


# National University of Computer and Emerging Sciences, Lahore Campus

	<b>Course Name:</b>	Operating Systems	<b>Section:</b>	BCS-4J
	<b>Program:</b>	BSCS	<b>Semester:</b>	Spring 2023
	<b>Duration:</b>	2 Hours	<b>Total Marks:</b>	60
	<b>Evaluation Type:</b>	Lab Mid Exam	<b>Weightage:</b>	30%
	<b>Name:</b>		<b>Roll Number:</b>	

## Instructions:

- The quality of the code will affect the marks.
- Students will receive **ZERO** marks if the answers are plagiarized.
- Use of ANY helping material/code, or cell phones, **INTERNET** and flash drive are strictly prohibited.
- You can use Linux man pages for help.
- You must ensure proper submission of your code following the file naming instructions (given below).
- No queries will be entertained.
- Your submission will contain your **CODE**.
- Submission location: Google Classroom.

## File Naming Instructions:

- Name your each individual file as **ROLLNUMBER\_QUESTION\_FILENAME.c** for example **20L\_1234\_question1\_speaker.c**
- 5 marks will be deducted if the naming instructions are not followed.

---

## Question 1:

(45 Marks)

- Write and use **makefile** to **compile** all the programs.
- Create an application using 4 C files named **create\_fifo.c**, **main.c**, **worker1.c**, and **worker2.c**.
- **create\_fifo.c** creates a pipe named **fifo\_pipe**.
- **main** reads a sentence from an **input file** via **read()** system call, but have to use **FD # 0** and call appropriate **redirection** system calls.

i.e. **read(0, buffer, 200)**

- **main** also executes two programs **worker1** and **worker2**.
- **worker1** calculates the **frequencies** of the stop words from the sentence.
- It also writes the frequencies to the **fifo\_pipe** with **write** system call, but have to use **FD # 1** and call appropriate redirection system calls.

i.e. **write(1, stopword\_frequencies, size)**

- **worker2** counts the **number** of words in the sentence and **average word length**.
- It also writes the both things to the **fifo\_pipe** with **write** system call, but have to use **FD # 1** and call appropriate redirection system calls.

i.e. **write(1, sent\_length\_and\_avg\_word\_length, size)**

- **main** has to read the **output of both workers** and **display** them in the **screen** with proper formatting.

List of stop words: **[a, the, an, of, to, in, and]**

*Note 1: FD means File Descriptor*

*Note 2: Strictly follow the naming instructions given at the start of paper.*

## Question 2:

**(15 Marks)**

- Create a C program named **main.c**.
- The **main** program reads a string from user, which can be passed through command line arguments.
- The string should contain **single digit integers** and **special characters**.
- **main** then **creates a child** process, and passes the string to the child via **ordinary pipe**.
- The **child** performs following tasks on the string:
  - Removes all the **special characters** from the string.
  - Sorts the integers in **descending** order
  - **Sends** the sorted integers back to **main** via **ordinary pipe**
- **main** reads the data from pipe and prints it on the screen.

*Note 1: Following strings can be passed for testing the program:*

*12&2\*99*

*%3212+2*

*Note 2: Strictly follow the naming instructions given at the start of paper.*

\*\*\*\*\* GOOD LUCK \*\*\*\*\*