High Performance Computing Objective Question

6CS005 HPC – S2023

Compiled By: Aatiz Ghimire

C- Programming for HPC

What does the acronym "C" stand for in C programming language?

a. Central

b. Common

c. Computer

d. Compiled

2. In C, what is the purpose of the "sizeof" operator?

a. Calculate the size of a file

b. Determine the size of a variable or data type

c. Find the size of an array

d. Measure the size of the code

Which keyword is used to define a constant in C?

a. const

b. constant

c. define

d. final

4. What is the syntax for a single-line comment in C?

a. // comment

b. /* comment */

c. # comment #

d. -- comment

How is memory allocated for a variable declared using the "malloc" function in C?

a. Automatically

b. Stack memory

c. Heap memory

d. Global memory

Which header file should be included to use the "printf" and "scanf" functions in C?

a. stdio.h

b. math.h

c. conio.h

d. string.h

7. What is the purpose of the "break" statement in C?

a. Exit the program

b. Terminate the loop or switch statement

c. Skip the current iteration in a loop

d. Return from a function

Which operator is used for pointer dereferencing in C?

a. *

b. &

c. ->

d. %

What is the output of the following code snippet?

printf("%d", 5 / 2);

a. 2.5

b. 2

c. 2.0

d. 2.25

How do you declare a function pointer in C?

a. int (*funcPtr)();

b. void pointerFunc();

c. function ptrFunc();

d. ptr functionPtr;

In C, what is the purpose of the "typedef" keyword?

a. Create a new variable

b. Define a new data type

c. Declare a function

d. Include a header file

What is the role of the "static" keyword in C?

a. Declare a global variable

b. Restrict access to a function

c. Limit the scope of a variable to the current file

d. Define a constant value

Which function is used to open a file in C?

a. openfile()

b. file_open()

c. fopen()

d. readfile()

How is a structure declared in C?

a. struct MyStruct {};

b. structure MyStruct {};

c. def MyStruct:

d. type MyStruct;

What is the purpose of the "union" keyword in C?

a. Combine two arrays

b. Merge two functions

c. Store different data types at the same memory location

d. Create a linked list

Which header file is used for dynamic memory allocation in C?

a. stdlib.h

b. memory.h

c. malloc.h

d. alloc.h

What is the default return type of a function in C if not specified?

a. int

b. void

c. char

d. float

Which loop in C is used to execute a block of code at least once?

a. for loop

b. do-while loop

c. while loop

d. repeat-until loop

How do you access the elements of an array in C?

a. Using pointers

b. Using indices

c. Using addresses

d. Using functions

What is the purpose of the "sizeof" operator when used with a data type?

a. Return the size of the variable

b. Determine the size of the data type

c. Calculate the sum of sizes in an array

d. Find the length of a string

What is the purpose of the "continue" statement in a loop in C?

a. Skip the rest of the code in the loop and move to the next iteration

b. Terminate the loop immediately

c. Jump to a specified label in the code

d. Return from the function

How is a constant string represented in C?

a. char* constantString = "Hello";

b. string constantString = "Hello";

c. const char* constantString = "Hello";

d. constantString = "Hello";

Which function is used to close a file in C?

a. fclose()

b. closefile()

c. file_close()

d. close()

What is the purpose of the "#define" directive in C?

a. Declare a constant

b. Define a new function

c. Include a header file

d. Declare a structure

How do you declare a two-dimensional array in C?

a. int[2][2] myArray;

b. int myArray[2, 2];

c. int myArray[2][2];

d. array myArray[2][2];

What is the output of the following code snippet?

c

Copy code

```
int x = 5;
printf("%d", x++);
```

a. 5

b. 6

c. 4

d. 10

In C, what is the purpose of the "typedef" keyword when used with structures?

a. Define a new data type

b. Create an alias for a data type

c. Declare a variable

d. Include a structure

What does the "NULL" pointer represent in C?

a. A pointer with no value

b. A pointer to the beginning of memory

c. A pointer with a value of 0

d. A pointer to the end of memory

How do you allocate memory for an array dynamically in C?

a. int arr = malloc(5 * sizeof(int));

b. int* arr = new int[5];

c. int* arr = malloc(5 * sizeof(int));

d. int arr[5];

What is the purpose of the "enum" keyword in C?

a. Define a new structure

b. Declare a constant

c. Create a new data type with named values

d. Implement exception handling

Which operator is used for bitwise XOR in C?

a. &

b. |

c. ^

d. ~

What is the function of the "strcat" in C?

a. Copy strings

b. Compare strings

c. Concatenate strings

d. Find the length of a string

In C, what is the purpose of the "volatile" keyword?

a. Declare a variable as read-only

b. Optimize the code for speed

c. Indicate that a variable's value can be changed by external factors

d. Prevent a variable from being modified

Which standard library function is used to generate random numbers in C?

a. random()

b. rand()

c. generate_random()

d. randomize()

How do you pass an array to a function in C?

a. Pass the array name

b. Pass the array size

c. Pass the array elements one by one

d. Pass the array address and size

What is the purpose of the "strtok" function in C?

a. Compare two strings

b. Tokenize a string

c. Concatenate two strings

d. Copy a string

Which header file should be included for mathematical functions in C?

a. mathlib.h

b. calculations.h

c. math.h

d. numbers.h

What is the purpose of the "const" keyword when used with a function parameter in C?

a. Indicate a constant function

b. Declare the function as void

c. Specify that the parameter cannot be modified within the function

d. Enable recursion in the function

How is the "do-while" loop different from the "while" loop in C?

a. "do-while" always executes at least once

b. "while" always executes at least once

c. "do-while" is not a loop structure in C

d. There is no difference between them

What is the purpose of the "#ifdef" preprocessor directive in C?

a. Include a header file

b. Conditionally include code if a macro is defined

c. Define a new macro

d. Declare a variable

What is the purpose of the "strncpy" function in C?

a. Concatenate two strings

b. Copy a substring from one string to another

c. Compare two strings

d. Reverse a string

In C, how do you declare a constant pointer?

a. const int *ptr;

b. int const *ptr;

c. int *const ptr;

d. const int *const ptr;

What is the significance of the "offsetof" macro in C?

a. Calculate the offset of a structure member

b. Determine the size of a structure

c. Find the memory address of a variable

d. Compute the offset of an array element

How is memory deallocated using the "free" function in C?

a. free(ptr);

b. delete ptr;

c. release(ptr);

d. dealloc(ptr);

What is the purpose of the "feof" function in C?

a. Check if the end-of-file indicator is set

b. Flush the file buffer

c. Retrieve the file size

d. Read the end-of-file character

Which function is used to convert a string to an integer in C?

a. atoi

b. itoa

c. strint

d. intstr

In C, what is the purpose of the "volatile" keyword when used with a variable?

a. Make the variable constant

b. Indicate that the variable cannot be changed

c. Optimize the variable for speed

d. Specify that the variable's value may change at any time

How do you declare a structure pointer in C?

a. struct MyStruct *ptr;

b. MyStruct pointer;

c. pointer struct MyStruct;

d. ptr *MyStruct;

What is the output of the following code snippet?

int arr[] = {1, 2, 3, 4, 5};

printf("%d", arr[3]);

a. 1

b. 2

c. 4

d. 5

Which header file should be included for handling date and time functions in C?

a. datetime.h

b. time.h

c. date.h

d. clock.h

What is the purpose of the "exit" function in C?

a. Close a file

b. Terminate the program and return a status code

c. Stop the execution of a loop

d. Return from a function

How do you define a macro in C?

a. #define MACRO_NAME { /* code here / }

b. define MACRO_NAME { / code here / }

c. #define MACRO_NAME / code here /

d. define MACRO_NAME / code here */

In C, what is the purpose of the "union" when used inside a structure?

a. Define a nested structure

b. Create a linked list

c. Save memory by allowing multiple data types in the same space

d. Declare a constant value

What is the purpose of the "offsetof" macro in C?

a. Calculate the size of a data type

b. Find the offset of a member within a structure

c. Determine the address of a variable

d. Measure the length of a string

Which function is used to read a character from the standard input in C?

a. getchar

b. readchar

c. scanchar

d. getc

How do you declare a global variable in C?

a. global int x;

b. int x;

c. extern int x;

d. public int x;

What is the purpose of the "getch" function in C?

a. Display a character on the screen

b. Read a character from the standard input

c. Get a character without waiting for the Enter key

d. Remove a character from the standard input buffer

Which keyword is used to define an enumeration in C?

a. enum

b. define

c. enumeration

d. typedef

What is the purpose of the "pow" function in C?

a. Find the nearest power of two

b. Raise a number to a specified power

c. Calculate the square root of a number

d. Return the next higher integer value

How do you include comments that span multiple lines in C?

a. // Comment

b. /* Comment */

c. # Comment #

d. - – Comment—

What is a function pointer and how is it used in C?

a. A pointer that holds the address of a function; used for dynamic function calls.

b. A pointer that points to a function's return value.

c. A pointer that holds the address of a variable within a function.

d. A pointer used exclusively for memory allocation functions.

In C, what is the purpose of the "inline" keyword when used with a function?

a. Forces the compiler to optimize the function for size.

b. Suggests the compiler to replace the function call with the actual code.

c. Specifies that the function should only be used in inline assembly.

d. Indicates that the function is to be excluded from the compilation.

What is the purpose of the "assert" macro in C?

a. Performs a bitwise AND operation.

b. Checks if a given condition is true; if not, it triggers an assertion failure.

c. Evaluates a mathematical expression.

d. Converts a variable to an assertion.

How is a linked list different from an array in C?

a. Linked lists have a fixed size, while arrays can dynamically grow.

b. Linked lists have constant-time access, while arrays have O(1) access time.

c. Linked lists use contiguous memory, while arrays use pointers.

d. Linked lists allow for dynamic memory allocation and insertion.

What is the purpose of the "memcpy" function in C?

a. Concatenates two strings.

b. Compares two strings.

c. Copies a specified number of bytes from one memory location to another.

d. Allocates memory for a string.

How is a structure different from a union in C?

a. Structures allow for storing multiple data types in the same memory location.

b. Unions allow for storing multiple data types in the same memory location.

c. Structures allocate memory based on the largest member, while unions allocate memory based on the total size.

d. Structures and unions are functionally identical.

What is the purpose of the "typedef" keyword when defining a structure in C?

a. It creates a new name for an existing data type.

b. It specifies the type of data a structure will hold.

c. It allocates memory for the structure.

d. It declares a new structure.

How can you dynamically allocate a two-dimensional array in C?

a. Using the "malloc" function.

b. Using the "calloc" function.

c. Using a combination of "malloc" and pointer arithmetic.

d. Only static allocation is possible for a two-dimensional array.

What is the purpose of the "fgets" function in C?

a. Reads a formatted string from a file.

b. Reads a character from the standard input.

c. Reads a line of text from a file, including newline characters.

d. Reads a line of text from a file, excluding newline characters.

How is a binary search different from a linear search in C?

a. Binary search works only on sorted arrays and has a higher time complexity.

b. Linear search works only on sorted arrays and has a lower time complexity.

c. Binary search works on sorted arrays and has a lower time complexity.

d. Linear search works on unsorted arrays and has a higher time complexity.

What is the purpose of the "volatile" keyword when used with a variable in C?

a. Indicates that the variable cannot be modified.

b. Prevents the variable from being accessed by other functions.

c. Specifies that the variable's value may change at any time without any action being taken by the code.

d. Optimizes the variable for speed.

What does the "const" qualifier in a function declaration indicate in C?

a. The function cannot be modified.

b. The function cannot modify any global variables.

c. The function does not return a value.

d. The function cannot have any local variables.

How can you pass a variable number of arguments to a function in C?

a. Using the "varargs" keyword.

b. By specifying a default value for the arguments.

c. Using the "stdarg.h" header and ellipsis (...) in the function declaration.

d. It is not possible to pass a variable number of arguments in C.

What is the purpose of the "strchr" function in C?

a. Finds the length of a string.

b. Locates the first occurrence of a character in a string.

c. Compares two strings.

d. Concatenates two strings.

How is a function defined as "static" different from a regular function in C?

a. A static function is accessible only within the file it is defined in.

b. A static function cannot be called from other files.

c. A static function has a fixed execution time.

d. A static function cannot be called recursively.

What is the purpose of the "setjmp" and "longjmp" functions in C?

a. Allocate and deallocate memory.

b. Jump to a specified label within the code.

c. Implement exception handling.

d. Save and restore the program's state.

How does the "qsort" function work in C?

a. Sorts an array using quicksort algorithm.

b. Sorts an array using bubblesort algorithm.

c. Sorts an array using mergesort algorithm.

d. Sorts an array using insertion sort algorithm.

What is the purpose of the "assert" function in C?

a. Prints a message to the console.

b. Exits the program immediately.

c. Checks if a given condition is true; if not, it triggers an assertion failure.

d. Returns a value indicating success or failure.

What is the role of the "const" keyword when used with a pointer in C?

a. It makes the pointer constant, preventing it from pointing to a different memory location.

b. It ensures that the memory the pointer points to cannot be modified.

c. It is used to declare a constant variable.

d. It is not possible to use "const" with a pointer.

How do you implement a shallow copy of a structure in C?

a. By using the "memcpy" function to copy the structure's memory.

b. By assigning the structure variable to another variable.

c. By manually copying each member of the structure.

d. It is not possible to implement a shallow copy of a structure.

Serial and Parallel Computing

What is the primary characteristic of serial computing?

i. Parallel execution

ii. Single execution path

iii. Distributed processing

iv. Dynamic task allocation

Define parallel computing.

i. The execution of tasks sequentially

ii. Simultaneous execution of tasks

iii. Serial processing of multiple tasks

iv. A form of multitasking

Explain the difference between concurrency and parallelism.

i. Concurrency is the same as parallelism.

ii. Concurrency implies sequential execution.

iii. Parallelism involves independent execution.

iv. Parallelism implies dependent execution.

What is task decomposition in the context of parallel computing?

i. Combining tasks into a single unit

ii. Breaking a task into smaller, independent subtasks

iii. Rearranging task order

iv. Executing tasks in parallel only

Name two types of parallel computer architectures.

i. Centralized and decentralized

ii. Shared-memory and distributed-memory

iii. Serial and parallel

iv. Sequential and non-sequential

What is message passing in parallel computing?

i. Passing messages through email

ii. Communicating between parallel processes

iii. Exchanging physical messages

iv. Sharing information through a common database

Name a language specifically designed for parallel programming.

i. C++

ii. Java

iii. Python

iv. MPI (Message Passing Interface)

Describe the difference between SIMD and MIMD processing models.

i. Single Instruction, Multiple Data; Multiple Instruction, Multiple Data

ii. Sequential Instruction, Single Data; Multiple Instruction, Single Data

iii. Symmetric Instruction, Multiple Data; Multiple Instruction, Single Data

iv. Single Instruction, Multiple Data; Multiple Instruction, Single Data

What is distributed computing?

i. The centralized execution of tasks

ii. Parallel execution on a single machine

iii. Simultaneous execution across multiple machines

iv. Serial execution on multiple machines

Name a sorting algorithm suitable for parallelization.

i. Bubble Sort

ii. Merge Sort

iii. Insertion Sort

iv. Quick Sort

What are some common challenges in designing parallel algorithms?

i. Load balancing, scalability, and task synchronization

ii. Serial execution, redundancy, and task decomposition

iii. Sequential execution, efficiency, and task granularity

iv. Message passing, fault tolerance, and task ordering

Define scalability in the context of parallel computing.

i. The ability to handle multiple tasks simultaneously

ii. The efficiency of a parallel algorithm

iii. The capability to increase performance as resources grow

iv. The adaptability of a program to different platforms

Why is task synchronization important in parallel computing?

i. To increase task granularity

ii. To avoid parallel execution

iii. To coordinate the execution of parallel tasks

iv. To slow down parallel processes

What is a supercomputer?

i. A highly efficient personal computer

ii. A computer with multiple GPUs

iii. A computer that performs at or near the currently highest operational rate

iv. A server with high storage capacity

Why is parallel I/O important in parallel computing?

i. To speed up the input phase of a program

ii. To handle multiple input devices

iii. To facilitate communication between parallel tasks

iv. To improve the output performance of a program

Define task granularity in the context of parallel computing.

i. The size of each individual task

ii. The order of task execution

iii. The speed of parallel tasks

iv. The priority of parallel tasks

Why is fault tolerance important in parallel and distributed computing?

i. To increase task granularity

ii. To coordinate parallel tasks

iii. To ensure the system remains operational despite failures

iv. To optimize the execution of parallel tasks

What is hybrid parallelism?

i. Combining parallel and serial execution in a program

ii. Exclusively using a shared-memory architecture

iii. Avoiding parallel execution altogether

iv. Implementing parallelism without communication

Briefly explain the concept of quantum computing.

i. A type of classical computing

ii. A form of parallel computing using quantum bits

iii. A process of serial computation in a quantum environment

iv. An algorithmic approach to machine learning

What are some potential advantages of quantum computing over classical computing?

i. Increased speed in solving certain problems

ii. Reduced energy consumption

iii. Enhanced graphics rendering

iv. Improved serial processing efficiency

What is a common model for parallel computing that involves dividing tasks into subtasks and solving them concurrently?

i. Divided-Conquer Model

ii. Fork-Join Model

iii. Divide-Combine Model

iv. Split-Merge Model

Which parallel sorting algorithm works by repeatedly dividing the data into subproblems and then merging the sorted results?

i. Quick Sort

ii. Merge Sort

iii. Bubble Sort

iv. Insertion Sort

What challenge arises when multiple parallel tasks require access to a shared resource simultaneously?

i. Task Granularity

ii. Load Balancing

iii. Race Condition

iv. Fault Tolerance

In a shared-memory parallel computing system, what type of memory is accessible to all processors?

i. Local Memory

ii. Global Memory

iii. Distributed Memory

iv. Cache Memory

What is the purpose of a parallel reduction operation in parallel computing?

i. To increase task granularity

ii. To minimize the number of parallel tasks

iii. To combine data from parallel tasks into a single result

iv. To synchronize parallel tasks

In parallel matrix multiplication, what approach involves dividing matrices into blocks and distributing the computation?

i. Row-wise Parallelism

ii. Column-wise Parallelism

iii. Block-wise Parallelism

iv. Element-wise Parallelism

How is efficiency calculated in the context of parallel computing?

i. Efficiency = Speedup / Number of Processors

ii. Efficiency = Speedup * Number of Processors

iii. Efficiency = 1 / Speedup

iv. Efficiency = Speedup - Number of Processors

What is the term for dividing a large task into smaller, independent subtasks for parallel execution?

i. Task Synchronization

ii. Task Granularity

iii. Task Decomposition

iv. Task Balancing

What is a Graphics Processing Unit (GPU) commonly used for in parallel computing?

i. Sequential Data Processing

ii. Floating-Point Arithmetic

iii. Serial Task Execution

iv. Disk I/O Operations

What does load balancing aim to achieve in parallel computing?

i. Unequal distribution of tasks

ii. Efficient allocation of tasks to processors

iii. Minimizing parallelism

iv. Dependency on a single processor

In parallel computing, what technique involves dividing a loop into smaller chunks and executing them concurrently?

i. Loop Balancing

ii. Loop Distribution

iii. Loop Unrolling

iv. Loop Parallelization

What is the term for arranging the execution order of parallel tasks to ensure correct results?

i. Task Sorting

ii. Task Ordering

iii. Task Synchronization

iv. Task Dependency

Which programming language is commonly used for parallel computing and supports message passing?

i. Java

ii. Python

iii. C++

iv. Fortran

In a distributed-memory parallel system, how are processors typically connected?

i. Shared Bus

ii. Star Topology

iii. Ring Topology

iv. Crossbar Switch

What architecture is commonly used in GPUs to support parallel processing on a large number of cores?

i. Centralized Architecture

ii. Vector Architecture

iii. Sequential Architecture

iv. Array Architecture

What is the term for a situation where the execution of one parallel task depends on the result of another?

i. Parallel Independence

ii. Task Interdependence

iii. Task Concurrency

iv. Task Isolation

What does asynchronous execution imply in the context of parallel computing?

i. Tasks execute in a synchronized manner

ii. Tasks can execute independently and without a fixed order

iii. Tasks execute with a constant time delay

iv. Tasks execute serially

In SIMD (Single Instruction, Multiple Data) execution, what happens to multiple data elements simultaneously?

i. Different instructions are executed on each data element

ii. The same instruction is executed on each data element

iii. Data elements are processed one at a time

iv. Data elements are processed in a random order

What does it mean for a parallel program to be thread-safe?

i. It can only be executed in a single thread

ii. It is resistant to deadlocks

iii. It can safely be executed by multiple threads without data corruption

iv. It requires explicit synchronization for every operation

When comparing serial and parallel performance, what does a speedup greater than 1 indicate?

i. Serial performance is better

ii. Parallel performance is better

iii. Equal performance in both cases

iv. Inefficiency in both cases

Pointers and Dynamic Memory Allocation

What is a pointer in C programming?

a. A variable that stores the memory address of another variable

b. A variable that stores a constant value

c. A variable that stores the result of a mathematical operation

d. A variable that stores a character value

How do you declare a pointer variable in C?

a. int ptr;

b. pointer int;

c. int *ptr;

d. ptr int;

What is the purpose of the "&" operator in C when used with pointers?

a. It represents the logical AND operation

b. It is used to declare a pointer

c. It is the address-of operator, used to get the address of a variable

d. It is used to access the value stored in a pointer

What does the malloc function do in C?

a. Allocates memory for a variable

b. Frees allocated memory

c. Copies the content of one variable to another

d. Initializes a variable with a default value

How do you free dynamically allocated memory in C?

a. delete

b. dealloc

c. free

d. release

What is the purpose of the sizeof operator in C?

a. Returns the size of a variable in bytes

b. Returns the value stored in a variable

c. Returns the address of a variable

d. Returns the type of a variable

What is the correct way to allocate an array of integers dynamically in C?

a. int arr[10];

b. int *arr = malloc(10 * sizeof(int));

c. int *arr[10];

d. malloc(arr, 10 * sizeof(int));

How do you access the value stored at a memory address pointed to by a pointer?

a. *ptr

b. ptr.value

c. &ptr

d. ptr->value

What is a NULL pointer in C?

a. A pointer with a value of 0

b. A pointer that points to the end of an array

c. A pointer that is not initialized

d. A pointer that points to a character string

What function is used to reallocate memory in C?

a. realloc

b. resize

c. reassign

d. resizealloc

What is the purpose of the calloc function in C?

a. Allocates memory for a variable

b. Allocates memory and initializes it to zero

c. Deallocates memory

d. Copies the content of one variable to another

How do you check if a dynamically allocated memory allocation was successful?

a. Use the success keyword

b. Check if the pointer is not NULL

c. Use the malloc_success function

d. Check if the pointer is NULL

What is the role of the free function in C?

a. Allocates memory

b. Releases dynamically allocated memory

c. Initializes memory to zero

d. Copies memory content

In C, what is a memory leak?

a. A type of error that causes the program to crash

b. Allocating memory without using malloc

c. Failing to deallocate memory after its use

d. A pointer that points to NULL

How do you declare a constant pointer in C?

a. const *ptr;

b. int const *ptr;

c. *const ptr;

d. const int *ptr;

What is the purpose of the memcpy function in C?

a. Allocates memory

b. Copies a block of memory from one location to another

c. Deallocates memory

d. Initializes memory to a specific value

What is the difference between malloc and calloc in C?

a. There is no difference; they are synonyms

b. malloc allocates memory and initializes it to zero, while calloc does not initialize

c. calloc allocates memory and initializes it to zero, while malloc does not initialize

d. calloc is used for arrays, and malloc is used for single variables

How do you iterate through an array using pointer arithmetic in C?

a. Use the for loop with an index variable

b. Use the while loop with a counter

c. Use pointer arithmetic with a loop

d. Use the do-while loop with a pointer variable

What is the purpose of the realloc function in C?

a. Allocates memory for a variable

b. Releases dynamically allocated memory

c. Resizes dynamically allocated memory

d. Copies the content of one variable to another

What does the expression ptr++ do in C?

a. Increments the value stored in the pointer

b. Moves the pointer to the next memory location

c. Decrements the value stored in the pointer

d. Multiplies the value stored in the pointer by 2

What is the purpose of the void pointer in C?

a. It is used to declare a pointer with no specific data type

b. It points to NULL by default

c. It is used to dereference a pointer

d. It is a placeholder for uninitialized pointers

How do you find the address of the first element in an array using a pointer?

a. &array

b. array[0]

c. *array

d. array

What is the difference between malloc and calloc regarding memory initialization?

a. malloc initializes memory to zero, while calloc does not initialize

b. calloc initializes memory to zero, while malloc leaves it uninitialized

c. Both malloc and calloc initialize memory to zero

d. Neither malloc nor calloc initializes memory

What is a double pointer in C?

a. A pointer that points to a variable of type double

b. A pointer that stores a double-precision floating-point number

c. A pointer that points to another pointer

d. A pointer that can hold two memory addresses

How do you declare a dynamic array of characters (string) in C?

a. char str[];

b. char *str;

c. char[] str;

d. char *str[];

What does the memset function do in C?

a. Allocates memory for a variable

b. Deallocates memory

c. Initializes a block of memory with a specific value

d. Copies a block of memory from one location to another

What is the purpose of the -> operator in C?

a. It is the address-of operator

b. It is used to declare a pointer

c. It is the structure member access operator for pointers

d. It is used to dereference a pointer

How do you create a pointer to a function in C?

a. function_ptr();

b. int (*function_ptr)();

c. *function_ptr;

d. function_ptr[];

What is the significance of the sizeof operator when working with pointers?

a. It returns the size of the pointer variable

b. It returns the size of the data type to which the pointer points

c. It returns the address of the pointer

d. It is not applicable to pointers

How do you swap the values of two variables using pointers in C?

a. swap(a, b);

b. swap(&a, &b);

c. *a = *b;

d. temp = *a; *a = *b; *b = temp;

In C, what is the purpose of the free function when working with arrays?

a. Deallocates the entire array

b. Frees up only the last element of the array

c. Deallocates the memory occupied by the array elements

d. Initializes the array elements to zero

What does the expression sizeof(int *) return in C?

a. The size of the integer data type

b. The size of the pointer to an integer

c. The size of the memory address

d. The size of the memory block allocated by malloc

How do you use the const keyword with pointers in C?

a. const int *ptr;

b. int *const ptr;

c. const *int ptr;

d. *const int ptr;

What is the purpose of the strncpy function in C?

a. Concatenates two strings

b. Compares two strings

c. Copies a specified number of characters from one string to another

d. Finds the length of a string

How do you declare a 2D array dynamically in C using pointers?

a. int **arr = malloc(rows * cols * sizeof(int));

b. int *arr[rows][cols];

c. int **arr = malloc(rows * sizeof(int *));

d. int *arr[rows];

What is the role of the realloc function when resizing a dynamically allocated array in C?

a. It deallocates the memory

b. It resizes the memory block without copying existing data

c. It resizes the memory block and initializes it to zero

d. It copies existing data to a new memory block of the requested size

How do you dereference a double pointer to access the value it points to?

a. **ptr

b. *ptr

c. &(*ptr)

d. &&ptr

What is the purpose of the strcat function in C?

a. Compares two strings

b. Copies a string to another string

c. Concatenates two strings

d. Finds the length of a string

When should you use the void pointer type in C?

a. To declare a pointer to a function

b. To declare a pointer with no specific data type

c. To declare a pointer to a structure

d. To declare a pointer to an integer

What is the difference between calloc and realloc in C?

a. calloc is used for dynamic memory allocation, while realloc is used for static memory allocation

b. calloc initializes memory to zero, while realloc resizes a previously allocated block of memory

c. calloc is used for arrays, while realloc is used for single variables

d. calloc is used for strings, while realloc is used for arrays

Multi-Programming and Threads

What is multitasking in the context of operating systems?

a) Running multiple processors simultaneously

b) Running multiple tasks concurrently on a single processor

c) Running a single task at a time

d) Running tasks in a sequential manner

What is the primary goal of multitasking in operating systems?

a) Maximizing CPU utilization

b) Minimizing CPU utilization

c) Minimizing memory usage

d) Maximizing disk space

What is a context switch?

a) Switching between different user accounts

b) Switching between different processes

c) Switching between different input devices

d) Switching between different network interfaces

Which of the following is a disadvantage of preemptive multitasking?

a) Improved responsiveness

b) Increased overhead due to frequent context switches

c) Reduced parallelism

d) Decreased system stability

What is the purpose of a process control block (PCB) in multitasking systems?

a) To store the contents of the CPU registers

b) To store information about a process's state

c) To manage input and output operations

d) To control the access to system resources

In a non-preemptive multitasking system, when does a context switch occur?

a) Only when a process voluntarily releases the CPU

b) Periodically at fixed time intervals

c) Whenever a higher-priority process becomes available

d) When a process exceeds its allotted time quantum

What is the role of the scheduler in a multitasking operating system?

a) Managing the allocation of system resources

b) Handling user input devices

c) Managing file systems

d) Controlling the execution of device drivers

Which scheduling algorithm is commonly used in preemptive multitasking systems?

a) First-Come-First-Serve (FCFS)

b) Shortest Job Next (SJN)

c) Round Robin

d) Priority Scheduling

What is the purpose of a ready queue in a multitasking system?

a) To store processes that are waiting for input/output operations

b) To store processes that are ready to execute but waiting for the CPU

c) To store processes that are blocked and cannot continue execution

d) To store completed processes

How does multitasking contribute to a responsive user interface in operating systems?

a) By reducing the overall system performance

b) By allowing multiple users to share the same resources

c) By quickly switching between different tasks

d) By increasing the system's memory footprint

Flynn's Taxonomy classifies parallel processing systems based on the number of instruction streams and data streams. How many categories does Flynn's Taxonomy have?

a) 2

b) 3

c) 4

d) 5

In Flynn's Taxonomy, what does SISD stand for?

a) Single Instruction, Single Data

b) Single Instruction, Multiple Data

c) Multiple Instruction, Single Data

d) Multiple Instruction, Multiple Data

Which category of Flynn's Taxonomy represents a traditional, sequential computer?

a) SISD

b) SIMD

c) MISD

d) MIMD

What is the key characteristic of SIMD (Single Instruction, Multiple Data) systems?

a) Multiple processors execute multiple instructions on multiple data sets

b) Multiple processors execute the same instruction on multiple data sets

c) Single processor executes a single instruction on multiple data sets

d) Single processor executes multiple instructions on a single data set

Which type of parallelism is often associated with vector processors in Flynn's Taxonomy?

a) SISD

b) SIMD

c) MISD

d) MIMD

In Flynn's Taxonomy, what does MIMD stand for?

a) Multiple Instruction, Multiple Data

b) Multiple Instruction, Single Data

c) Multiple Instruction, Sequential Data

d) Multiple Instruction, Data Sequence

What is the main advantage of MISD (Multiple Instruction, Single Data) systems in Flynn's Taxonomy?

a) Improved parallelism

b) Enhanced fault tolerance

c) Higher clock speeds

d) Simplified programming models

Which category of Flynn's Taxonomy is commonly associated with parallel processing in modern multi-core processors?

a) SISD

b) SIMD

c) MISD

d) MIMD

In Flynn's Taxonomy, which category represents a system where multiple processors execute multiple instructions independently on different data sets?

a) SISD

b) SIMD

c) MISD

d) MIMD

Flynn's Taxonomy provides a classification for parallel architectures based on instruction and data streams. What aspect of parallelism does it not explicitly consider?

a) Instruction-level parallelism

b) Task-level parallelism

c) Data-level parallelism

d) Thread-level parallelism

Concurrent Programming

What is a thread?

a) A lightweight process

b) A heavy-duty process

c) A data structure

d) A file in the file system

What is the primary advantage of using threads in a program?

a) Improved modularity

b) Increased parallelism

c) Reduced memory usage

d) Enhanced security

In a multithreaded environment, what is a race condition?

a) A condition where threads compete to execute the same code

b) A condition where threads synchronize perfectly

c) A condition where threads deadlock

d) A condition where threads crash the system

What is thread safety in the context of multithreading?

a) Ensuring that only one thread can execute at a time

b) Ensuring that multiple threads can execute simultaneously

c) Ensuring that threads never execute

d) Ensuring that threads always execute in a specific order

Which of the following statements about user-level threads is true?

a) They are managed by the operating system kernel

b) They are faster than kernel-level threads

c) They are visible to the operating system as independent processes

d) They cannot be scheduled independently of the process they belong to

What is a critical section in the context of concurrent programming?

a) A section of code where race conditions are guaranteed

b) A section of code that needs to be executed atomically

c) A section of code that is never executed

d) A section of code that causes deadlocks

How can you prevent race conditions in a critical section?

a) By avoiding the use of locks

b) By using atomic operations or locks

c) By increasing the number of threads

d) By introducing intentional delays

What is a data race?

a) A race condition involving threads accessing shared data

b) A race between two threads in a running competition

c) A race involving data transmission over a network

d) A condition where data is lost during execution

What is the purpose of a mutex in concurrent programming?

a) To increase race conditions

b) To avoid deadlocks

c) To synchronize thread execution

d) To create intentional delays

In the context of multithreading, what is the meaning of the term "atomic"?

a) Indivisible and uninterruptible

b) Large and complex

c) Slow and inefficient

d) Easily divisible and interruptible

What is a deadlock in concurrent programming?

a) A situation where threads run too quickly

b) A situation where threads compete for resources

c) A situation where threads wait indefinitely for each other

d) A situation where threads terminate unexpectedly

What are the necessary conditions for a deadlock to occur?

a) Mutual exclusion, no preemption, circular wait, and hold and wait

b) Mutual exclusion, preemption, circular wait, and hold and wait

c) Mutual exclusion, no preemption, no circular wait, and hold and wait

d) Mutual exclusion, preemption, no circular wait, and hold and wait

What is the difference between a thread and a process?

a) Threads cannot be scheduled independently

b) Threads share the same address space and resources

c) Processes are always more lightweight than threads

d) Processes cannot run concurrently

What is thread synchronization?

a) Ensuring threads never run concurrently

b) Coordinating the execution of threads to avoid conflicts

c) Forcing threads to run concurrently

d) Ignoring the need for coordination between threads

What is the purpose of the join() method in threading?

a) To create a new thread

b) To terminate a thread

c) To wait for a thread to complete

d) To synchronize threads

What is the difference between parallelism and concurrency?

a) Parallelism involves multiple processes; concurrency involves multiple threads

b) Parallelism involves multiple threads; concurrency involves multiple processes

c) Parallelism and concurrency are synonymous

d) Parallelism is a hardware concept; concurrency is a software concept

What is a thread pool?

a) A collection of threads sharing the same priority

b) A group of threads that run in parallel

c) A limited set of threads that are reused for multiple tasks

d) A thread that manages other threads

What is the purpose of a semaphore in managing race conditions?

a) To encourage race conditions

b) To prevent race conditions

c) To increase the likelihood of deadlocks

d) To delay the execution of threads

What is the difference between a race condition and a deadlock?

a) A race condition involves competition for resources; a deadlock involves blocked threads

b) A race condition involves threads waiting indefinitely; a deadlock involves quick competition

c) A race condition involves circular wait; a deadlock involves mutual exclusion

d) A race condition involves preemption; a deadlock involves no preemption

How does the use of locks contribute to managing race conditions?

a) Locks guarantee the absence of race conditions

b) Locks introduce intentional race conditions

c) Locks ensure that multiple threads can access shared resources simultaneously

d) Locks prevent multiple threads from accessing shared resources simultaneously

What is the purpose of the "volatile" keyword in the context of multithreading?

a) To increase the visibility of variables across threads

b) To decrease the visibility of variables across threads

c) To prevent the use of locks

d) To force variables to be constant

What is a thread-safe data structure?

a) A data structure that only supports single-threaded access

b) A data structure that automatically locks itself

c) A data structure that can be accessed by multiple threads without causing data corruption

d) A data structure that intentionally causes race conditions

What is the difference between a deadlock and livelock?

a) A deadlock involves competing threads; a livelock involves threads that are blocked

b) A deadlock involves threads that are blocked; a livelock involves threads actively trying to resolve the situation

c) A deadlock is always fatal; a livelock is not necessarily fatal

d) A deadlock can be resolved automatically; a livelock requires manual intervention

What is resource preemption in the context of deadlock prevention?

a) Allowing resources to be held indefinitely

b) Forcing threads to release resources

c) Introducing intentional delays

d) Ignoring the hold and wait condition

In the context of deadlock detection, what is a wait-for graph used for?

a) To allocate resources dynamically

b) To represent the order in which threads are waiting

c) To detect cycles and potential deadlocks

d) To force threads to wait indefinitely

What is the role of the "timeout" strategy in deadlock prevention?

a) Allowing threads to wait indefinitely

b) Forcing threads to release resources after a certain period

c) Introducing intentional delays

d) Ignoring the hold and wait condition

How does the use of multiple resource instances contribute to deadlock prevention?

a) It increases the likelihood of deadlocks

b) It decreases the complexity of the system

c) It decreases the chance of circular wait

d) It does not affect deadlock prevention strategies

How can you prevent deadlocks in a system?

a) By allowing circular wait

b) By using a preemptive scheduling algorithm

c) By ignoring the hold and wait condition

d) By avoiding one or more of the necessary conditions

What is the Banker's algorithm used for in the context of deadlocks?

a) To allocate resources dynamically

b) To preemptively terminate threads

c) To detect deadlocks

d) To cause intentional deadlocks

What is the role of a resource allocation graph in deadlock detection?

a) To allocate resources to threads

b) To prevent circular wait

c) To detect and represent the state of resources and processes

d) To intentionally cause deadlocks for testing purposes

Mutex, Semaphores and others

What does the term "mutex" stand for?

A. Multiple Text

B. Mutual Exclusion

C. Multi-thread Execution

D. Memory Extension

What is the primary purpose of semaphores?

A. Memory Allocation

B. Process Synchronization

C. File I/O

D. Network Communication

What is a critical section in concurrent programming?

A. A segment of code where errors are allowed

B. A part of the program that is executed only once

C. A code segment that must be executed atomically by one thread at a time

D. A section where debugging is not allowed

What is a key difference between a mutex and a semaphore?

A. Mutex allows multiple threads to access the critical section simultaneously.

B. Semaphore is used only for binary synchronization.

C. Semaphore provides exclusive access to the critical section.

D. Mutex supports signaling operations.

What is a deadlock in multithreading?

A. A situation where two threads exchange data simultaneously.

B. A condition where a thread is terminated unexpectedly.

C. A state where two or more threads are blocked indefinitely.

D. A scenario where threads execute out of order.

In semaphore terminology, what does the "wait" operation do?

A. Increment the semaphore value.

B. Decrement the semaphore value.

C. Put the thread to sleep.

D. Wake up all waiting threads.

What distinguishes a counting semaphore from a binary semaphore?

A. Counting semaphores are used for file I/O.

B. Counting semaphores allow multiple threads to access the critical section.

C. Binary semaphores can have values greater than one.

D. Counting semaphores are not used in concurrent programming.

Why are atomic operations important in critical sections?

A. They make the code run faster.

B. They ensure that the operations are executed in a single, uninterruptible step.

C. They prevent the use of mutexes.

D. They allow for parallel execution without synchronization.

What is a race condition?

A. A competition between threads to finish execution.

B. A situation where two threads try to access shared data simultaneously, leading to unpredictable results.

C. A condition where threads race to acquire a semaphore.

D. An error that occurs when a thread finishes last.

What does the term "mutual exclusion" mean in the context of synchronization?

A. Threads must exclude each other from execution.

B. Threads must mutually agree on program execution.

C. Only one thread can execute a critical section at a time.

D. Threads must have exclusive access to shared memory

What is the downside of using busy waiting (spinning) in synchronization mechanisms?

A. It consumes less CPU resources.

B. It increases energy efficiency.

C. It can lead to high CPU usage without making progress.

D. It eliminates the need for locks.

What does it mean for a piece of code to be thread-safe?

A. The code can only be executed by one thread at a time.

B. The code can be executed by multiple threads simultaneously without causing issues.

C. The code is guaranteed to execute first in a multithreaded program.

D. The code is immune to race conditions.

CLI Passing in C

How are command-line arguments passed to a C program?

- A. Through a separate configuration file

- B. Via a function call

- C. Through the main function parameters

- D. Through global variables

In a C program, how can you access individual command-line arguments?

- A. Using the getopt function

- B. Through the argc and argv parameters of the main function

- C. By reading from a file named "command-line-args.txt"

- D. Using the args global array

What does the variable argc represent in the main function when dealing with command-line arguments?

- A. The total number of command-line arguments

- B. The number of options specified in the command line

- C. The index of the current command-line argument being processed

- D. The address of the first command-line argument

Which C library function is commonly used for parsing command-line options in a more structured manner?

- A. parse_options

- B. getopt

- C. read_options

- D. scan_args

How can a C program handle errors in the command-line arguments provided by the user?

- A. By using a try-catch block

- B. By directly terminating the program

- C. By displaying an error message and exiting gracefully

- D. By ignoring the invalid arguments and continuing execution

---

Parallel Programming

What is parallel computing?

a. The execution of multiple tasks sequentially

b. The execution of multiple tasks simultaneously

c. The execution of a single task on multiple processors

d. The execution of a single task on a single processor

Which of the following is a key advantage of parallel computing?

a. Increased sequential execution time

b. Decreased overall computational power

c. Improved performance and speedup

d. Limited scalability

What is Amdahl's Law used for in parallel computing?

a. Measuring parallel efficiency

b. Assessing sequential execution time

c. Calculating speedup

d. Analyzing memory hierarchy

What is the purpose of parallelism in computer architecture?

a. To slow down computational processes

b. To enhance memory capacity

c. To improve overall system performance

d. To reduce the number of processors needed

Which type of parallel computing architecture is based on the idea of dividing the input data into chunks and processing them simultaneously?

a. SIMD (Single Instruction, Multiple Data)

b. MIMD (Multiple Instruction, Multiple Data)

c. SPMD (Single Program, Multiple Data)

d. MISD (Multiple Instruction, Single Data)

What is a race condition in parallel computing?

a. A condition where multiple processes compete for resources

b. A condition where processes avoid each other

c. A condition where only one process executes at a time

d. A condition where processes run independently

What is the purpose of a barrier in parallel computing?

a. To synchronize processes and ensure they reach a certain point together

b. To slow down the execution of processes

c. To introduce a delay between processes

d. To stop the execution of processes

What does the term "load balancing" refer to in parallel computing?

a. Distributing computational tasks evenly across processors

b. Increasing the computational load on a single processor

c. Reducing the overall computational load

d. Eliminating parallelism in a system

Which parallel programming model allows for the explicit communication between processes?

a. Shared-memory model

b. Message-passing model

c. Data parallelism model

d. Task parallelism model

In parallel computing, what is meant by the term "scalability"?

a. The ability of a system to increase its performance with the addition of processors

b. The ability of a system to decrease its performance with the addition of processors

c. The ability of a system to execute only one task at a time

d. The ability of a system to reduce its memory requirements

What is the role of a parallel algorithm in parallel computing?

a. To slow down the execution of processes

b. To distribute tasks among processors

c. To eliminate parallelism

d. To increase sequential execution time

Which parallel computing architecture is commonly used in graphics processing units (GPUs)?

a. SIMD (Single Instruction, Multiple Data)

b. MIMD (Multiple Instruction, Multiple Data)

c. SPMD (Single Program, Multiple Data)

d. MISD (Multiple Instruction, Single Data)

What is the purpose of a critical section in parallel programming?

a. To execute a section of code atomically

b. To increase parallelism

c. To introduce race conditions

d. To slow down computation

What is the difference between synchronous and asynchronous parallel programming models?

a. Synchronous models use explicit communication, while asynchronous models do not.

b. Asynchronous models use explicit communication, while synchronous models do not.

c. Synchronous models require barriers, while asynchronous models do not.

d. Asynchronous models always have a higher speedup than synchronous models.

What is the goal of parallel task decomposition?

a. To increase sequential execution time

b. To divide a task into smaller, parallelizable subtasks

c. To avoid parallelism in a system

d. To synchronize processes at a certain point

Which metric is used to measure the efficiency of a parallel algorithm?

a. Speedup

b. Sequential execution time

c. Barrier count

d. Load imbalance

What is the purpose of a data dependency in parallel computing?

a. To slow down computation

b. To introduce race conditions

c. To synchronize processes

d. To determine the order of execution of tasks

What does Amdahl's Law suggest about the scalability of a parallel algorithm?

a. Scalability is only determined by the speed of the processors.

b. Scalability is limited by the sequential portion of the algorithm.

c. Scalability is not affected by the number of processors.

d. Scalability is inversely proportional to the number of processors.

Which parallel programming model allows multiple threads to share a common address space?

a. Shared-memory model

b. Message-passing model

c. Data parallelism model

d. Task parallelism model

What is the primary purpose of a parallel virtual machine (PVM) in parallel computing?

a. To simulate parallel algorithms

b. To manage virtual memory in parallel systems

c. To provide a common interface for parallel programming

d. To slow down the execution of processes

What is data parallelism in the context of parallel programming?

a. Executing multiple tasks independently on different processors

b. Executing a single task using multiple processors simultaneously

c. Executing tasks sequentially on a single processor

d. Executing tasks with explicit communication between processors

Which programming model is commonly associated with data parallelism?

a. Shared-memory model

b. Message-passing model

c. Task parallelism model

d. Data parallelism model

What is the purpose of a data parallel algorithm?

a. To introduce race conditions

b. To synchronize processes

c. To distribute data across processors for simultaneous processing

d. To eliminate parallelism

In data parallelism, what does the term "SIMD" stand for?

a. Single Instruction, Multiple Data

b. Single Instruction, Single Data

c. Multiple Instruction, Multiple Data

d. Multiple Instruction, Single Data

What is the role of a data parallel compiler in parallel programming?

a. To increase the sequential execution time

b. To eliminate parallelism

c. To automatically parallelize code for execution on multiple processors

d. To slow down computation

How is data parallelism different from task parallelism?

a. Data parallelism executes multiple tasks simultaneously, while task parallelism executes tasks sequentially.

b. Data parallelism executes a single task on multiple processors, while task parallelism executes multiple tasks independently.

c. Data parallelism and task parallelism are the same concepts.

d. Data parallelism uses explicit communication, while task parallelism does not.

What is the purpose of a parallel loop in data parallel programming?

a. To introduce race conditions

b. To synchronize processes

c. To parallelize the execution of a loop across multiple processors

d. To eliminate parallelism

Which parallel programming model allows for implicit parallelism in loops?

a. Shared-memory model

b. Message-passing model

c. Task parallelism model

d. Data parallelism model

What is a data parallelism "scatter" operation?

a. Distributing data across processors

b. Collecting data from processors

c. Synchronizing data access

d. Eliminating data parallelism

In data parallelism, what does the term "gather" refer to?

a. Distributing data across processors

b. Collecting data from processors

c. Synchronizing data access

d. Eliminating data parallelism

Memory Models:

What is the main advantage of the shared-memory model in parallel programming?

a. Explicit communication between processes

b. Simplicity in programming with shared variables

c. Automatic load balancing

d. Independent address spaces for each processor

What is the primary challenge of the shared-memory model?

a. Inefficient use of memory

b. Difficulty in managing explicit communication

c. Difficulty in maintaining data consistency

d. Limited scalability

In the context of shared memory, what is a "cache coherence" problem?

a. Inefficient use of memory caches

b. Difficulty in managing explicit communication

c. Inconsistent views of shared data in different processor caches

d. Limited scalability

Which memory consistency model ensures that the order of memory operations observed by each processor is consistent with the program's order?

a. Sequential consistency

b. Release consistency

c. Weak consistency

d. Causal consistency

What is the purpose of a memory barrier in parallel programming?

a. To eliminate parallelism

b. To synchronize processes

c. To increase sequential execution time

d. To manage explicit communication

What is the primary advantage of the message-passing model in parallel programming?

a. Simplicity in programming with shared variables

b. Efficient use of memory

c. Explicit communication between processes

d. Automatic load balancing

Which memory model is associated with the idea of each processor having its own local memory?

a. Shared-memory model

b. Message-passing model

c. Distributed-memory model

d. Hybrid-memory model

What is the purpose of a memory consistency model in parallel programming?

a. To eliminate parallelism

b. To synchronize processes

c. To define the order in which memory operations become visible to other processors

d. To increase sequential execution time

In a distributed-memory model, how do processors communicate?

a. Through a shared address space

b. Through explicit message passing

c. By synchronizing memory access

d. By using a single global memory

What does the term "NUMA" stand for in the context of parallel computing?

a. Non-Uniform Memory Access

b. Near Uniform Memory Allocation

c. Networked Universal Memory Architecture

d. Nested Unified Memory Access

What is the primary advantage of a multicore processor architecture?

a. Increased clock speed

b. Improved single-threaded performance

c. Enhanced parallel processing capabilities

d. Reduced power consumption

What does the term "SIMD" stand for in the context of parallel computing architecture?

a. Single Instruction, Multiple Data

b. Single Instruction, Single Data

c. Multiple Instruction, Multiple Data

d. Multiple Instruction, Single Data

Which architectural feature allows multiple processors to access a shared pool of memory in parallel?

a. Cache coherence

b. NUMA (Non-Uniform Memory Access)

c. SIMD instruction set

d. Instruction pipelining

In the context of parallel processing, what is the purpose of a vector processor?

a. To process scalar operations sequentially

b. To execute instructions in parallel on multiple cores

c. To accelerate operations on arrays or vectors

d. To manage cache coherence

What is the significance of instruction-level parallelism (ILP) in processor architecture?

a. It allows multiple processors to share a common address space.

b. It enables parallel execution of multiple instructions within a single processor.

c. It facilitates communication between distributed-memory processors.

d. It determines the order of memory operations.

Which architectural feature is designed to enhance the memory access speed by predicting and pre-fetching data?

a. Out-of-order execution

b. Branch prediction

c. Memory interleaving

d. Speculative execution

What does the term "pipeline hazard" refer to in processor architecture?

a. Efficient use of instruction pipelines

b. Delays in instruction execution due to dependencies or conflicts

c. Parallel execution of multiple instructions

d. Synchronization between processors

In the context of parallel architecture, what is a benefit of using a heterogeneous computing model?

a. Simplified programming models

b. Improved cache coherence

c. Increased power consumption

d. Specialized processing for different tasks

Which type of memory architecture is characterized by having a uniform memory access time for all processors?

a. NUMA (Non-Uniform Memory Access)

b. UMA (Uniform Memory Access)

c. SIMD (Single Instruction, Multiple Data)

d. MIMD (Multiple Instruction, Multiple Data)

What is the purpose of a memory hierarchy in parallel computing architecture?

a. To eliminate parallelism

b. To increase sequential execution time

c. To manage data storage and access efficiently

d. To synchronize processes

How does hyper-threading contribute to parallelism in processor architecture?

a. By increasing clock speed

b. By adding more processor cores

c. By allowing multiple threads to share the same core

d. By optimizing branch prediction

What is the role of a memory controller in a parallel computing architecture?

a. To execute computational tasks

b. To manage the cache hierarchy

c. To handle memory access and communication with RAM

d. To perform vector operations

Which architectural feature is designed to improve parallel processing efficiency by overlapping the execution of multiple instructions?

a. Pipelining

b. Out-of-order execution

c. Superscalar architecture

d. Speculative execution

What is the purpose of a barrier synchronization mechanism in parallel architectures?

a. To increase sequential execution time

b. To eliminate parallelism

c. To synchronize processes at a specific point

d. To introduce race conditions

In a parallel computing architecture, what does the term "co-processor" refer to?

a. A processor with a large cache size

b. A specialized processor designed for specific tasks

c. A processor with a high clock speed

d. A processor with multiple cores

How does cache coherence impact the performance of a parallel computing system?

a. It improves memory access speed.

b. It introduces delays in memory access.

c. It ensures consistency in the data stored in different caches.

d. It eliminates the need for memory hierarchy.

What is the purpose of a speculative execution in processor architecture?

a. To optimize branch prediction

b. To synchronize memory access

c. To eliminate parallelism

d. To increase sequential execution time

What is the primary advantage of a distributed-memory architecture in parallel computing?

a. Efficient shared-memory access

b. Simplified programming models

c. Scalability to a large number of processors

d. Reduced communication overhead

How does the concept of "task parallelism" relate to the architecture supporting parallelism?

a. It describes the organization of processor cores in a chip.

b. It refers to the efficient use of memory hierarchy.

c. It involves executing multiple tasks independently on different processors.

d. It focuses on optimizing instruction-level parallelism.

What is the significance of a cache line in parallel computing architecture?

a. It defines the number of processor cores in a chip.

b. It determines the clock speed of the processor.

c. It represents a block of contiguous memory that is loaded into the cache.

d. It specifies the number of instructions executed in a pipeline.

Cmake and Header File

CMake:

What is CMake used for in C programming?

a. Compiler optimization

b. Code documentation

c. Build system generation

d. Dynamic memory allocation

Which command is used to generate build files with CMake?

a. cmake build

b. make

c. cmake generate

d. cmake configure

What is the purpose of the CMakeLists.txt file?

a. To store source code

b. To configure compiler options

c. To specify project dependencies

d. To define custom header files

How do you set the CMake minimum required version in a CMakeLists.txt file?

a. set(CMAKE_MINIMUM_VERSION)

b. cmake_minimum_required()

c. minimum_version_required()

d. set(MINIMUM_CMAKE_VERSION)

What does the cmake .. command do in the context of CMake?

a. Configures the build system in the current directory

b. Builds the project in the parent directory

c. Generates Makefiles in the parent directory

d. Installs the project in the parent directory

Which CMake command is used to set compiler flags?

a. set_compiler_flags()

b. add_compile_options()

c. compiler_flags()

d. set_flags()

What is the purpose of the add_executable command in a CMakeLists.txt file?

a. Adds a new source file to the project

b. Configures compiler options

c. Specifies project dependencies

d. Defines the executable target

How do you specify additional include directories in CMake?

a. add_include_directories()

b. include_directories()

c. set_include_paths()

d. include_paths()

What does the target_link_libraries command do in CMake?

a. Links object files

b. Links external libraries

c. Configures compiler options

d. Includes header files

How do you define and set a variable in CMake?

a. set_variable()

b. define()

c. set()

d. variable_define()

Custom Header File Creation in C:

What is the purpose of a header file in C programming?

a. To store object code

b. To provide function prototypes and declarations

c. To define main() function

d. To include external libraries

How do you include a header file in a C source file?

a. include "header.h"

b. include <header.h>

c. import "header.h"

d. import <header.h>

What is the convention for naming header files in C?

a. All lowercase with underscores

b. All uppercase with underscores

c. CamelCase

d. PascalCase

What is the purpose of include guards in a header file?

a. To prevent the inclusion of external libraries

b. To avoid duplicate inclusion of the same header file

c. To hide the contents of the header file

d. To limit the scope of variables

How do you declare a function prototype in a header file?

a. function_prototype();

b. declare function_prototype();

c. void function_prototype();

d. int function_prototype();

Which directive is used to conditionally include code in a header file?

a. #ifdef

b. #endif

c. #define

d. #include

What is the purpose of the extern keyword in a header file?

a. To define a variable

b. To declare a variable without defining it

c. To include external libraries

d. To specify compiler options

How do you create a guard macro for a header file named "example.h"?

a. #ifndef EXAMPLE_H #define EXAMPLE_H #endif

b. #define EXAMPLE_H #ifndef EXAMPLE_H #endif

c. #define EXAMPLE_H #endif #ifndef EXAMPLE_H

d. #ifndef EXAMPLE_H #endif

What is the role of a function definition in a header file?

a. To declare the function prototype

b. To implement the function

c. To specify compiler options

d. To include external libraries

How can you organize multiple header files in a C project?

a. Include all declarations in a single header file

b. Create separate header files for each function

c. Embed header files in the source code

d. Avoid using header files in C programming

OpenMP

What does OpenMP stand for?

a. Open Multi-Processing

b. Open Memory Processing

c. Open Message Passing

d. Open Multi-Programming

In OpenMP, which directive is used to parallelize a loop?

a. #pragma omp master

b. #pragma omp parallel

c. #pragma omp loop

d. #pragma omp for

What is the purpose of the omp_get_num_threads() function in OpenMP?

a. Retrieve the total number of threads in the system

b. Get the number of threads in the parallel region

c. Determine the maximum number of threads

d. Return the thread ID of the calling thread

In OpenMP, what does the "master" thread do in a parallel region?

a. Executes the parallel region

b. Synchronizes other threads

c. Manages memory allocation

d. Handles I/O operations

Which OpenMP clause is used to specify that a variable should be private to each thread in a parallel region?

a. private

b. shared

c. firstprivate

d. threadprivate

How is data-sharing accomplished in OpenMP?

a. By using global variables

b. Through explicit communication between threads

c. By using the "private" and "shared" clauses

d. Data-sharing is not supported in OpenMP

Which OpenMP directive is used to create a parallel region in C/C++?

a. #pragma omp parallel

b. #pragma omp section

c. #pragma omp master

d. #pragma omp task

What is the purpose of the omp_get_thread_num() function in OpenMP?

a. Retrieve the total number of threads in the system

b. Get the number of threads in the parallel region

c. Determine the maximum number of threads

d. Return the thread ID of the calling thread

Which OpenMP clause is used to control the scheduling of loop iterations?

a. schedule

b. parallel

c. threadprivate

d. atomic

In OpenMP, which directive is used to specify a critical section?

a. #pragma omp parallel

b. #pragma omp section

c. #pragma omp critical

d. #pragma omp master

What is the purpose of the omp_set_num_threads() function in OpenMP?

a. Set the number of threads for the entire program

b. Dynamically adjust the number of threads

c. Specify the number of threads for a parallel region

d. Disable multithreading

Which OpenMP clause is used to control the reduction operation in a parallel loop?

a. reduction

b. private

c. shared

d. critical

What is the role of the omp barrier directive in OpenMP?

a. To synchronize threads at a specific point

b. To create a new parallel region

c. To specify the loop schedule

d. To declare a critical section

How is the default schedule for a parallel loop determined in OpenMP?

a. By the operating system

b. By the compiler

c. By the programmer

d. OpenMP does not support loop scheduling

In OpenMP, what does the "firstprivate" clause do?

a. Initializes the variable to the first element in an array

b. Copies the value from the first thread to all other threads

c. Initializes the variable with the value from the first iteration

d. Specifies that the variable is private to the first thread

Which OpenMP directive is used to specify a parallel loop with dynamic scheduling?

a. #pragma omp for

b. #pragma omp parallel for

c. #pragma omp sections

d. #pragma omp taskloop

What is the purpose of the omp_get_wtime() function in OpenMP?

a. Get the current wall-clock time

b. Return the time spent in a parallel region

c. Measure the execution time of a specific loop

d. Determine the time elapsed since program start

Which OpenMP clause is used to declare a variable as shared among all threads in a parallel region?

a. shared

b. private

c. firstprivate

d. threadprivate

What is the effect of the nowait clause in OpenMP?

a. It forces the threads to wait for each other at the end of a parallel region.

b. It allows threads to continue without waiting at the end of a parallel region.

c. It specifies a wait time for each thread before proceeding.

d. It has no impact on thread synchronization.

How can you specify that a section of code should be executed by a single thread in OpenMP?

a. #pragma omp master

b. #pragma omp single

c. #pragma omp critical

d. #pragma omp parallel

Which OpenMP directive is used to create parallel sections in a program?

a. #pragma omp parallel

b. #pragma omp sections

c. #pragma omp master

d. #pragma omp task

What is the purpose of the omp_get_max_threads() function in OpenMP?

a. Retrieve the total number of threads in the system

b. Get the maximum number of threads allowed by the system

c. Determine the maximum number of threads in the parallel region

d. Return the thread ID of the calling thread

In OpenMP, which clause is used to specify a reduction operation on a variable in a parallel loop?

a. reduction

b. private

c. shared

d. critical

How is the collapse clause used in OpenMP?

a. To specify the number of loops to collapse into a single loop

b. To define the loop schedule in a parallel region

c. To declare a variable as private

d. To synchronize threads at the end of a parallel region

Which OpenMP directive is used to parallelize a single task block in a program?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp sections

What is the purpose of the schedule clause in OpenMP?

a. To declare a loop variable as private

b. To control the scheduling of loop iterations

c. To specify the maximum number of threads

d. To synchronize threads at the end of a parallel region

Which OpenMP clause is used to specify a private copy of a variable for each thread in a parallel region?

a. private

b. shared

c. firstprivate

d. threadprivate

In OpenMP, what does the flush directive do?

a. Forces synchronization of all threads

b. Clears the cache memory

c. Initializes variables to a default value

d. Specifies a barrier at the end of a parallel region

What is the purpose of the nowait clause in the omp for directive?

a. It forces a wait at the end of the loop.

b. It allows threads to continue without waiting at the end of the loop.

c. It introduces a delay before the loop starts.

d. It has no impact on loop execution.

In OpenMP, what is the purpose of the reduction clause in a parallel loop?

a. To reduce the number of iterations in the loop

b. To simplify the loop structure

c. To perform a reduction operation on a variable

d. To eliminate the need for a loop

Which OpenMP directive is used to create a parallel task in a program?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp task

How can you specify a custom reduction operation in OpenMP?

a. Use the custom_reduction clause

b. Define a function and use it with the reduction clause

c. Specify the reduction operation directly in the loop

d. OpenMP does not support custom reduction operations

What does the omp_get_dynamic() function in OpenMP return?

a. The number of threads in the dynamic team

b. The status of dynamic thread adjustment

c. The total number of threads in the system

d. The thread ID of the calling thread

In OpenMP, what is the purpose of the omp_set_dynamic() function?

a. To set the number of threads in the dynamic team

b. To control dynamic thread adjustment

c. To specify the loop schedule

d. To initialize variables to default values

What is the purpose of the omp_get_nested() function in OpenMP?

a. Retrieve the total number of threads in the system

b. Get the nested parallelism status

c. Determine the maximum number of threads

d. Return the thread ID of the calling thread

Which OpenMP directive is used to enable or disable nested parallelism?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp parallel sections

How does the omp_get_thread_limit() function in OpenMP help?

a. Retrieve the total number of threads in the system

b. Get the maximum number of threads allowed by the system

c. Determine the maximum number of threads in the parallel region

d. Return the thread ID of the calling thread

What is the purpose of the omp_set_schedule() function in OpenMP?

a. To specify the loop schedule

b. To control dynamic thread adjustment

c. To set the schedule for nested parallelism

d. OpenMP does not provide such a function

How does the schedule(runtime) clause in OpenMP affect loop scheduling?

a. It lets the operating system determine the schedule at runtime.

b. It statically schedules loop iterations.

c. It dynamically adjusts the schedule during runtime.

d. It disables loop scheduling.

Which OpenMP directive is used to declare a section of code that should only be executed by one thread?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp sections

What is the purpose of the omp_get_num_procs() function in OpenMP?

a. Retrieve the total number of processes in the system

b. Get the number of processes in the parallel region

c. Determine the maximum number of processes

d. Return the process ID of the calling process

How does the omp_set_num_threads() function impact the execution of an OpenMP program?

a. It sets the number of processors to be used.

b. It dynamically adjusts the number of threads.

c. It statically sets the number of threads for the entire program.

d. It has no impact on thread management.

In OpenMP, what does the collapse clause do when applied to nested loops?

a. Combines the nested loops into a single loop

b. Separates the nested loops into individual loops

c. Specifies the number of iterations for each nested loop

d. Controls the loop scheduling

Which OpenMP directive is used to parallelize a single loop with dynamic scheduling?

a. #pragma omp for

b. #pragma omp parallel for

c. #pragma omp sections

d. #pragma omp taskloop

What does the nowait clause in OpenMP imply when applied to a parallel loop?

a. Threads wait for each other at the end of the loop.

b. Threads do not wait for each other at the end of the loop.

c. It introduces a delay before the loop starts.

d. It has no impact on thread synchronization.

How is the default schedule determined in OpenMP if the schedule clause is not specified in a parallel loop?

a. By the compiler

b. By the operating system

c. By the programmer

d. OpenMP does not support default scheduling

What is the purpose of the omp_get_schedule() function in OpenMP?

a. Retrieve the loop schedule type

b. Get the number of threads in the parallel region

c. Determine the schedule for nested parallelism

d. Return the thread ID of the calling thread

In OpenMP, what does the ordered directive ensure in a parallel loop?

a. Enforces an ordered execution of loop iterations

b. Randomizes the order of loop iterations

c. Disables loop scheduling

d. Declares a loop as unordered

Which OpenMP directive is used to create a parallel region with thread-private variables?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp parallel sections

How can you specify that a variable should be shared among threads in OpenMP?

a. Declare it as a global variable

b. Use the shared clause in the parallel region

c. Include it in the private clause

d. Use the firstprivate clause

What is the role of the omp atomic directive in OpenMP?

a. To disable atomic operations

b. To specify atomic data types

c. To ensure that a block of code is executed atomically

d. To synchronize threads at the end of a parallel region

Which OpenMP clause is used to specify a reduction operation for multiple variables in a parallel loop?

a. reduction

b. private

c. shared

d. critical

How does the omp_get_dynamic() function impact the behavior of dynamic thread adjustment in OpenMP?

a. Enables dynamic thread adjustment

b. Disables dynamic thread adjustment

c. Returns the current status of dynamic thread adjustment

d. Dynamically adjusts the number of threads

In OpenMP, what does the master directive signify?

a. Designates a master thread that executes a specific block of code

b. Assigns a specific task to the primary thread

c. Ensures that the master thread is the only thread in the parallel region

d. Synchronizes all threads in the parallel region

Which OpenMP directive is used to declare a variable as private to each thread in a parallel region?

a. #pragma omp shared

b. #pragma omp private

c. #pragma omp firstprivate

d. #pragma omp threadprivate

What is the purpose of the omp get_nested() function in OpenMP?

a. Retrieve the nested parallelism status

b. Get the number of nested loops

c. Determine the maximum number of nested parallel regions

d. Return the thread ID of the calling thread

How does the schedule(static, chunk_size) clause in OpenMP affect loop scheduling?

a. It dynamically adjusts the chunk size during runtime.

b. It statically schedules loop iterations with a specified chunk size.

c. It introduces a delay before the loop starts.

d. It has no impact on loop scheduling

Which OpenMP directive is used to specify that a section of code should be executed by a single thread?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp sections

What does the reduction(+:sum) clause do in the context of an OpenMP parallel loop?

a. Computes the sum of all loop iterations and assigns it to the variable sum

b. Specifies a custom reduction operation on the variable sum

c. Declares the variable sum as private to each thread

d. Eliminates the need for a reduction operation on the variable sum

In OpenMP, what is the purpose of the flush directive?

a. Forces synchronization of all threads

b. Clears the cache memory

c. Initializes variables to a default value

d. Specifies a barrier at the end of a parallel region

In OpenMP, what does the omp_get_thread_limit() function return?

a. The total number of threads in the system

b. The maximum number of threads allowed by the system

c. The number of threads in the parallel region

d. The thread ID of the calling thread

How can you enable dynamic adjustment of the number of threads in an OpenMP program?

a. Use the omp_set_dynamic() function

b. Include the dynamic clause in the parallel directive

c. Set the environment variable OMP_DYNAMIC to true

d. Use the omp_get_dynamic() function

Which OpenMP directive is used to declare a parallel loop with the simd clause?

a. #pragma omp parallel

b. #pragma omp for simd

c. #pragma omp single

d. #pragma omp master

What is the purpose of the omp_get_cancellation() function in OpenMP?

a. Retrieve the status of thread cancellation

b. Enable thread cancellation

c. Disable thread cancellation

d. Return the thread ID of the calling thread

In OpenMP, what does the cancellation point refer to?

a. A point in the program where threads are canceled

b. A point in the program where cancellation is not allowed

c. A synchronization point in a parallel region

d. A point in the program where dynamic adjustment occurs

Which OpenMP directive is used to specify the maximum number of threads for a parallel region?

a. #pragma omp max_threads

b. #pragma omp parallel

c. #pragma omp num_threads

d. #pragma omp thread_limit

What is the purpose of the omp_set_schedule() function in OpenMP?

a. To specify the loop schedule

b. To control dynamic thread adjustment

c. To set the schedule for nested parallelism

d. OpenMP does not provide such a function

In OpenMP, which clause is used to declare a variable as firstprivate to each thread in a parallel region?

a. firstprivate

b. private

c. shared

d. threadprivate

What is the effect of the nowait clause in the omp sections directive?

a. Threads wait for each other at the end of each section.

b. Threads do not wait for each other at the end of each section.

c. It introduces a delay before the sections start.

d. It has no impact on section execution.

Which OpenMP directive is used to create a parallel region with a fixed number of threads?

a. #pragma omp parallel

b. #pragma omp single

c. #pragma omp master

d. #pragma omp parallel sections

In OpenMP, what does the copyin clause do when applied to a variable in a parallel region?

a. Initializes the variable to a default value

b. Copies the value from the master thread to all other threads

c. Initializes the variable with a specified value

d. Copies the value from the enclosing parallel region to all threads

Which OpenMP directive is used to declare a variable as shared among threads in a parallel region?

a. #pragma omp shared

b. #pragma omp private

c. #pragma omp firstprivate

d. #pragma omp threadprivate

What is the purpose of the omp_get_max_active_levels() function in OpenMP?

a. Retrieve the maximum number of active levels for nested parallelism

b. Get the number of active levels in the current parallel region

c. Determine the maximum number of threads

d. Return the thread ID of the calling thread

In OpenMP, what is the function of the simd clause in the omp parallel for directive?

a. Specifies a reduction operation for the loop

b. Enables single instruction, multiple data (SIMD) optimization

c. Declares a loop variable as private

d. Disables loop scheduling

How does the cancel directive in OpenMP impact the cancellation of threads?

a. Enables thread cancellation at specific points in the program

b. Disables thread cancellation entirely

c. Forces immediate cancellation of all threads

d. Specifies a wait time before cancellation

Which OpenMP directive is used to create a parallel loop with the simd and collapse clauses?

a. #pragma omp parallel

b. #pragma omp for simd

c. #pragma omp sections

d. #pragma omp taskloop

In OpenMP, what does the cancel construct do?

a. Specifies a cancellation point

b. Cancels all threads in the parallel region

c. Disables thread cancellation

d. Forces immediate cancellation of the master thread

What is the purpose of the omp_get_proc_bind() function in OpenMP?

a. Retrieve the processor binding policy

b. Set the processor binding policy

c. Determine the maximum number of threads

d. Return the thread ID of the calling thread

How does the ordered clause in OpenMP impact the execution of sections in a parallel region?

a. Ensures an ordered execution of sections

b. Randomizes the order of section execution

c. Disables section execution entirely

d. Specifies a wait time before section execution

In OpenMP, what does the nowait clause in the omp parallel sections directive imply?

a. Threads wait for each other at the end of each section.

b. Threads do not wait for each other at the end of each section.

c. It introduces a delay before the sections start.

d. It has no impact on section execution.

CUDA

True or False: Coprocessors are designed to assist the main processor in performing specific types of computations.

a. True
b. False

Which term refers to the practice of using both general-purpose processors and specialized coprocessors in a single system?

a. Parallel computing

b. Heterogeneous computing

c. Serial computing

d. Distributed computing

What is the primary purpose of accelerators in HPC?

a. Increase clock speed

b. Enhance memory capacity

c. Boost overall system performance

d. Improve network connectivity

Which of the following is a common coprocessor architecture used in HPC applications?

a. x86

b. ARM

c. CUDA

d. Java

What type of computations are typically offloaded to coprocessors in HPC systems?

a. Graphics rendering

b. Integer operations

c. Floating-point calculations

d. File I/O operations

In heterogeneous computing, what is the term for the process of assigning specific tasks to the most suitable processing unit, such as a coprocessor or a GPU?

a. Task switching

b. Task offloading

c. Task parallelism

d. Task scheduling

Which programming model is commonly used for developing applications that leverage GPU accelerators?

a. OpenMP

b. MPI

c. CUDA

d. OpenACC

What is the impact of coprocessor integration on the power consumption of an HPC system?

a. Increased power consumption

b. Decreased power consumption

c. No impact on power consumption

d. Variable impact depending on the coprocessor type

Which term describes the technique of breaking down a large computational task into smaller subtasks that can be processed simultaneously?

a. Parallel processing

b. Pipelining

c. Vectorization

d. Decomposition

What role do coprocessors play in addressing memory bandwidth limitations in HPC applications?

a. Increase memory capacity

b. Improve memory access speed

c. Reduce the need for memory

d. Enhance memory reliability

Which of the following is NOT a common type of coprocessor used in HPC?

a. FPGA

b. DSP

c. SSD

d. GPU

What programming language is often used for developing applications for FPGA-based accelerators?

a. C++

b. Verilog

c. Python

d. Java

True or False: Coprocessors are primarily designed for serial computing tasks.

a. True
b. False

In the context of HPC, what does the term "coherence" refer to?

a. Consistency of data across multiple coprocessors

b. Speed of coprocessor clock cycles

c. Memory capacity of the coprocessor

d. Coprocessor's ability to handle parallel tasks

Which of the following is an example of a widely used parallel programming framework for HPC applications?

a. DirectX

b. OpenCL

c. OpenGL

d. CUDA

What is the primary advantage of using GPUs as accelerators in HPC?

a. Lower cost

b. Higher memory capacity

c. High parallel processing capabilities

d. Lower power consumption

Which coprocessor architecture is commonly associated with artificial intelligence (AI) workloads in HPC?

a. CUDA

b. FPGA

c. Tensor Core

d. DSP

What is the role of vectorization in optimizing code for coprocessors?

a. Parallelizing code execution

b. Increasing the coprocessor's clock speed

c. Reducing the coprocessor's memory requirements

d. Utilizing vector processing units efficiently

What is the purpose of using a coprocessor with a high number of cores in HPC applications?

a. Increase clock speed

b. Enhance parallelism

c. Improve single-threaded performance

d. Boost memory access speed

Which of the following is a challenge associated with the heterogeneous nature of coprocessors in HPC systems?

a. Simplified programming models

b. Homogeneous memory access

c. Efficient workload distribution

d. Software compatibility issues

What does GPU stand for?

a) General Processing Unit

b) Graphics Processing Unit

c) General Purpose Unit

d) Gaming Processing Unit

Which architecture is used in NVIDIA's latest GPU models as of 2023?

a) Pascal

b) Kepler

c) Ampere

d) Turing

What is the primary function of CUDA cores in a GPU?

a) Video decoding

b) Ray tracing

c) Parallel processing

d) Memory management

What does VRAM stand for in the context of GPU specifications?

a) Video RAM

b) Virtual RAM

c) Visual RAM

d) Vector RAM

Which technology is associated with real-time ray tracing in NVIDIA GPUs?

a) G-Sync

b) ShadowPlay

c) DLSS

d) RTX

What is the purpose of Tensor Cores in NVIDIA's GPU architecture?

a) Enhance gaming graphics

b) Improve machine learning performance

c) Accelerate video encoding

d) Optimize power consumption

Which memory type is used in GDDR6X in recent NVIDIA GPUs?

a) GDDR5

b) GDDR6

c) HBM2

d) DDR4

What is the purpose of NVIDIA NVLink technology?

a) Crossfire support

b) Improved multi-GPU communication

c) Power efficiency optimization

d) Video streaming acceleration

What does SLI stand for in the context of NVIDIA GPUs?

a) Scalable Link Interface

b) Single Link Integration

c) System-Level Interconnect

d) Simulated Load Index

What does NVIDIA GPU Boost technology aim to achieve?

a) Overclocking

b) Power consumption reduction

c) Dynamic clock adjustment

d) Ray tracing optimization

Name two features that distinguish the NVIDIA Ampere architecture from the previous Turing architecture.

a) CUDA Cores and GDDR6X

b) RT Cores and Tensor Cores

c) DLSS and Ray Tracing

d) NVLink and SLI

What is the purpose of CUDA parallel processing in NVIDIA GPUs?

a) Improved VRAM allocation

b) Enhanced multi-monitor support

c) Parallel execution of tasks

d) Better power management

What is the maximum resolution supported by the HDMI 2.1 standard in recent NVIDIA GPUs?

a) 1080p

b) 1440p

c) 4K

d) 8K

Describe the significance of the PCIe interface in connecting GPUs to a computer system.

a) Faster data transfer between GPU and CPU

b) Improved power efficiency

c) Enhanced graphics rendering

d) Better cooling capabilities

What is the purpose of the NVIDIA NVENC encoder in GPU hardware?

a) Ray tracing acceleration

b) Video encoding and streaming

c) Parallel processing of game physics

d) VRAM optimization

Name two technologies that enhance the performance of NVIDIA GPUs in virtual reality applications.

a) G-Sync and FreeSync

b) DLSS and Ray Tracing

c) VRWorks and VR SLI

d) NVLink and SLI

Explain the role of PCIe power connectors on NVIDIA GPUs.

a) Data transfer with the motherboard

b) Power supply to the GPU

c) Cooling fan control

d) Monitor connectivity

CUDA Programming

What does CUDA stand for in the context of parallel computing?

a) Central Unified Device Architecture

b) Compute Unified Device Architecture

c) Concurrent Unified Data Acceleration

d) CUDA is not an acronym

Which programming language is commonly used for CUDA programming?

a) Java

b) C++

c) Python

d) CUDA has its own programming language

What is the primary purpose of the CUDA parallel computing platform?

a) CPU optimization

b) GPU optimization

c) Memory management

d) Networking

What are CUDA cores primarily responsible for in a GPU?

a) Memory allocation

b) Parallel processing

c) Graphics rendering

d) Network communication

What does a CUDA thread represent in the context of parallel execution?

a) A CPU core

b) A GPU core

c) A parallel task

d) A memory address

Which CUDA function is used to allocate memory on the GPU?

a) cudaMalloc

b) gpuAlloc

c) allocateGPU

d) mallocGPU

What is the purpose of the CUDA kernel in a parallel program?

a) GPU initialization

b) CPU-GPU communication

c) Sequential processing

d) Parallel computation

How is data transferred between the CPU and GPU in CUDA programming?

a) Direct memory access

b) CPU and GPU share the same memory space

c) cudaMemcpy function

d) GPU memory mapping

What is the significance of the threadIdx variable in CUDA programming?

a) GPU thread index

b) CPU thread index

c) Global memory index

d) Shared memory index

Which CUDA keyword is used to declare a function to be executed on the GPU?

a) device

b) global

c) cuda

d) gpu

What is the purpose of the CUDA grid in the execution model?

a) Memory allocation

b) Thread synchronization

c) Define the structure of parallel tasks

d) GPU clock control

Which CUDA API call is used to launch a kernel on the GPU?

a) cudaRunKernel

b) gpuLaunch

c) kernelExecute

d) cudaLaunchKernel

How are threads organized into blocks in the CUDA programming model?

a) One-dimensional

b) Two-dimensional

c) Three-dimensional

d) All of the above

Which CUDA function is used to synchronize the CPU with the GPU?

a) cudaSync

b) gpuSync

c) cudaDeviceSynchronize

d) syncGPU

What is the purpose of the shared memory in CUDA programming?

a) Global data storage

b) Communication between threads

c) CPU-GPU communication

d) Temporary storage for the CPU

Which CUDA keyword is used to declare a variable in shared memory?

a) global

b) shared

c) local

d) thread

What is warp in the context of CUDA programming?

a) A bug in the code

b) A parallel task

c) A group of threads

d) A memory space

Which CUDA function is used to free memory allocated on the GPU?

a) cudaFree

b) gpuFree

c) freeGPU

d) deallocGPU

What is the purpose of the warp size in CUDA architecture?

a) GPU clock speed

b) Number of threads in a block

c) Size of the shared memory

d) Smallest unit of thread execution

Which NVIDIA tool is commonly used for profiling and optimizing CUDA applications?

a) CUDA Debugger

b) NVIDIA Inspector

c) CUDA Visual Profiler

d) NVIDIA System Monitor

What is the purpose of the blockIdx variable in CUDA programming?

a) GPU block index

b) CPU block index

c) Global memory index

d) Shared memory index

Which CUDA datatype is used to represent a 3D vector in device code?

a) float3

b) vector3

c) triple

d) dim3

What is the significance of the warp divergence issue in CUDA programming?

a) Decreased GPU performance

b) Increased memory consumption

c) Thread synchronization problems

d) Unpredictable program behavior

In CUDA, what is the purpose of the warp shuffle operation?

a) Thread communication within a warp

b) Memory allocation for a warp

c) Warp synchronization

d) GPU clock control

Which CUDA API call is used to set the device on which the CUDA code will execute?

a) cudaSetDevice

b) gpuSetDevice

c) setCUDADevice

d) setGPUDevice

What does the term "cooperative groups" refer to in CUDA programming?

a) A group of cooperating threads

b) Synchronized CPU and GPU execution

c) Memory coherency optimization

d) Dynamic thread allocation

Which CUDA keyword is used to declare a variable in constant memory?

a) constant

b) readonly

c) const

d) readonly

What is the purpose of the warp ballot operation in CUDA?

a) Counting the number of threads in a warp

b) Broadcasting a condition to all threads in a warp

c) Distributing work among threads in a warp

d) Sorting threads within a warp

Which CUDA function is used to perform atomic operations on global memory?

a) atomicAdd

b) globalAtomic

c) atomicModify

d) memAtomic

What does the term "thread divergence" mean in the context of CUDA programming?

a) Threads within a block taking different code paths

b) Threads in a warp all executing the same instruction

c) Thread synchronization issues

d) Thread allocation problems

What is the purpose of the warp-level reduction operation in CUDA?

a) Reducing memory usage within a warp

b) Combining results from threads within a warp

c) Achieving higher clock speeds in a warp

d) Sorting threads within a warp

What does the term "grid-stride loop" refer to in CUDA programming?

a) A loop that spans multiple blocks

b) A loop that spans multiple grids

c) A loop that iterates over the entire grid

d) A loop with a variable stride

Which CUDA keyword is used to declare a variable in global memory?

a) global

b) device

c) globalmem

d) memory

Which CUDA API call is used to query the number of available GPUs on a system?

a) cudaGetDeviceCount

b) gpuQueryDeviceCount

c) getGPUCount

d) cudaDeviceQuery

In CUDA, what is the purpose of the constant memory cache?

a) Faster access to global memory

b) Storage of constant values for all threads

c) Improved warp synchronization

d) Temporary storage for thread variables

What is the purpose of the warp shuffle intrinsic function in CUDA programming?

a) Inter-thread communication within a warp

b) Sorting threads within a warp

c) Dynamic thread allocation

d) Global memory access optimization

Which CUDA datatype is used to represent a 2D matrix in device code?

a) matrix2

b) float2x2

c) float2

d) dim2

What does the term "thread synchronization" mean in CUDA programming?

a) Coordinating the execution of threads

b) Assigning tasks to different threads

c) Balancing workload among threads

d) Allocating memory for threads

In CUDA, what is the purpose of the warp-level shuffle operation?

a) Efficient data exchange between threads in a warp

b) Sorting threads within a warp

c) Random access to global memory

d) Dynamic thread allocation

Which CUDA API call is used to determine the maximum number of threads per block?

a) cudaGetMaxThreadsPerBlock

b) gpuMaxThreadsPerBlock

c) getMaxThreadsPerBlock

d) cudaMaxThreadsPerBlock

What is the purpose of the CUDA Warp Vote Functions?

a) Warp synchronization

b) Data reduction within a warp

c) Counting the number of active threads in a warp

d) Sorting threads within a warp

Which CUDA API call is used to asynchronously copy data from host to device?

a) cudaMemcpy

b) cudaMemcpyAsync

c) cudaCopyHostToDevice

d) gpuMemcpyAsync

What does the term "thread-level parallelism" mean in CUDA programming?

a) Parallel execution of threads within a block

b) Parallel execution of blocks within a grid

c) Sequential execution of threads

d) Memory allocation for threads

Which CUDA function is used to perform atomic operations on shared memory?

a) atomicAdd

b) sharedAtomic

c) atomicModifyShared

d) memAtomicShared

In CUDA, what is the purpose of the warp-level broadcast operation?

a) Distributing data to all threads within a warp

b) Reducing data within a warp

c) Sorting threads within a warp

d) Dynamic thread allocation

What is the role of the threadIdx.x variable in CUDA programming?

a) GPU thread index

b) CPU thread index

c) Warp index

d) Block index

Which CUDA function is used to allocate dynamically managed memory on the GPU?

a) cudaMallocManaged

b) gpuMallocManaged

c) mallocGPUManaged

d) cudaAllocManaged

What is the purpose of the CUDA constant memory cache?

a) Improved access to global memory

b) Storage of constant values for all threads

c) Dynamic allocation of constant memory

d) Temporary storage for thread variables

In CUDA, what is the significance of the warp shuffle XOR operation?

a) Combining data from threads within a warp

b) Sorting threads within a warp

c) Optimizing memory access

d) Dynamic thread allocation

Which CUDA API call is used to set the maximum size of the shared memory per block?

a) cudaSetSharedMemorySize

b) gpuSetSharedMemorySize

c) setSharedMemorySize

d) cudaDeviceSetSharedMemConfig

Which CUDA datatype is used to represent a 3D point in device code?

a) point3

b) float3

c) dim3

d) vector3

What is the purpose of the CUDA warp shuffle up and down operations?

a) Data exchange between threads within a warp

b) Sorting threads within a warp

c) Random access to global memory

d) Dynamic thread allocation

What does the term "thread divergence avoidance" mean in CUDA programming?

a) Threads within a block executing different code paths

b) Threads in a warp all executing the same instruction

c) Optimizing warp synchronization

d) Preventing memory allocation issues

Which CUDA keyword is used to declare a variable in texture memory?

a) texture

b) text

c) texmem

d) tex

In CUDA, what is the significance of the constant memory cache?

a) Faster access to global memory

b) Storage of constant values for all threads

c) Improved warp synchronization

d) Dynamic allocation of constant memory

What is the purpose of the CUDA warp vote all and any functions?

a) Counting the number of active threads in a warp

b) Sorting threads within a warp

c) Checking conditions for all or any threads within a warp

d) Reducing data within a warp

Which CUDA datatype is used to represent a 4D vector in device code?

a) vector4

b) float4

c) dim4

d) quad

What is the role of the blockIdx.y and blockIdx.z variables in CUDA programming?

a) Block indices in the y and z dimensions

b) Thread indices in the y and z dimensions

c) Global memory indices in the y and z dimensions

d) Warp indices in the y and z dimensions

Which CUDA API call is used to query the amount of shared memory per block?

a) cudaGetSharedMemorySize

b) gpuQuerySharedMemorySize

c) getSharedMemorySize

d) cudaDeviceGetSharedMemConfig

What does the term " warp-synchronous programming" mean in CUDA?

a) Programming that optimizes warp synchronization

b) Programming that avoids warp divergence

c) Utilizing features specific to CUDA warps

d) Parallel execution across multiple warps

Which CUDA function is used to launch a kernel with multiple blocks and threads?

a) cudaLaunch

b) gpuLaunchKernel

c) launchKernel

d) cudaLaunchKernel

What is the purpose of the warp-level shuffle operations in CUDA programming?

a) Data exchange between threads within a warp

b) Sorting threads within a warp

c) Dynamic thread allocation

d) Reducing data within a warp

In CUDA, what does the term "thread granularity" refer to?

a) The number of threads in a block

b) The size of shared memory per block

c) The number of warps in a block

d) The size of global memory per block

Which CUDA API call is used to synchronize the CPU with the GPU and ensures that all the GPU work is completed?

a) cudaSynchronize

b) gpuSync

c) cudaDeviceSynchronize

d) syncGPU

What is the role of the gridDim.x variable in CUDA programming?

a) Grid index in the x-dimension

b) Block index in the x-dimension

c) Thread index in the x-dimension

d) Global memory index in the x-dimension

What is the purpose of the CUDA warp vote balloting functions?

a) Counting the number of active threads in a warp

b) Sorting threads within a warp

c) Distributing data to threads within a warp

d) Reducing data within a warp

Which CUDA function is used to set the preferred cache configuration for a device?

a) cudaSetCacheConfig

b) gpuSetCacheConfig

c) setCacheConfig

d) cudaDeviceSetCacheConfig

Which CUDA API call is used to set the size of the L1 cache per multiprocessor?

a) cudaSetCacheSize

b) gpuSetCacheSize

c) setCudaCacheSize

d) cudaDeviceSetCacheConfig

What does the term "loop unrolling" refer to in the context of CUDA programming?

a) Expanding a loop to reduce overhead

b) Reducing the number of loops for optimization

c) Combining multiple loops into one

d) Reversing the iteration order of a loop

What is the significance of the CUDA device properties in GPU programming?

a) Information about the GPU hardware and capabilities

b) Device-specific runtime parameters

c) Kernel execution parameters

d) Shared memory allocation details

In CUDA, what is the purpose of the warp shuffle XOR and broadcast operations?

a) Combining data from threads within a warp

b) Distributing data to all threads within a warp

c) Sorting threads within a warp

d) Reducing data within a warp

In CUDA, what is the role of the gridDim.y and gridDim.z variables?

a) Block indices in the y and z dimensions

b) Thread indices in the y and z dimensions

c) Global memory indices in the y and z dimensions

d) Warp indices in the y and z dimensions

Which CUDA API call is used to set the size of the shared memory per block?

a) cudaSetSharedMemorySize

b) gpuSetSharedMemorySize

c) setSharedMemorySize

d) cudaDeviceSetSharedMemConfig

What is the purpose of the CUDA cooperative groups in parallel programming?

a) Efficient communication and synchronization among threads

b) Reducing data within a warp

c) Sorting threads within a warp

d) Dynamically allocating threads to different tasks

Which CUDA datatype is used to represent a 1D array in device code?

a) array1

b) float1

c) dim1

d) vector1

What is the significance of the CUDA cooperative groups "thread subset" concept?

a) Selecting a subset of threads within a block for parallel tasks

b) Synchronizing threads within a warp

c) Reducing data within a warp

d) Sorting threads within a warp

In CUDA, what is the purpose of the __syncthreads() function?

a) Synchronize threads within a block

b) Synchronize threads across blocks

c) Pause the GPU execution

d) Terminate the CUDA kernel

Which CUDA API call is used to query the maximum number of threads per multiprocessor?

a) cudaGetMaxThreadsPerMultiProcessor

b) gpuMaxThreadsPerMultiprocessor

c) getMaxThreadsPerMultiprocessor

d) cudaDeviceGetAttribute

What is the purpose of the CUDA dynamic parallelism feature?

a) Launching kernels from within other kernels

b) Dynamically allocating shared memory

c) Sorting threads within a warp

d) Reducing data within a warp

Which CUDA function is used to perform atomic operations on device memory?

a) atomicDevice

b) atomicModifyDevice

c) atomicDeviceAdd

d) atomicExch

Question in High Performance Computing – Short Answer Question

---

Serial Computing:

a. What is the primary characteristic of serial computing?

b. In serial computing, how are tasks typically executed?

c. What is the impact of Amdahl's Law on serial computing?

Parallel Computing:

a. Define parallel computing.

b. What is the main advantage of parallel computing over serial computing?

c. Name a programming model commonly used in parallel computing.

Concurrency vs. Parallelism:

a. Explain the difference between concurrency and parallelism.

b. How can a program exhibit concurrency in a parallel computing environment?

Speedup and Efficiency:

a. Define speedup in the context of parallel computing.

b. How is efficiency calculated in parallel computing?

Task Decomposition:

a. What is task decomposition in the context of parallel computing?

b. Why is task decomposition important for parallelization?

Parallel Architectures:

a. Name two types of parallel computer architectures.

b. What is a shared-memory architecture?

Message Passing:

a. What is message passing in parallel computing?

b. Name a commonly used communication library for message passing.

Parallel Programming Languages:

a. Name a language specifically designed for parallel programming.

b. How does OpenMP facilitate parallel programming?

Parallel Processing Models:

a. Describe the difference between SIMD and MIMD processing models.

b. Give an example of an application suitable for SIMD processing.

Distributed Computing:

a. What is distributed computing?

b. How does distributed computing differ from parallel computing?

Parallel Algorithms:

a. Name a sorting algorithm suitable for parallelization.

b. How can parallelism be applied to matrix multiplication?

Parallel Computing Challenges:

a. What are some common challenges in designing parallel algorithms?

b. How does load balancing impact the performance of parallel computing systems?

Scalability:

a. Define scalability in the context of parallel computing.

b. What factors can affect the scalability of a parallel program?

Task Synchronization:

a. Why is task synchronization important in parallel computing?

b. Name a synchronization primitive used in parallel programming.

Supercomputing:

a. What is a supercomputer?

b. Name a well-known supercomputing architecture.

Parallel I/O:

a. Why is parallel I/O important in parallel computing?

b. Name a parallel file system.

Task Granularity:

a. Define task granularity in the context of parallel computing.

b. How does task granularity affect the performance of parallel programs?

Hybrid Parallelism:

a. What is hybrid parallelism?

b. Give an example of a programming model that supports hybrid parallelism.

Quantum Computing:

a. Briefly explain the concept of quantum computing.

b. What are some potential advantages of quantum computing over classical computing?