Operating Systems - Monsoon 2020 Take Home Quiz 2 (Total points: 70)

Sambuddho Chakravarty, Arani Bhattacharya

November 16, 2020

Submission deadline: November 18, 2020 @ 2359hrs.

In each of the following cases:

- You must handle spaces and other special characters. A string ends when an endline (‘\n’) is given as input.
- The length of the input string cannot be more than 500 characters.
- You should NOT print anything apart from the expected output, since grading might be done using a script. All prints used for debugging should be removed.
- Please use the same filenames as given in the repository, and use the makefile given.
- You may use ONLY C standard or UNIX libraries. Use of any other libraries is not allowed.

Question 1

Pipes: Write a C program which forks a child process. The parent process is responsible for accepting an input, and sending it to the child process through a pipe. The child process converts it into upper case, and sends the modified character back to the parent process. The parent process then prints the character received to a console.

Question 2

Message Queues: Write two C programs, each of which reads the files “para1.txt” and “para2.txt” respectively. Split the text into tokens based on spaces. Create a message queue and use the **same** message queue to send the individual tokens to a third program. This third program should print the received message as is on the console.

Question 3

Sockets: Repeat the same process, but this time send it using datagram sockets. You can use different sockets to send messages, but the same port number should be used to receive them.

Question 4

Shared Memory: Write a program that forks a child and then creates two distinct shared memory segments to communicate with each other. One memory segment is used to write text, while the other segment is used to write numbers. To coordinate the reading and writing, a process always locks (using semaphore or mutexes) **both** the segments. You also need to think of a mechanism by which a reader process should know that a memory location has been written to or not.

Question 5

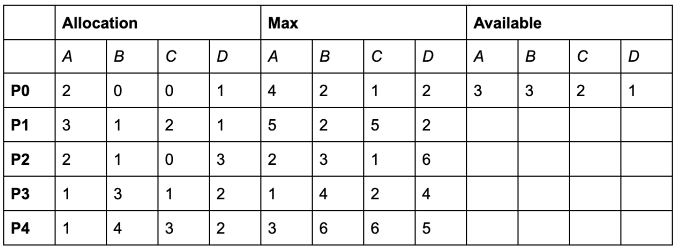
Reader-Writer Problem with Synchronization: Write a C program that involves multiple processes/threads where one of the processes/threads is a writer while there are multiple reader processes. You need to take care of problems related to race conditions using POSIX mutexes and semaphores. The processes may use a shared memory for 8 bytes (64-bits) that could be written to or read from.

Question 6

Dining philosopher problem: Write a C program to solve the dining philosopher problem where the philosophers are represented by threads. You need to ensure that your solution is free of deadlocks, i.e. it should be designed such that it avoids/prevents deadlocks.

Question 7

Deadlock avoidance: Assume that there are five processes P0, P1, P2, P3 and P4, along with four types of resources A, B, C and D. The table shows the Allocation, Max and Available matrices. Is the system in a safe state ? If yes then please present a safe order/sequence in which processes may be completed. If no, give the reason why.



	Allocation				**Max**				**Available**			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*
P0	2	0	0	1	4	2	1	2	3	3	2	1
P1	3	1	2	1	5	2	5	2				
P2	2	1	0	3	2	3	1	6				
P3	1	3	1	2	1	4	2	4				
P4	1	4	3	2	3	6	6	5				

Submission Instructions

- All your code files and the text file for Question 7 must be enclosed in a zip file whose file name should be rollNo.zip. Any other folder name will be rejected. Do not make any subfolders.

- We have provided the git repository template for your submission.You need to clone the template and use it.
  `https://github.com/arani89/quiz-2-cse231`

- A correct submission should look like
  `https://drive.google.com/file/d/1sSbruT3wh9skBVvy__0Zo84Xp27iADsU/view?usp=sharing`

Any submission which does not follow the submission rules will not be evaluated.