Which of the following is false about deadlocks? *

● Deadlocks cannot happen if resources are always requested in a specific sequence.

○ Deadlocks can be avoided if resources can be preemptively taken away from other processes.

○ The Linux kernel by default has a module that observes the resources requested, and avoids deadlocks based on it.

○ Deadlocks occur due to circular requests of resources by processes.

Which of the following is true in general about paging and segmentation in computers? *

● Paging is supported by all architectures, but segmentation is not

○ Segmentation is present by all architectures, but paging is not

○ Both paging and segmentation are supported by all architectures

○ None of the other given options

Suppose you want to share a lot of data among multiple processes, but there is no strict requirement of synchronization. What is the best data structure for this requirement? *

○ Message queue

○ Pipe

● Shared memory

○ None of the other options

Which of the following is a disadvantage of an inverted page table? *

○ Size of page table is too large

○ Not all page entries are present in the page table

● Its hardware is much more complex to implement

○ It is vulnerable to more page faults

Suppose a designer of a memory system finds that the size of page table is too large. Which of the following options are possible? *

○ Increase the size of page frames

○ Use multi-level page table

○ Use inverted page table

● All the other options are possible

○ None of the other options are possible

Suppose you want to execute a shell script? Which of the permissions is NOT necessary for it to execute? *

○ All the options are necessary

○ Read

● Write

○ Execute

Which of the following does not slow down file reads? *

● Slower movement of mechanical head

○ Fragmentation of files

○ Disk caches

○ Random requests to access data from disk

Suppose you write a single C program that prints integers from 1 to 100, and then computes their sum. Now, you run this program simultaneously from two different shells. How many processes (ignoring the shell processes) and how many additional threads (excluding the main thread) were created to run this C program? *

● 1, 1

○ 2, 2

○ 1, 0

○ 2, 0

Which of the following events are NOT handled by the interrupt handler? *

○ Divide by zero

○ Overflow of a number

○ Page fault

● Access to memory outside the process address space

Which of the following techniques do NOT utilize the principle of locality? *

○ Accessing data on disks

○ Accessing page frames

○ Load/Store instructions to get data

● All of the above utilize locality

In which of the following data sharing scheme is it possible to actually transfer data without using a system call? *

○ Pipe

○ Message Queue

● Shared memory

○ Socket

Why do Linux filesystems have a maximum file size? *

○ To avoid excess disk fragmentation

○ To save disk space

● To limit the number of inode indirections

○ None of the above

The memory ranges visible to Linux Modules are: *

○ a. Virtual address ranges.

○ b. Corresponding to kernel address spaces.

○ c. Use the same virtual memory management system (such as demand paging system) like the rest of the system does.

○ d. The real addresses and not the virtual address, as the latter is only for application programs.

○ e. A&B

● f. A&C

○ g. None of the above

The main reason why interrupts are often not disabled in interrupt handling is because: *

○ It is not supported by all CPU architectures.

● So as to allow a single interrupt handler to handle interrupts from various devices.

○ For more system responsiveness, we require preemptive interrupt handling.

○ Because disabling interrupts is often not an atomic operation.

The memory ranges visible to Interrupt handlers are: *

○ Virtual address ranges.

○ Corresponding to kernel address spaces.

○ Use the same virtual memory management system (such as demand paging system) like the rest of the system does.

● All the above.

The main reason to store attributes of a files with the inodes (instead of managing centrally with a common data structures) is: *

○ To be able to keep attributes intact when the files are moved from one physical storage to another.

● To be able to maintain consistency across all files. The OS need not worry about the attributes.

○ The attributes are actually stored in different data structures within the filesystem

○ None of the other given options

Which of the following is true for Non-Maskable Interrupts (NMIs): *

○ One cannot implement an interrupt handler for NMIs, the CPU handles them through built-in instructions.

○ If a NMI and a regular interrupt arrive together, the regular interrupt generally has higher precedence.

● If a NMI and a regular interrupt arrive together, the NMI takes precedence over regular interrupt.

○ A regular interrupt could take precedence over NMI by appropriately re-programming the Programmable Interrupt Controller (PIC).

The following is the difference between pthread_mutex_lock() and advisory locks (e.g. flock()): *

○ Pthread_mutex_lock() can be both blocking and non-blocking while flock() is always blocking.

○ Advisory locks like flock() and pthread_mutex() work exactly the same way, they differ only in the arguments they use.

○ Pthread_mutex_lock() can only be used in multithreaded programs, while flock() can be used in all cases.

● Pthread_mutex_lock() causes the calling thread to sleep, if some other process has acquired the lock, while flock() does not.

○ None of the other given options

Which of the following is true about library functions and processes. *

○ Just the way programs can run library routines, they may also deliberately or inadvertently corrupt the local and global variable of library functions.

○ Library functions don’t segfault as they use their own heap memory and do not use statically allocated variables, nor do they use dangling pointers.

● Pages containing library functions are treated like pages of other processes, as in they could be replaced by page eviction strategies, much like the pages of regular processes (say following a LFU strategy).

○ You can load libraries in the main memory during boot-up by configuring appropriate rc scripts, so that they stay resident in RAM when process invoke the functions therein.

○ Library functions have addresses that are mapped to code pages of all processes. Therefore, a malicious user could access the code, stack and data of other processes, using the access to library function addresses as conduit.

○ None of the other given options

Which of the following operations is well suited to be handled using DMA operations: *

○ Handling several quick keystrokes and mouse pointer movements in applications like computer games.

● Reading or writing data from the network card device -- e.g. while web browsing that involves exchanging millions of bits per second.

○ Reading or writing to the sound card say when making audio calls over the Internet.

○ Writing to a printer connected via USB (Universal Serial Bus).