

1. Which of the following regarding an operating system ( OS ) is not true?

- A. It allocates resources.
- B. It provides the runtime environment.
- C. Its primary goal is the efficiency of operation and secondary goal is the convenience of operation.**
- D. It hides heterogeneous hardware from users.
- E. An OS may be regarded as a virtual machine providing higher level abstraction to application programs.

2. Which of the following is the not true about the OS Multics?

- A. It is a joint project started by IBM.**
- B. It is a joint project started by Massachusetts Institute of Technology (MIT).
- C. It is short for multiplexed information and computing services.
- D. It is a time sharing system.
- E. General Electric (GE) was one of the starting partners of the project.

3. Which of the following is the not true about distributed systems?

- A. They consist of machines connected by network.
- B. They are loosely coupled systems.
- C. They share resources.
- D. They often use clientserver model.
- E. They are tightly coupled systems.**

4. Which of the following about OS is not true?

- A. Most of the top 500 supercomputers are built with Linux.
- B. More than 5% of the top 500 supercomputers are built with MSWindows.**
- C. Less than 3% of the top 500 supercomputers are built with Mac OS/X.
- D. All realtime systems have very fast response.
- E. Embedded systems are used in mobile phones.

5. Consider the following program:

```
int main()
{
    int i;
    for ( i = 0; i < 3; ++i ) {
        fork();
        cout << i << endl;
    }
    return 0;
}
```

How many 2's will be printed by the program? Enter your answer, an integer, here (do not put in any punctuation mark): **8**

6. What is the fivecharacters Unix command that creates a directory? Enter your answer here (do not put in any punctuation mark; note also that Unix commands are case sensitive):

**mkdir**

7. Which of the following Unix command will make the file test1 executable? (Choose the best answer.)

- A. chmod +x test1**
- B. chmod +w test1
- C. g++ test1
- D. cp test1 test1.exe
- E. mv test1 test1.exe

8. The following is the source code of progA:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "A "
    return 0;
}
```

and the following is the source code of progB:

```
#include <unistd.h>
#include <sys/wait.h>
#include <stdio.h>
using namespace std;
int main()
{
    int pid = fork();
    if ( pid == 0 ) {
        cout << "C ";
        fflush( stdout ); //flush standard output
        execlp ( "./progA", "", NULL );
        cout << "D ";
        fflush( stdout );
    } else {
        wait ( NULL );
        cout << "B ";
    }
    cout << endl;
    return 0;
}
```

What is the output when you execute ./progB?

- A. A B C D
- B. C A D B
- C. B C A D
- D. C A B**
- E. C D B

9. Suppose the executable shell scripts, sh1, sh2, and sh3 contain the statements

x=1 #sh1 ,

x=2 #sh2 , and

x=`expr + 3` #sh3

respectively.

The following sequence of commands are executed:

\$ ./sh1

\$ ./sh2

\$ ./sh3

\$ ./sh1

\$ ./sh3

\$ echo \$x

What is the output?

A. 3

B. 4

**C. 5**

D. 7

E. 8

1. Signals may be initiated by which of the following? Choose all the correct answers.

**A. hardware**

**B. kernel**

**C. process**

**D. users**

2. This question relates to the labs you have done. Which of the following is not a signal function (a function that is used by a process to send a signal to another process or to catch a signal)?

A. kill

B. raise

C. alarm

**D. ding**

E. signal

3. How many processes does the following piece of code create? (Note: without any fork(), the code creates one process.)

```
int main()
```

```
{
```

```
fork();
```

```
fork();
```

```
fork();
```

```
return 0;
```

```
}
```

Enter your answer, an integer, here (do not put in any punctuation mark): **8**

4. When a process is blocked,
- A. it is waiting in the ready queue.
  - B. it is finishing execution.
  - C. it is newly created.
  - D. it has used up its current quantum (timeslice).
  - E. it is waiting for some events such as I/O processing.**

5. Suppose headers have been correctly included in the following program:

```
int main()
{
    int fd[2];
    char buffer[100];
    pid_t pid;
    memset(buffer, 0, sizeof(buffer)); //fill whole buffer with 0
    strcpy ( buffer, "xyz" );
    pipe(fd); //creates pipe
    pid = fork();
    if (pid == 0) { //child
        char buf[100];
        int n = read ( fd[0], buf, 100 );
        buf[n] = 0;
        cout << buf;
        n = write(fd[1], buffer, strlen(buffer));
    } else { //parent
        strcpy ( buffer, "abc" );
        int n = write(fd[1], buffer, strlen(buffer));
        strcpy ( buffer, "123" );
        wait ( NULL );
        n = read ( fd[0], buffer, 100 );
        cout << buffer << endl;
    }
    exit(EXIT_SUCCESS);
}
```

What is the output of the program when executed?

- A. abc123
- B. xyz123
- C. abcxyz**
- D. 123abc
- E. abcab

6. Which of the following systems often uses a microkernel?

- A. clientserver system**
- B. virtual machine
- C. layered system
- D. monolithic system
- E. mainframe computer

7. Processes in a system arrive in the order P1,P2,P3,P4,P5 with burst time and arrival time

shown below:

Process	Arrival Time	Burst Time
P1	0	5
P2	0	3
P3	1	1
P4	1	3
P5	2	2

If firstcomefristserved FCFS scheduling algorithm is used, what is the average waiting time of the processes?

- A. 6.2
- B. 6**
- C. 7
- D. 8
- E. 7.4
- F. 4.8

8. In the above question, if nonpreemptive shortestjobfirst (SJF) scheduling algorithm is used, what is the average turn around time of the processes?

- A. 6.2
- B. 6
- C. 7
- D. 8**
- E. 7.4
- F. 4.8

1. The scheduler of a simple UNIX OS prints out the name of the currently running process and its process id (pid). The following messages have been printed with line numbers labeled at the left:

- 1. Process initcode with pid 1 running
- 2. Process initcode with pid 1 running
- 3. Process initcode with pid 1 running
- 4. Process initcode with pid 1 running
- 5. Process initcode with pid 1 running
- 6. Process initcode with pid 1 running
- 7. Process init with pid 1 running
- 8. Process init with pid 1 running
- 9. Process init with pid 1 running
- 10. Process init with pid 2 running
- 11. Process init with pid 2 running
- 12. Process init with pid 2 running

13. Process init with pid 2 running  
14. Process init with pid 2 running

From line 6 to line 7, the name of the process with pid 1 has been changed from initcode to init. What has happened?

- A. The quantum of the process initcode has expired and the scheduler switches to run the new process init.  
B. The process initcode has executed the system call `fork()` to load the new process init.  
**C. The process initcode has executed the system call `exec()` to load init to replace itself.**  
D. The process initcode has been blocked.  
E. The process initcode has renamed itself.

2. In the above question, from line 9 to line 10, the process id (pid) of init has been changed from 1 to 2. What has happened? (Choose the best answer.)

- A. The process init has been blocked.  
**B. The process init has executed the system call `fork()` to create a child process.**  
C. The process init has executed the system call `exec()` to load another init to replace itself.  
D. The original process init has died.  
E. The process init changes its id.

3. Processes in a system arrive in the order P1,P2,P3,P4,P5 with burst time and arrival time shown below:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	3	1

If the firstcomefristserved FCFS scheduling algorithm is used, what is the average waiting time of the processes?

- A. 4.33  
B. 4  
C. 5.33  
D. 5  
**E. 3**  
F. 3.33

4. In the above question, if the nonpreemptive shortestjobfirst (SJF) scheduling algorithm is used, what is the average waiting time of the processes?

- A. 2.33**  
B. 2  
C. 3.33

- D. 3
- E. 4.33
- F. 4

5. In the above question, if a roundrobin (RR) scheduling algorithm with quantum=2 is used, what is the average waiting time of the processes? (Ignore the switching time.)

- A. 2.33
- B. 2
- C. 3.33
- D. 3**
- E. 4.33
- F. 4

6. Suppose H, M, and L are three tasks running in a realtime system with high, medium and low priorities respectively, and priority inversion has occurred. The issue is often resolved by the priorityinheritance protocol. Which of the following best describes this protocol?

- A. Change the priority of L to that of H while L is executing the critical section.**
- B. Change the priority of H to that of L while L is executing the critical section.
- C. Change the priority of M to that of H while L is executing the critical section. adaptor
- D. Change the priority of L to that of M while L is executing the critical section. iterator
- E. Change the priority of M to that of L while L is executing the critical section.

7. Which of the following about multiprocessor scheduling is not true?

- A. Modern desktop OS such as Linux and MS Windows often use symmetric multiprocessing.
- B. Microkernel systems often use asymmetric multiprocessing.**
- C. Symmetric multiprocessing may load the OS multiple times, one for each CPU.
- D. Each processor in symmetric multiprocessing scheduling does its own schedule.
- E. The CPUs of asymmetric multiprocessing have a masterslave relation with the master server doing the scheduling.

8. Suppose headers have been correctly included in the following program:

```
static int counter = 0;
SDL_mutex *counter_mutex;
/* The threads will run until this flag is set. */
static int exit_flag = 0;
/* This function is a thread entry point. */
int ThreadEntryPoint(void *data)
{
    char *threadname;
    /* Anything can be passed as thread data.
    We will use it as a thread name. */
    threadname = (char *) data;
    /* Loop until main() sets the exit flag. */
    while (exit_flag == 0) {
        // Lock so that no other thread can access subsequent code until unlock
        SDL_mutexP(counter_mutex);
        printf("%s:%d", threadname, counter);
```

```

counter++;
SDL_mutexV(counter_mutex); //unlock
thread 2
} r
return 0;
} i
int main()
{
    SDL_Thread *thread1, *thread2;
    /* Create a mutex to protect the counter. */
    counter_mutex = SDL_CreateMutex();
    /* Create two threads. The thread names are '1' and '2'*/
    thread1 = SDL_CreateThread( ThreadEntryPoint, ( void *) "1");
    thread2 = SDL_CreateThread(ThreadEntryPoint, ( void *) "2");
    /* Let the threads run until the counter reaches 20. */
    while (counter < 4)
    ;
    /* Signal the threads to exit. */
    exit_flag = 1;
    printf("
");
    /* Give threads time to notice the flag and exit. */
    SDL_Delay(1000);
    /* Destroy the counter mutex. */
    SDL_DestroyMutex(counter_mutex);
    return 0;
}

```

What is the most likely output of the program?

- A. The program will hang in an infinite while loop.
- B. The output is: (1:0)(2:1)(1:2)(1:3)
- C. The output is: (1:0)(2:1)(1:2)(2:3)**
- D. The output is: (1:0)(1:1)(2:2)(1:3)
- E. The output is: (1:0)(2:1)(2:2)(1:3)

1. Which of the following is true?

- A. Operating systems manage only hardware.
- B. It is possible that at any given time, more than one process are executing instructions on a computer.**
- C. A process must have a parent process.
- D. Distributed systems rely on signals instead of message passing.
- E. Both asleep and awake processes compete for CPU time.

2. Which of the following is not a goal of CPU scheduling?

- A. minimize turnaround time
- B. minimize response time
- C. minimize throughput**
- D. maximize CPU utilization (i.e. keep CPU as busy as possible )



E. maintain fairness

3. Which of the following scheduling algorithms minimizes the average response time.

A. Firstcome Firstserved

B. RoundRobin

C. Multilevel Queue

**D. Shortest Job First**

E. Twolevel Scheduling

4. Which of the following scheduling algorithms may cause priority inversion?

**A. Real Time Scheduling**

B. Multilevel Queue Scheduling

C. Firstcome Firstserved

D. RoundRobin

E. Multiprocessor scheduling

5. Fill in the blank. (Your answer must be all lowercase letters with correct spelling.)

There are two parts of CPU scheduling. The **scheduler** is a piece of OS code that decides the priorities of processes and how long each will run. The dispatcher provides the basic mechanism for running processes.

6. Processes in a system arrive in the order P1,P2,P3 with burst time and arrival time shown below:

Process Arrival Time Burst Time

P

1 0 5

P

2 1 3

P

3 3 1

If a roundrobin (RR) scheduling algorithm with quantum=2 is used, what is the average waiting time of the processes? (Ignore the switching time.)

A. 2.33

B. 2

C. 3.33

**D. 3**

E. 4.33

F. 4

7. Which of the following is not true about semaphores? (Choose the best answer.)

A. Semaphores are machine independent.

B. Semaphores can work with many processes.

C. Semaphores permit multiple processes into the critical section at once, if that is desirable.

D. Semaphores allow a process to acquire many shared resources simultaneously.

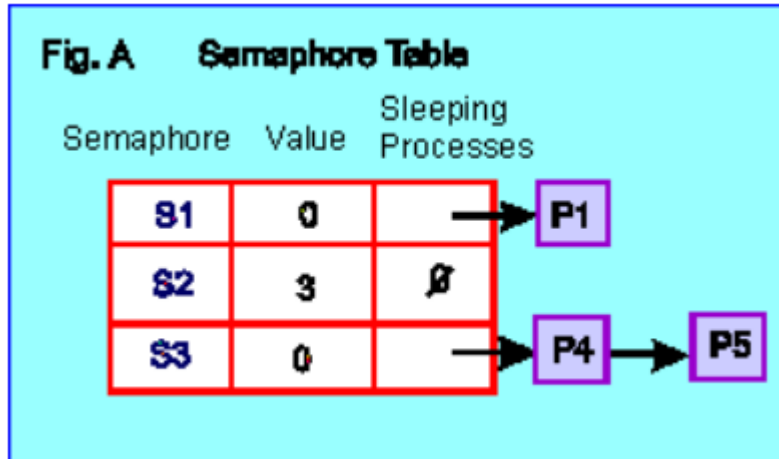
**E. Semaphores are provided by hardware.**

8. Processes in a system share three common semaphores, namely, S1, S2, and S3. At a certain

instance, the values of the semaphores are:

$S1 = 0$ ,  $S2 = 3$ ,  $S3 = 0$

Process P1 is waiting on S1, no process is waiting on S2, and processes P4 and P5 are waiting on S3 as shown below:



Now a process leaves the critical section and executes an UP on S1. Process P1 then enters the critical section. What will be the values of S1, S2 and S3 when P1 is executing the critical section?

- A.  $S1 = 1$ ,  $S2 = 3$ ,  $S3 = 0$
- B.  $S1 = 0$ ,  $S2 = 3$ ,  $S3 = 0$**
- C.  $S1 = 0$ ,  $S2 = 0$ ,  $S3 = 0$
- D.  $S1 = 0$ ,  $S2 = 2$ ,  $S3 = 0$
- E.  $S1 = 1$ ,  $S2 = 3$ ,  $S3 = 0$

1. Consider the following solution, which is presented in guarded commands, to the readerswriters problem:

```
void reader()
```

```
{
```

```
  when ( writers == 0 ) [
```

```
    readers++;
```

```
  ] /
```

```
  /read
```

```
  [readers--;]
```

```
}
```

```
void writer()
```

```
{
```

```
  when ( (readers == 0) && (writers == 0) ) [
```

```
    writers++;
```

```
  ] /
```

```
  /write
```

```
  [writers--;]
```

```
}
```

What is the shortcoming of this solution?

- A. It might not synchronize readers and writers threads.
- B. It might cause deadlock.
- C. It might cause starvation of the writers.**
- D. It might allow more than one writer to write at the same time.
- E. It might not allow more than two readers to read at the same time.

2. In the above question, how could one correct the defect of the solution?

- A. assigning higher priorities to the readers
- B. assigning higher priorities to the writers**
- C. using a shortestjobfirst policy
- D. using a firstcomefirstserved policy
- E. allowing more than one reader to read at the same time

3. Which of the following synchronization mechanisms has been adopted and provided by Java? ( choose the best answer )

- A. Testandsetlock
- B. Semaphore
- C. Disabling Interrupts
- D. Monitor**
- E. Lamport's Bakery Algorithm

4. Fill in the blank. (Your answer must be all lowercase letters with correct spelling. Do not enter any punctuation mark.)

A semaphore S is an integer variable; apart from initialization, it is accessed through standard **atomic** (meaning indivisible) operations.

5. Which of the following is considered as a highlevel synchronization tool? (Choose the best answer.)

- A. semaphore
- B. condition variables
- C. Lamport's Bakery Algorithm
- D. Testandsetlock
- E. monitor**

6. The following is a solution to the dininingphilospher problem, represented in the figure at the right. In the code,

mutex is a binary semaphore shared by all philosophers.

```
//philosopher i
while ( true ) {
    think();
    wait ( mutex ); //first check if someone is eating
    take_chops( i ); //take left chopstick
    take_chops((i+1)%5); //take right chopstick
    eat();
    put_chops( i ); //put down left chops
    put_chops( (i+1)5); //put down right chops
```

```
signal( mutex ); //wake up anyone who's waiting
}
```

What is the shortcoming of this solution?

- A. It allows only one philosopher to eat at a time though there are five chopsticks.**
- B. It may lead to deadlock.
- C. It may result in starvation.
- D. Some philosophers may eat too much.
- E. Two philosophers may eat at the same time.

7. Which of the following is not a common alternative name for the down() operation of a binary semaphore?

- A. p()
- B. lock()
- C. wait()
- D. decrement()**

8. Which of the following is not a correct description of the barriers problem shown in the figure below?

- A. It is a synchronization problem of N threads (or processes).
- B. Threads enter a barrier but are not allowed to exit until all threads have arrived at the barrier.
- C. After all threads have exited, the barrier resets itself.
- D. The problem can be solved using binary semaphores and condition variables.
- E. Once the threads exit the barrier, they cannot reenter again.**

1. Suppose we have four resources, A, B, C, and D, shared among 3 processes, P1, P2 and P3. The processes make the requests in the orders as shown below. Which set of the requests will guarantee that the system is deadlock free?

- A.  
P1: A, B, C, D  
P2: B, C, D  
P3: C, D, A
- B.  
P1: A, C, B, D  
P2: B, C, D  
P3: C, D
- C.  
P1: D, C, B, A  
P2: B, A  
P3: D, C**
- D.  
P1: A, C  
P2: B, C, D  
P3: C, D, A
- E. none of the above is deadlock free

2. In the above question, if the processes make the following requests, which set of requests may result a possible deadlock?

- A.**

**P1: A, B, C, D**

**P2: B, C, D**

**P3: B, C, D, A**

B.

P1: A, B, C, D

P2: B, C, D

P3: A, B, C

C.

P1: D, C, B, A

P2: D, C, B

P3: C, B, A

D.

P1: D, C, A, B

P2: A, B

P3: D, C, B

E. none of the above will result in a deadlock

3. Which of the following is not true?

A. A semaphore has an integer variable.

B. The down and up functions of a semaphore must be atomic.

C. Semaphores can avoid busy waiting if they are implemented in kernel.

D. If a semaphore allows the thread with a higher priority to proceed when UP() is called, starvation may occur.

**E. A thread can be in two semaphore's waiting queues at a time.**

4. Two threads, T1 and T2 share two semaphores S1 and S2. Consider the following code segments that are executed by T1 and T2 as shown below; cs() represents accessing a critical section.

T1:

down ( S1 );

down ( S2 );

cs();

up ( S2 );

up ( S1 );

T2:

down ( S2 );

down ( S1 );

cs();

up ( S1 );

up ( S2 );

Is deadlock possible between the two threads? If yes, how could you modify the code to prevent deadlock?

A. Deadlock is not possible.

B. Deadlock is possible. It can be prevented by swapping the order of executing down ( S1 ) and down ( S2 ) statements of T1 and leave the code of T2 unchanged.

**C. Deadlock is possible. It can be prevented by swapping the order of executing down ( S1 ) and down ( S2 ) statements of both T1 and T2.**

D. Deadlock is possible. It can be prevented by swapping the order of executing up ( S1 )

and up ( S2 ) statements of T1 and leave the code of T2 unchanged.

E. Deadlock is possible. It can be prevented by swapping the order of executing up ( S1 ) and up ( S2 ) statements of both T1 and T2.

5. What are the four necessary conditions for a deadlock to occur in a computing system?

Choose four answers.

**A. no preemption**

**B. holdandwait**

**C. circular wait**

**D. mutual exclusion**

E. starvation

6. Which of the following concerning deadlock of a system and its resource graph is not true?

A. If the graph can be reduced by all the processes, then there is no deadlock.

B. If the graph cannot be reduced by all the processes, the irreducible processes constitute the set of deadlock processes in the graph.

C. If there is no cycle in the graph, no deadlock exists in the system.

**D. If there is a cycle in the graph, deadlock exists in the system.**

E. If there is deadlock in the system, the graph has a cycle.

7. This question relates to your labs. What is the SDL function that corresponds to the POSIX pthread\_join function?

A. SDL\_CreateThread

**B. SDL\_WaitThread**

C. SDL\_SemPost

D. SDL\_SemWait

E. SDL\_CondWait

8. Which of the following SDL functions locks a mutex?

A. SDL\_CreateMutex

B. SDL\_DestroyMutex

C. SDL\_mutexV

**D. SDL\_mutexP**

E. SDL\_WaitThread

1. Which of the following is true?

A. The low cost of main memory coupled with the increase in memory capacity in most systems has obviated the need for memory management strategies.

**B. The number of faults for a particular process always decreases as the number of page frames allocated to a process increases when firstin firstout ( FIFO ) replacement is used.**

C. LRU is designed to benefit processes that exhibit spatial locality.

D. Looping through an array exhibits both spatial and temporal locality.

E. The firstfit memory allocation strategy requires that the free memory list be sorted.

2. Consider a system consisting of  $m$  resources of the same type, being shared by  $n$  processes. Resources can be requested and released by processes only one at a time. Consider the case of

$m = 6$

$n = 7$

Which of the following cases is deadlock free?

A. The maximum need of each process is 2.

**B. The maximum need of each of 4 processes is 1; the maximum need of each of another two processes is 2, and the maximum need of the remaining process is 4.**

C. Each process needs no more than 2 units of resources.

D. The maximum need of each of three processes is 4 and the maximum need of each of the remaining processes is 1.

E. The maximum need of each of three processes is 3 and the maximum need of each of the remaining processes is 2.

3. This question relates to your labs. Which of the following is the correct POSIX function that creates a semaphore.

**A. sem\_open()**

B. sem\_create()

C. POSIX\_OpenSemaphore()

D. POSIX\_CreateSemaphore()

E. POSIX\_CreateSem()

4. What is the CPU utilization of a system with degree of multiprogramming equals 10 if the probability of a process in the waiting state is 0.5?

A. about 60%

B. about 70%

C. about 80%

D. about 90%

**E. about 99.9%**

5. Which of the following memory allocation methods allocates the largest hole?

A. quick fit

B. first fit

**C. worst fit**

D. best fit

E. next fit

6. Suppose we use the Banker's Algorithm to avoid deadlock. Consider a system that has 5 processes P1, P2, P3, P4, P5 and four types of resources a, b, c and d; it has 6 units of a, 3 units of b, 4 units of c, 2 units of d.

The matrix EXISTING[] denotes the existing resource units and is always ( 6 3 4 2 ). Matrix POSSESSED[] denotes the resource units held by the processes; AVAILABLE[] denotes the units available for allocation; ALLOCATED[] denotes the units that have been allocated to the processes; NEED[] denotes the units that may still be needed by the processes to complete the tasks.

Suppose at a certain state the matrices are as shown below.

EXISTING = ( 6 3 4 2 )

ALLOCATED=

2 0 1 0

0 1 0 1

1 0 1 0

1 1 0 0

1 0 1 0

NEED =

1 1 0 1

0 1 1 1

2 0 1 0

1 2 1 0

1 1 0 0

Which of the following is the correct AVAILABLE matrix?

A. (1, 1, 0, 0 )

**B. ( 1, 1, 1, 1 )**

C. ( 6, 3, 4, 2 )

D. ( 5, 2, 3, 1 )

E. ( 5, 5, 3, 3 )

7. A system of 64K virtual memory with page size 4K is mapped to a 32K main memory as shown below.

What physical location will the virtual address 0010 000000000011 be mapped to?

A. 010 000000000011

B. 011 000000000010

C. 0011 000000000010

**D. 110 000000000011**

E. 010 000000000000

8. In the above question, what will happen if the CPU executes an instruction that needs to access the location 20488(= 20K + 8)?

A. A page fault is generated.

B. The MMU will put the address 20488 on the address bus.

C. The MMU will generate the physical address 20488.

D. The MMU will translate the address to 10248 (=10K + 8).

**E. The MMU will translate the address to 12288 (=12K + 8)**