CS302 Operating System Midterm Exam

April 10, 2021

CS302 Operating Systems Dr. Bo Tang & Dr. Yinqian Zhang

Your Name:		**Student ID:**	

General Information:

This is a **closed book and one 2-sided handwritten note (A4-size)** examination. You have 120 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points for that question. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. ***Make your answers as concise as possible.*** If there is something in a question that you believe is open to interpretation, then please ask us about it! Good Luck!

QUESTION	POINTS ASSIGNED	POINTS OBTAINED
P1: CPU scheduling	**12**	
P2: Deadlock	**12**	
P3: Synchronization	**14**	
P4: True/False and why	**22**	
P5: Short answer	**30**	
P6: Multiple choices	**10**	
TOTAL	**100**	

P1 (12 points) CPU Scheduling

Analyze scheduling algorithms for the following 4 processes given the CPU burst (time requirement), priority (lower number means higher priority) and arrival time.

Processes	Time Requirement	Priority	Arrival Time
P1	4	3	0
P2	2	4	1
P3	5	1	5
P4	4	2	3

For each of the following scheduling algorithms, calculate the waiting time for each process and calculate the average waiting time.

(1) Non-preemptive Shortest Job First (4 points)

(2) Non-preemptive Priority Scheduling (4 points)

(3) Preemptive Priority Scheduling (4 points)

*Note: Arrival Times should be considered for each of the scheduling algorithms. It's possible that the algorithms will not need all the information given in the table.

Correct result: lower number means higher priority

a) p1: 0 p2: 3 p3: 5 p4: 3 aver = (0+3+5+3)/4 = 11/4 = 2.75

b) p1: 0, p2: 12, p3: 3, p4: 1 aver = (0+12+3+1)/4 = 4

c) p1: 9, p2 12, p3, 0, p4:5 aver = (9+12+0+5)/4 = 6.5

Incorrect but OK (take 1 point off) if consider lower number means lower priority

a) p1: 0 p2: 3 p3: 5 p4: 3 aver = (0+3+5+3)/4 = 11/4 = 2.75

b) p1: 0 p2: 3. p3: 5 p4: 3 aver = (0+3+5+3)/4= 2.75

c) p1: 2 p2: 0. p3: 5 p4: 3 aver = (2+0+5+3) /4 = 2.5

P2 (12 points) Deadlock

Given four processes P1 through P4 and four resource types in the operating systems: A, B, C, D. The total number of instances of these resource types are (2, 4, 3, 3). The following matrices show a snapshot of the resource allocation at time T0.

	Allocation	Max
	A B C D	A B C D
P1	0 2 1 0	2 3 1 0
P2	0 1 0 1	0 1 2 2
P3	0 0 1 0	1 0 1 1
P4	1 1 0 0	1 2 1 1

(1) Is the operating system in a safe state? (4pts)

(2) If P4 requests (0,0,1,1), please run the Banker's algorithm to determine if the request should be granted. (4pts)

(3) Let's assume P4's request was granted anyway (regardless of the answer to question 2). If then the processes request additional resources as follows, is the system in a deadlock state? (4pts)

	Request A B C D
P1	2 1 0 0
P2	0 0 1 0
P3	1 0 0 0
P4	0 1 0 0

(1) Yes, the system is in a safe state. There is a sequence of process: p3->p2->p4->p1
(2) No, the request should not be granted, because if so, the system goes into unsafe state.
(3) No, the system is not in deadlock state, because there is a sequence of processes (per deadlock detection algorithm) P3->P2->P4->P1.

Note: The details need to be written. question 3 does not use MAX and NEED matrix.

P3 (14 points) Synchronization

The sleeping teaching assistant at CS302 (answer it with pseudo-code)

CS302 has a teaching assistant (TA) who helps CS302 students with their project assignment during regular office hours. The TA's office is rather small and has room for only one desk with a chair and computer. There are n chairs in the hallway outside the office where students can sit and wait if the TA is currently helping another student. When there are no students who need help during office hours the TA sits at the desk and take a nap. If a student arrives during office hours and finds TA sleeping, the student must awaken the TA to ask for help. If a student arrives and finds TA currently helping another student, the student sits on one of the chairs in the hallway and waits. If no chairs are available, the student will come back later.

Using mutex locks and semaphores implement a solution that coordinates the activities of the TA and the students. (Full points will only give the optimal solution)

Shared object: 3 points

```
int waiting =0;
semaphore mutex = 1;
semaphore tachair = 1;
semaphore wchair = n;
semaphore ready = finish = 0;
```

TA function: 3 points

```
TA()
{
        while(true)
        {
                sem_wait(ready);
```

```
                    // helping student
                    sem_post(finish);
            }
    }

    Student function 6 points:
    Student()
    {
            sem_wait(mutex);
            if(waiting <=n);
            {
                    waiting ++;
                    sem_post(mutex);
            }
            else
            {
                    sem_post(mutex);
                    // leave and come later
            }
    sem_wait(wchair);
    sem_wait(tachair);
    sem_post(wchair);
    sem_post(ready);
    sem_wait(finish);
    sem_post(tachair);
    sem_wait(mutex);
    waiting--;
    sem_post(mutex);
}
```

P4 (22 points,) True / False and why (2 pts for each problem, 1 pt for T/F and 1 pt for why)?

TFTFF TTTTF F

1. The priority-based CPU scheduling algorithm may cause issues such as starvation.
2. According to the process state diagram, a process may move from ready state to waiting state.
3. Process control blocks contain information like program states, program counters, CPU registers, etc.
4. Multiple threads of the same process share the same stack.
5. When semaphore is used to protect a critical section, it must be initialized as 0.

6. Applications usually call standard APIs instead of system calls, but they can also use system calls directly.
7. In a resource allocation graph with single instances of resources, a cycle is sufficient to conclude that there is deadlock.
8. CPU-bound process spends more time doing computations with a few very long CPU bursts.
9. In multi-threaded programs, it is possible to deliver the signal to every thread in the process or to certain threads of the process.
10. Suppose a local variable ***num*** is initialized to 5 at the beginning of a process. Further along in the code, after the child process is created, the child process added 6 to ***num***. If the parent process then prints out the value of ***num***, the printed result will be 11.
11. The 4 necessary conditions of a deadlock are mutual exclusion, hold and wait, progress, circular wait.

P5 (30 points) Short answers, please answer the question within 6 sentences or less (6 pts for each problem)

(a) What is the difference between "system call" and "function call"?
The main difference between system call and function call is that a system call is a request for the kernel to access a resource while a function call is a request made by a program to perform a specific task.
(b) Why CPU provides dual mode operation? Please list three methods to change user mode to kernel mode.
Dual-mode operation forms the basis for I/O protection, memory protection and CPU protection. In dual-mode operation, there are two separate modes: kernel mode and user mode. In kernel mode, the CPU can use all instructions and access all areas of memory. In user mode, the CPU is restricted to unprivileged instructions and a specified area of memory. When responding to system calls, other traps/exceptions, and interrupts, OS code is run, the CPU automatically switches to kernel mode whenever an interrupt or trap occurs.

(1)System call, (2) interrupt, (3) exception

(c) Please describe the actions in operating system when process A executes "exit()" function (suppose A's parent "wait()" is called).

(1) The kernel frees all the allocated memory, the list of opened files are all closed. (so it's okay to skip fclose(); though not recommended)
(2) The kernel frees everything on the user-space memory about the concerned process, including program code and allocated memory.

(3) The kernel notifies the parent of the child process about the termination of its child. The kernel sends a SIGCHLD signal to the parent
(4) When SIGCHLD comes, the corresponding signal handling routine is invoked!
(5) Last, default handler of SIGCHLD accept and remove the SIGCHLD signal; destroy the child process in the kernel-space (remove it from process table, task-list, etc.)

(d) What is a race condition? Give an example.

A race condition occurs when there is an uncoordinated concurrent access to shared resources (e.g. memory or device registers), which leads to incorrect behavior of the program, deadlocks or lost work.

An example would be two process updating a counter simultaneously. Say the counter has a value of 1. Process A reads the variable but before updating Process B reads and updates the counter to 2. Process A, having no knowledge of what Process B does, updates the counter to 2. The correct behaviour of the program would have the counter being incremented to 3, not 2.

(e) Describe four general strategies for dealing with deadlocks.

1) ignore it

2) detect and recover

3) deadlock avoidance

4) deadlock prevention

P6 (10 points) Multiple Choice (2 pts for each question)

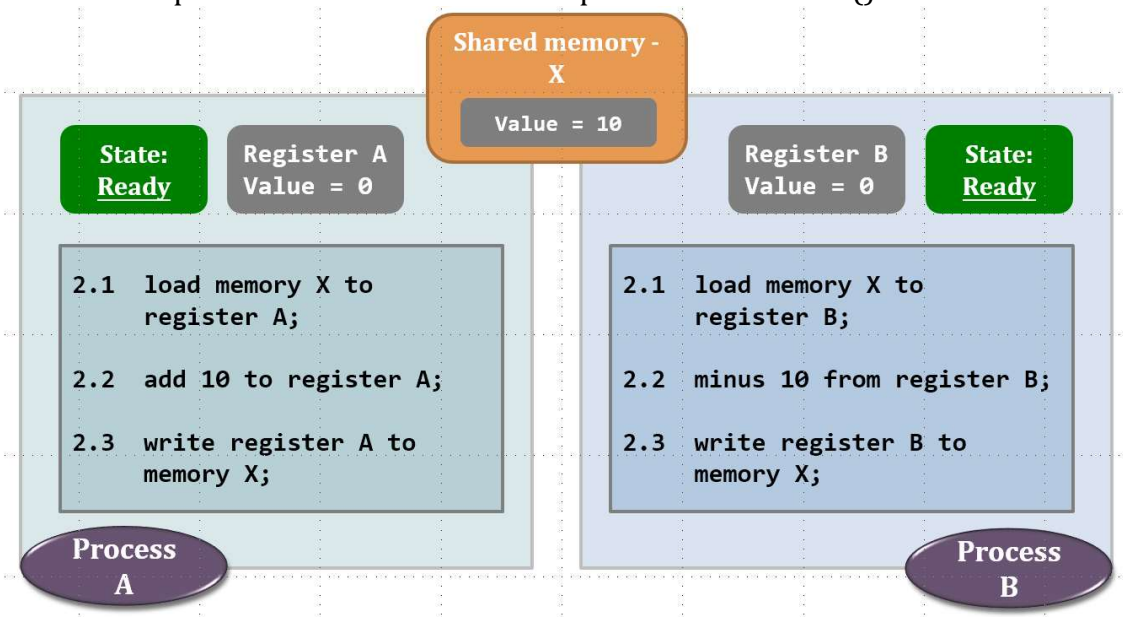
1. What is(are) not shared among different threads in a process?
   A. PID B. File descriptor
   C. Execution Stack D. Program Counter

2. Which system calls are employed to implement a C library call "system()"?
   A. fork() B. exec*()
   C. wait() D. pipe()

3. Consider the lifecycle of a process, which status transferring is(are) impossible?
   A. Ready → Running B. Running → Ready
   C. Running → Waiting D. Ready → Waiting

4. Suppose there are three jobs J1, J2, J3 arrive at time step 0. The execution cost are T1, T2, and T3 (suppose T1<T2<T3). The scheduling algorithm is shortest job first, what is the average turnaround time ()
   A. T1+T2+T3 B. (T1+T2+T3)/3 C. (3T1+2T2+T3)/3 D. (T1+2T2+3T3)/3

5. What are the possible x value after the two processes executed () ?



A. -10 and 30, B. 10 C. 0, 10, 20, D. -10, 0, 10, 20