|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **Operating System** | **Course Code:** | **CS-2006** |
| **Program:** | **BS** | **Semester:** | **Fall 2023** |
| **Duration:** | **20 Minutes** | **Total Marks:** | **15** |
| **Paper Date:** | **19-10-23** | **Weight** | **3.3 %** |
| **Quiz:** | **Quiz:2 sample 1** | **Page(s):** | **3** |
| **Name:** |  | **Roll No.** |  |

**Q:1-** Write a C program called “sample.c” that does the following: **[10]**

1. Takes an integer argument (say, **3**) from the command line OR Hard-coded user values.
2. The main process will create two threads by using (**pthread\_create**) command:
   1. First thread function will compute let say **z=**(**a+b+c)** (sum of positive integers up to 3) and pass out the result to main process.
   2. Second thread function will apply square root on the output produce by first thread function **sqrt(z)** and pass out the result to main process.
3. Main process thread will have to wait for the spawned threads to complete its work, then prints out the message “*Done*” and both Sum and **sqrt** of **3** numbers.

*NOTE: You have to do proper error handling.*

**Sample execution** (assuming the executable is called sample): bash$ ./sample

[ID = 101] Sum of positive integers a,b and c = 9

[ID = 102] square root of 9 is= 3

[ID = 100] Done

**Q:2-** Write the output of the following code, also justify your answer for some multithreaded and multicore systems: **[5]**

|  |  |
| --- | --- |
| void \*thread\_main(void \*p)  {  int \*p\_int = (int\*) p;  int x = \*p\_int;  x += x;  pid\_t pid= fork();  if(pid>0)  {  x=x+10;  \*p\_int = x;  }  return NULL;  }  int main()  {  int data = 1;  pthread\_t one, two;  pthread\_create(&one, NULL, thread\_main, &data);  pthread\_join(one, NULL);  printf("after first join %d\n", data);  pthread\_create(&two, NULL, thread\_main, &data);  pthread\_join(two, NULL);  printf("After second join %d\n", data);  return 0;  } | Process tree:  Output: |

Ans: