Operating Systems Course (CS39001)

Spring Semester 2021-2022

Take home assignment 1

Assignment given on: February 23, 2022
Assignment deadline: March 06, 2022, 11:55 PM
Total marks: 60
Weightage: 15% of theory

Submission instructions

- This is a time-bound take home assignment to test your understanding of the concepts developed in the course so far.

- YOU HAVE TO USE PEN AND PAPER TO COMPLETE ALL QUESTIONS OF THE ASSIGNMENT (including the coding questions).

- You need to submit your INDIVIDUAL solutions to CSE Moodle in the following way: Scan and put all the pages sequentially in a pdf file and upload the pdf to CSE Moodle.

- The pdf file should be named "Take_home_Assgn1_<roll no>_submission.pdf"

- Note that, if we face problems with your answer script e.g., cannot open your submitted pdf file, cannot read the text (due to bad resolution), cannot determine the page order (i.e., the pages in the pdf are jumbled up), or if we find you copying from other students, it will affect your marks.

Problems

1. State three ways in which the processor can transition from user mode to kernel mode while a user process is being executed. [3 marks]

2. We discussed in the class that a thread library can be implemented at the user level or at the kernel level. State two advantages and one disadvantage of implementing a threading library at the user level rather than the kernel level. [2 + 1 = 3 marks]

3. A program contains a single loop that executes 50 times. The loop contains a computation that lasts 50 msec followed by an I/O operation that consumes 200 msec. This program is executed in a time-sharing system with 9 other identical programs. All programs start their execution at the same time. The scheduling overhead of the OS is 3

msec. Compute the response time in the first and subsequent iterations if (a) The time slice is 50 msec, and (b) The time slice is 20 msec. [4 + 4 = 8 marks]

4. Consider the different states in the life cycle of a process. For each state transition mentioned below, mention if the transition is possible directly (i.e., not via some other intermediate state). If yes, give one example of a situation when the said transition can take place directly. If not, explain your answer: [5 x 2 = 10 marks]

- (i) Ready state to Running state
- (ii) Running state to Ready state
- (iii) Running state to Waiting state
- (iv) Ready state to Waiting state
- (v) Waiting state to Ready state.

5. Consider the following set of processes, given along with their arrival times and their CPU burst times (both in units of msec). Calculate the ***average waiting time*** of the processes, for the following scheduling algorithms: [5 + 5 = 10 marks]

- (i) preemptive Shortest Job First (SJF), which is also known as Shortest Remaining Time First (SRT) scheduling,
- (ii) round-robin scheduling with a time quantum of 3 msec.

Process	P1	P2	P3	P4	P5	P6
Arrival Time (msec)	0	2	3	5	6	8
CPU Burst (msec)	7	4	6	2	8	5

6. State the different modes of operation of an operating system. Which mode of operation does an Operating System enter for executing a system call? Why is it necessary to have the different operating modes (two reasons)? [2 + 1 + 3 = 6 marks]

7. Please answer True or False for each of the following questions with brief reasons (no marks without reason): [2.5 x 4 = 10 marks]

- During execution if a user process issues a system call the processor switches to kernel mode
- During execution of a user process the processor can switch to kernel mode even when the user process did not issue a system call.
- The kernel must maintain a separate heap for each thread in each user process.
- The kernel must maintain a separate stack for each thread in each user process.

8. How many new processes will be created by the following code segments? [6 marks]

(a)

```
for (i=0; i<6; i++)
  fork();
```

(b)

```
pid = fork();
if (pid == 0) { fork(); fork(); }
else  fork();
```

9. Write a code segment in C/C++ to do the following: [4 marks]

- *Create a shared memory segment of size 100 bytes, and attach it to the current process.*