MTech Refresher Module

System Programming Lab-1

Deadline: - 26/07/2024

- **Instruction:**

1. All the doubts regarding the lab should be posted on the Google Classroom.

2. Late submissions will not be considered.

3. Submit a zip file with the name <YourName>_<RollNo>.zip. It should contain 5 files: two .c files from the first question, along with the screenshot of its output, and one .asm file from question 2, along with a screenshot of its output.

4. Name the files in question 1 as question_1_a.c, question_1_b.c, question_1_screenshot and files in question 2 as question_2.asm and question_2_screenshot.

- Task before questions
  - Install gcc
  - Install VIM
  - Create a file using VIM
  - Execute a program using gcc

Solution:

On Ubuntu/Debian-based systems:

sudo apt install gcc vim

Create file in vim:

vim file_name.extension

Press i to enter insert mode and write your code.

```
#include <stdio.h>

int main() {
 printf("Hello, World!\n");
 return 0;
 }
```

Press Esc to exit insert mode.

Type :wq and press Enter to save and exit vim.

Compile Your Program

gcc -o executable_file_name filename.extension

./executable_file_name

Question 1:

Design a program using ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and

outputs it to the terminal. For example, if the first process sends the message ‘Sp Lab’, the second process will output ‘sP lAB’.

Question 2:
Write an assembly program to calculate the sum of two integers. Use **gdb** to see the final sum.

You need to save the file as <filename>.asm and compile by:-

yasm -f elf64 -g dwarf2 <filename>.asm
ld <filename>.o

Then use gdb to debug the program by:-

gdb ./a.out

While debugging, use ‘b _start’ to add a breakpoint to your program, then enter ‘r’ to run the program. Keep entering ‘n’ till you reach the end of your program. Now you can print the value in a register by ‘p (int)$<register name>’ or in a variable by ‘p (int)<variable name>‘.

Note: semi-colons in assembly code are used to signify the start of comments. Don’t add it to your programs.

Basic code for assembly program:-

```
        section .data   ;used to define and initialize global variables
a       dd 1            ;a is a variable, dd is a datatype called double word, which is of 4 bytes.
b       dd 4

        section .text   ;instructions of the program go here
        global _start   ;defining main point of enter

_start:
        ; code goes here
        ;Move values from memory to registers, add them and run using gdb
```