CSCE 312 LAB 1

•••

By Dr. Eun Jung Kim

Welcome To

CSCE 312: Computer Organization

Please Sign Up

Piazza: https://piazza.com/tamu/spring2024/csce312

Lab Distribution

Lab contains total 35% of your course grade

- Lab assignments = 20%
    - 5 assignments (4% each)
    - Deadline (approx.): 1 week for Lab 1-2, 2 weeks for lab 3-5.
- Project = 15%
    - 4 weeks

Class Participation

- Class participation is mandatory in this lab.
- If your total participation is less than 80%(except university excused leaves)
    - You will get 5% deduction from your course grade.

Bonus Point Rules

There will be some bonus point

- Some for labs
- Some for projects

Bonus points will help you to recover any missing points

Group Formation

Final project and some labs will be group work.

Group size: 2-3 people. We encourage to form groups of 3.

Tentative Schedule

- Lab 1 - Jan 23, 2024 - Jan 31, 2024(1 week)
  - No Demo, Report deadline: Jan 31, 2024
- Lab 2 - Feb 01, 2024 - Feb 08, 2024(1 week)
  - Demo: Feb 08, 2024, Report Deadline: Feb 11, 2024
- Lab 3 - Feb 13, 2024 - Feb 22, 2024(2 weeks)
  - Demo: Feb 22, 2024, Report Deadline: Feb 25, 2024
- Lab 4 - Feb 27, 2024 - Mar 07, 2024(2 weeks)
  - Demo: Mar 07, 2024, Report Deadline: Mar 10, 2024

!!!Spring Break!!!!!

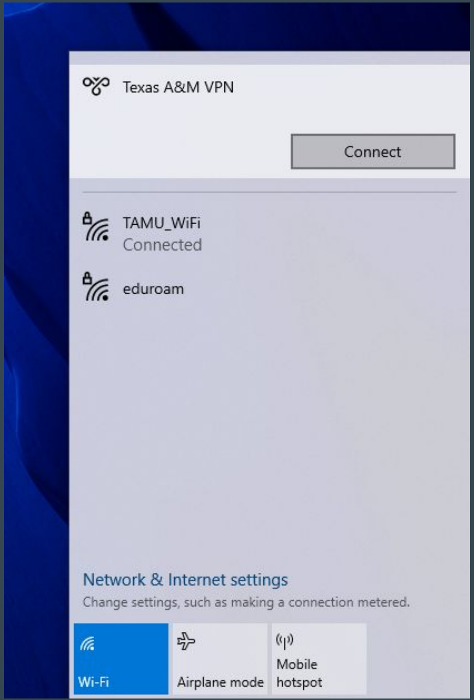
- Lab 5 - Mar 19, 2024 - Mar 28, 2024(2 weeks)
  - No Demo, Report Deadline: Mar 31, 2024
- Project: Apr 2, 2024 - Apr 30, 2024(4 weeks)

Demo and Report Policy

- For each lab, you need to submit a report and for some lab you need to show demo.
- For group works, one demo and report per group.
- Demo required for: Lab 2, Lab 3, Lab 4.
- Details regarding project will be shared later.

Setting up

- Set up tamu VPN to work at home
    - Link



Change settings, such as making a connection metered.

Using Linux Server

For Lab 1 and Lab 5, we will use ‘linux.cse.tamu.edu’ server to run some codes.

As ‘linux.cse.tamu.edu’ is linux based, we can

- Use the terminal of linux server to run any program
- But before that we need to copy file from the local machine to the server

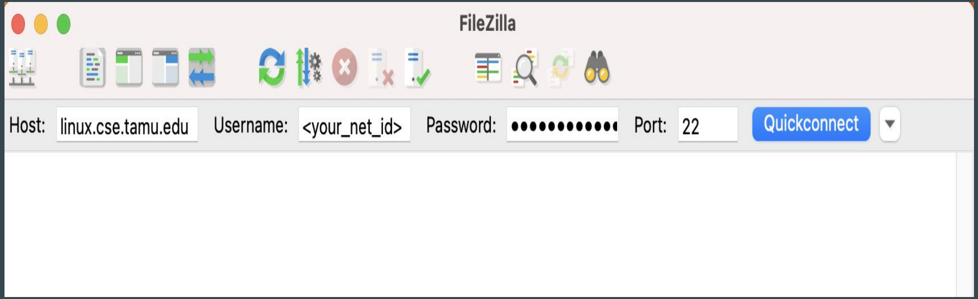
Both of these steps will require a home directory.

Copy files from Local machine to Server

We will use FileZilla as our FTP client. FileZilla is a free FTP solution for both client and server. We can copy file from a local machine to remote server using this tool.

Steps:

- Download FileZilla(link)
- To connect to the server, open FileZilla.
  - Add the following information
    - Host: linux.cse.tamu.edu
    - Username: <your_net_id>
    - Password: Your net_id password
    - Port Number 22
  - Click on QuickConnect. You should be connected to your home directory in the server.
  - Now you can drag and drop to copy files from local machine to your server.



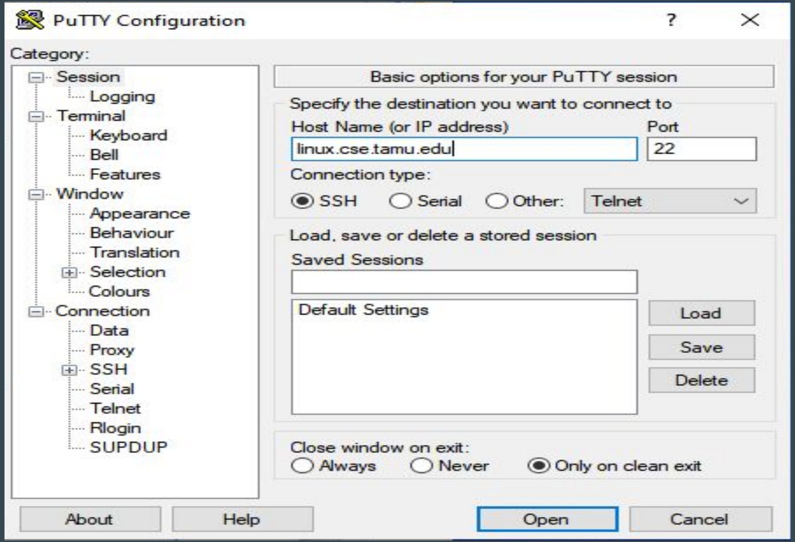
Host: linux.cse.tamu.edu Username: <your_net_id> Password: ●●●●●●●●●●●● Port: 22

Connecting Remote Terminal(Windows)

Install Putty.

Connection steps:

- Host Name: <net_id>@linux.cse.tamu.edu
- Port: 22
- Connection type: SSH
- Click ‘Open’
- Provide your <net_id> password and press enter.
- You should be logged in to your homepage



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): linux.cse.tamu.edu

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Connecting Remote Terminal(Mac/Linux)

You can use your terminal as the SSH client.

Connection Steps:

- Run “ssh <net-id>@linux.cse.tamu.edu” and enter your password.
- You should be logged in to your homepage

Create Your Home Directory

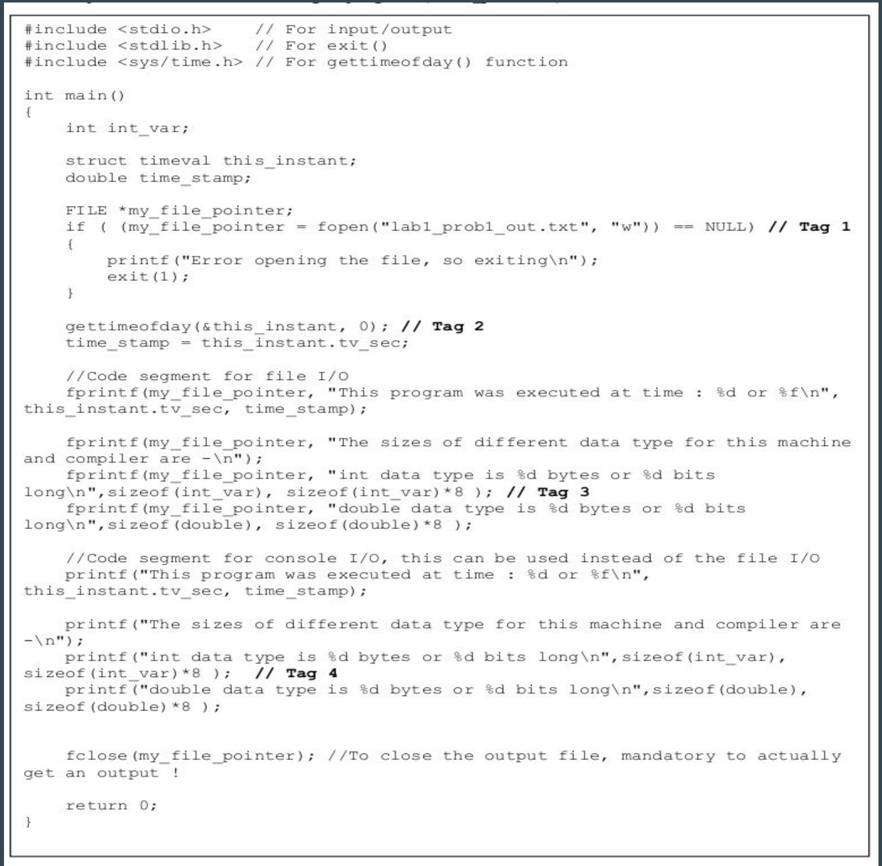
When connecting to the server, if you face errors such as ‘No home directory found’,

Please follow the steps mentioned here to set up the home directory.

Objectives

- Relate computer programs to a given real-life application
- Design C code to solve real-life problems using Boolean algebra
- Optimize code for real-life embedded systems

Problem 1



```
#include <stdio.h>     // For input/output
#include <stdlib.h>    // For exit()
#include <sys/time.h> // For gettimeofday() function

int main()
{
    int int_var;

    struct timeval this_instant;
    double time_stamp;

    FILE *my_file_pointer;
    if ( (my_file_pointer = fopen("lab1_prob1_out.txt", "w")) == NULL) // Tag 1
    {
        printf("Error opening the file, so exiting\n");
        exit(1);
    }

    gettimeofday(&this_instant, 0); // Tag 2
    time_stamp = this_instant.tv_sec;

    //Code segment for file I/O
    fprintf(my_file_pointer, "This program was executed at time : %d or %f\n",
this_instant.tv_sec, time_stamp);

    fprintf(my_file_pointer, "The sizes of different data type for this machine
and compiler are -\n");
    fprintf(my_file_pointer, "int data type is %d bytes or %d bits
long\n",sizeof(int_var), sizeof(int_var)*8 ); // Tag 3
    fprintf(my_file_pointer, "double data type is %d bytes or %d bits
long\n",sizeof(double), sizeof(double)*8 );

    //Code segment for console I/O, this can be used instead of the file I/O
    printf("This program was executed at time : %d or %f\n",
this_instant.tv_sec, time_stamp);

    printf("The sizes of different data type for this machine and compiler are
-\n");
    printf("int data type is %d bytes or %d bits long\n",sizeof(int_var),
sizeof(int_var)*8 );  // Tag 4
    printf("double data type is %d bytes or %d bits long\n",sizeof(double),
sizeof(double)*8 );


    fclose(my_file_pointer); //To close the output file, mandatory to actually
get an output !

    return 0;
}
```

Problem 1

1. Explain Tag 1, Tag 2, Tag 3 and Tag 4.
2. Run the code and show the output
   a. Command to compile the code: gcc <filename>.c -o <filename>.out
   b. Command to run the code: ./<filename>.out
3. Find out:
   a. Structure of 'timeval'
   b. Data type of 'tv_sec'

Problem 2

Same as problem 1, but need to find out bit and byte lengths of couple of more data types

Problem 3

Designing a decision control sub-system for a car

Problem 3

Input Sensors:

1. Driver Seat Belt Fastened(DSBF)
2. Engine Running(ER)
3. Doors Closed(DC)
4. Door Lock Lever(DLC)
5. Driver On Seat(DOS)
6. Key In Car(KIC)
7. Brake Pedal(BP)
8. Car Moving(CM)

Output Sensors:

1. BELL
2. Door Lock Actuator(DLA)
3. Brake Actuator(BA)

Our target is Enable/Disable the output sensors based on Input Sensors values
Ex: BELL = ER && !DSBF

Problem 3

What are the conditions????

- The BELL should chime/sound when the driver starts the engine without fastening his seatbelt
- The BELL should sound when the driver starts the engine without closing all the doors
- The BELL should be off as soon as the conditions change to normal, i.e. the two cases above do not occur.

There are couple of more conditions in the handout,
please take a look into that.

Problem 3

What you have to do here??

1. Make 3 truth tables for each of the actuator
2. Make 3 boolean expressions for each of the actuators
3. Write a C code to implement the subsystem(Initial code is given)
4. Run your code with given test cases.

For full requirements, check the handout.

Problem 3

Example of a truth table:

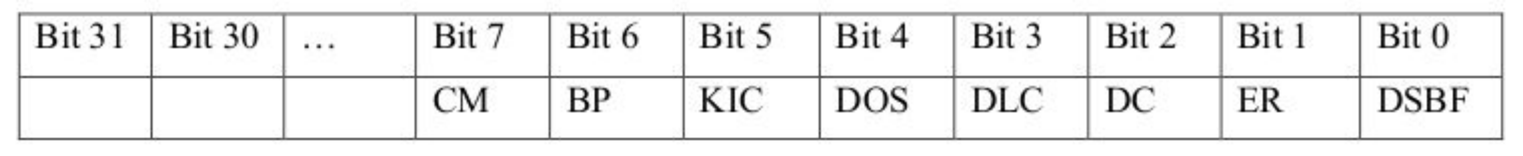
- The BELL should chime/sound when the driver starts the engine without fastening his seatbelt
- The BELL should be off as soon as the conditions change to normal, i.e. the above case doesn’t occur.

DSBF	ER	DC	DLC	DOS	KIC	BP	CM	BELL
0	0	x	x	x	x	x	x	0
0	1	x	x	x	x	x	x	1
1	0	x	x	x	x	x	x	0
1	1	x	x	x	x	x	x	0

Problem 4

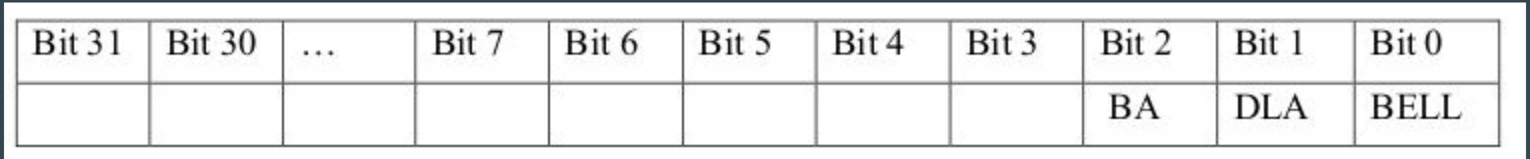
Same as Problem 3, just use bit masking to solve the problem

Sensor Inputs



Bit 31	Bit 30	…	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			CM	BP	KIC	DOS	DLC	DC	ER	DSBF

Actuator Outputs



Bit 31	Bit 30	…	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
								BA	DLA	BELL

Problem 4

Let’s say, we have sensor input = 2

What does it mean??

2 in binary = 0000 0000 0000 0000 0000 0000 0000 0010

Means:

- Engine is Running
- Driver seat belt is not fastened

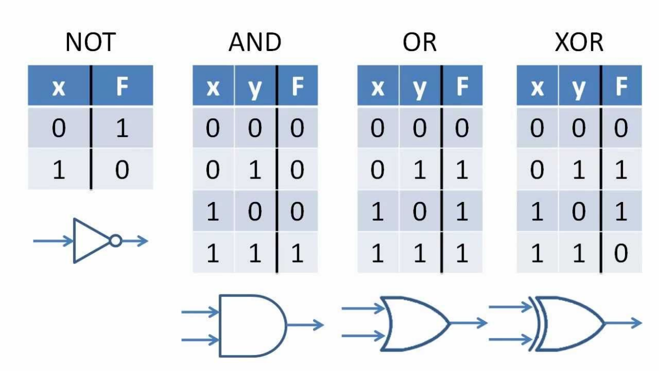
What should be the output??

- 1(0000 0000 0000 0000 0000 0000 0000 0001)

Problem 4

What concepts you should master for this problem?

- Different logical operators(And, Or, XOR, Not, etc)
- How to set a specific bit position(Hint: Use Or)
- How to reset a specific bit position(Hint: Use And)
- How to get the values of specific bit positions.
- Using left shift and right shift operators



NOT

x	F
0	1
1	0

AND

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Problem 4 Continue

- BELL = ER && !DSBF
- Let input 0011 1010 (driver seatbelt not fastened (1st bit = 0), engine is running (2nd bit = 1))
    - So we want to only consider 1st &2nd bit (mask)

- Using bitwise AND operation :
- Case 1:

```
    0011 1010  (input)
&  0000 0011  (mask);  considering 1st&2nd bit
—-----------------
    0000 0010   (result)  ==  bell ringing condition,  bit 0 (bell) of the output should be changed. So, bitwise or of 001 with
output last 3 bits will be XXX | 001 = XX1, which will set the BELL.
```

- Case 2:

```
    0011 1011 (driver seatbelt fastened & engine is running)
&  0000 0011 (mask)
—-------------------------
    0000 0011 (result)  != bell ringing condition, so no need to update output bit
```

- Therefore, if (input & 3) == 2, then output = output | 1

Problem 5

- Measure and compare the execution time between the implementations in problem 3 and 4.
- Timing code provided.

Thank You.

Any Questions??