

Question No. 1

- 1a) How can a process handle signals? (6) CO1

Question No. 2

- 2a) What is the difference between hard links and symbolic links in Linux? (6) CO2

Question No. 3

- 3a) What are the key components of a device driver, and what are their responsibilities? (8) CO3

OR

- 3b) Write a C program that simulates a low-level device driver, like a GPIO driver. Discuss how it interacts with higher layers in the kernel. (8) CO3

- 3c) Why are character drivers generally used for devices like terminals and serial ports, and how do they handle data transmission? (8) CO3

OR

- 3d) Write a program to demonstrate blocking and non-blocking I/O calls in a device driver. (8) CO3

Question No. 4

- 4a) What is the purpose of the `del_timer()` function, and why is it important to properly clean up timers? (8) CO4

OR

- 4b) Write a C program to create a kernel thread that handles work queued from an interrupt handler. Compare the implementation using a kernel thread with using a work queue. (8) CO4

- 4c) How does the ARM7 architecture allow for both vectorized and non-vectorized interrupt handling? (8) CO4

OR

- 4d) Write a kernel module that uses jiffies to measure time. Implement a delay loop in your module that waits for a specific number of jiffies and print a message before and after the delay. (8) CO4

Question No. 5

- 5a) How does the Linux kernel implement resource limiting for CPU, memory, and I/O? (8) CO5

OR

- 5b) What are the key differences between TCP and UDP sockets in terms of reliability, ordering, and connection management? (8) CO5

- 5c) What are namespaces in Linux, and how do they help with process isolation and resource control in containerized environments? (8) CO5

OR

- 5d) What are the key security considerations when designing client-server applications (e.g., encryption, authentication, input validation)? (8) CO5

..... **End of question paper**.....