CPE 460: OPERATING SYSTEMS DESIGN

Second Exam, May 8, 2017

- This is a **75-minute** OPEN BOOK exam, with a total of **30 marks**. There are **30 questions**, and **9 pages** (including this cover page).
- All your answers to multiple choice questions must be marked on this answer sheet. We will **not** take into consideration anything written on the question booklet or if multiple markings are made on the answer sheet. Make sure to mark only one answer.

GOOD LUCK

Total Grade
/30

Serial No

Name

1 (A) (B) (C) (D) 18 (A) (B) (C) (D) 27 (A) (B) (C) (D)
2 (A) (B) (C) (D) 19 (A) (B) (C) (D) 28 (A) (B) (C) (D)
3 (A) (B) (C) (D) 20 (A) (B) (C) (D) 29 (A) (B) (C) (D)
4 (A) (B) (C) (D) 21 (A) (B) (C) (D) 30 (A) (B) (C) (D)
5 (A) (B) (C) (D) 22 (A) (B) (C) (D)
6 (A) (B) (C) (D) 23 (A) (B) (C) (D)
7 (A) (B) (C) (D) 24 (A) (B) (C) (D)
8 (A) (B) (C) (D) 25 (A) (B) (C) (D)
9 (A) (B) (C) (D) 26 (A) (B) (C) (D)

cpe460-second-exam-spring2017 (0594)

10 (A) (B) (C) (D)
11 (A) (B) (C) (D)
12 (A) (B) (C) (D)
13 (A) (B) (C) (D)
14 (A) (B) (C) (D)
15 (A) (B) (C) (D)
16 (A) (B) (C) (D)
17 (A) (B) (C) (D)

Student ID

0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9

(A) (B) (C) (D) (E)

1. Consider the methods used by processes `P1` and `P2` for accessing their critical sections whenever needed, as given below. The initial values of shared boolean variables `S1` and `S2` are randomly assigned.

Process `P1`	**Process** `P2`
`while(S1==S2);`	`while(S1!=S2);`
`Critical section`	`Critical section`
`S1 = S2;`	`S2 = not(S1);`

Which of the following statements describes properties achieved ?

(A) Mutual exclusion but not progress

(B) Progress but not mutual exclusion

(C) Neither mutual exclusion nor progress

(D) Both mutual exclusion and progress

2. Each process $P_i \quad \forall i = 0, 1, 2, 3, \cdots, 9$ is coded as follows:

```
while(true){
    lock(&mutex);
    {
        //Critical Section
    }
    unlock(&mutex);
}
```

The code for P_{10} is identical except that it uses `unlock(&mutex)` instead of `lock(&mutex)`. What is the largest number of processes that can be inside the critical section at any moment (the `mutex` being initialized to 1)?

(A) 1

(B) 2

(C) 3

(D) 4

3. The following pair of processes share a common variable `X` as well as the binary semaphore `T`:

Process `A`	**Process** `B`
`int Y;`	`int Z;`
`A1: Y = X*2;`	`B1: wait(T);`
`A2: X = Y;`	`B2: Z = X+1;`
`signal(T);`	`X = Z;`

`T` is set to 0 and `X` is set to 5 before either process begins execution. Now, how many different values of `X` are possible after both processes finish executing?

(A) one

(B) two

(C) three

(D) four

For questions 4–5, consider the below code for `Process 0` and `Process 1` and the `man` page for `msgrcv`:

```
struct msgbuf {
       long mtype;
       char mtext[200];
};

// Process 0
void main(){
    int queue_id;
    struct msgbuf msg;
    int i;
    queue_id = msgget(1234, IPC_CREAT | IPC_EXCL | 0600);

    msg.mtype = 1;
    strcpy(msg.mtext, "I love you Mom once!\n");
    msgsnd(queue_id, &msg, sizeof(msg.mtext), 0);

    msg.mtype = 2;
    strcpy(msg.mtext, "I love you Mom twice!\n");
    msgsnd(queue_id, &msg, sizeof(msg.mtext), 0);

    msg.mtype = 3;
    strcpy(msg.mtext, "I love you Mom thrice!\n");
    msgsnd(queue_id, &msg, sizeof(msg.mtext), 0);
}

// Process 1
void main(){
    int queue_id;
    struct msgbuf msg;
    int msg_type;
    int i;
    queue_id = msgget(1234, 0);
    for (i = 1; i <= 3; i++) {
        msg_type = (i % 3);
        msgrcv(queue_id, &msg, sizeof(msg.mtext), msg_type, 0);
        printf("%s\n", msg.mtext);
    }
}

$ man msgrcv
ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);

The msgrcv() system call removes a message from the queue specified by msqid
and places it in the buffer pointed to by msgp. The argument msgsz specifies
the maximum size in bytes for the member mtext of the structure pointed to by
the msgp argument.
```

```
The argument msgtyp specifies the type of message requested as follows:
- If msgtyp is 0, then the first message in the queue is read.
- If msgtyp is greater than 0, then the first message in the queue of type
  msgtyp is read,  unless MSG_EXCEPT was specified in msgflg, in which case
  the first message in the queue of type not equal to msgtyp will be read.
- If msgtyp is less than 0, then the first message in the queue with
  the lowest type less than or equal to the absolute value of msgtyp will be read.

The msgflg argument is a bit mask constructed by ORing together zero or more of
the following flags:
- IPC_NOWAIT: Return immediately if no message of the requested type is in the queue.
- MSG_EXCEPT: Used with msgtyp greater than 0 to read the first message in the queue
  with message type that differs from msgtyp.

If no message of the requested type is available and IPC_NOWAIT isn't specified
in msgflg, the calling process is blocked until one of the following conditions occurs:
- A message of the desired type is placed in the queue.
- The message queue is removed from the system.
```

4. If we executed the code for `Process 0` followed by the code for `Process 1`, what will be output printed on the screen from `Process 1`:

   (A)
   ```
   I love you Mom once!
   I love you Mom twice!
   I love you Mom thrice!
   ```

   (B)
   ```
   I love you Mom once!
   I love you Mom twice!
   ```

   (C)
   ```
   I love you Mom once!
   ```

   (D)
   ```
   I love you Mom twice!
   I love you Mom thrice!
   ```

5. According to the `man msgrcv` page, which of the following values for `msg_type` makes the recieving of messages occurs on FIFO (first-in first-out) basis:

   (A) 0

   (B) > 0

   (C) < 0

   (D) Do not choose this answer

6. The IPC mechanism that does **not** suffer from blocking problems is _____

   (A) ordinary pipes

   (B) named pipes

   (C) message queues

   (D) shared memory

7. Consider the following code to answer the question below:

Process `P0`**:**

```
void main(){
    int i = 0;
    int *values = &i;
    int shmid = shmget(3456, 10 * sizeof(int), IPC_CREAT|0777);
    values = (int *) shmat(shmid, 0, 0);
    for(i = 5; i < 15; i++){
         *values = i * i;
         values++;
    }
}
```

Process `P1`**:**

```
void main(){
    int i = 0;
    int *values = &i;
    int shmid = shmget(3456, 10 * sizeof(int), IPC_CREAT|0777);
    printf("The first value in shared memory is: %d\n", *values);
}
```

If we ran the code for `P0` followed by the code for `P1`, the output of process `P1` is:

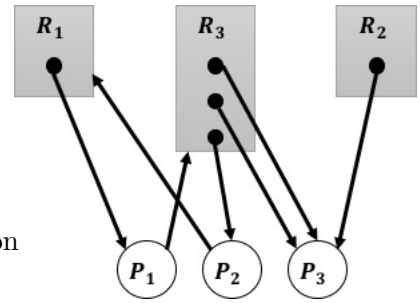
(A) `The first value in shared memory is: 0`

(B) `The first value in shared memory is: 25`

(C) `The first value in shared memory is: 36`

(D) `The first value in shared memory is: 1`

8. Given the resource allocation graph depicted below, does a deadlock exist?



(A) yes, deadlock exists

(B) no, the system is deadlock free

(C) it depends on the order of execution

(D) do not choose this answer

9. Which process can be affected by other processes executing in the system?

(A) cooperating process

(B) child process

(C) parent process

(D) init process

10. The `signal` operation of the semaphore basically works on the basic _____ system call.
    (A) `continue()`
    (B) `wakeup()`
    (C) `getup()`
    (D) `start()`
11. If the semaphore value is negative :
    (A) its magnitude is the number of processes waiting on that semaphore
    (B) it is invalid
    (C) no operation can be further performed on it until the signal operation is performed on it
    (D) none of these
12. _____ is an IPC mechanism that **cannot** be used for communication between two **unrelated** processes.
    (A) Ordinary pipes
    (B) Named pipes
    (C) Message queues
    (D) Shared memory
13. When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called _____
    (A) dynamic condition
    (B) race condition
    (C) essential condition
    (D) critical section
14. For two processes accessing a shared variable, Peterson's algorithm provides:
    (A) mutual exclusion
    (B) progress
    (C) bounded waiting
    (D) all of the above
15. Which of the following is **correct** when using zero capacity queue in message passing IPC:
    (A) the queue has non-zero capacity
    (B) the sender blocks until the receiver receives the message
    (C) the sender keeps sending and the messages do not wait in the queue
    (D) the queue can store at least one message

For questions 16–18, consider the code below:

```
1:void main(){
2:   int p[2];
3:   char ch = 'a';
4:   pipe(p);
5:   if(fork() == 0){
```

```
 6:      write(p[1],"c",1);
 7:      write(p[1],"d",1);
 8:   }else{
 9:      wait(NULL);
10:      read(p[0], &ch, 1);
11:      printf("I read %c\n", ch);
12:   }
13:}
```

16. The output of the parent process is _____

    (A) `I read a`

    (B) `I read c`

    (C) `I read d`

    (D) none of the above

17. The pipe created in this code is _____

    (A) ordinary pipe

    (B) named pipe

    (C) full duplex pipe

    (D) no pipe is created

18. If the statement `printf("I read %c\n", ch);` has been added in between lines 7 and 8, the output of the child process will be _____

    (A) `I read a`

    (B) `I read c`

    (C) `I read d`

    (D) `I read c` OR `I read d`

19. A minimum of _____ variable(s) is/are required to be shared between processes to solve the critical section problem.

    (A) one

    (B) two

    (C) three

    (D) four

20. A mutex :

    (A) is a binary semaphore

    (B) must be accessed from only one process

    (C) can be accessed from multiple processes

    (D) none of these

21. The following program consists of 3 concurrent processes and 3 **binary semaphores**. The semaphores are initialized as `S0 = 1`, `S1 = 0`, and `S2 = 0`.

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

`while(true){`
`  wait(S0);`
`  printf("0\n");`
`  signal(S1);`
`  signal(S2);`
`}`
	`wait(S1);`
	`signal(S0);`
		`wait(S2);`
		`signal(S0);`

How many times will process `P0` print `0` ?

(A) At least twice

(B) Exactly twice

(C) Exactly thrice

(D) Exactly once

22. The segment of code in which the process may change common variables, update tables, write into files is known as _____

(A) program

(B) critical section

(C) noncritical section

(D) synchronizing

23. How many seasons was "Bab Al-Hara" on the air?

(A) 6

(B) 7

(C) 8

(D) 9

24. What was the name of the school's principal where Sally spent her childhood?

(A) Ms. Emilia

(B) Ms. Mention

(C) Ms. Armingard

(D) Ms. Viki

25. A deadlock free solution to the dining philosophers problem :

(A) necessarily eliminates the possibility of starvation

(B) does not necessarily eliminate the possibility of starvation

(C) eliminates any possibility of any kind of problem further

(D) I do not care about the philosophers

26. In the bounded buffer problem, there are the `empty` and `full` semaphores that:

    (A) count the number of empty and full buffers

    (B) count the number of empty and full memory spaces

    (C) count the number of empty and full queues

    (D) None of these

27. At a particular time of computation, the value of a **counting semaphore** is 7. Then 20 `wait` operations and 15 `signal` operations were completed on this semaphore.The resulting value of the semaphore is:

    (A) 42

    (B) 2

    (C) 7

    (D) 12

28. Which TV series that I mentioned in the slides on last lecture?

    (A) Bab Al-Hara

    (B) Breaking Bad

    (C) House of Cards

    (D) Arabs Got Talent

29. The `test_and_set` and `compare_and_swap` instructions are executed:

    (A) after a particular process

    (B) periodically

    (C) atomically

    (D) staticly

30. A computer system has 6 tape drives, with n processes competing for them. Each process may need 3 tape drives. The maximum value of n for which the system is guaranteed to be deadlock free is:

    (A) 1

    (B) 2

    (C) 3

    (D) 4