B.Sc. Engg. CSE 6th Semester 24 May 2024

28

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION SUMMER SEMESTER, 2022-2023
DURATION: 3 HOURS FULL MARKS: 150

CSE 4649: Systems Programming

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.
[For all the questions, assume **64-bit** system unless otherwise mentioned]

1. a) Identify and mention all the dead codes of the program given in Code Snippet 1. 6 (CO2) (PO2)

```
int main() {
    int num = 0x1569 * 0x69;
    int c = 0x69;
    if (c > 0)
        printf("Hello\n");
    if (num < 0x48)
        printf("Hi\n");
    else
        return 0x48;
    printf("World\n");
}
```

Code Snippet 1: A C program for Question 1.a

b) Explain integer conversion rules with code examples in C. You can write multiple code examples if needed. The code examples must demonstrate all the rules in action. 10 (CO1) (PO1)

c) Explain the concept of memory aliasing with appropriate code example in C. 5 (CO1) (PO1)

d) State whether the following statements are correct or incorrect. 4 (CO1) (PO1)

i. Calling pthread_exit() in main function will terminate the whole process immediately without checking anything.
ii. Opening the same file multiple times with open() system call will return different file descriptors.
iii. After termination, the init process is reaped by its parent.
iv. fork() returns Process ID (PID) of child in parent and 0 in child.

2. a) Explain the concept of reaping of child processes by the parent process. How the Operating System (OS) handles zombie processes? 4 + 3 (CO1) (PO1)

b) Write a program which will print the contents of current directory in a child process and then print the string "Hello World" in the parent process. The string must be printed by the parent only after the child process is finished. 8 (CO2) (PO2)

c) Considering a 32-bit system, discuss the output of the program in Code Snippet 2. 10 (CO2) (PO2)

```c
int main() {
    long a[3] = {1, 2, 3};
    long long ll = 6;
    printf("%d %d %d %d\n", sizeof(&a), sizeof(a), sizeof(a[0]),
        sizeof(ll));

    short int i = -1569;
    unsigned short ui = i;
    int j = i;
    char c = j;
    printf("%d %x %d\n", ui, j, c);
}
```

Code Snippet 2: A C program for Question 2.c

3. a) Consider the following C program in Code Snippet 3 where the function `calcSum()` will be run inside multiple threads.

```c
struct {
    int sum;
    int prod;
} result;

void *calcProd(void *arg) {
    result.prod = result.sum * (int)arg;
    pthread_exit(NULL);
}

void *calcSum(void *arg) {
    result.sum += (int)arg;
    calcProd(arg);
    pthread_exit(NULL);
}

int main() {
    pthread_t t[2]; // for creating two threads
    // Thread creation
    // and
    // joining code
    printf("%d %d\n", result.sum, result.prod);
}
```

Code Snippet 3: A C program for Question 3.a

i. What is race condition in concurrent programming? Explain the reason of race condition and show possible scheduling of the two threads to demonstrate race condition in the program of Code Snippet 3. You can consider dummy values as function arguments. Note that, function `calcSum()` would be the first running function inside all threads. 2 + 8 (CO3) (PO2)

ii. Rewrite the program given in Code Snippet 3 using a synchronization primitive so that the race condition does not happen anymore. Only showing necessary lines would be sufficient. 5 (CO3) (PO2)

b) Describe file descriptor table. Explain with necessary diagrams why any write operation on a file by a child process is also reflected in the parent process. 10 (CO1) (PO1)

4. a) How pipes facilitate Inter Process Communication (IPC)? Explain with appropriate example and diagrams. 5 (CO1) (PO1)

b) Consider the partial C program given in Code Snippet 4. 20 (CO2) (PO2)

```c
int pipefd[2]; // global variable

void *readFunc(void *arg) {
    char *filename = (char *)arg;
    /*
        TODO
    */
    pthread_exit(NULL);
}

void *writeFunc(void *arg) {
    char *fp = (char *)arg;
    /*
        TODO
    */
    pthread_exit(NULL);
}
int main() {
    char *readFileName = "peace.txt";
    char *writeFileName = "ranger.txt";
    pipe(pipefd);

    pthread_t threads[2];
    /*
        TODO
        thread creation and joining code
    */
    close(pipefd[0]);
    close(pipefd[1]);
    return 0;
}
```

Code Snippet 4: A C program for Question 4.b

Complete the above program which will create two threads and share data between the threads using a pipe. One thread will run the function `readFunc()` which will read from the file "peace.txt" and write its contents to a pipe. Another thread will run `writeFunc()` which will read the data sent from `readFunc()` through the pipe and write the data to file "ranger.txt". Thread running `readFunc()` must finish its execution before the other thread starts. You can assume the files already exist and the size of data passed between two threads will not exceed 40 bytes. Note that, you must write the whole program including the given lines.

5. a) Explain the differences between processes and threads. Mention one scenario where a process is a better choice and one scenario where a thread is better. Provide appropriate justification in favor of your answer. 7 + 6 (CO1) (PO1)

   b) Suppose, a program uses `open()` system call to open a file and `pthread_create()` function to create a thread. Assume both of these functions encountered some error while executing. As a programmer, you want to know what caused the error. Write appropriate C code to see the corresponding error messages from both of these functions. 6 (CO2) (PO2)

   c) Explain the concept of link and reference counts with appropriate examples. Mention when these counts increase or decrease. 6 (CO1) (PO1)

6. a) Explain how mutex prevents race condition in multithreaded programming. Provide appropriate code example in C. 10 (CO1) (PO1)

   b) Draw a diagram showing all the phases of C compilation system and their corresponding input/output files. 5 (CO1) (PO1)

   c) Write short notes on any two of the following with relevant examples: 5 × 2 (CO1) (PO1)

      i. Concurrency and Parallelism
      ii. Endianness
      iii. Instruction Set Architecture (ISA)