COMP8049 – Embedded Systems Engineering - Labs

Completion Date: 21st October 2024

Value: 15 marks

On completion please zip up your files and upload to Canvas.

Q1

Write a c program which reads in a text file and finds all the identifiers in the file. An identifier is a token (sequence of characters) that starts with an underscore or a letter and continues with underscores or letters or digits.

You are to sort all the identifiers in the input text file and output them in alphabetical order.

You are also to output the line numbers on which the identifier occurred.

For example – given the following input file:

Hello yy
good (yy)
Hello

The program will output
Hello: 1,3
good: 2
yy: 1,2

(4 marks)

Q2

Create a model for the traffic light simulation and state machine that is detailed in section 10.4 of the traffic-light-simulator.pdf also here

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C10_FiniteStateMachines.htm

This model is a c program that runs on your laptop in Linux. You can use the code in traffic-model.c. You can then emulate a button press by simple key presses on the keyboard. For example:-

a) enter the letter N to indicate that the North button is on.

b) enter the letter E to indicate that the East button is on.

c) enter the letter B to indicate that both buttons are on.

d) any other key to indicate that both buttons are off

Please note you will need to use the Inputavailable function to first check that there is input from the keyboard. Otherwise the scanf will block waiting for input and your model will be unable to transition to a new state when the elapsed time designated for the state has been surpassed.

(4 marks)

Q3

Using the technical reference manual for the PL011 UART (pl011.pdf), give a brief explanation of the pl011_read function in the QEMU implementation of the PL011 UART - https://github.com/qemu/qemu/blob/master/hw/char/pl011.c.

(3 marks)

Q4

Using your code in Q2 build a traffic light controller model and simulation that is cross compiled for an ARM based SOC see here .

You can use the qemu-system-arm emulator https://wiki.qemu.org/Documentation/Platforms/ARM.

See here for steps to build a binary for the qemu emulated versatilepb board.

https://balau82.wordpress.com/2010/02/28/hello-world-for-bare-metal-arm-using-qemu/

See emulating-arm-pl011-serial-ports.pdf for an example which emulates the serial communications to the qemu emulated `versatilepb board`.

https://balau82.wordpress.com/2010/11/30/emulating-arm-pl011-serial-ports/

(4 marks)