

Q1. लिनक्स में wait() सिस्टम कॉल का उद्देश्य क्या है?

- (A) चाइल्ड प्रोसेस की संख्या गिनना
- (B) सभी चाइल्ड प्रोसेस को किल करना
- (C) ज़ॉम्बी प्रोसेस को रिप करना
- (D) पहले पैरेंट प्रोसेस को इनिट करना और फिर प्रत्येक चाइल्ड प्रोसेस को किल करना

What is the purpose of wait() system call in Linux?

- (A) Count the number of child processes
- (B) Kill all child processes
- (C) Reap zombie processes
- (D) First make init the parent process and then kill each child process

Q2. म्यूटेक्स, सेमाफोर और स्पिनलॉक के बारे में, निम्नलिखित में से कौन सा कथन सही नहीं है?

- (A) म्यूटेक्स एकल इंस्टेंस संसाधन की सुरक्षा करता है और इसे प्राप्त करने वाले थ्रेड के पास इसका स्वामित्व होता है
- (B) सेमाफोर एक संसाधन के कई इंस्टेंस तक पहुँच का प्रबंधन करते हैं, इनका स्वामित्व नहीं होता
- (C) स्पिनलॉक का उपयोग बहुत छोटे महत्वपूर्ण खंडों के लिए किया जाता है और थ्रेड बार-बार जाँच करता है कि बिजी-वेट लूप में कोई लॉक उपलब्ध है या नहीं। ये कर्नेल-स्तरीय प्रोग्रामिंग के लिए आदर्श हैं
- (D) बिजी-वेट के कारण स्पिनलॉक का ओवरहेड अधिक होता है। थ्रेड पर ब्लॉकिंग के कारण म्यूटेक्स और सेमाफोर का ओवरहेड कम होता है और ये स्पिनलॉक की तुलना में अधिक कुशल होते हैं

About mutex, semaphore, and spinlock, which of the following is NOT a correct statement?

- (A) Mutex protects a single instance resource and is owned by the thread that acquires it
- (B) Semaphores manage access to multiple instances of a resource, do not have ownership
- (C) Spinlocks are used for very short critical sections and thread repeatedly checks if a lock is available in busy-wait loop. They are ideal for kernel level programming
- (D) Spinlocks have higher overhead due to busy-wait. Mutexes and semaphore have low overhead due to blocking on thread and are more efficient than spinlocks

Q3. जलवायु मॉडलिंग, औषधि खोज और पदार्थ विज्ञान जैसे क्षेत्रों में सिमुलेशन को _____ से त्वरित किया जा सकता है।

- (A) इन-मेमोरी कंप्यूटिंग
- (B) शेयर्ड मेमोरी
- (C) ट्रांजेक्शनल मेमोरी
- (D) टर्शियरी स्टोरेज कंट्रोलर

Simulations in fields like climate modeling, drug discovery, and materials science can be accelerated with:

- (A) In-Memory Computing
- (B) Shared Memory
- (C) Transactional Memory
- (D) Tertiary Storage controllers

Q4. लिनक्स में, फ़ाइल सिस्टम विवरण जानने के लिए निम्नलिखित में से कौन सा उपयुक्त नहीं है?

In Linux, which of the following is not suitable to know file system details?

- (A) fdisk -l
- (B) df -hT
- (C) lsblk
- (D) diskpart -lv

Q5. pthread के बारे में, निम्नलिखित में से कौन सा कथन सही नहीं है?

- (A) pthread_create() का उपयोग एक नया थ्रेड बनाने के लिए किया जाता है
- (B) pthread_join() एक थ्रेड को दूसरे थ्रेड के निष्पादन पूरा होने तक प्रतीक्षा करने की अनुमति देता है
- (C) pthread_mutex_lock() का उपयोग म्यूटेक्स का उपयोग करके रेस कंडीशन को रोकने के लिए किया जाता है
- (D) pthread_attr_t() का उपयोग थ्रेड विशेषताएँ, स्टैक आकार और शेड्यूलिंग नीति सेट करने के लिए किया जाता है

About pthread, which of the following is NOT a correct statement?

- (A) pthread_create() is used to create a new thread
- (B) pthread_join() allows one thread to wait for another thread to complete its execution
- (C) pthread_mutex_lock() is used to prevent race conditions using mutex
- (D) pthread_attr_t() is used to set thread attributes, stack size and scheduling policy

Q6. पोसिक्स सेमाफोर के बारे में, निम्नलिखित में से कौन सा कथन सही नहीं है?

- (A) sem_wait() सेमाफोर के मान को घटाता है। यदि मान वर्तमान में शून्य है, तो कॉलिंग थ्रेड या प्रक्रिया तब तक ब्लॉक रहती है जब तक कि सेमाफोर का मान शून्य से अधिक न हो जाए
- (B) sem_post() सेमाफोर के मान को बढ़ाता है। यह क्रिया सेमाफोर पर प्रतीक्षारत थ्रेड या प्रक्रिया को अनब्लॉक कर सकती है
- (C) sem_trywait(): बिना ब्लॉक किए सेमाफोर को घटाने का प्रयास करता है
- (D) sem_trypost(): सेमाफोर को बढ़ाने का प्रयास करता है और थ्रेड या प्रक्रिया को प्रतीक्षा में रखता है

About POSIX semaphores, which of the following is NOT a correct statement?

- (A) sem_wait() decrements the semaphore's value. If the value is currently zero, the calling thread or process blocks until the semaphore's value becomes greater than zero
- (B) sem_post() increments the semaphore's value. This operation can unblock a thread or process that was waiting on the semaphore
- (C) sem_trywait(): Attempts to decrement the semaphore without blocking
- (D) sem_trypost(): Attempts to increment the semaphore and keeps the thread or process in waiting

Q7. ऑपरेटिंग सिस्टम में, म्यूटेक्स _____ सुनिश्चित करने के लिए एक मूलभूत सिंक्रोनाइज़ेशन प्रिमिटिव है।

- (A) इन्स्ट्रक्शन पैरललिज़्म
- (B) डेटा पैरललिज़्म
- (C) डेटा इंटेग्रिटी
- (D) डेटा इम्यूटेबिलिटी

In OS, mutex is a fundamental synchronization primitive to ensure:

- (A) instruction parallelism
- (B) data parallelism
- (C) data integrity
- (D) data immutability

Q8. थ्रेड सिंक्रोनाइज़ेशन के लिए उपयोग किए जाने वाले कंडीशनल वैरिएबल _____ की चुनौती प्रस्तुत नहीं करते हैं।

- (A) बीजी वेटिंग
- (B) स्पूरियस वेक-अप
- (C) डेडलॉक
- (D) रेस कंडीशन

Conditional variables used for thread synchronization does not present the challenge of :

- (A) busy waiting
- (B) spurious wake-ups
- (C) deadlocks
- (D) race conditions

Q9. शेयर्ड मेमोरी के बारे में निम्नलिखित में से कौन सा कथन गलत है?

- (A) शेयर्ड मेमोरी में रेस कंडीशन को रोकने के लिए जटिल सिंक्रोनाइज़ेशन की आवश्यकता होती है
- (B) शेयर्ड मेमोरी सुरक्षा जोखिम पैदा कर सकती है क्योंकि कई प्रक्रियाएँ डेटा तक पहुँच सकती हैं
- (C) शेयर्ड मेमोरी सबसे धीमा आईपीसी तंत्र है क्योंकि इसके लिए कर्नेल हस्तक्षेप की आवश्यकता होती है
- (D) शेयर्ड मेमोरी आईपीसी कार्यान्वयन प्लेटफ़ॉर्म पर निर्भर है

Which of the following is incorrect about shared memory?

- (A) Shared memory requires complex synchronization to prevent race conditions
- (B) Shared memory can pose security risks because multiple processes can access data
- (C) Shared memory is slowest IPC mechanism because this requires kernel intervention
- (D) Shared memory IPC implementation is platform-dependent

Q10. निम्नलिखित में से कौन सा क्रम सही है?

- (A) पावर ऑन, बूट लोडर शुरू होता है, पोस्ट जाँच, बायोस/यूइएफआई सक्रिय होता है, ओएस सबसिस्टम लोड करता है
- (B) पावर ऑन, पोस्ट जाँच, बायोस/यूइएफआई सक्रिय होता है, बूट लोडर शुरू होता है, ओएस सबसिस्टम लोड करता है
- (C) पावर ऑन, बायोस/यूइएफआई सक्रिय होता है, पोस्ट जाँच, बूट लोडर शुरू होता है, ओएस सबसिस्टम लोड करता है
- (D) पावर ऑन, बायोस/यूइएफआई सक्रिय होता है, पोस्ट जाँच, ओएस सबसिस्टम लोड करता है, बूट लोडर शुरू होता है

Which of the following sequence is in the correct order?

- (A) Power On, Boot Loader starts, POST checks, BIOS/UEFI wakes up, OS loads subsystems
- (B) Power On, POST checks, BIOS/UEFI wakes up, Boot Loader starts, OS loads subsystems
- (C) Power On, BIOS/UEFI wakes up, POST checks, Boot Loader starts, OS loads subsystems
- (D) Power On, BIOS/UEFI wakes up, POST checks, OS loads subsystems, Boot Loader starts

Q11. लिनक्स कर्नेल किस रूप में कार्य करता है:

- (A) मोनोलिथिक कर्नेल
- (B) माइक्रोकर्नेल
- (C) एक्सोकर्नेल
- (D) सभी विकल्प

Linux kernel functions as:

- (A) monolithic kernel
- (B) microkernel
- (C) exokernel
- (D) All of the options

Q12. लिनक्स में, निम्नलिखित में से कौन सा सही नहीं है?

- (A) उपयोगकर्ता इनपुट को ग्लोबल सिंक /dev/null पर रीडायरेक्ट किया जाता है
- (B) लिनक्स में स्वेप पार्टिशन अनिवार्य नहीं है
- (C) /dev में डिवाइस फ़ाइलें होती हैं, जो प्रत्येक हार्डवेयर डिवाइस को एक फ़ाइल के रूप में दर्शाती हैं
- (D) रूट फ़ाइल सिस्टम, समग्र फ़ाइल सिस्टम पदानुक्रम का विस्तार करने के लिए / से जुड़ा होता है

In Linux, which of the following is NOT correct?

- (A) user input is redirected to global sink /dev/null
- (B) a swap partition is not mandatory in Linux
- (C) /dev contains device files, representing each hardware device as a file
- (D) root file system is attached to / to extend the overall file system hierarchy

Q13. निम्नलिखित में से कौन सा लिनक्स वर्चुअलाइजेशन सॉफ्टवेयर नहीं है?

- (A) केवीएम
- (B) इएसएक्स
- (C) वर्चुअलबॉक्स
- (D) वीएमवेयर वर्कस्टेशन

Which of the following is not a Linux virtualization software?

- (A) KVM
- (B) ESX
- (C) VirtualBox
- (D) VMWare Workstation

4_HPC System software development

- Q14.** सी प्रोग्रामिंग में, प्रकार रूपांतरण के संबंध में निम्नलिखित में से कौन सा कथन असंगत है?
- (A) अंतर्निहित या स्पष्ट प्रकार रूपांतरण के साथ, यदि लक्ष्य प्रकार मान धारण नहीं कर सकता, तो अतिप्रवाह या अल्पप्रवाह हो सकता है
 - (B) अंतर्निहित या स्वचालित प्रकार रूपांतरण ओएस द्वारा किया जाता है
 - (C) अंतर्निहित या स्पष्ट दोनों प्रकार के रूपांतरण के परिणामस्वरूप डेटा हानि होती है, क्योंकि लक्ष्य प्रकार छोटा डेटा प्रकार होता है
 - (D) वास्तविक संख्याओं से पूर्णाकों में रूपांतरण से हमेशा परिशुद्धता की हानि होती है

In C programming, about type conversion what is odd one from the following?

- (A) With implicit or explicit type conversion, if the target type cannot hold the value, overflow or underflow can occur
- (B) Implicit or automatic type conversion is performed by OS
- (C) Both implicit or explicit type conversion result into data loss with target being smaller data type
- (D) Conversion from real numbers to integers always results in a loss of precision

- Q15.** निम्नलिखित फ़ंक्शन क्या करेगा?

```
unsigned int myfunction(unsigned int value)
```

```
{  
    return ((value & 0xFF000000) >> 24) |  
           (((value & 0x00FF0000) >> 16) << 8) |  
           (((value & 0x0000FF00) >> 8) << 16) |  
           ((value & 0x000000FF) << 24);  
}
```

- i. myfunction एक पूर्णांक के 4 बाइट्स को स्वैप करता है (32-बिट प्लेटफ़ॉर्म पर)
- ii. myfunction लिटिल-एंडियन निरूपण को बिग-एंडियन में परिवर्तित कर सकता है
- iii. myfunction बिग-एंडियन निरूपण को लिटिल-एंडियन में परिवर्तित कर सकता है

- (A) केवल i
- (B) केवल i और ii
- (C) केवल ii और iii
- (D) सभी i, ii और iii

What is the following function doing?

```
unsigned int myfunction(unsigned int value)
```

```
{  
    return ((value & 0xFF000000) >> 24) |  
           (((value & 0x00FF0000) >> 16) << 8) |  
           (((value & 0x0000FF00) >> 8) << 16) |  
           ((value & 0x000000FF) << 24);  
}
```

- i. myfunction swaps the 4 bytes of an integer (on a 32-bit platform)
- ii. myfunction can convert the little-endian representation to big-endian
- iii. myfunction can convert the big-endian representation to little-endian

- (A) Only i
- (B) Only i and ii
- (C) Only ii and iii
- (D) All i, ii, and iii

Q16. सी प्रोग्रामिंग में निम्नलिखित में से कौन सा सही आइडेंटिफायर है?

Which one of the following is correct identifier in C programming?

- (A) `_init_`
- (B) `1stSeat`
- (C) `##token`
- (D) `return-value`

Q17. सी प्रोग्रामिंग में, निम्नलिखित में से कौन सा एक सामान्य प्री-प्रोसेसर निर्देश है जिसका उपयोग हेडर फाइलों में उन्हें एकल संकलन इकाई में कई बार शामिल होने से रोकने के लिए किया जाता है?

In C programming, which of the following is a common pre-processor directive used in header files to prevent them from being included multiple times in a single compilation unit?

- (A) `#pragma`
- (B) `#pragma pack`
- (C) `#pragma startup`
- (D) `#pragma once`

Q18. सी, ऐरे को स्टोर करने के लिए _____ का उपयोग करता है।

- (A) रो-मेजर ऑर्डर
- (B) कॉलम-मेजर ऑर्डर
- (C) स्पार्स-फ्रिक्वेंसी ऑर्डर
- (D) स्टैटिक एडजासेंसी लिस्ट

C uses _____ to store arrays.

- (A) row-major order
- (B) column-major order
- (C) sparse-frequency order
- (D) static adjacency list

Q19. सी में पॉइंटर्स के बारे में क्या सही है?

- (A) 32-बिट या 64-बिट सिस्टम पर, जेनेरिक पॉइंटर हमेशा 4 बाइट्स लेता है
- (B) किसी फ़ंक्शन का पॉइंटर और पॉइंटर रिटर्न करने वाला वाला फ़ंक्शन एक ही चीज़ हैं
- (C) पॉइंटर एक्सेस हमेशा ऐरे से तेज़ होता है
- (D) ऐरे के लिए: `arr[i]`, `*(arr+i)`, `*(i+arr)` और `i[arr]` एक ही चीज़ हैं

What is true about pointers in C?

- (A) On a 32-bit or 64-bit systems generic pointer always takes 4 bytes
- (B) Pointer to a function and function returning a pointer are one and same thing
- (C) Pointer access is always faster than an array
- (D) For an array: `arr[i]`, `*(arr+i)`, `*(i+arr)` and `i[arr]` are same thing

Q20. निम्नलिखित में से कौन सा पॉइंटर्स का एक ऐरे है?

Which of the following is an array of pointers?

- (A) `int *ptr_array[5];`
- (B) `int (*ptr_array())[5];`
- (C) `int (*ptr_array)[5];`
- (D) `int *ptr_array()[5];`

- Q21. निम्नलिखित में से कौन सा एब्स्ट्रैक्ट डाटा टाइप (एडीटी) का उदाहरण है?
- (A) सी में एक ऐरे
 - (B) पायथन में एक लिंकड सूची कार्यान्वयन
 - (C) एक स्टैक
 - (D) एक विशिष्ट मेमोरी आवंटन योजना

Which of the following is an example of an Abstract Data Type (ADT)?

- (A) An array in C
- (B) A linked list implementation in Python
- (C) A stack
- (D) A specific memory allocation scheme

- Q22. लिनियर सेरच एल्गोरिदम की सबसे खराब टाइम कॉम्प्लेक्सीटी क्या है?

What is the worst-case time complexity of a linear search algorithm?

- (A) $O(1)$
- (B) $O(\log n)$
- (C) $O(n)$
- (D) $O(n^2)$

- Q23. निम्नलिखित में से कौन-सा ऑपरेशन लिनियर लिंकड लिस्ट की तुलना में डबली-लिंकड लिस्ट द्वारा अधिक कुशलता से किया जाता है?

- (A) दिए गए स्थान वाले नोड को डिलीट करना
- (B) किसी दिए गए आइटम के लिए एक अक्रमित सूची में खोज करना
- (C) दिए गए स्थान वाले नोड के बाद एक नोड सम्मिलित करना
- (D) प्रत्येक नोड को संसाधित करने के लिए सूची को पार करना

Which of the following operations is performed more efficiently by doubly linked list than by linear linked list?

- (A) Deleting a node whose location is given
- (B) Searching an unsorted list for a given item
- (C) Inserting a node after the node with a given location
- (D) Traversing the list to process each node

- Q24. किसी कतार को क्रियान्वित करने के लिए आवश्यक स्टैक की न्यूनतम संख्या है:

The minimum number of stacks needed to implement a queue is:

- (A) 1
- (B) 4
- (C) 3
- (D) 2

4_HPC System software development

- Q25. क्विक सॉर्ट निम्नलिखित में से किस विधि का उपयोग सॉर्टिंग को लागू करने के लिए करती है?
- | | |
|-----------------|--------------|
| (A) पार्टीशनिंग | (B) सेलेक्शन |
| (C) एक्सचेंजिंग | (D) मर्जिंग |

Quick sort uses which of the following method to implement sorting?

- | | |
|------------------|---------------|
| (A) partitioning | (B) selection |
| (C) exchanging | (D) merging |

- Q26. हैश ट्री किसका सामान्यीकरण है?

- | | |
|------------|---------------|
| (A) हीप | (B) हैश लिस्ट |
| (C) बीएसटी | (D) बी-ट्री |

What is a hash tree a generalization of?

- | | |
|----------|---------------|
| (A) Heap | (B) Hash list |
| (C) BST | (D) B-tree |

- Q27. कौन सा जीसीसी फ़्लैग जीडीबी के लिए डिबगिंग जानकारी उत्पन्न करता है?

Which GCC flag generates debugging information for gdb?

- | |
|------------|
| (A) -g |
| (B) -d |
| (C) -debug |
| (D) -dbg |

- Q28. जीसीसी, सी/सी++ प्रोग्रामों के लिए _____ कंपाइलर है।

- | |
|--|
| (A) जस्ट-इन-टाइम (जेआईटी) |
| (B) प्लेटफॉर्म-इंडिपेंडेंट-एग्जीक्यूशन (पीआईई) |
| (C) बिल्ड-टाइम |
| (D) अहेड-ऑफ-टाइम (एओटी) |

GCC is _____ compiler for C/C++ programs.

- | |
|--|
| (A) Just-in-Time (JIT) |
| (B) Platform-Independent-Execution (PIE) |
| (C) Build-Time |
| (D) Ahead-of-Time (AOT) |

- Q29. Vi editor में, "Average" टेक्स्ट की सभी घटनाओं को "Good" के साथ प्रतिस्थापित करने का सही तरीका क्या है?

In Vi editor, what is the correct way to substitute all occurrences of text "Average" with "Good"?

- | | |
|-----|-----------------------|
| (A) | : %s/Average/Good |
| (B) | : %s/Average/Good/g |
| (C) | : %sub/Average/Good |
| (D) | : %sub/Average/Good/g |

Q30. निम्नलिखित में से कौन-सा लिनक्स में टेक्स्ट प्रोसेसिंग उपयोगिता नहीं है?

Which one of the following is not a text processing utility in Linux?

- (A) grep
- (B) sed
- (C) awk
- (D) pipe

Q31. निम्नलिखित में से कौन सा टूल सी प्रोग्राम में बग ढूँढने और उन्हें ठीक करने में मदद करने के लिए बनाया गया है?

- (A) कंपाइलर
- (B) डीबगर
- (C) एडिटर
- (D) लिंकर

Which of the following tools is purpose-built to help in finding and fixing bugs in a C program?

- (A) Compiler
- (B) Debugger
- (C) Editor
- (D) Linker

Q32. निम्नलिखित में से कौन सा कथन सही है?

- (A) यदि सावधानी से उपयोग न किया जाए, तो कंपाइलर ऑप्टिमाइज़ेशन कभी-कभी सूक्ष्म बग उत्पन्न कर सकते हैं
- (B) कंपाइलर ऑप्टिमाइज़ेशन हमेशा कोड को काफी तेज़ बनाते हैं
- (C) कंपाइलर ऑप्टिमाइज़ेशन केवल बहुत बड़े प्रोग्रामों पर ही लागू होते हैं
- (D) कंपाइलर ऑप्टिमाइज़ेशन अच्छे एल्गोरिदम डिज़ाइन का विकल्प हैं

Which of the following statements is true?

- (A) Compiler optimizations can sometimes introduce subtle bugs if not used carefully
- (B) Compiler optimizations always make the code significantly faster
- (C) Compiler optimizations are only applicable to very large programs
- (D) Compiler optimizations are a substitute for good algorithm design

Q33. जब `fopen()` किसी फ़ाइल को सफलतापूर्वक खोलता है तो उसका रिटर्न प्रकार क्या होता है?

What is the return type of `fopen()` when it successfully opens a file?

- (A) int
- (B) char*
- (C) FILE*
- (D) void*

Q34. निम्नलिखित में से कौन सा सिग्नल किसी प्रक्रिया द्वारा पकड़ा या अनदेखा नहीं किया जा सकता है?

Which of the following signals cannot be caught or ignored by a process?

- (A) SIGKILL
- (B) SIGINT
- (C) SIGSEGV
- (D) SIGTERM

4 HPC System software development

Q35. निम्न में से कौन सी कमांड बैश में किसी फ़ाइल की अनुमतियों को बदलने के लिए उपयोग की जाती है?

Which of the following command is used to change permissions of a file in bash?

- (A) chown (B) chmod
(C) chgrp (D) mv -p

Q36. निम्नलिखित में से कौन सा वर्तमान कार्यशील डिक्शनरी को सही ढंग से प्रतिस्थापित करेगा?

Which of the following will correctly substitute current working directory?

- (A) \$pwd
(B) [[\$pwd]]
(C) <pwd>
(D) `pwd`

Q37. मान लीजिए कि एक सिस्टम पर कई अनुप्रयोगों को एक ही सामान्य लाइब्रेरी का उपयोग करने की आवश्यकता होती है। मेमोरी एफ़िसीयेन्सी के लिए कौन सी लिंकिंग विधि अधिक लाभदायक है?

- (A) स्टैटिक लिंकिंग
(B) डायनामिक लिंकिंग
(C) स्टैटिक और डायनामिक लिंकिंग दोनों समान मेमोरी दक्षता प्रदान करते हैं
(D) न तो स्टैटिक और न ही डायनामिक लिंकिंग मेमोरी दक्षता को प्रभावित करती है

Consider multiple applications on a system require to use the same common library.

Which linking method is more advantageous for memory efficiency?

- (A) Static linking
(B) Dynamic linking
(C) Both static and dynamic linking offer similar memory efficiency
(D) Neither static nor dynamic linking impacts memory efficiency

Q38. निम्नलिखित दो कथनों पर विचार करें:

S1: जब किसी फ़ाइल को किसी निष्पादन योग्य फ़ाइल में स्थिर रूप से लिंक किया जाता है, तो लिंक समय के दौरान फ़ाइल की सामग्री शामिल होती है।

S2: जब किसी फ़ाइल को गतिशील रूप से लिंक किया जाता है, तो केवल रन टाइम के दौरान ही उस फ़ाइल का पॉइंटर, जो लिंक की गई है, निष्पादन योग्य फ़ाइल में शामिल होता है और फिर उस फ़ाइल की सामग्री लोड होती है।

उपरोक्त में से कौन सा/से कथन सही है/हैं?

- (A) S1 और S2 दोनों सही हैं (B) S1 और S2 दोनों गलत हैं
(C) केवल S1 सही है (D) केवल S2 सही है

Consider the following two statements:

S1: **When a file is linked statically into an executable, the contents of the file are included during link time.**

S2: **When a file is linked dynamically, it is only during run time when a pointer to the file, which is linked in, is included in the executable and then the contents of said file are loaded.**

Which of the above statements is/are true?

- (A) Both S1 and S2 are true (B) Both S1 and S2 are false
(C) Only S1 is true (D) Only S2 is true

- Q39. आप आमतौर पर किस परिदृश्य में क्रॉस-कंपाइलर का उपयोग करेंगे?**
- (A) जब आप उसी ऑपरेटिंग सिस्टम और आर्किटेक्चर के लिए प्रोग्राम कंपाइल करना चाहते हैं जिसका आप वर्तमान में उपयोग कर रहे हैं
 - (B) किसी भिन्न प्लेटफॉर्म के लिए निष्पादन योग्य कोड बनाने के लिए, जैसे कि x86 मशीन से आर्म प्रोसेसर के लिए कंपाइल करना
 - (C) जब आप किसी नए प्लेटफॉर्म के लिए एक नेटिव कंपाइलर बना रहे हों और पहले ऑपरेटिंग सिस्टम टूल्स को कंपाइल करने की आवश्यकता हो
 - (D) उसी हार्डवेयर पर बेहतर प्रदर्शन के लिए मौजूदा कोड को अनुकूलित करने के लिए

In which scenario would you typically use a cross-compiler?

- (A) When you want to compile a program for the same operating system and architecture you are currently using
- (B) To create executable code for a different platform, such as compiling for an ARM processor from an x86 machine
- (C) When you are building a native compiler for a new platform and need to compile the operating system tools first
- (D) To optimize existing code for better performance on the same hardware

- Q40. यदि आप किसी भिन्न कर्नेल संस्करण के लिए कंपाइल किए गए कर्नेल मॉड्यूल को पुनः संकलित किए बिना लोड करने का प्रयास करते हैं, तो क्या होगा?**
- (A) मॉड्यूल लोड तो होता है, लेकिन कंपैटीबिलिटी मोड में चलता है
 - (B) कर्नेल स्वचालित रूप से मॉड्यूल को रिकंपाइल करता है
 - (C) इन्समोड "इनवैलिड मॉड्यूल फ़ारमैट" के साथ फ़ेल हो जाता है
 - (D) मॉड्यूल लोड तो होता है, लेकिन तुरंत कर्नेल पैनिक उत्पन्न कर सकता है

What happens if you attempt to load a kernel module compiled for a different kernel version without recompiling?

- (A) Module loads but runs in compatibility mode
- (B) Kernel automatically recompiles the module
- (C) insmod fails with "Invalid module format"
- (D) Module loads but may cause kernel panic immediately

- Q41. निम्नलिखित में से कौन सी फ़ाइल में वर्तमान में लोड किए गए लिनक्स कर्नेल मॉड्यूल (एलकेएम) की सूची है?**

Which of the following files contains a list of currently loaded Linux Kernel Modules (LKMs)?

- (A) /dev/modules
- (B) /proc/modules
- (C) /etc/modules
- (D) /sys/kernel/modules

- Q42.** एक कैरेक्टर डिवाइस ड्राइवर गतिशील रूप से आवंटित मेजर नंबर कैसे प्राप्त कर सकता है?
- (A) मैनुअल रूप से /proc/devices की जाँच करके
 - (B) register_chrdev() को मेजर नंबर के रूप में शून्य पास करके
 - (C) register_chrdev_dyn() को कॉल करने से पहले एक सीड मेजर नंबर सेट करके
 - (D) उपलब्ध नंबरों का पता लगाने के लिए एक कस्टम PROBE फ़ंक्शन लागू करके

How can a character device driver obtain a dynamically allocated major number?

- (A) By manually checking /proc/devices
- (B) By passing a zero as the major number to register_chrdev()
- (C) By setting a seed major number before calling register_chrdev_dyn()
- (D) By implementing a custom PROBE function to detect available numbers

- Q43.** लिनक्स कर्नेल में kmalloc() और vmalloc() फ़ंक्शन के बीच मुख्य अंतर क्या है?

- (A) kmalloc() भौतिक रूप से सन्निहित मेमोरी आवंटित करता है, जबकि vmalloc() वस्तुतः सन्निहित मेमोरी आवंटित करता है
- (B) kmalloc() वस्तुतः सन्निहित मेमोरी आवंटित करता है, जबकि vmalloc() भौतिक रूप से सन्निहित मेमोरी आवंटित करता है
- (C) kmalloc() का उपयोग उपयोगकर्ता-स्थान आवंटन के लिए किया जाता है, जबकि vmalloc() का उपयोग कर्नेल-स्थान आवंटन के लिए किया जाता है
- (D) kmalloc() हमेशा शून्य मेमोरी लौटाता है, जबकि vmalloc() ऐसा नहीं करता है और इसके लिए __GFP_ZERO फ़्लैग की आवश्यकता होती है

What is the main difference between kmalloc() and vmalloc() functions in the Linux kernel?

- (A) kmalloc() allocates physically contiguous memory, while vmalloc() allocates virtually contiguous memory
- (B) kmalloc() allocates virtually contiguous memory, while vmalloc() allocates physically contiguous memory
- (C) kmalloc() is used for user-space allocation, while vmalloc() is used for kernel-space allocation
- (D) kmalloc() always returns zeroed memory, while vmalloc() does not and requires __GFP_ZERO flag

- Q44.** बिना स्लीपिंग इंटरप्ट संदर्भ में लघु विलंब (दसियों माइक्रोसेकंड) को क्रियान्वित करने के लिए कौन सा कर्नेल फ़ंक्शन सबसे उपयुक्त है?

Which kernel function is most appropriate to implement a short delay (tens of microseconds) in interrupt context without sleeping?

- (A) msleep()
- (B) udelay()
- (C) schedule_timeout_interruptible()
- (D) mdelay()

- Q45. स्ट्रक्चर टाइमर_लिस्ट में समाप्ति के लिए प्रयुक्त समय माप की इकाई क्या है?
- (A) मिलीसेकंड (B) माइक्रोसेकंड
(C) जिफ्रीज़ (D) नैनोसेकंड

What is the unit of time measurement used for expires in struct timer_list?

- (A) Milliseconds (B) Microseconds
(C) Jiffies (D) Nanoseconds

- Q46. डिवाइस ड्राइवर में निम्नलिखित में से कौन सा सिंक्रोनाइज़ेशन मैकेनिज्म समय-महत्वपूर्ण सेक्शन और इंटरप्ट हैंडलर के लिए सबसे उपयुक्त है, जो बिजी-वेटिंग का उपयोग करके एकाधिक थ्रेड्स को एक ही कोड सेक्शन तक पहुँचने से रोकता है?
- (A) सेमाफोर (B) म्यूटेक्स
(C) स्पिनलॉक (D) एटॉमिक ऑपरेशन

Which one of the following synchronization mechanisms in a device driver is best suited for time-critical sections and interrupt handlers, to prevent multiple threads from accessing the same code section by using busy-waiting?

- (A) Semaphores (B) Mutexes
(C) Spinlocks (D) Atomic operations

- Q47. निम्नलिखित में से कौन सा सॉफ्टवेयर इंटरप्ट का उदाहरण है?
- (A) टाइमर इंटरप्ट
(B) डिविजन बाइ ज़ीरो एरर
(C) डिस्क कंट्रोलर इंटरप्ट
(D) इनवैलिड ऑपकोड

Which one of the following is an example of a software interrupt?

- (A) A timer interrupt
(B) A division-by-zero error
(C) A disk controller interrupt
(D) An invalid opcode

- Q48. निम्नलिखित में से कौन सा परिदृश्य डिवाइस ड्राइवर में विलंबित निष्पादन से सबसे अधिक लाभान्वित होता है?
- (A) हार्डवेयर डिवाइस को एक जटिल, बहु-चरणीय कमांड जिसमें काफी समय लगता है
(B) बफर में एक त्वरित, सरल लेखन ऑपरेशन
(C) एक छोटे डेटा ट्रांसफर के लिए किसी रुकावट पर तत्काल प्रतिक्रिया
(D) सिस्टम स्टार्टअप पर डिवाइस को प्रारंभ करना

Which one of the following scenarios benefits most from deferred execution in a device driver?

- (A) A complex, multi-step command to a hardware device that takes significant time
(B) A quick, simple write operation to a buffer
(C) An immediate response to an interrupt for a small data transfer
(D) Initializing a device upon system startup

- Q49.** लिनक्स कर्नेल मॉड्यूल (एलकेएम) या ड्राइवरों को डीबग करने के लिए निम्नलिखित में से कौन सी विधि सामान्य है?
- (A) printk() कथनों का उपयोग करना
 - (B) केजीडीबी या जीडीबी जैसे कर्नेल डीबगर का उपयोग करना
 - (C) एसट्रेस, आईट्रेस, या एफट्रेस जैसे सिस्टम टूल्स का उपयोग करना
 - (D) सभी विकल्प

Which of the following is a common method for debugging Linux Kernel Modules (LKMs) or drivers?

- (A) Using printk() statements
- (B) Using a kernel debugger like KGDB or GDB
- (C) Using system tools such as strace, ltrace, or ftrace
- (D) All of the options

- Q50.** एम्बेडेड सी में वोलाटाइल कीवर्ड सामान्य प्रयोजन इनपुट/आउटपुट (जीपीआईओ) ड्राइवरों के लिए महत्वपूर्ण है क्योंकि यह:
- (A) किसी वैरिएबल को फ्लैश मेमोरी में स्टोर करने की अनुमति देता है
 - (B) कंपाइलर को उन वैरिएबल तक पहुँच को अनुकूलित करने से रोकता है जिन्हें हार्डवेयर द्वारा बदला जा सकता है
 - (C) निर्दिष्ट करता है कि किसी वैरिएबल को केवल रीड किया जा सकता है, राइट नहीं किया जा सकता
 - (D) री-एंट्रेंट कोड से बचाव के लिए वैरिएबल को एक फ़ंक्शन पॉइंटर के रूप में परिभाषित करता है

The volatile keyword in Embedded C is crucial for General Purpose Input/Output (GPIO) drivers because it:

- (A) Allows a variable to be stored in a flash memory
- (B) Prevents the compiler from optimizing access to variables that can be changed by hardware
- (C) Specifies that a variable can only be read, not written
- (D) Defines the variable as a function pointer to guard against re-entrant code

- Q51.** एसपीआई और आइ2सी के बारे में निम्नलिखित में से कौन सा कथन गलत है?
- (A) आइ2सी, एसडीए (सीरियल डेटा) और एससीएल (सीरियल क्लॉक) का उपयोग करता है
 - (B) एसपीआई, एमओएसआई (मास्टर आउट, स्लेव इन), एमआईएसओ (मास्टर इन, स्लेव आउट), एससीएलके (सीरियल क्लॉक), और एक चिप सेलेक्ट (सीएस) या एसएस (स्लेव सेलेक्ट) लाइन का उपयोग करता है
 - (C) एसपीआई, आइ2सी की तुलना में आमतौर पर उच्च क्लॉक आवृत्तियों का समर्थन करता है
 - (D) एसपीआई और आइ2सी दोनों ही मल्टी-मास्टर और मल्टी-स्लेव कॉन्फिगरेशन की अनुमति देते हैं क्योंकि प्रत्येक डिवाइस स्वतंत्र रूप से एड्रेसेबल होता है

Which one of the following is incorrect statement about SPI and I2C?

- (A) I2C uses SDA (Serial Data) and SCL (Serial Clock)
- (B) SPI uses MOSI (Master Out, Slave In), MISO (Master In, Slave Out), SCLK (Serial Clock), and a Chip Select (CS) or SS (Slave Select) line
- (C) SPI generally supports higher clock frequencies compared to I2C
- (D) SPI and I2C both allow multi-master and multi-slave configurations because each device is independently addressable

- Q52. निम्नलिखित में से कौन सा प्रॉक फ़ाइल सिस्टम के उद्देश्य का सबसे अच्छा वर्णन करता है?**
- (A) प्रॉक फ़ाइल सिस्टम कर्नेल कॉन्फ़िगरेशन फ़ाइलों के लिए एक फिजिकल स्टोरेज स्थान प्रदान करता है
 - (B) प्रॉक फ़ाइल सिस्टम कर्नेल डेटा संरचनाओं और चल रही प्रक्रियाओं के लिए एक इंटरफ़ेस प्रदान करता है
 - (C) प्रॉक फ़ाइल सिस्टम निष्पादन योग्य प्रोग्राम और लाइब्रेरीज़ स्टोर करता है
 - (D) प्रॉक फ़ाइल सिस्टम ऑपरेटिंग सिस्टम के बूट अनुक्रम का प्रबंधन करता है

Which of the following best describes the purpose of the proc file system?

- (A) proc file system provides a physical storage location for kernel configuration files
- (B) proc file system offers an interface to kernel data structures and running processes
- (C) proc file system stores executable programs and libraries
- (D) proc file system manages the boot sequence of the operating system

- Q53. कौन सा एआरएम निर्देश मेमोरी से एक शब्द को रजिस्टर R0 में लोड करता है?**

Which ARM instruction loads a word from memory into register R0?

- (A) MOV R0, #1
- (B) LDR R0, [R1]
- (C) STR R0, [R1]
- (D) ADD R0, R1, R2

- Q54. एम्बेडेड सी में किस प्रकार के चर स्वचालित रूप से शून्य पर आरंभ हो जाते हैं?**

- (A) लोकल
- (B) रजिस्टर
- (C) स्टैटिक
- (D) स्वचालित

Which type of variables are automatically initialized to zero in Embedded C?

- (A) Local
- (B) Register
- (C) Static
- (D) Automatic

- Q55. माइक्रोकंट्रोलर में प्रोग्राम स्टोरेज के लिए फ्लैश मेमोरी का उपयोग क्यों किया जाता है?**

- (A) यह वोलाटाइल होती है
- (B) इसकी लेखन क्षमता कम होती है
- (C) यह बिना बिजली के डेटा को सुरक्षित रखती है
- (D) यह रैम से तेज़ होती है

Why is flash memory used for program storage in microcontrollers?

- (A) It is volatile
- (B) It has low write endurance
- (C) It retains data without power
- (D) It is faster than RAM

Q56. निम्नलिखित में से कौन सिस्टम ऑन ए चिप (एसओसी) को परिभाषित करता है?

- (A) उच्च प्रोसेसिंग पावर और मेमोरी वाला एक एकल सीपीयू
- (B) चिप पर डेटा और निर्देशों को प्रबंधित करने के लिए डिज़ाइन किया गया एक हार्डवेयर घटक
- (C) सीपीयू, मेमोरी और बाह्य उपकरणों जैसे कई घटकों को मिलाकर एक एकल एकीकृत चिप
- (D) हार्डवेयर को नियंत्रित करने के लिए उपयोग किया जाने वाला एक प्रकार का सॉफ्टवेयर

Which of the following defines a System on a Chip (SoC)?

- (A) A single CPU with high processing power and memory
- (B) A hardware component designed to manage data and instructions on a chip
- (C) A single integrated chip combining multiple components like a CPU, memory, and peripherals
- (D) A type of software used to control hardware

Q57. माइक्रोप्रोसेसर और एसओसी के बीच मुख्य अंतर क्या है?

- (A) माइक्रोप्रोसेसर हमेशा एसओसी से ज़्यादा शक्तिशाली होते हैं
- (B) एसओसी आमतौर पर माइक्रोप्रोसेसर से बड़े होते हैं
- (C) माइक्रोप्रोसेसर एकल-उद्देश्य वाले सीपीयू होते हैं, जबकि एसओसी एक ही चिप पर कई सिस्टम घटकों को एकीकृत करते हैं
- (D) एसओसी का उपयोग विशेष रूप से मोबाइल उपकरणों में किया जाता है, जबकि माइक्रोप्रोसेसर का उपयोग कंप्यूटिंग-गहन उपकरणों में किया जाता है

What is a key difference between a microprocessor and an SoC?

- (A) Microprocessors are always more powerful than SoCs
- (B) SoCs are typically larger than microprocessors
- (C) Microprocessors are single-purpose CPUs, whereas SoCs integrate multiple system components onto a single chip
- (D) SoCs are exclusively used in mobile devices, while microprocessors are used in compute intensive devices

Q58. रिस्क आर्किटेक्चर प्रायः सिस्क की तुलना में उच्च क्लॉक दर क्यों प्राप्त करते हैं?

- (A) कम निर्देश प्रकार पाइपलाइन को सरल बनाते हैं
- (B) बड़े इन्स्ट्रक्शन सेट कंट्रोल लॉजिक को कम करते हैं
- (C) रिस्क में अधिक जटिल डिकोडिंग होती है
- (D) सिस्क पाइपलाइन छोटी होती है

Why do RISC architectures often achieve higher clock rates than CISC?

- (A) Fewer instruction types simplify the pipeline
- (B) Larger instruction sets reduce control logic
- (C) RISC has more complex decoding
- (D) CISC pipelines are shorter

Q59. पीडब्ल्यूएम सिग्नल उत्पन्न करने के लिए आप किस टाइमर मोड का उपयोग करेंगे?

- (A) कैप्चर मोड
- (B) कंपेयर मोड
- (C) ओवरफ्लो मोड
- (D) वॉचडॉग मोड

Which timer mode would you use to generate a PWM signal?

- (A) Capture mode
- (B) Compare mode
- (C) Overflow mode
- (D) Watchdog mode

Q60. सीरियल पेरिफेरल इंटरफ़ेस (एसपीआई) एक _____ प्रकार का सीरियल कम्युनिकेशन इंटरफ़ेस है।

- (A) सिंक्रोनस
- (B) एसिंक्रोनस
- (C) डुप्लेक्स
- (D) संदेश कतार

Serial Peripheral Interface (SPI) is a _____ type of serial communication interface.

- (A) synchronous
- (B) asynchronous
- (C) duplex
- (D) message queue

Q61. एम्बेडेड सिस्टम में रियल-टाइम क्लॉक (आरटीसी) मॉड्यूल का मुख्य कार्य क्या है?

- (A) प्रोसेसर की गति को वास्तविक समय में मापना
- (B) वास्तविक समय में बिजली की खपत का प्रबंधन करना
- (C) सिस्टम बंद होने पर भी सटीक समय और दिनांक रखना
- (D) एलइडी और एलसीडी के संचालन को नियंत्रित करना

What is the main function of a Real-Time Clock (RTC) module in an embedded system?

- (A) To measure the speed of the processor in real time
- (B) To manage power consumption in real time
- (C) To keep accurate time and date, even when the system is powered off
- (D) To control the operation of LEDs and LCDs

Q62. एक माइक्रोकंट्रोलर को हाई-स्पीड डिस्प्ले से जोड़ने के लिए कौन सी बस सबसे उपयुक्त है?

For connecting a microcontroller to a high-speed display, which bus is most suitable?

- (A) I²C
- (B) SPI
- (C) UART
- (D) CAN

4_HPC System software development

- Q63. जब प्रोसेसर आर्म अवस्था में कार्य कर रहा होता है, तो सभी इन्स्ट्रक्शन _____ वाइड होते हैं।
- (A) 8-बिट (B) 16-बिट
(C) 32-बिट (D) 64-बिट

When the processor is executing in ARM state, then all instructions are _____ wide.

- (A) 8-bits (B) 16-bits
(C) 32-bits (D) 64-bits

- Q64. निम्नलिखित में से कौन सा भारत में विकसित प्रोसेसर रिस्क-वी आर्किटेक्चर पर आधारित है?
- (A) आर्म प्रोसेसर (B) वेगा प्रोसेसर
(C) शक्ति प्रोसेसर (D) एक्स86 प्रोसेसर

Which of the following Indian-developed processors is based on the RISC-V architecture?

- (A) ARM processors (B) VEGA processors
(C) Shakti processors (D) x86 processors

- Q65. टीसीपी/आईपी मॉडल को ओएसआई रिफरेंस मॉडल _____ विकसित किया गया था।
- (A) से पहले (B) के बाद
(C) के साथ-साथ (D) बिना किसी लिंक के

TCP/IP model was developed _____ the OSI reference model.

- (A) prior to (B) after
(C) simultaneous to (D) with no link to

- Q66. सामान्य नेटवर्क मॉडल में प्रक्रिया से प्रक्रिया वितरण के लिए कौन सी लेयर जिम्मेदार होती है?
- (A) नेटवर्क लेयर (B) परिवहन लेयर
(C) अनुप्रयोग लेयर (D) डेटा लिंक लेयर

Which layer is responsible for process to process delivery in a general network model?

- (A) network layer (B) transport layer
(C) application layer (D) data link layer

- Q67. निम्नलिखित में से किस रूटिंग एल्गोरिदम का उपयोग नेटवर्क लेयर डिज़ाइन के लिए किया जा सकता है?
- (A) ओपन शॉर्टेस्ट पाथ फर्स्ट (B) डिस्टेंस वेक्टर रूटिंग
(C) लिंक स्टेट रूटिंग (D) सभी विकल्प

Which one of the following routing algorithms can be used for network layer design?

- (A) Open Shortest Path First (B) Distance Vector Routing
(C) Link State Routing (D) All of the options

Q68. यदि समान डेटा दर के n सिग्नल सोर्स हैं, तो टीडीएम लिंक में _____ स्लॉट हैं।

If there are n signal sources of same data rate, then the TDM link has _____ slots.

- (A) n
- (B) $n/2$
- (C) $n*2$
- (D) 2^n

Q69. सॉकेट प्रोग्रामिंग में, क्लाइंट को यह जानना आवश्यक है:

- (A) सर्वर का आइपी एड्रेस
- (B) पोर्ट संख्या
- (C) सर्वर का आइपी एड्रेस और पोर्ट संख्या दोनों
- (D) सर्वर का आइपी एड्रेस, पोर्ट संख्या और एसएसएल प्रमाणपत्र

In socket programming, the client must know:

- (A) IP address of the server
- (B) Port number
- (C) Both IP address of the server and Port number
- (D) IP address of the server, Port number and SSL certificate

Q70. दो डिवाइस एक नेटवर्क में होते हैं यदि:

- (A) एक डिवाइस में एक प्रक्रिया दूसरे डिवाइस में एक प्रक्रिया के साथ सूचना का आदान-प्रदान करने में सक्षम है
- (B) एक प्रक्रिया दोनों डिवाइस पर चल रही है
- (C) विभिन्न डिवाइस पर चल रही प्रक्रियाओं के पीआईडी समान हैं
- (D) एक प्रक्रिया मास्टर है और हमेशा सक्रिय है और दूसरी स्लेव है

Two devices are in a network if:

- (A) A process in one device is able to exchange information with a process in another device
- (B) A process is running on both devices
- (C) PIDs of the processes running on different devices are same
- (D) A process is master and always active and another is slave

Q71. निम्नलिखित में से कौन सी विधि सर्वर द्वारा आने वाले क्लाइंट कनेक्शन की तैयारी के लिए उपयोग की जाती है?

Which one of the following methods is used by a server to prepare for incoming client connections?

- (A) connect()
- (B) accept()
- (C) bind()
- (D) listen()

4_HPC System software development

- Q72.** सॉकेट प्रोग्रामिंग में ट्रांसमिशन कंट्रोल प्रोटोकॉल (टीसीपी) और यूज़र डेटाग्राम प्रोटोकॉल (यूडीपी) के बीच मुख्य अंतर क्या है?
- (A) टीसीपी कनेक्शन-रहित है, जबकि यूडीपी कनेक्शन-उन्मुख है
 - (B) टीसीपी विश्वसनीय, व्यवस्थित डेटा वितरण प्रदान करता है, जबकि यूडीपी तेज़ है, लेकिन वितरण की गारंटी नहीं देता है
 - (C) टीसीपी का उपयोग सरल डेटा स्ट्रान्सफर के लिए किया जाता है, और यूडीपी का उपयोग जटिल अनुप्रयोगों के लिए किया जाता है
 - (D) यूडीपी के लिए हैंडशेक की आवश्यकता होती है, जबकि टीसीपी के लिए नहीं

What is the main difference between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in socket programming?

- (A) TCP is connection-less, while UDP is connection-oriented
- (B) TCP provides reliable, ordered data delivery, while UDP is faster but does not guarantee delivery
- (C) TCP is used for simple data transfer, and UDP is used for complex applications
- (D) UDP requires a handshake, while TCP does not

- Q73.** निम्नलिखित में से कौन सा एक प्राइवेट आईपी एड्रेस रेंज नहीं है?

- (A) 192.168.0.0 –से- 192.168.255.255
- (B) 10.0.0.0 –से- 10.255.255.255
- (C) 172.16.0.0 –से- 172.31.255.255
- (D) 169.254.0.0 –से- 169.254.255.255

Which one of the following is NOT a private IP address range?

- (A) 192.168.0.0 -to- 192.168.255.255
- (B) 10.0.0.0 -to- 10.255.255.255
- (C) 172.16.0.0 -to- 172.31.255.255
- (D) 169.254.0.0 -to- 169.254.255.255

- Q74.** आइपीवी6 एड्रेस का साइज बिट्स में कितना होता है?

- (A) 32 बिट्स
- (B) 64 बिट्स
- (C) 128 बिट्स
- (D) 256 बिट्स

What is the address size of an IPv6 address in bits?

- (A) 32 bits
- (B) 64 bits
- (C) 128 bits
- (D) 256 bits

- Q75.** निम्नलिखित में से किसका उपयोग यह निर्धारित करने के लिए किया जाता है कि कोई आइपी पैकेट हटाए जाने से पहले नेटवर्क में कितनी देर तक रह सकता है?

- (A) आइडेंटिफिकेशन
- (B) ऑप्शन्स
- (C) टाइप ऑफ सर्विस (टीओएस)
- (D) टाइम टू लिव (टीटीएल)

Which of the following is used to determine how long an IP packet can reside in the network before it is discarded?

- (A) Identification
- (B) Options
- (C) Type Of Service (TOS)
- (D) Time to Live (TTL)

4_HPC System software development

- Q76.** 7 बिट के सबनेट मास्क वाले आइपी 151.242.16.49 नेटवर्क के लिए कितने होस्ट (असाइन करने योग्य आइपी एड्रेस के साथ) और सबनेट संभव हैं?
- (A) 512 होस्ट और 126 सबनेट (B) 512 होस्ट और 128 सबनेट
(C) 126 सबनेट और 510 होस्ट (D) 128 सबनेट और 510 होस्ट

How many hosts (with assignable IP addresses) and subnets are possible for an IP 151.242.16.49 network with a subnet mask of 7 bits?

- (A) 512 hosts and 126 subnets (B) 512 hosts and 128 subnets
(C) 126 subnets and 510 hosts (D) 128 subnets and 510 hosts

- Q77.** एमपीआइ में, एमपीआइ बैरियर क्या हासिल करता है?
- (A) एक कम्प्यूनिकेटर के भीतर सभी प्रक्रियाओं को सिंक्रनाइज़ करता है
(B) रैंक 0 से सभी प्रक्रियाओं तक एक संदेश प्रसारित करता है
(C) सभी प्रक्रियाओं को समाप्त करता है
(D) संदेशों को एक ही प्रक्रिया में एकत्रित करता है

In MPI, what does MPI_Barrier achieve?

- (A) Synchronizes all processes within a communicator
(B) Broadcasts a message from rank 0 to all processes
(C) Terminates all processes
(D) Gathers messages into a single process

- Q78.** आप एक एमपीआइ प्रोग्राम की प्रोफ़ाइल बनाते हैं और देखते हैं कि अधिकांश समय एमपीआइ बैरियर में व्यतीत होता है। यह क्या दर्शाता है?
- (A) उच्च संचार-से-गणना अनुपात
(B) प्रक्रियाओं के बीच लोड असंतुलन
(C) छोटे संदेशों से नेटवर्क संकुलन
(D) अत्यधिक मेमोरी उपयोग

You profile an MPI program and see most time spent in MPI_Barrier. What does this indicate?

- (A) High communication-to-computation ratio
(B) Load imbalance between processes
(C) Network congestion from small messages
(D) Excessive memory usage

- Q79.** एमपीआइ में, MPI_Gather, MPI_Scatter से भिन्न है क्योंकि:
- (A) Gather, प्रक्रियाओं में डेटा वितरित करता है
(B) Gather, सभी प्रक्रियाओं से डेटा को एक में एकत्रित करता है
(C) Scatter को बैरियर सिंक्रोनाइज़ेशन की आवश्यकता होती है
(D) Scatter केवल रैंक 0 के साथ ही काम कर सकता है

In MPI, MPI_Gather differs from MPI_Scatter because:

- (A) Gather distributes data across processes
(B) Gather collects data from all processes into one
(C) Scatter requires barrier synchronization
(D) Scatter can only work with rank 0

4_HPC System software development

- Q80.** आप ओपेनएमपी के साथ एक समानांतर लूप का अनुकूलन कर रहे हैं जहाँ प्रत्येक आइटरेशन एक शेयर्ड हिस्टोग्राम एरे को अपडेट करती है। कौन सा दृष्टिकोण न्यूनतम प्रदर्शन हानि के साथ शुद्धता सुनिश्चित करता है?
- (A) प्रत्येक अपडेट के आसपास #pragma omp critical का उपयोग करें
 - (B) प्रत्येक हिस्टोग्राम प्रविष्टि के लिए एक अटॉमिक अपडेट का उपयोग करें
 - (C) हिस्टोग्राम को निजी बनाएँ और लूप के बाद मर्ज करें
 - (D) एरे को बिना सिंक्रनाइज़ेशन के शेयर्ड मेमोरी में रखें

You are optimizing a parallel loop with OpenMP where each iteration updates a shared histogram array. Which approach ensures correctness with minimal performance loss?

- (A) Use #pragma omp critical around each update
- (B) Use an atomic update for each histogram entry
- (C) Make the histogram private and merge after the loop
- (D) Place the array in shared memory without synchronization

- Q81.** आपको अत्यधिक अनियमित आइटरेशन टाइम वाले लूप को संसाधित करना होगा। कौन सी ओपेनएमपी शेड्यूलिंग रणनीति सबसे उपयुक्त है?

- (A) बड़े खंडों के साथ स्टैटिक शेड्यूलिंग
- (B) छोटे खंडों के साथ स्टैटिक शेड्यूलिंग
- (C) डायनामिक शेड्यूलिंग
- (D) घटते खंडों के आकार के साथ गाइडेड शेड्यूलिंग

You must process a loop with highly irregular iteration times. Which OpenMP scheduling strategy is most appropriate?

- (A) Static scheduling with large chunks
- (B) Static scheduling with small chunks
- (C) Dynamic scheduling
- (D) Guided scheduling with decreasing chunk sizes

- Q82.** आपको ओपेनएमपी प्रोग्राम में प्रत्येक थ्रेड को एक ही प्रारंभिक वैरिएबल वैल्यू से शुरू करना होगा, लेकिन फिर उसे स्वतंत्र रूप से संशोधित करना होगा। आप किस क्लॉज़ का उपयोग करते हैं?

- (A) शेयर्ड
- (B) प्राइवेट
- (C) फ़र्स्ट प्राइवेट
- (D) लास्ट प्राइवेट

You need each thread in an OpenMP program to start with the same initial variable value, but then modify it independently. Which clause do you use?

- (A) shared
- (B) private
- (C) firstprivate
- (D) lastprivate

Q83. पैरलल प्रोग्राम में बहुत अधिक थ्रेड्स का उपयोग करने का संभावित नुकसान क्या है?

- (A) अत्यधिक कांटेक्ट स्विचिंग ओवरहेड के कारण कम प्रदर्शन
- (B) अधिक समानांतरता के कारण बेहतर प्रदर्शन
- (C) बढ़ा हुआ मेमोरी उपयोग
- (D) कम डेटा लोकैलिटी

What is a potential drawback of using too many threads in a parallel program?

- (A) Reduced performance due to excessive context switching overhead
- (B) Increased performance due to more parallelism
- (C) Increased memory usage
- (D) Reduced data locality

Q84. जीपीयू प्रोग्रामिंग के लिए आमतौर पर किस प्रोग्रामिंग मॉडल का उपयोग किया जाता है?

- (A) ओपेनएमपी
- (B) सीयूडीए
- (C) एमपीआई
- (D) पोसिक्स थ्रेड्स

Which programming model is commonly used for programming GPUs?

- (A) OpenMP
- (B) CUDA
- (C) MPI
- (D) POSIX Threads

Q85. यदि आपको एएमडी और एनविडिया जीपीयू पर कार्य निष्पादित करने हैं, तो कौन सा एपीआई अधिक पोर्टेबल है?

- (A) सीयूडीए
- (B) ओपेनसीएल
- (C) वल्कन कम्प्यूट
- (D) डायरेक्टकम्प्यूट

If you must execute tasks on AMD and NVIDIA GPUs, which API is more portable?

- (A) CUDA
- (B) OpenCL
- (C) Vulkan Compute
- (D) DirectCompute

Q86. एक मल्टी-थ्रेडेड एप्लिकेशन कभी-कभी गलत परिणाम देता है। रेस स्थितियों का पता लगाने के लिए आप सबसे पहले कौन सा टूल इस्तेमाल करेंगे?

A multi-threaded application occasionally produces incorrect results. Which tool would you apply first to detect race conditions?

- (A) gprof
- (B) perf
- (C) strace
- (D) valgrind

Q87. कौन सी तकनीक समानांतर प्रोग्रामों में डेटा रेस का पता लगाने में मदद करती है?

- (A) मेमोरी कैशिंग
- (B) स्टैटिक एनालिसिस
- (C) डायनामिक इंस्ट्रूमेंटेशन टूल्स
- (D) वेक्टराइज़ेशन

Which technique helps detect data races in parallel programs?

- (A) Memory caching
- (B) Static analysis
- (C) Dynamic instrumentation tools
- (D) Vectorization

4 HPC System software development

- Q88.** समानांतर कंप्यूटिंग में एम्डल के नियम का उपयोग किसका अनुमान लगाने के लिए किया जाता है?
- (A) किसी प्रोग्राम को समानांतर करके प्राप्त की जा सकने वाली गति
 - (B) किसी दिए गए कार्य के लिए प्रोसेसरों की इष्टतम संख्या
 - (C) प्रोसेसरों के बीच संचार की विलंबता
 - (D) समानांतर प्रोग्राम के लिए आवश्यक मेमोरी की मात्रा

What is Amdahl's Law used to estimate in parallel computing?

- (A) The speedup achievable by parallelizing a program
- (B) The optimal number of processors for a given task
- (C) The latency of communication between processors
- (D) The amount of memory required for a parallel program

- Q89.** किस प्रकार का आइएसए आमतौर पर कम संख्या में सरल निर्देशों का उपयोग करता है?

- (A) सीआइएससी (कॉम्प्लेक्स इन्स्ट्रक्शन सेट कंप्यूटिंग)
- (B) आरआइएससी (रेड्यूसड इन्स्ट्रक्शन सेट कंप्यूटिंग)
- (C) इपीआईसी (एक्सप्लीसिटली पैरलल इन्स्ट्रक्शन कंप्यूटिंग)
- (D) कोई भी विकल्प नहीं

Which type of ISA typically uses a smaller number of simple instructions?

- (A) CISC (Complex Instruction Set Computing)
- (B) RISC (Reduced Instruction Set Computing)
- (C) EPIC (Explicitly Parallel Instruction Computing)
- (D) None of the options

- Q90.** एक रिस्क प्रोसेसर को अधिक जटिल एड्रेसिंग मोड शामिल करने के लिए संशोधित किया गया है। कौन सा ट्रेड-ऑफ सबसे अधिक संभावित है?

- (A) बढ़ी हुई पाइपलाइन थ्रूपुट
- (B) सरलीकृत डिकोडिंग लॉजिक
- (C) धीमी इन्स्ट्रक्शन डिकोडिंग
- (D) कम ट्रांजिस्टर संख्या

A RISC processor is modified to include more complex addressing modes. Which trade-off is most likely?

- (A) Increased pipeline throughput
- (B) Simplified decoding logic
- (C) Slower instruction decoding
- (D) Lower transistor count

- Q91.** आप हाई-फ्रिक्वेंसी ट्रेडिंग वर्कलोड के लिए एक सीपीयू डिज़ाइन कर रहे हैं। कौन सा कैश कॉन्फ़िगरेशन इष्टतम है?

- (A) उच्च संबद्धता वाला बड़ा L3 कैश
- (B) तेज़ हिट टाइम के लिए बहुत छोटा L1 कैश
- (C) कम लैटेंसी वाला बड़ा L2 कैश
- (D) उच्च संबद्धता और कम लैटेंसी वाला मध्यम L1

You are designing a CPU for high-frequency trading workloads. Which cache configuration is optimal?

- (A) Large L3 cache with high associativity
- (B) Very small L1 cache for faster hit time
- (C) Large L2 cache with low latency
- (D) Moderate L1 with high associativity and low latency

- Q92.** सॉफ्टवेयर अपडेट के बाद, वर्कलोड की एल1 हिट दर 90% से घटकर 70% हो जाती है। प्रोफाइलिंग स्ट्राइड-आधारित पहुँच में वृद्धि दर्शाती है। इसका सबसे संभावित कारण क्या है?
- (A) बढ़ी हुई टेम्पोरल लोकलिटी
 - (B) बढ़ी हुई स्थानिक लोकलिटी
 - (C) कम कैश एसोसिएटिविटी उपयोग
 - (D) बड़े स्ट्राइड के कारण कैश लाइन का कम उपयोग

A workload's L1 hit rate drops from 90% to 70% after a software update. Profiling shows increased stride-based accesses. Which is the most likely cause?

- (A) Increased temporal locality
- (B) Increased spatial locality
- (C) Reduced cache associativity usage
- (D) Poor cache line utilization due to large stride

- Q93.** अर्थमेटिक ऑपरेशनों के अतिरिक्त, एल्यू अन्य किस प्रकार के ऑपरेशन कर सकता है?
- (A) लॉजिकल ऑपरेशन
 - (B) फ्लोटिंग-पॉइंट ऑपरेशन
 - (C) मेमोरी एक्सेस ऑपरेशन
 - (D) इंटरप्ट हैंडलिंग

In addition to arithmetic operations, what other type of operations can an ALU perform?

- (A) Logical operations
- (B) Floating-point operations
- (C) Memory access operations
- (D) Interrupt handling

- Q94.** फ्लोटिंग पॉइंट में डॉट गुणनफल की गणना करते समय, कौन सी तकनीक राउंडिंग एरर को कम करती है?
- | | |
|-------------------------|------------------------------------|
| (A) रैंडमाइज्ड राउंडिंग | (B) काहन योग एल्गोरिदम |
| (C) इंटीजर का उपयोग | (D) डिसेबलिंग कंपाइलर ओप्टिमाइजेशन |

When computing dot products in floating point, which technique reduces rounding errors?

- | | |
|----------------------------|--------------------------------------|
| (A) Randomized rounding | (B) Kahan summation algorithm |
| (C) Using integers instead | (D) Disabling compiler optimizations |

- Q95.** आइइइ 754 मानक मुख्य रूप से किस प्रकार की संख्याओं के निरूपण और अंकगणित को परिभाषित करता है?
- | | |
|-------------------------|-----------------|
| (A) फिक्स्ड-पॉइंट नंबर | (B) इंटीजर |
| (C) फ्लोटिंग-पॉइंट नंबर | (D) बाइनरी नंबर |

The IEEE 754 standard primarily defines the representation and arithmetic of what kind of numbers?

- | | |
|----------------------------|--------------------|
| (A) Fixed-point numbers | (B) Integers |
| (C) Floating-point numbers | (D) Binary numbers |

4_HPC System software development

- Q96. एक पारंपरिक 5-चरणीय पाइपलाइन में, आप बिना रुके, रीड आफ्टर राइट खतरे का समाधान कैसे कर सकते हैं?
- (A) इन्स्ट्रक्शन रीऑर्डिंग (B) फॉरवर्डिंग/बाईपासिंग
(C) क्लॉक स्पीड बढ़ाकर (D) एनओपी निर्देशों का उपयोग करके

In a classic 5-stage pipeline, how can you resolve a Read After Write hazard without stalling?

- (A) Instruction reordering (B) Forwarding/bypassing
(C) Increasing clock speed (D) Using NOP instructions

- Q97. सीपीयू के ब्रांच प्रेडिक्टर की सटीकता दर 90% है। कंपाइलर को अपडेट करने के बाद, गलत प्रेडिक्शन के कारण परफॉर्मंस कम हो जाती है। ऐसा किस कंपाइलर बदलाव के कारण हुआ होगा?
- (A) लूप अनरोलिंग में वृद्धि
(B) फंक्शन इनलाइनिंग में वृद्धि
(C) कंडीशनल ब्रांचिंग जटिलता में वृद्धि
(D) इन्स्ट्रक्शन शेड्यूलिंग में कमी

CPU's branch predictor has a 90% accuracy rate. After updating the compiler, performance drops due to incorrect predictions. Which compiler change likely caused this?

- (A) Increased loop unrolling
(B) Increased function inlining
(C) Increased conditional branching complexity
(D) Reduced instruction scheduling

- Q98. सीयूडीए में, कौन-सा मेमोरी स्पेस एक ब्लॉक में सभी थ्रेड्स के बीच साझा किया जाता है और इसकी विलंबता कम होती है?
- (A) ग्लोबल मेमोरी (B) शेयर्ड मेमोरी
(C) कॉन्स्टेंट मेमोरी (D) लोकल मेमोरी

In CUDA, which memory space is shared among all threads in a block and has low latency?

- (A) Global memory (B) Shared memory
(C) Constant memory (D) Local memory

- Q99. मान लीजिए जीपीयू कर्नेल कम ऑक्यूपेंसी प्राप्त करता है। इसका सबसे संभावित कारण क्या है?
- (A) प्रति थ्रेड अत्यधिक रजिस्टर उपयोग
(B) प्रति ब्लॉक बहुत अधिक थ्रेड
(C) साझा मेमोरी का उपयोग
(D) उच्च मेमोरी कोलेसिंग

Suppose GPU kernel achieves low occupancy. Which factor is most likely the cause?

- (A) Excessive register usage per thread
(B) Too many threads per block
(C) Use of shared memory
(D) High memory coalescing

4_HPC System software development

Q100. _____ के साथ एक समय में केवल एक ही प्रक्रिया निष्पादित हो सकती है; इस बीच अन्य सभी प्रक्रियाएँ प्रोसेसर की प्रतीक्षा कर रही होती हैं। _____ के साथ एक से अधिक प्रक्रियाएँ एक साथ, प्रत्येक अलग-अलग प्रोसेसर पर चल सकती हैं।

- (A) मल्टीप्रोसेसिंग, मल्टीप्रोग्रामिंग
- (B) मल्टीप्रोग्रामिंग, यूनिप्रोसेसिंग
- (C) मल्टीप्रोग्रामिंग, मल्टीप्रोसेसिंग
- (D) यूनिप्रोग्रामिंग, मल्टीप्रोसेसिंग

With _____ only one process can execute at a time; meanwhile all other process are waiting for the processor. With _____ more than one process can be running simultaneously each on a different processor.

- (A) Multiprocessing, Multiprogramming
- (B) Multiprogramming, Uniprocessing
- (C) Multiprogramming, Multiprocessing
- (D) Uniprogramming, Multiprocessing

