**Step 1 – Knowledge Question (50-80 words)**

**Research Inter-Process Communication (IPC). Provide at least three different examples of techniques that you can use to implement IPC, and, in your own words, describe how each one of them works.**

Inter-process communication is a mechanism provided by the operating system that allows processes to communicate with each other and synchronise their actions. The processes can communicate with each other through shared memory and message passing.

The three examples of methods in IPC are pipes, message queuing and sockets.

Pipesallow the flow of data in one direction only. Data from the output is usually buffered until the input process receives it, which must have a common origin.

Message queueing allows messages to be passed between processes using either a single queue or several message queues. These messages are coordinated using an API.

Sockets are the most common method for communicating over a network between a client and a server. They allow for a standard connection that is computer and OS independent.

**Step 3 – Knowledge Question (30-60 words)**

**Operating System “signals” can be used as a form of sending commands to other processes. Describe, in your own words, what signals are, and which signals are available (this differs for each Operating System). Also, provide at least three examples how signals can be used in real-world applications.**

Signals are a component in operating systems that enable communication and control between processes and between a process and the operating system itself. Through signals, processes can interact and coordinate their actions, such as termination, interruption etc. Each operating system provides a set of signals, such as SIGINT, for keyboard interruptions. Signals are more limited in Windows compared to Unix-like operating systems. Windows uses a different mechanism called asynchronous procedure calls(APCs) for similar functionality. The signals available in Windows are SIGINT, SIGBREAK, SIGTERM and SIGKILL. The signals available in Linux and macOS are SIGTERM, SIGHUP, SIGKILL, SIGPIPE, and more.

Signals are used for process manipulation in shell scripting. When we type commands on the terminal and want to interrupt long-running commands or scripts, we can press control + c, which sends a SIGINT signal and terminates the execution.

Signals are used in web servers like Apache on Unix-like systems for process management. Sending a SIGHUP signal to a web server process triggers a configuration reload without interrupting ongoing connections. This allows administrators to update server configuration without restarting the server.

Database management systems often use signals for graceful shutdown and backup operations. Sending SIGTERM signal to a database server initiates a graceful shutdown, allowing active transactions to complete and data to be properly flushed to disk before shutting down.