Operating Systems, Evaluation 2

CS30002, Spring 2021

12:00 noon to 1:10 pm, 17th March, 2021
Full marks: 40
Answer ALL questions

IMPORTANT INSTRUCTIONS

Taking the exam: You need to log into Google meet (personalized links will be provided close to the exam date), keep your video on during taking the test (so that we can monitor you during the exam). **YOU HAVE TO USE PEN AND PAPER TO GIVE THE EXAM.**

Decorum: Throughout the examination, you are strictly expected to have your cameras on, directing towards your work-space including yourself. Arrange your laptops/desktops/mobiles beforehand to save time during the examination. Disconnecting video for a long duration will be grounds for suspecting malpractice.

You need to keep your workplace and your hands visible to us. However, avoid the visibility of your answers to the rest of students. Once you open your question paper, refrain yourself from using your PC/laptop/mobile for searching for anything or typing during the exam.

Submission: You can do either of two things -- (i) take pictures of your answer script pages, name the pictures page1.jpg, page2.jpg, page3.jpg, etc., zip the pictures and upload the zipped file via CSE Moodle, or (ii) Put all the pages sequentially in a pdf file and upload the pdf to CSE Moodle. Note that the scheduled time includes both writing time and time needed to upload your answer-script.

Submissions must be through the course Moodle, by 1:10 PM (according to the Moodle clock). The actual exam will be from 12:00 to 1:00 PM and you have a dedicated 10 minutes time to upload the exam to moodle.

If you miss this deadline for Moodle submission, then you need to email your submission to TA Anju Punuru (anjupunuru@gmail.com) within 1:20 PM; there will be a penalty of 10 marks for such late submissions. No submission will be allowed after 1:20 PM. If any submission reaches Anju's mailbox after 1:20 PM (according to the timestamp of the mail), it will be rejected.

Tip: Install Adobe scan on your mobile phone to make the whole process easier. In that case, your laptop acts as a camera, while you are using your mobile for checking the questions, scanning and uploading the answers. [This is only a suggestion that can make things simpler for you, not a necessity.]

Policies: Note that, if we face problems with your answer script, e.g., cannot open your submitted zipped file, cannot read the text in pictures (due to bad resolution), cannot determine the page order from the file names (or if the pages in the pdf are jumbled up), it will affect your marks.

Malpractice: If any group of students is found to have similar answers/working in their answer sheets, ALL of them will receive the maximum penalty with no grace. We will not distinguish between who supplied answers and who copied; everyone involved will receive the maximum penalty. We expect you to NOT take help from the internet, your copies, textbooks, slides or video recordings during the exam. Note that this is NOT an open-book exam. Also, you should NOT discuss answers with anyone during the scheduled exam time. If found otherwise, you will be penalized.

PLESE WRITE YOUR NAME AND ROLL NO. ON THE TOP OF THE FIRST PAGE OF YOUR ANSWER SCRIPT. WE WILL NOT EVALUATE YOUR ANSWER SCRIPT WITHOUT IT.

Question 1: For each question below, select all of the choices that apply and justify your answer within 1—3 sentences. *No marks will be awarded if no justification is given / the justification is not correct*. The Justification should be between 1-3 sentences (marks might be deducted for longer justifications). Assume all function calls (pthread, file read/write etc.) succeed. **[4 × 2 = 8]**

1.1. Which of the following could be the contents of out.txt after all processes of the code snippet below got executed?

```
int main(int argc, char** argv) {
   if (fork() == 0) {
     int fd1 = open("out.txt", O_WRONLY);
      write(fd1, "a", 1);
   } else {
      int fd2 = open("out.txt", O_WRONLY);
      write(fd2, "b", 1);
   }
}
```

(i) empty
(ii) a
(iii) b
(iv) aa
(v) ab
(vi) ba
(vii) bb
(viii) none of these

1.2. Which of the following could be the contents of out.txt after all processes of the code snippet below got executed?

```
int fd;
void* helper(void* arg) {
           write(fd, "a", 1);
       }
int main(int argc, char** argv) {
    fd = open("out.txt", O_WRONLY);
    pthread_t thread;
    pthread_create(&thread, NULL, helper, NULL);
    write(fd, "b", 1);
}
```

(i) empty
(ii) a
(iii) b
(iv) aa
(v) ab
(vi) ba
(vii) bb
(viii) none of these

1.3. Which of the following could be the contents of out.txt after all processes of the code snippet below got executed?

```
       int fd;
       void* helper(void* arg) {
           write(fd, "a", 1);
           pthread_exit(NULL);
       }
       int main(int argc, char** argv) {
           fd = open("out.txt", O_WRONLY);
           pthread_t thread;
           pthread_create(&thread, NULL, helper, NULL);
           write(fd, "b", 1);
          pthread_exit(NULL);
       }
```

(i) empty
(ii) a
(iii) b
(iv) aa
(v) ab
(vi) ba
(vii) bb
(viii) none of these

1.4. Which of the following could be the contents of out.txt after all processes of the code snippet below got executed?

```
int fd;
char buffer;
void* helper(void* arg) {
    buffer = ’a’;
    write(fd, &buffer, 1);
    pthread_exit(NULL);
}
int main(int argc, char** argv) {
    fd = open("out.txt", O_WRONLY);
    pthread_t thread;
    pthread_create(&thread, NULL, helper, NULL);
    buffer = ’b’;
    write(fd, &buffer, 1);
    pthread_exit(NULL);
 }
```

(i) empty
(ii) a
(iii) b
(iv) aa
(v) ab
(vi) ba
(vii) bb
(viii) none of these

Question 2: Answer the following. **[4 + 4 + 4 = 12]**

2.1. Consider a program consisting of 3 concurrent processes P0, P1 and P2 as shown below, and 3 binary semaphores that are initialized as: S0 = 1, S1 = 0; S2 = 0. How many times will process P0 print ‘0’?

Process P0	**Process P1**	**Process P2**

`while (true) {`
`  wait (S0);`
`  print ‘0’;`
`  signal (S1);`
`  signal (S2);`
`}`
	`wait (S1);`
	`signal (S0);`
		`wait (S2);`
		`signal (S0);`

2.2. Consider the following solution for the Producer-Consumer problem that uses two binary semaphores **n** and **s**, initialized to 0 and 1 respectively. Is this given solution correct? If yes, please explain why, if not, please explain why not.

```
void producer()
{
  while (true) {
    produce();
    wait (s);
    addToBuffer();
    signal (s);
    signal (n);
  }
}
```

```
void consumer()
{
  while (true) {
    wait (s);
    wait (n);
    removeFromBuffer();
    signal (s);
    consume();
  }
}
```

2.3. Consider two processes P1 and P2 accessing the shared variables X and Y protected by two binary semaphores SX and SY respectively, both initialized to 1. Complete the entry and exit sections of the following codes such that the processes can update the shared variables atomically.

```
P1: while (true) do {
    <entry section>
    X = X + 10;
    Y = Y – 20;
    <exit section>
}
```

```
P2: while (true) do {
    <entry section>
    Y = Y + 20;
    X = X – 10;
    <exit section>
}
```

Question 3. Answer the following: **[3 + 4 + 5 = 12]**

3.1. Suggest a solution to the deadlock avoidance problem using resource allocation graph, where there exist single instances of each resource type.

3.2. A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
Process A:	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1
Process B:	2 0 1 1 0	2 2 2 1 0	
Process C:	1 1 0 1 0	2 1 3 1 0	
Process D:	1 1 1 1 0	1 1 2 2 1	

What is the smallest value of x for which this is a safe state?

3.3. N processes share M resource units that can be reserved and released only one at a time. The maximum need of each process does not exceed M, and the sum of all maximum needs is less than M+N. Show that a deadlock cannot occur.

Question 4. Answer the following: **[3 + (2.5 + 2.5) = 8]**

4.1. Three concurrent processes P1, P2 and P3 are concurrently updating a shared variable var (with initial value of 110) as follows:

```
P1:     var = var – 15;
P2:     var = var + 25;
P3:     var = var * 3;
```

What can be the possible maximum and minimum values of *var* after execution of the three processes? Show the execution order of P1, P2, P3 in the cases of the possible maximum and minimum values.

4.2. Consider three threads in a concurrent program uses semaphores S1, S2, S3

Thread 1	Thread 2	Thread 3

L1: wait(S3);
Print “2";
signal(S2);
goto L1;
	L2: wait(S1);
	Print “6";
	signal(S3);
	goto L2;
		L3: wait(S2);
		Print “1";
		signal(S1);
		goto L3;

a) Are there initial values of S1, S2, S3 so that the threads together can print a string that begins with 16216216216216? If yes, give the initial values of S1, S2, S3 and tell your reasoning. If not justify why not (no marks without justification)

b) If the initial values are S1 = 2, S2 = 6, S3 = 1, is it possible for the threads to print 1122622 together? Justify why or why not.