**Lateral Self-Study**

1. Conduct a Google search on what the software development life cycle (SDLC) is and document your findings in a Google Word document.

The Software Development Life Cycle (SDLC) is a framework used to describe the processes involved in the development and maintenance of information systems and software, covering the stages necessary to create and maintain applications and systems efficiently.

Phases of the SDLC:

**Planning:** This crucial phase involves defining the goals and scope of the software or system to be developed. It also involves planning the budget, allocating resources, and establishing a project execution schedule. Proper planning is essential to ensure that project expectations match the final outcomes.

**Requirements Analysis:** At this stage, the needs of the end users are identified and evaluated to ensure that the system design fully meets these needs. This often requires meetings with stakeholders to agree on the specific features the software should have.

**Design:** Here, the identified requirements are transformed into a detailed design. The software architecture is defined, selecting technologies, data structures, processes, and interface protocols, in addition to developing models and prototypes.

**Implementation:** This phase involves the actual construction of the software. Programmers write the code based on the provided design, making this one of the most extensive phases of the SDLC.

**Testing:** After the software is developed, a series of tests are conducted to verify that there are no failures or errors. These include unit tests, integration tests, full system tests, and user acceptance tests.

**Deployment:** Once the software passes all tests, it is implemented in the production environment or launched to the market.

**Maintenance and Support:** After deployment, the software requires constant maintenance to address emerging problems and improve its functionality based on user feedback. This includes performing regular updates and optimizations.

**Retirement:** Over time, the system or software may be replaced by newer versions or completely retired. It is necessary to migrate data and processes as required.

2. Conduct another Google search and understand what a LAMP stack is.

The set of technologies known as the LAMP stack is essential for hosting websites and web applications, comprising Linux, Apache, MySQL, and PHP, which can be substituted with Python or Perl. Each component of the stack plays a vital role:

**Linux:** Acts as the underlying operating system, chosen for its stability, security, and effectiveness in server environments.

**Apache:** Is the web server responsible for managing and serving web pages, known for its high performance and configurability.

**MySQL:** Serves as the database management system, responsible for storing and managing essential data such as user information and page content. It is valued for its speed and simplicity.

**PHP:** This programming language, though interchangeable with Python or Perl, is preferred for developing server-side business logic.

The LAMP stack is widely adopted for its superior performance, cost-effectiveness, and versatility. The advantages include:

**Cost:** Its open-source nature significantly reduces development and maintenance costs.

**Community:** The components benefit from robust support thanks to their extensive developer communities.

**Flexibility:** Allows for specific customizations, easily adapting to the developer's needs.

**Compatibility:** Offers great compatibility with numerous tools and applications, facilitating integration.

Additionally, implementing LAMP in the cloud, using platforms like AWS, simplifies deployment through services such as Amazon EC2 for computing, Amazon RDS for managing MySQL databases, and AWS Elastic Beanstalk for application management, ensuring efficient and scalable infrastructure management.

3. Read about the 'chmod' and 'chown' commands in Linux and understand how file and directory access and ownership work.

**chmod Command:** This command is used to modify the permissions of files and directories. Permissions include read (r), which allows viewing the content of a file or the elements of a directory; write (w), which authorizes the alteration of the file content or the modification of the elements within a directory; and execute (x), which enables the execution of the file or access to the directory to perform operations. Permissions are assigned to three categories of users: the owner of the file, the group to which the file belongs, and other system users. Permissions can be defined using numeric or symbolic notation, such as 755 or rwxr-xr-x, respectively.

**chown Command:** This command changes the ownership of files or directories, adjusting the owner and/or the associated group.

4. Learn what TCP and UDP mean and how they differ. List the most commonly used ports on the web (http, https, ssh, telnet, ftp, sftp).

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are essential protocols in the transport layer of the OSI (Open Systems Interconnection) model. Both are used for sending data across the Internet, but they differ significantly in their communication approach.

**TCP (Transmission Control Protocol)**

**Connection-oriented:** Before transmitting data, TCP establishes a connection through a three-step "handshake" process between the sender and the receiver.

**Reliable:** Ensures complete and orderly delivery of packets. If packets are lost during transmission, TCP retransmits them.

**Flow and congestion control:** Implements mechanisms to regulate data flow and prevent network congestion, adjusting the transmission rate based on the receiver's capacity and network condition.

**UDP (User Datagram Protocol)**

**Connectionless:** UDP does not require establishing a prior connection, allowing immediate packet sending without a "handshake".

**Unreliable:** Does not ensure packet delivery, order, or data integrity. Packets may arrive out of order or be lost.

**Fast and efficient:** Due to the absence of connection management, retransmissions, and congestion control, UDP is lighter and faster, ideal for applications that require efficiency and speed, such as video streaming and online gaming.

**Common Web Ports**

Ports are numerical identifiers that facilitate packet routing in a network, distinguishing both the host and the service within it. Some of the most common ports in web applications are:

**HTTP (Hypertext Transfer Protocol):** Port 80, used for unencrypted web pages.

**HTTPS (HTTP Secure):** Port 443, encrypts communication between the browser and the server for enhanced security.

**SSH (Secure Shell):** Port 22, allows secure access to servers for remote administration.

**Telnet:** Port 23, used for remote server access, but without encryption, making it insecure.

**FTP (File Transfer Protocol):** Ports 20 (data transfer) and 21 (control).

**SFTP (SSH File Transfer Protocol):** Operates over SSH on port 22, providing a secure method for transferring files.













