IMPORTANT QUESTION

Unix System Programming (BCS515C)

Module 1: Introduction to Unix:

1. Explain the architecture of the Unix operating system with a neat diagram.
2. Discuss the features of Unix and its environment.
3. Describe the command structure in Unix with examples.
4. Differentiate between internal and external commands with examples.
5. Explain the use of the type command and its significance.
6. Illustrate the concept of relative and absolute pathnames with examples.
7. What are hidden files? How can you list them? Provide examples.
8. Explain the usage of directory commands like pwd, cd, mkdir, and rmdir.
9. Write a shell script to copy a file to a new directory.
10. Demonstrate the use of cat, mv, rm, cp, wc, and od commands.

Module 2: File Attributes, Permissions, and Shell Programming:

1. Explain file attributes in Unix. How do you use the ls command with options to view them?
2. Describe the methods to change file permissions using absolute and relative methods.
3. Discuss the concept of wildcards in Unix and their usage with examples.
4. Explain the redirection of standard files with suitable examples.
5. Illustrate the use of pipes in connecting commands with real-world use cases.
6. What are regular expressions? Differentiate between basic and extended regular expressions with examples.
7. Write a shell program to calculate the factorial of a number using while and if.
8. Explain the use of the test command and its shortcuts in shell programming.
9. Describe the here document and its application in Unix.
10. Create a shell script to accept command-line arguments and display them in reverse order.

Module 3: Unix Standardization, File I/O, and Environment:

1. Discuss the standardization of Unix and its implementations.
2. Explain the usage of open, create, read, write, and close functions in file handling.
3. Differentiate between relative and absolute paths using chdir and fchdir functions.
4. Write a program to demonstrate the use of getcwd and mkdir functions.
5. What are device special files? Discuss their importance.
6. Explain the memory layout of a C program in Unix.
7. Discuss environment variables and their significance in Unix.
8. Illustrate the use of setjmp and longjmp functions with a program.
9. Explain the getrlimit and setrlimit functions with examples.
10. Write a C program to list all files in the current directory.

Module 4: Process Control and Inter-Process Communication (IPC):

1. Explain the concepts of process identifiers in Unix with examples.
2. Differentiate between fork and vfork system calls.
3. Describe the wait and exec family of functions with examples.
4. What are race conditions? How can they be avoided in Unix?
5. Explain the concept of pipes and their usage in IPC.
6. Discuss FIFOs and how they are used for communication.
7. Explain the popen and pclose functions with real-world examples.
8. Discuss the methods to implement shared memory in Unix.
9. Write a C program to demonstrate the usage of semaphores.
10. Explain the client-server model and its properties in Unix.

Module 5: Signals and Daemon Processes:

1. Explain the concept of signals in Unix and list the commonly used signal functions.
2. Write a C program to handle the SIGINT signal.
3. Describe the use of the kill and raise functions with examples.
4. Explain the difference between sigaction and sigprocmask functions.
5. Discuss the implementation of sigsuspend with an example.
6. Illustrate the use of alarm, pause, and nanosleep functions.
7. What are daemon processes? List their characteristics and significance.
8. Explain the concept of job-control signals in Unix.
9. Discuss the role of the system function in Unix programming.
10. Write a program to implement error logging for a daemon process.