United International University
Department of Computer Science and Engineering

CSI 309/CSE 4509: Operating System Concepts/Operating Systems

Midterm Examination: Spring 2024

Total Marks: 30 Time: 1 hour 30 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1. (a) We are pleased to announce the successful completion of the 7th UIU convocation. In this process, every student undergoes registration, payment, and the acquisition of a token. These services are facilitated through a system where multiple students can concurrently access it, and each student is accorded equal priority. Which operating system (OS) is suitable for that system - **justify** your answer. [2]

   (b) A game server needs to manage communication between players in an online multiplayer game, ensuring smooth gameplay and efficient data transfer. **Explain** how communication will happen if the specific address is unknown to the game server. [3]

2. Consider the following code: Given the initial process ID is 164071, subsequent process IDs will increment by one with each new process created.

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#define SIZE 5

int nums[SIZE] = {1,2,3,4,5};

int main(){
    int sum = 0;
    pid_t pids[2];
    for(int i=0;i<2;i++){
        wait(NULL);
        pids[i] = fork();
    }
    if(pids[0] && pids[1]){
        sleep(10);
        for(int i=0;i<SIZE;i++){
            sum += nums[i];
        }

    }
    else if(pids[0]==0 && pids[1]){
        for(int i=0;i<SIZE;i++){
            sum -= nums[i];
        }
    }
    else if(pids[0] && pids[1]==0){
        for(int i=0;i<SIZE;i++){
            sum *= nums[i];
        }
        exit(0);
    }
    else{
        sleep(5);
        printf("You are In  a Trap!\n");
    }
    printf("Sum:%d\n",sum);
    return 0;
}
```

   (a) **Find** the possible output for the above code. [3]

(b) **Write** the process ID of the process that is a zombie or orphan process by drawing process tree. [2]

3. **Find** the output of the following: [5]

```
#include <bits/stdc++.h>
using namespace std;
#define NUM_OF_PERSON 3
bool socks[NUM_OF_PERSON]={1,1,1};

void acquire(bool *available){
    while(*available==false);
    *available = false;
}
void release(bool *available){
    *available = true;
}
void outing(){
    printf("Friends are outing now\n");
}
void jumping(){
    printf("Friends are jumping now\n");
}
void friends_gang(int i){
    if(i+1 != NUM_OF_PERSON){
        printf("Friend #:%d\n",i+1);
        outing();
        acquire(&socks[i]);
        printf("One sock is picked\n");
        sleep(10);
        acquire(&socks[(i+1)%NUM_OF_PERSON]);
        jumping();
        release(&socks[i]);
        release(&socks[(i+1)%NUM_OF_PERSON]);
        outing();
    }
    else{
        printf("Friend #:%d\n",i+1);
        outing();
        acquire(&socks[(i+1)%NUM_OF_PERSON]);
        printf("One sock is picked\n");
        acquire(&socks[i]);
        jumping();
        release(&socks[i]);
        release(&socks[(i+1)%NUM_OF_PERSON]);
        outing();
    }
}
int main()
{
    thread threads[NUM_OF_PERSON];
    // Create Friend threads
    for (int i = 0; i < NUM_OF_PERSON; ++i) {
        threads[i] = thread(friends_gang,i);
    }

    // Wait for threads to finish
    for (int i = 0; i < NUM_OF_PERSON; ++i) {
        threads[i].join();
    }

    return 0;
}
```

4. Suppose a multiprocessor system has 4 processors labeled P_1, P_2, $P_3$and P_4 respectively. The queue ID for processor P_i is Q_i. The details of the processes can be found in 1a and queues in 1b. Additionally, the constraints posed by the system can be found in 1c.

Process ID	Process type	Priority	Arrival Time	Burst Time
1010	Batch	1	0	7
920	Real Time	1	2	4
420	Batch	2	1	3
69	Interactive	4	2	5
4444	Real Time	3	4	4
220	Interactive	5	5	5

(a) Details of the Processes

Queue	Algorithm	Remarks
Q_1	Shortest Job First	Non-preemptive version
Q_2	First Come First Serve	-
Q_3	Round Robin	Time Quantum=3
Q_4	Priority	-

(b) Details of the Queues

Constraints
The order of filling up the queues is $Q1 \rightarrow Q2 \rightarrow Q3 \rightarrow Q4$
The smallest numerical priority represents the highest priority value.
Processor 3 cannot have interactive processes.

(c) Constraints of the system

Table 1: Necessary information for question 4

(a) **Calculate** turnaround time, response time, and wait time for all the processes [3]

(b) If all four processors start at timestep 0 and run until the completion of all the processes, **calculate** CPU utilization for all four processors. [2]

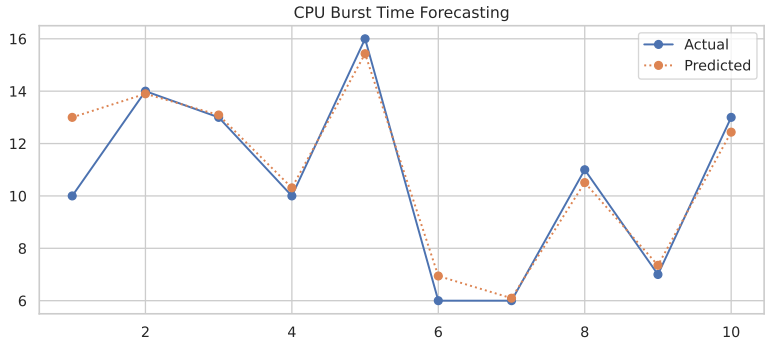
5. For the following piece of code, answer the following questions.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    pid_t p1 = fork() + fork();
    pid_t p2 = fork() && fork();

    if (p1 > 0 && p2 > 0){
        printf("This is fine.");
    }
    else if (p1 == 0 && p2 > 0){
        printf("This is workable.");
    }
    else if (p1 > 0 && p2 == 0){
        printf("Are you serious?");
    }
    else{
        printf("This is ****** insane.");
    }

}
```

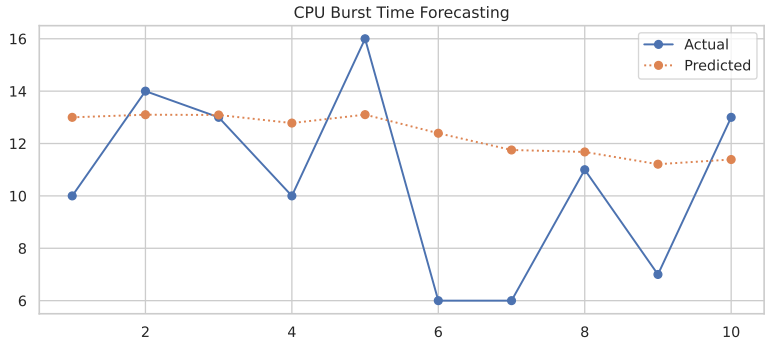
(a) **Draw** the process tree. [2]

(b) **Find** one of the possible outputs. [3]

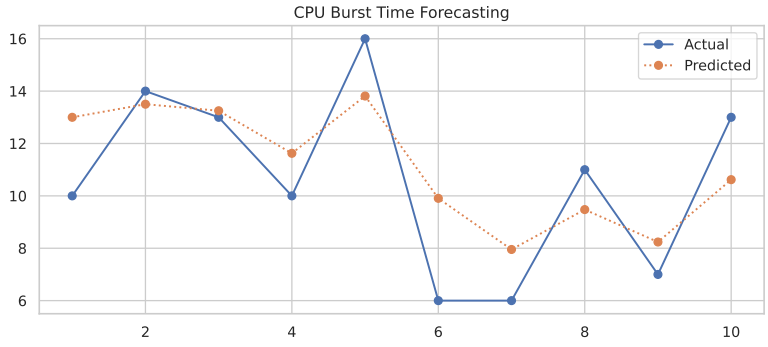
6. (a) For a system, the actual and predicted burst times for ten time-steps for 3 different values of α are as follows. The three values of α are 0.1, 0.5, and 0.9. Which graph corresponds to which value of alpha? **Explain** within five sentences. [3]



(a) Alpha = α_1



(b) Alpha = α_2



(c) Alpha = α_3

Figure 1: CPU Burst Time Forecasting

(b) How many children processes will be created in the following code? **Explain** within two sentences. [2]

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n = 1024;
    for (int i = 0; fork() < n; i++);
    return 0;
}
```