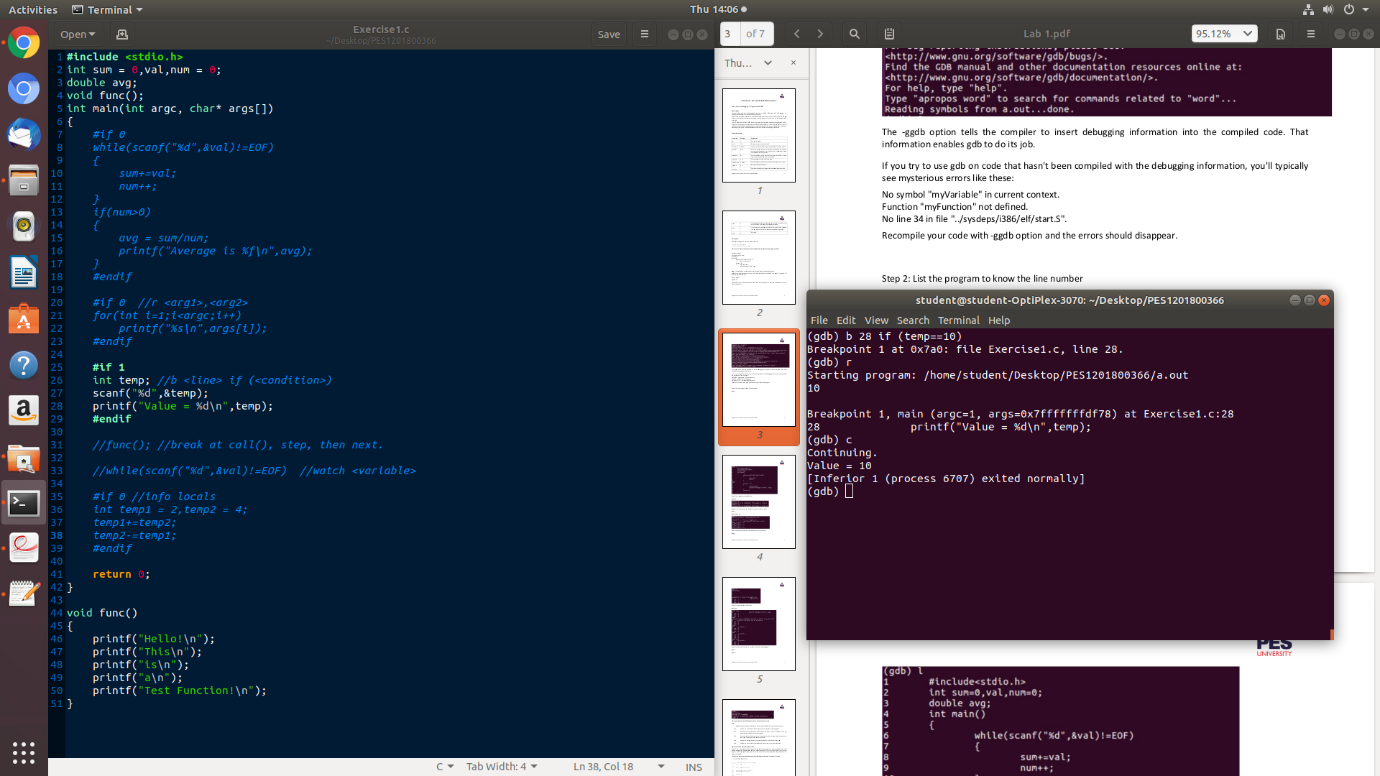


1. How do you set a breakpoint which only occurs when a set of conditions is true (eg when certain variables are a certain value)?

Ans – Conditional breakpoints can be set by suffixing a condition to the break statement.

Syntax: b <*line number*> if (<*condition*>)

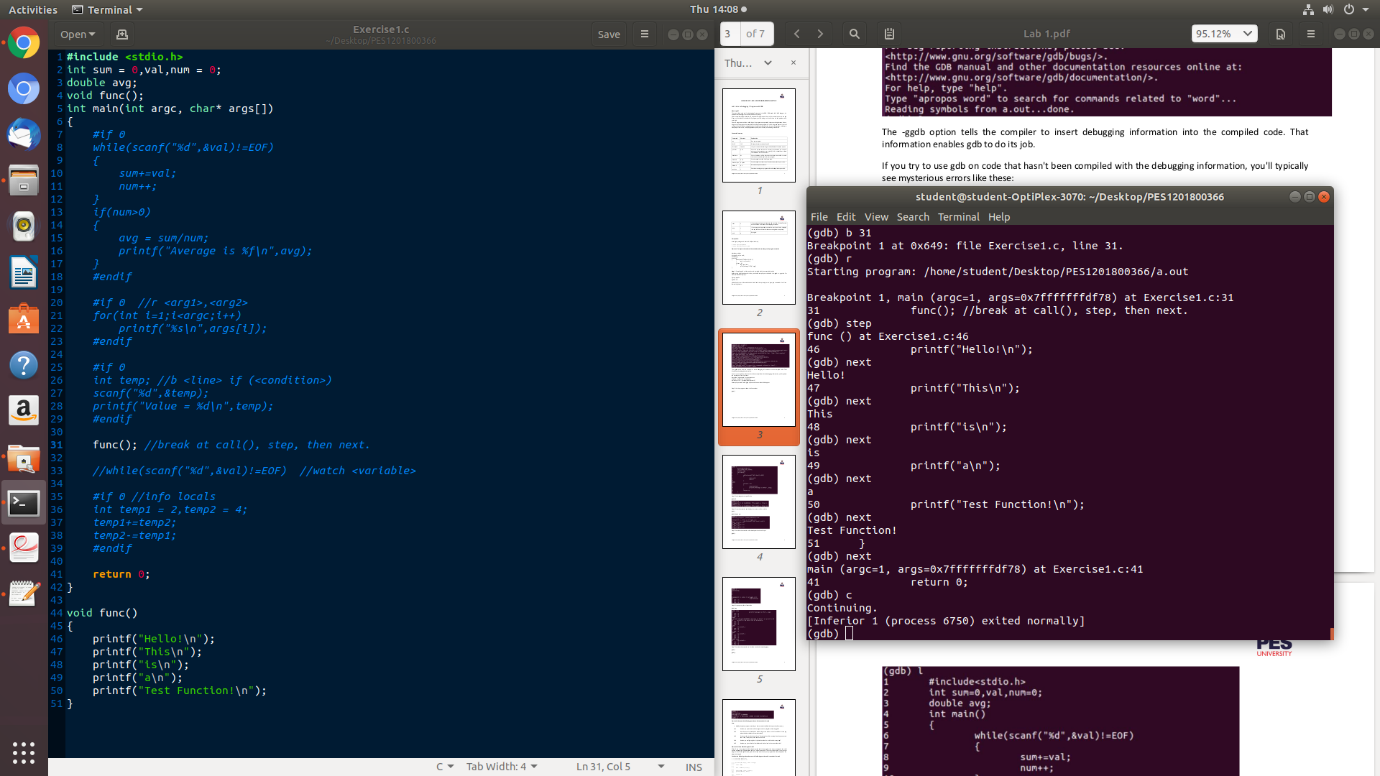
Example: b 20 if (temp == 10)



1. If the next line is a function call, you'll execute the call in one step. How do you execute the C code, line by line, inside the function call?

Ans – We can step through a function call and definition by placing a breakpoint at the function call and then using the step command to step into the function. The next or step commands can then be used to execute each statement in the function, one line at a time.

Syntax: step

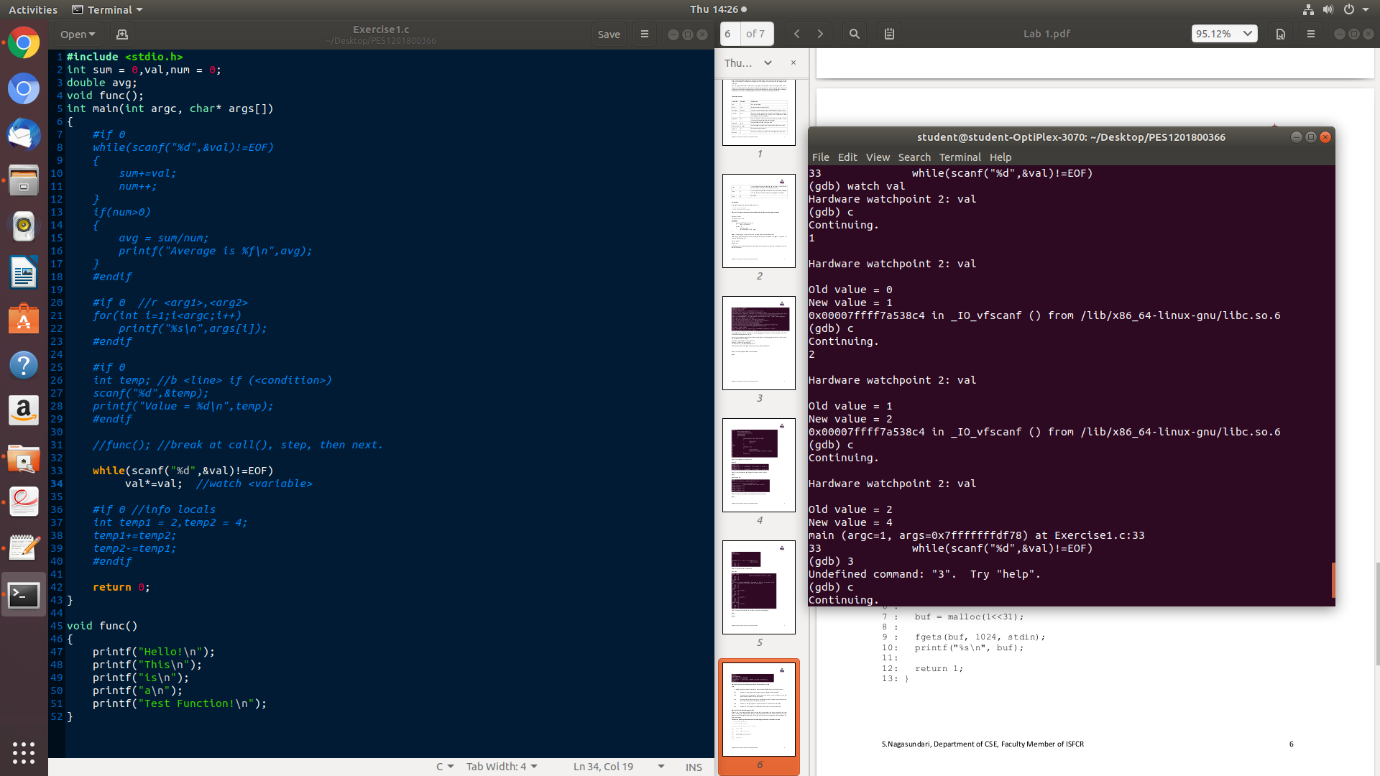


1. How do you configure gdb so it prints the value of a variable after every step?

Ans – We can display the value of a variable after every step by using the watch command, which will place a watchpoint on a variable and display the value it holds after executing till a pause or a breakpoint in the gdb.

Syntax: watch <*variable*>

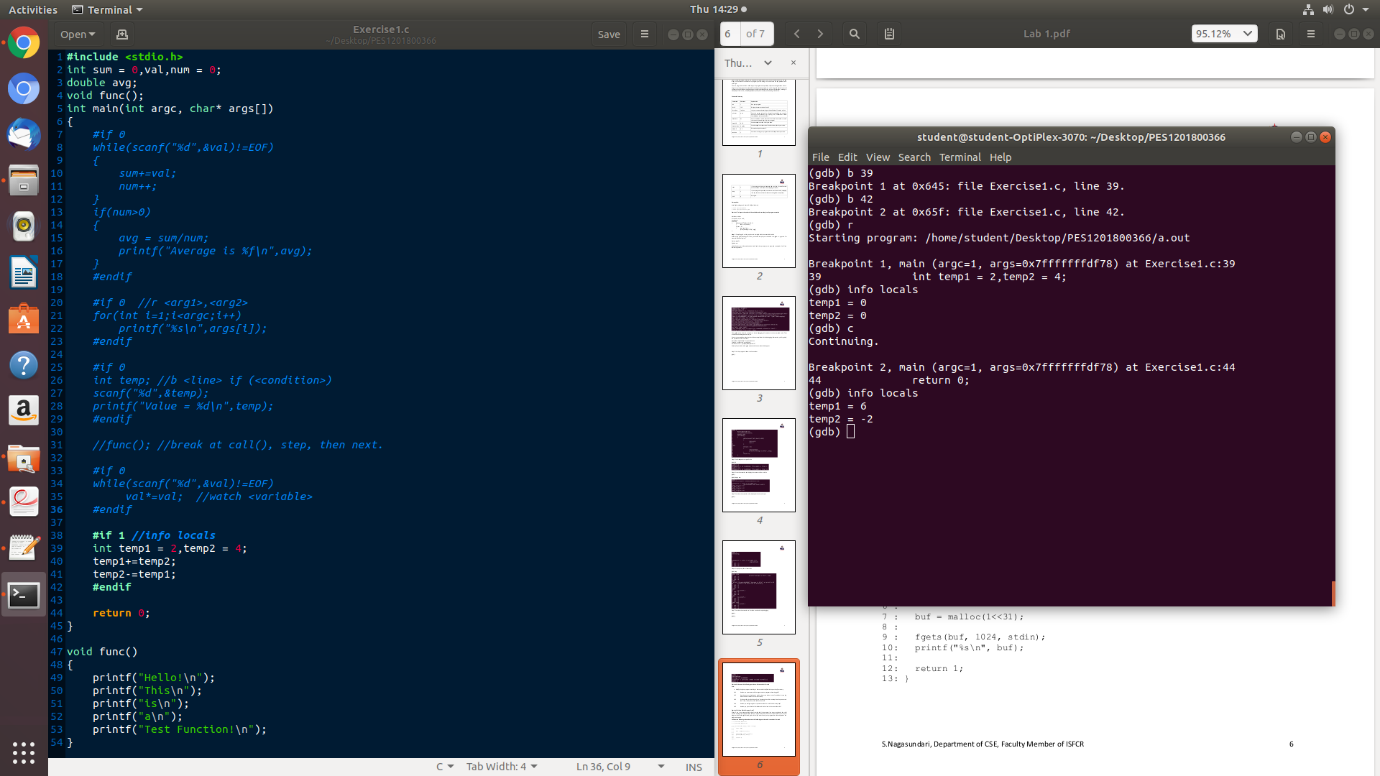
Example: watch val



1. How do you print a list of all variables and their values in the current function?

Ans – Information on all variables including their names and values can be done using the info command. By calling info on locals, all the local variables inside a function definition are printed along with their values.

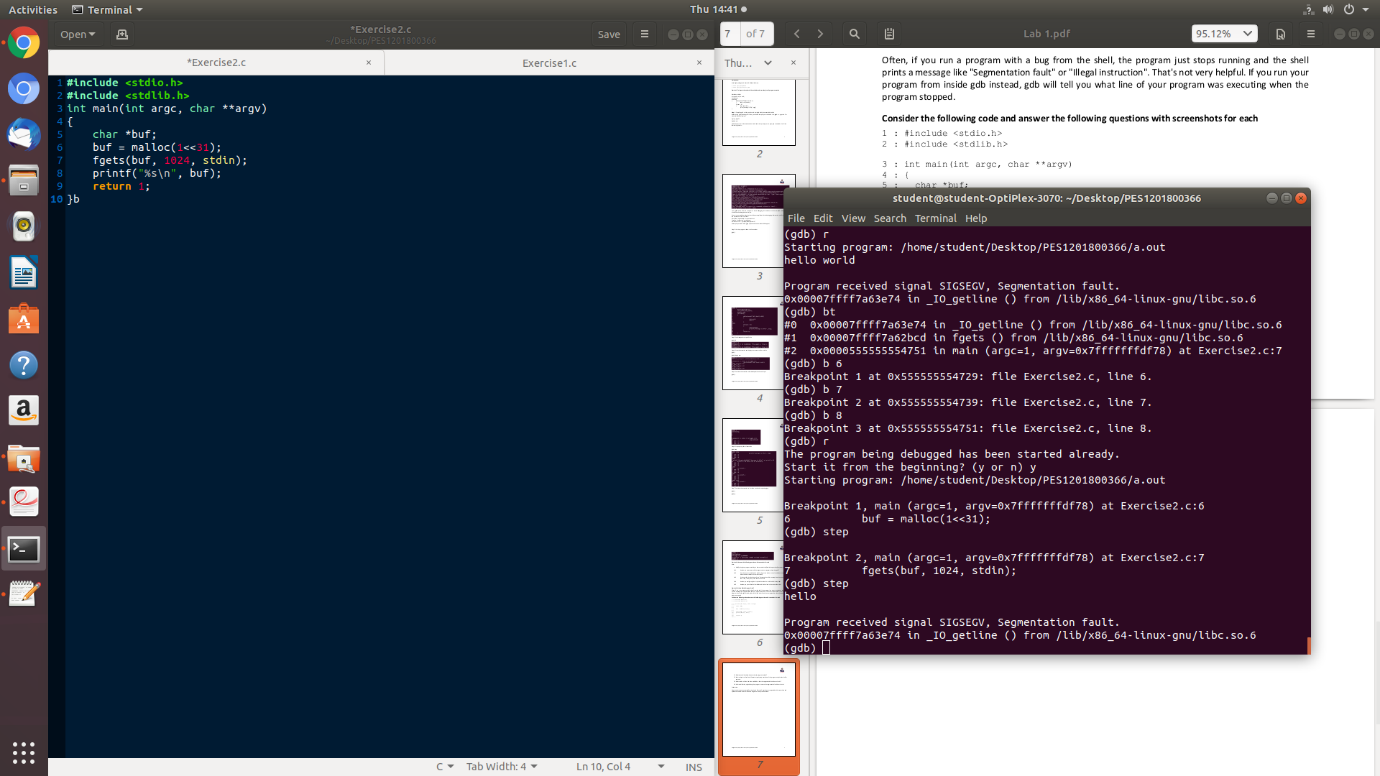
Syntax: info locals



# Exercise 2

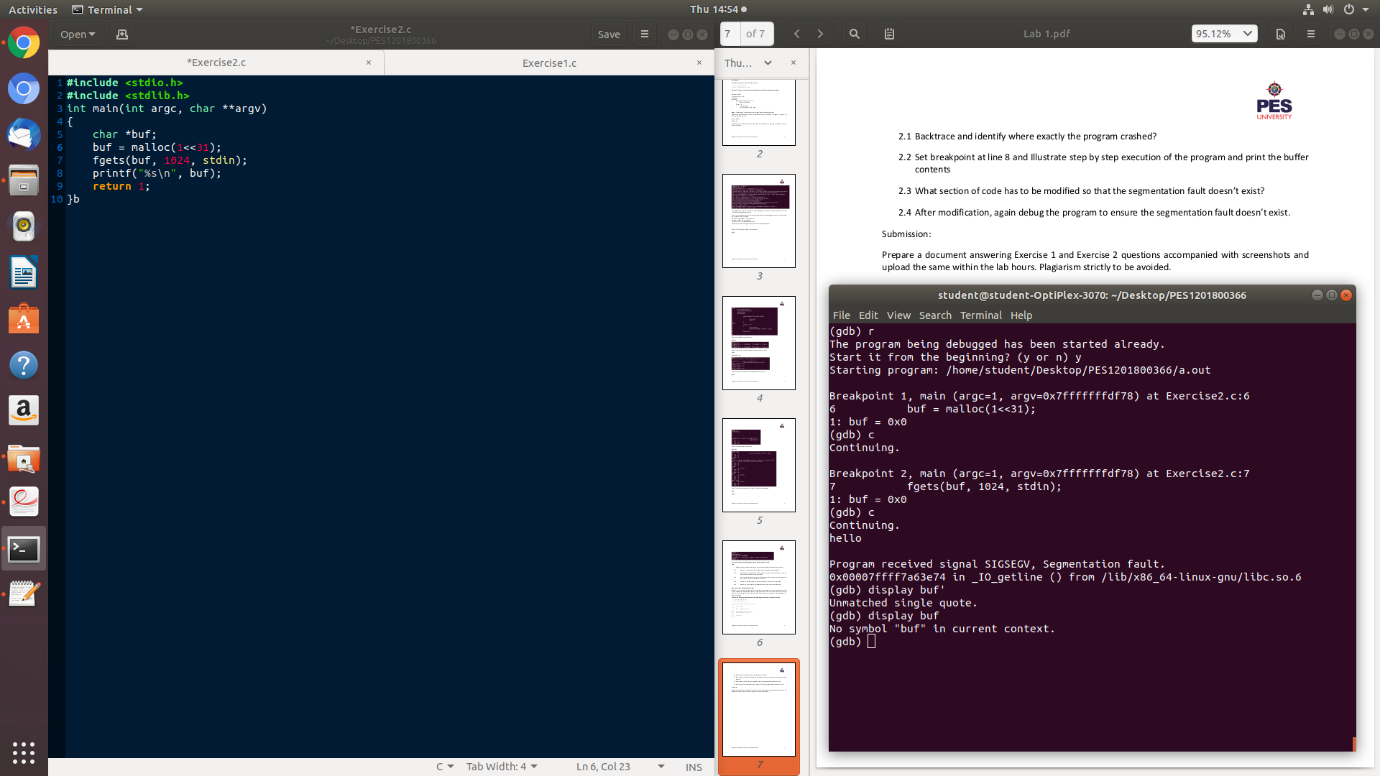
1. Backtrace and identify where exactly the program crashed?

Ans – On backtracing with the backtrace command, we find that the fgets() statement causes a Segmentation Fault. We verify this by stepping into every line and checking for termination.



1. Set breakpoint at line 8 and Illustrate step by step execution of the program and print the buffer contents

Ans – buf contents can be printed on every line using the watch command and then executing the program line by line.



1. What section of code has to be modified so that the segmentation fault doesn’t exist?

Ans – The given code allocates 231 bytes of memory for the character pointer. Since this exceeds the permissible size that can be allocated, on changing the argument inside the malloc() to 1<<30 which is 230 bytes, we can avoid the segmentation fault.

1. After modification, again debug the program to ensure the segmentation fault doesn’t exist.

Ans –

