We follow the order of Version B
Question 1

What is the major disadvantage of not having privileged instructions?

A. Unrestricted user access to the data on the hard drive.
B. Too many context switches.
C. Both A and B.
D. None of the above.
Question 1

What is the major disadvantage of *not having privileged instructions*?

A. Unrestricted user access to the data on the hard drive.
B. Too many context switches.
C. Both A and B.
D. None of the above.
Question 2

In a vectorized interrupt system, an interrupt can only be interrupted by an interrupt of higher priority.

A. TRUE  
B. FALSE
Question 2

In a vectorized interrupt system, an interrupt can only be interrupted by an interrupt of higher priority.

A. TRUE
B. FALSE
Question 3

A program in execution is called

A. A channel.
B. A function.
C. A procedure.
D. A process.
Question 3

A program in execution is called

A. A channel.
B. A function.
C. A procedure.
D. A process.
Question 4

What is the major reason for the success of modular kernels?

A. They make kernels more secure.
B. They let users add extensions to the kernel.
C. Both A and B.
D. None of the above.
Question 4

What is the major reason for the success of modular kernels?

A. They make kernels more secure.
B. They let users add extensions to the kernel.
C. Both A and B.
D. None of the above.
Question 5

Each process has its own Process Control Block.

A. TRUE
B. FALSE
Question 5

- Each process has its own Process Control Block.

A. TRUE
B. FALSE
Question 6

Which of these events can move a process from the *running* state to the *blocked* state?

A. The process performs a system call.
B. The process is swapped out.
C. A timer interrupt.
D. All of the above.
The big diagram
Question 6

Which of these events can move a process from the *running* state to the *blocked* state?

A. The process performs a system call.
B. The process is swapped out.
C. A timer interrupt.
D. All of the above.
Question 7

Which of the following statements apply to the program.cs.uh.edu server?

A. It is an interactive system.
B. It is a time-sharing system.
C. Both A and B.
D. None of the above.
Question 7

Which of the following statements apply to the program.cs.uh.edu server?

A. It is an interactive system.
B. It is a time–sharing system.
C. Both A and B.
D. None of the above.
Question 8

How many lines will the following program print?

```cpp
main()
{
    cout << "Hello!\n";
    fork();
    cout << "Goodbye!\n";
}
```

A. One line   B. Two lines
C. Three lines D. Four lines
Let us check (I)

$ more quiz1.cpp
#include <iostream>
#include <unistd.h>
using namespace std;

main(){
    cout << "Hello!\n";
    fork();
    cout << "Goodbye!\n";
} // main
Let us check (II)

```bash
$ g++ quiz1.cpp -o quiz1.exe

$ ./quiz1.exe
Hello!
Goodbye!
Goodbye!
```
Question 8

How many lines will the following program print?

```cpp
main()
{
    cout << "Hello!\n";
    fork();
    cout << "Goodbye!\n";
}
```

A. One line  B. Two lines  
C. Three lines  D. Four lines
Question 9

Which system call is used to send a signal to another process?

A. `exec()`  
B. `kill()`  
C. `notify()`  
D. `signal()`
Which system call is used to send a signal to another process?

A. exec()
B. kill()
C. notify()
D. signal()
Question 10

Which of the following operating systems is **not derived** from UNIX?

A. Android.
B. Chrome.
C. MacOS.
D. Windows.
Question 10

Which of the following operating systems is \textit{not derived} from UNIX?

A. Android.
B. Chrome.
C. MacOS.
D. Windows.
Question 11

Which one of the following is not shared by threads that share the same address space?

A. Their stacks.
B. Their program counters.
C. Both A and B.
D. Neither A nor B.
Question 11

Which one of the following is \textbf{not shared} by threads that share the same address space?

A. Their stacks.
B. Their program counters.
C. Both A and B.
D. Neither A nor B.
Which of the following statements about the `fork()` system call is **false**?

A. Both child and the parent processes share the same opened file descriptors.
B. `fork()` returns zero in the child process.
C. `fork()` returns zero in the parent process.
D. The child process has an identical copy of the address space of the parent.
Question 12

Which of the following statements about the `fork()` system call is **false**?

A. Both child and the parent processes share the same opened file descriptors.
B. `fork()` returns zero in the child process.
C. `fork()` returns zero in the parent process.
D. The child process has an identical copy of the address space of the parent.
Question 13

Which system call returns the process ID of a terminated child?

A. exit ()
B. fork()
C. signal()
D. wait()
Question 13

Which system call returns the process ID of a terminated child?

A. exit ()
B. fork()
C. signal()
D. wait()
Question 14

The time required to create a new thread in an existing process is:

A. A function of the number of threads already created by the process.
B. Greater than the time required to create a new process.
C. Less than the time required to create a new process.
D. More or less equal to the time required to create a new process.
Question 14

The time required to create a new thread in an existing process is:

A. A function of the number of threads already created by the process.
B. Greater than the time required to create a new process.
C. Less than the time required to create a new process.
D. More or less equal to the time required to create a new process.
Question 15

- The `execv()` system call specifies which new program a process should execute.

A. TRUE  
B. FALSE
The `execv()` system call specifies which new program a process should execute.

A. TRUE
B. FALSE
Alternate question on Version A

- The **execv()** system call creates a new process.

  A. TRUE  
  B. FALSE
Alternate question on Version A

- The `execv()` system call creates a new process.

A. TRUE
B. FALSE
Question 16

Which hardware mechanism allows a device to notify the CPU of an event?

A. Interrupts.
B. Polling.
C. System calls.
D. Upcalls.
Question 16

Which hardware mechanism allows a device to notify the CPU of an event?

A. Interrupts.
B. Polling.
C. System calls.
D. Upcalls.
Question 17

Which of these events can move a process from the *running* state to the *ready* state?

A. A timer interrupt.
B. The arrival in the ready state of a higher-priority process.
C. Both A and B.
D. None of the above.
Question 17

Which of these events can move a process from the *running* state to the *ready* state?

A. A timer interrupt.
B. The arrival in the ready state of a higher-priority process.
C. Both A and B.
D. None of the above.
Alternate question on Version A

Which of these events can move a process from the *running* state to the *blocked* state?

A. A timer interrupt.
B. The process performs a system call.
C. The process is swapped out.
D. All of the above.
Question 18

In which queue is a newly created process initially put?

A. Device queue.
B. I/O queue.
C. Ready queue.
D. Waiting queue.
Question 18

In which queue is a newly created process initially put?

A. Device queue.
B. I/O queue.
C. Ready queue.
D. Waiting queue.
Question 19

Memory protection is normally done through *privileged instructions*.

A. TRUE
B. FALSE
Question 19

- Memory protection is normally done through *privileged instructions*.

A. TRUE
B. FALSE
Question 20

Delaying disk—or SSD—writes

A. Increases the number of context switches.
B. May result in lost data if the system crashes.
C. Both A and B.
D. Neither A nor B.
Question 20

- Delaying disk—or SSD—writes

A. Increases the number of context switches.
B. May result in lost data if the system crashes.
C. Both A and B.
D. Neither A nor B.
Question 21

Which of the following applications is a real-time application with soft deadlines?

A. An interactive computing session.
B. Industrial process control.
C. Watching a video.
D. Missile guidance.
Question 21

Which of the following applications is a real-time application with soft deadlines?

A. An interactive computing session.
B. Industrial process control.
C. Watching a video.
D. Missile guidance.
Question 22

Which of the following statements does not apply to microkernels?

A. They are extensible.
B. They are faster than most other kernel organizations.
C. They are more reliable than most other kernel organization.
D. None of the above.
Question 22

Which of the following statements does *not* apply to microkernels?

A. They are extensible.
B. They are faster than most other kernel organizations.
C. They are more reliable than most other kernel organization.
D. None of the above.
A process in the *ready state* can only move from that state to the:

A. Blocked state.
B. New state.
C. Running state.
D. Terminated state.
The big diagram

- **New**
  - Admit process

- **Ready**
  - Get CPU
  - Completion

- **Running**
  - Exit
  - System request

- **Blocked**
  - Interrupt

- **Terminated**
Question 23

A process in the **ready state** can only move from that state to the:

A. Blocked state.
B. New state.
C. Running state.
D. Terminated state.
Alternate question on Version A

- A process in the *blocked state* can only move from that state to the:

  A. Running state.
  B. Ready state.
  C. New state.
  D. Terminated state.
The big diagram

- New
- Ready
- Blocked
- Running
- Terminated

- Admit process
- Get CPU
- Completion
- Interrupt
- Exit
- System request
Alternate question on Version A

- A process in the **blocked state** can only move from that state to the:

  A. Running state.
  B. Ready state.
  C. New state.
  D. Terminated state.
Question 24

Which of the following actions are the normal result of a system call?

A. An interrupt occurs.
B. The calling process is moved to the suspended state.
C. Both A and B.
D. None of the above.
Question 24

Which of the following actions are the normal result of a system call?

A. An interrupt occurs.
B. The calling process is moved to the suspended state.
C. Both A and B.
D. None of the above.
Question 25

What is the default action a Linux process takes when it receives a signal from another process?

A. It acknowledges it.
B. It catches it.
C. It ignores the signal.
D. It terminates.
Question 25

What is the **default action** a Linux process takes when it receives a signal from another process?

A. It acknowledges it.
B. It catches it.
C. It ignores the signal.
D. It terminates.
Version A Key

- Page 1:  C  C  A  A  A  A  B
- Page 2:  B  C  D  B  C  B
- Page 3:  A  A  C  A  C  B  B
- Page 4:  C  B  C  A  A  B
Version B Key

- Page 1: A A D B A A
- Page 2: C B B D C C
- Page 3: D C A A C C B
- Page 4: B C B C A D