

**BAHRIA UNIVERSITY (KARACHI CAMPUS**)

OPEN ENDED LAB II – **Fall22**

**(**System Programing (LAB) **CSC-454)**

Class: BSE [4]-5 (B) (Morning)

Course Instructor: **Engr Rizwan Fazal / Engr Rehan Baig** Time Allowed: **1.5 Hour**

Max Marks: **6**

Student’s Name: M Muaz Shahzad Reg. No: 02-131202-081

Instructions:

1. Submit your answers within file against each question with screenshot of both code and solution output.
2. File must be submitted in .pdf.

**[CLO#05, 6 marks]**

# SCENARIO:

**You are working as a system engineer in a Microsoft vendor company that creates Apps for Microsoft store.**

# Your Project manager assigned you a task to design an application for code editor for Microsoft store. For that you need to analyze the basics of NotePad/WordPad applications that comes built-in with Microsoft windows. You need to create a process and analyze the following for notepad and WordPad.

**Q1:** Run a loop or Use Recursion which enable program to print 5 times following for both Notepad and WordPad **(versionId, ThreadId, processId)**, meanwhile use exit thread function that-should be interrupt when counter reaches on 4rth iteration. (**4 Marks**)

**Code**

#include <iostream>

#include <iostream>

#include <thread>

#include <unistd.h>

void print\_info(int iteration) {

if (iteration == 4) {

pthread\_exit(NULL);

}

std::string AppName;

if (iteration % 2 == 0)

AppName = "Note\_pad";

else

AppName = "Word\_pad";

std::cout << "AppName: " << AppName << ", Thread\_id: " << std::this\_thread::get\_id() << ", Process\_id: " <<

getpid() << std::endl;

}

int main() {

std::cout << "System Programmin OEL 2:\n" << std::endl;

std::cout << "Create By M Muaz Shahzad" << std::endl;

for (int id = 0; id <= 4; id++) {

std::thread put(print\_info, id);

put.join();

}

return 0;

}

**Output**

**Text, letter

Description automatically generated**

**Q2:** Write a code for any two synchronization objects from following. (**2 Marks**)

* 1. Events
  2. Semaphores
  3. Mutexes

Code:

#include <iostream>

#include <thread>

#include <mutex>

#include <semaphore.h>

#include<vector>

sem\_t semaphore;

std::mutex mutex;

std::vector<int> shared\_resource;

void addToResource(int value) {

sem\_wait(&semaphore);

std::lock\_guard<std::mutex> lock(mutex);

shared\_resource.push\_back(value);

sem\_post(&semaphore);

}

int main() {

std::cout << "System Programmin OEL 2:\n" << std::endl;

std::cout << "Create By M Muaz Shahzad" << std::endl;

sem\_init(&semaphore, 0, 1);

std::thread t1(addToResource, 1);

std::thread t2(addToResource, 2);

t1.join();

t2.join();

std::cout << "Final value of shared resource: [ ";

for (auto i : shared\_resource)

std::cout << i << " ";

std::cout << "]" << std::endl;

sem\_destroy(&semaphore);

return 0;

}

**Output**

