National University of Computer and Emerging Sciences

Operating Systems Lab (CL2006)

Lab Mid Exam (A)

Date: March 19th 2024

Course Instructor(s)

Ms. Anaum Hamid, Ms. Mubashra Fayyaz, Ms. Hania Usman, Mr. Muhammed Monis, Mr. Usama bin Umar, Ms. Zainab Asif Jawed, Ms. Fatima Gado, Mr. Ubaidullah, Ms. Hira Tunio, Mr. Ali Haider, Mr. Sameer Faisal, Mr. Nauman Rajpoot

Total Time: 90 minutes
Total Marks: 25
Total Questions: 03

Semester: SP-2024
Campus: Karachi
Dept: Computer Science

Submission Instructions:

- You must comment your student ID on top of each file. (Line#1 of your code).
- Name the file for each question according to Roll_No e.g. **k22-xxxx_Q1.c, k22- xxxx _Q2.c** etc.
- Create a ZIP folder of all your solutions and copy it in the local storage with the title **K22-xxxx_A** (Your paper Type).
- Submission are on local storage that can be accessed via the other location tab in explorer and then entering the address as **smb://172.16.5.43/** address in the dialog box.
- Enter your username as **khifast/K22xxxx** and its assigned password (**Default is Fast1234**).
- Zip folder needs to be pasted in the "**Exam Folder2024\teacherName\Your_Roll_No"** folder

____________________	________	________	________________
Student Name	Roll No	Section	Student Signature

CLO # 1: Understand and Analyze Command Line tools for Linux OS and Shell scripts for system level programming to automate tasks such as file management, system backups and software installations. (Lab # 1 and Lab # 3)

Q1. [= 9.5 marks]

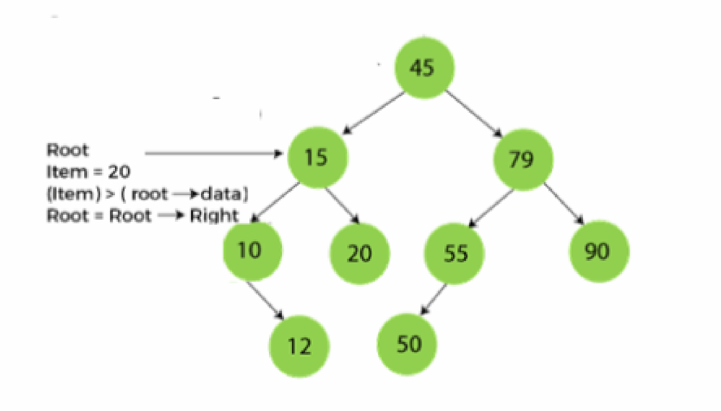
1. Create three shell script files in the arrangement of file1.sh, file2.sh, file3.sh.
2. File1.sh should create a c/cpp file with a code to sort an array using a sorting algorithm.
3. File2.sh should compile the said c/cpp file that was created to two files out.o and output.
4. File3.sh should provide parameters via the main function to the original c/cpp file.

Execution Step:

1. File1.sh will start first and once executed will start file2.sh
2. File2.sh will then execute and once completed will move to file3.sh
3. After file3.sh is completed the process call file.sh to display the file and its properties and to create a copy under a different name and delete the original file.
4. Order will be file1.sh > file2.sh > file3.sh > file2.sh

CLO # 2: Gain hands on experience in writing code that interacts with operating system services related process and files system, multi-thread programing and different synchronization primitives.

Q2. [= 6 marks]



Implement a tree like structure but the output of this tree will be in post order **i.e. 12,10,20,15,50,55,90,79,45.** Before traversing create a .txt file that will display how many children, grandchildren and parent nodes exists in the tree. When traversing each node append the output to a file as well. Note: Nodes should be in sync with their parent and grandparent nodes only the output of the child/parent nodes will change.

CLO # 3: Understand and implement code that handles information sending between different processes via various means.

Q3. [= 9.5 marks]

Consider a scenario where there exists a team of 4 members. The 2 members are in a parent/child configuration while the other two are just in their parent configuration. You are task of sending information (in our case sending morse code in the form of an string as **. ___ …. __**) via the child process of the first two member to the 3 member of the team. The third member of the team would intercept the code and translate it to its desired output (In our case it would be Hello World) and send it to the 4 member of the team. The 4 member would then append another string to the translated string (in our case that would be Hello Linux World) and send it back to the 2 parent/child member where the parent would receive it and print the message and the text message received. Note: You would need **three file minimum** to achieve this and an **appropriate IPC Method.**

Example:

File 1 Parent = Waiting
File 1 Child = **. ___ …. __**
File 2 Parent = Message Received
File 2 Parent = Hello World
File 3 Parent = Message Received
File 3 Parent = Hello Linux World
File 1 Parent = Message Received
File 1 Parent = Hello Linux World