Lab 1 Report

Huy Quang Lai
132000359

Texas A&M University

6 September 2022

An Aggie does not lie, cheat or steal.
Nor does an Aggie tolerate those who do.

Problem 1

a) Tag 1 tells the compiler to allocate 2-bytes to the stack and treat this value as a 16-bit signed integer.
Tag 2 tells the compiler to allocate a variable amount of bytes to the stack. This value would be treated as an address to FILE data. Additionally, Tag 2 also allocates space on the heap for the File itself
Tag 3 will send the formatted string to the file output stream.

b) Code compilation with output

```
gcc -std=c99 -o main.out lab1_prob1.c
./main.out
This program was executed at time : 1662492770 or 1662492770.000000
The sizes of different data type for this machine and compiler are -
int data type is 4 bytes or 32 bits long
double data type is 8 bytes or 64 bits long
```

Figure 1: Compilation and Output

c) The data type of "timeval" is a signed 16-byte integer. It indicates the number of second that have elapsed since 00:00:00 on January 1, 1970, Coordinated Universal Time.

Problem 2

Lab 1 > lab1_prob2_out.txt

```
1  Size of unsigned int is 4.
2  Size of double is 8.
3  Size of long is 8.
4  Size of long long is 8.
5  Size of char is 1.
6  Size of float is 4.
7  Size of struct timeval is 16.
```

Figure 2: Output

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

int main() {
    FILE* file;
    if ((file=fopen("lab1_prob2_out.txt", "w")) == NULL) {
        printf("Error opening the file, so exiting\n");
        exit(1);
    }

    fprintf(file, "Size of unsigned int is %lu.\n", sizeof(unsigned int));
    fprintf(file, "Size of double is %lu.\n", sizeof(double));
    fprintf(file, "Size of long is %lu.\n", sizeof(long));
    fprintf(file, "Size of long long is %lu.\n", sizeof(long long));
    fprintf(file, "Size of char is %lu.\n", sizeof(char));
    fprintf(file, "Size of float is %lu.\n", sizeof(float));
    fprintf(file, "Size of struct timeval is %lu.\n", sizeof(struct timeval));

    return 0;
}
```

Figure 3: Code

Problem 3

a) The Five Requirements

i. BELL = ER && !DSBF

ii. BELL = ER && !DC

iii. BELL = DSBF && !ER && DC

iv. DLA = !DOS && !KIC

v. BA = BP && CM

b) Truth Table

DOS	DSBF	ER	DC	KIC	DLC	BP	CM	BELL	DLA	BA
X	0	1	X	X	X	X	X	1	X	X
X	X	1	0	X	X	X	X	1	X	X
X	1	0	1	X	X	X	X	0	X	X
0	X	1	X	X	X	X	X	X	1	X
X	X	X	X	X	X	1	1	X	X	1

c) Code

```
void read_inputs_from_ip_if() {
    printf("Is the Driver on the Seat?\t");
    scanf("%u", &driver_on_seat);

    printf("Is the Driver Seat Belt Fastened?\t");
    scanf("%u", &driver_seat_belt_fastened);

    printf("Is the Enginer Running?\t");
    scanf("%u", &engine_running);

    printf("Are the Doors Closed?\t");
    scanf("%u", &doors_closed);

    printf("Is the Key in Car?\t");
    scanf("%u", &key_in_car);

    printf("Is the Door Lock Leaver activated?\t");
    scanf("%u", &door_lock_lever);

    printf("Is the Break Pedal activated?\t");
    scanf("%u", &brake_pedal);

    printf("Is the Car Moving?\t");
    scanf("%u", &car_moving);
}
```

Figure 4: Read Inputs

```c
void write_output_to_op_if() {
    printf("\n(BELL, DLA, BA):\t%u %u %u\n", bell, door_lock_actu, brake_actu);
}
```

Figure 5: Write Output

```c
// The code segment which implements the decision logic
void control_action() {

    /*
       The code given here sounds the bell when driver is on seat
       AND hasn't closed the doors. (Requirement-2)

       3. Provide your own code to do problems 3, which satisfies 5 requirements
    */
    if (engine_running && !doors_closed)
        bell = 1;
    else if (engine_running && !driver_seat_belt_fastened)
        bell = 1;
    else
        bell = 0;

    if (!(!driver_on_seat && key_in_car))
        door_lock_actu = 1;
    else if (driver_on_seat && door_lock_lever)
        door_lock_actu = 1;
    else
        door_lock_actu = 0;

    if (brake_pedal && car_moving)
        brake_actu = 1;
    else
        brake_actu = 0;
}
```

Figure 6: Logic

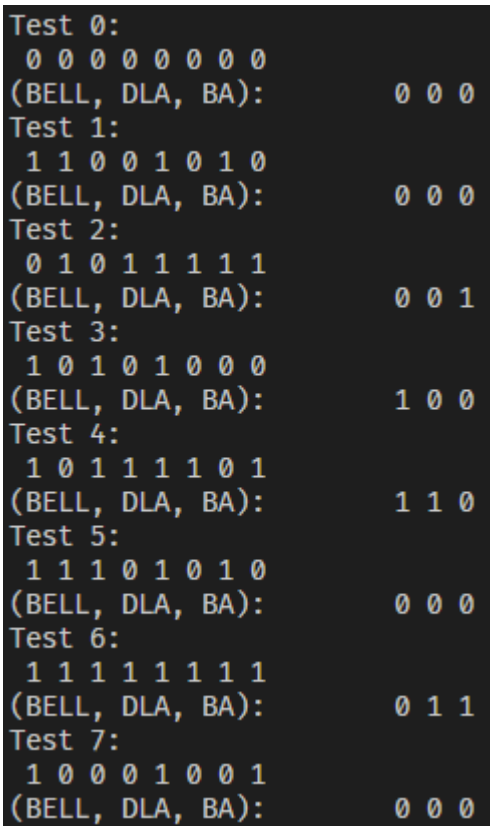

Figure 7: Output

Problem 4

```
enum Inputs { DOS = 1, DSBF = 2, ER = 4, DC = 8, KIC = 16, DLC = 32, BP = 64, CM = 128 };

//The code segment which implements the decision logic
void control_action(){

    /*
       The code given here sounds the bell when driver is on seat
       AND hasn't closed the doors. (Requirement-2)
       Replace this code segment with your own code to do problems 3 and 4.
    */

    //if (engine_running && !doors_closed) bell = 1;
    if ((input & 12) == 4)
        output = output | 1;
    if ((input & (ER + DSBF)) == 4)
        output |= 1;
    if ((input & (KIC + DOS)) == 14)
        output |= 2;
    if ((input & (BP + CM)) == BP + CM)
        output |= 4;
}
```

Figure 8: Code

```
Case 0:  0 0 0
Case 1:  0 0 0
Case 2:  0 0 1
Case 3:  1 0 0
Case 4:  1 1 0
Case 5:  0 0 0
Case 6:  0 1 1
Case 7:  0 0 0
```

Figure 9: Output

Problem 5

```
[lai.huy]@linux2 ~/CSCE 312/Lab 1> (14:33:16 09/06/22)
:: make build_5
rm -f *.out *.debug
gcc -o main.out lab1_prob5_4.c -lrt
./main.out
input signal: 0
output signal: 0
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 2328 nanoseconds
 The measured code took 0 seconds and  4000 nano seconds to run
rm -f *.out *.debug
gcc -o main.out lab1_prob5_3.c -lrt
./main.out
Is the Driver on the Seat?      0
Is the Driver Seat Belt Fastened?       0
Is the Enginer Running? 0
Are the Doors Closed?   0
Is the Key in Car?      0
Is the Door Lock Leaver activated?      0
Is the Break Pedal activated?   0
Is the Car Moving?      0

(BELL, DLA, BA):        0 0 0
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 3467 nanoseconds
 The measured code took 0 seconds and  2660 nano seconds to run
```

Figure 10: Execution time on linux.cse.tamu.edu

```
[lai.huy]@compute ~/CSCE 312/Lab 1> (14:33:52 09/06/22)
:: make build_5
rm -f *.out *.debug
gcc -o main.out lab1_prob5_4.c -lrt
./main.out
input signal: 0
output signal: 0
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 4647 nanoseconds
 The measured code took 0 seconds and  4294967075 nano seconds to run
rm -f *.out *.debug
gcc -o main.out lab1_prob5_3.c -lrt
./main.out
Is the Driver on the Seat?      0
Is the Driver Seat Belt Fastened?       0
Is the Enginer Running? 0
Are the Doors Closed?   0
Is the Key in Car?      0
Is the Door Lock Leaver activated?      0
Is the Break Pedal activated?   0
Is the Car Moving?      0

(BELL, DLA, BA):        0 0 0
Timer Resolution = 1 nanoseconds
 Calibrartion time = 0 seconds and 4384 nanoseconds
 The measured code took 0 seconds and  2713 nano seconds to run
```

Figure 11: Execution time on compute.cse.tamu.edu