**Programación sobre redes**

**Trabajo Práctico teórico**

**Consignas**

**1. ¿Qué es una red de computadoras y cuál es su propósito principal?**

**Red de computadoras:** Es un conjunto de dispositivos conectados entre sí que pueden comunicarse e intercambiar datos. Los dispositivos pueden ser computadoras, impresoras, servidores, routers, etc.

**Propósito principal:** Facilitar la comunicación y el intercambio de recursos (como archivos y dispositivos) entre los dispositivos conectados. Permite compartir información y servicios, como acceso a Internet, impresoras y archivos, y facilita la colaboración entre usuarios.

**2. Describe las diferencias entre una red local (LAN) y una red de área amplia (WAN).**

**LAN (Local Area Network):** Red que abarca una área geográfica limitada, como una oficina o un edificio. Ofrece alta velocidad y bajo costo en comparación con las WAN. Ejemplos incluyen redes domésticas o redes de oficinas.

**WAN (Wide Area Network):** Red que cubre un área geográfica extensa, como ciudades, países o incluso continentes. Las WAN suelen ser más lentas que las LAN y suelen utilizar tecnologías como enlaces de fibra óptica y satélites para conectar redes distantes.

**3. Explica la función de una tarjeta de red (NIC) en una computadora.**

**Tarjeta de Red (NIC - Network Interface Card):** Es un hardware que permite a la computadora conectarse a una red. Se encarga de la transmisión y recepción de datos en la red, así como del manejo de direcciones de hardware (MAC) y, a menudo, del control de errores.

**4. Describe el modelo OSI y sus siete capas. ¿Cuál es el propósito de cada capa?**

**-Capa 1: Física:** Transmisión de datos brutos a través del medio físico (cables, fibra óptica, etc.).

**-Capa 2: Enlace de Datos:** Establecimiento, mantenimiento y liberación de conexiones de datos, manejo de errores y control de flujo en un enlace físico.

**-Capa 3: Red:** Encaminamiento de datos a través de la red, determinando la mejor ruta desde el origen hasta el destino.

**-Capa 4: Transporte:** Garantiza la transferencia completa y correcta de datos entre dos puntos finales. Maneja la segmentación, el reensamblaje y el control de errores.

**-Capa 5: Sesión:** Establece, gestiona y termina las sesiones de comunicación entre aplicaciones.

**-Capa 6: Presentación:** Traduce, encripta y comprime los datos para que puedan ser entendidos por la capa de aplicación.

**-Capa 7: Aplicación:** Interactúa directamente con el software de aplicación para proporcionar servicios de red a los usuarios.

**5. Explica cómo se comunican dos dispositivos utilizando el modelo OSI.**

**Proceso de comunicación:**

**1. Capa de Aplicación:** La aplicación en el dispositivo de origen genera los datos que se desean enviar.

**2. Capa de Presentación**: Los datos son formateados, codificados y encriptados si es necesario.

**3. Capa de Sesión:** Establece y mantiene la conexión con el dispositivo receptor.

**4. Capa de Transporte: Segmenta los datos en paquetes y proporciona control de flujo y corrección de errores.**

**5. Capa de Red:** Encamina los paquetes a través de la red usando direcciones IP.

**6. Capa de Enlace de Datos:** Enmarca los paquetes para su transmisión en el enlace físico.

**7. Capa Física:** Transmite los datos como señales eléctricas, ópticas o de radio a través del medio físico.

El proceso inverso ocurre en el dispositivo receptor, donde los datos se desenvuelven capa por capa hasta llegar a la aplicación destino.

**6. Describe el modelo TCP/IP y sus capas. ¿En qué se diferencia del modelo OSI?**

* **Modelo TCP/IP:**

**1. Capa de Aplicación:** Proporciona servicios de red directamente a las aplicaciones (equivalente a las capas de Aplicación, Presentación y Sesión del modelo OSI).

**2. Capa de Transporte:** Asegura la entrega de datos a través de protocolos como TCP y UDP (equivalente a la capa de Transporte del modelo OSI).

**3. Capa de Internet:** Maneja el enrutamiento y la entrega de paquetes a través de redes, utilizando IP (equivalente a la capa de Red del modelo OSI).

**4. Capa de Enlace:** Gestiona la comunicación en un enlace físico, incluyendo direccionamiento MAC y control de acceso al medio (equivalente a la capa de Enlace de Datos del modelo OSI).

* **Diferencias con el modelo OSI:**

El modelo TCP/IP es más simplificado, con menos capas. Combina las funciones de varias capas del modelo OSI en una sola capa de Aplicación.

TCP/IP está basado en un enfoque práctico y se utiliza ampliamente en Internet, mientras que el modelo OSI es más teórico.

**7. Explica la función de la capa de aplicación en el modelo TCP/IP. ¿Qué protocolos trabajan en esta capa?**

**-Función:** La capa de aplicación proporciona servicios de red directamente a las aplicaciones del usuario, como correo electrónico, transferencia de archivos y navegación web.

Protocolos en esta capa:

**HTTP/HTTPS:** Para la navegación web.

**FTP:** Para la transferencia de archivos.

**SMTP/POP3/IMAP:** Para el correo electrónico.

**DNS:** Para la resolución de nombres de dominio.

**8. Describe la capa de transporte en el modelo TCP/IP. ¿Cuáles son los dos principales protocolos en esta capa y en qué se diferencian?**

**Capa de Transporte:** Asegura la entrega correcta y completa de datos entre los dispositivos.

Protocolos principales:

**TCP (Transmission Control Protocol)**: Proporciona una conexión orientada, garantiza la entrega de datos con control de errores y reensamblaje de paquetes. Ideal para aplicaciones que requieren fiabilidad, como la transferencia de archivos o la navegación web.

**UDP (User Datagram Protocol)**: Proporciona una comunicación sin conexión, sin garantías de entrega. Es más rápido y se utiliza para aplicaciones donde la velocidad es crucial y la pérdida de algunos paquetes es aceptable, como en streaming de video o juegos en línea.

**9. Explica la función de la capa de red en el modelo TCP/IP. ¿Cuál es el protocolo principal en esta capa?**

-**Función:** La capa de red es responsable del enrutamiento de los paquetes a través de la red y del direccionamiento de los mismos, determinando la mejor ruta desde el origen hasta el destino.

**Protocolo principal: IP (Internet Protocol):** Encargado de direccionar y enrutar los paquetes de datos a través de la red.

**10. Describe la función de la capa de enlace de datos en el modelo TCP/IP.**

**Función:** La capa de enlace de datos proporciona la comunicación en el enlace físico entre dispositivos de una red local. Se encarga del direccionamiento a nivel de enlace (MAC), el control de acceso al medio y el manejo de errores en la transmisión de datos.

**11. ¿Qué es una dirección IP y por qué es importante en la comunicación de redes?**

**Dirección IP:** Es una dirección única asignada a cada dispositivo en una red que permite su identificación y comunicación con otros dispositivos.

**Importancia:** Permite que los datos sean enviados al dispositivo correcto en una red, ya sea a través de una red local o en Internet, y facilita el enrutamiento de la información a través de diferentes redes.

**12. Explica la diferencia entre una dirección IP estática y una dinámica.**

**Estática:** Una dirección IP que no cambia y es asignada permanentemente a un dispositivo. Es útil para servidores y dispositivos que necesitan una dirección fija.

**Dinámica:** Una dirección IP que es asignada temporalmente por un servidor DHCP. Puede cambiar cada vez que el dispositivo se conecta a la red, y es común en redes domésticas y la mayoría de las redes empresariales.

**13. ¿Qué es una máscara de subred y cómo se utiliza en una red?**

**Máscara de subred:** Es una secuencia de bits que divide una dirección IP en una parte de red y una parte de host. Permite identificar qué parte de la dirección IP corresponde a la red y qué parte a los dispositivos individuales dentro de esa red.

**Uso:** Facilita la organización de direcciones IP en subredes, mejora la administración y el enrutamiento dentro de una red.

**14. Explica el proceso de encapsulación de datos desde la capa de aplicación hasta la capa física en el modelo OSI.**

**Aplicación:** Los datos son generados por la aplicación.

**Presentación:** Los datos son formateados, codificados y encriptados si es necesario.

**Sesión:** Se establece una conexión y se gestionan los datos.

**Transporte:** Los datos se segmentan en paquetes y se les añaden cabeceras para control de errores y secuencias.

**Red:** Los paquetes son encapsulados en paquetes IP y se les asignan direcciones IP.

**Capa de Enlace de Datos:** Los paquetes se encapsulan en tramas con direcciones físicas (MAC) para la transmisión en una red local.

**Capa Física:** Las tramas se convierten en señales eléctricas, ópticas o de radio para la transmisión física a través del medio.

**15. ¿Qué es la concurrencia en programación y por qué es importante en el contexto de redes?**

**Concurrencia:** Se refiere a la capacidad de un sistema para manejar múltiples tareas o procesos al mismo tiempo. En programación, esto se logra mediante la ejecución de múltiples hilos (threads) o procesos simultáneamente.

**Importancia en redes:** En el contexto de redes, la concurrencia es crucial para manejar múltiples conexiones de red al mismo tiempo. Permite a los servidores y aplicaciones procesar varias solicitudes simultáneamente, mejorando la eficiencia y la capacidad de respuesta del sistema.

**16. Explica qué es un thread en programación y cómo se utiliza para manejar la concurrencia.**

**Thread (Hilo):** Es una unidad de ejecución dentro de un proceso. Los hilos dentro del mismo proceso comparten el mismo espacio de direcciones y recursos, pero se ejecutan de manera independiente.

**Uso para manejar la concurrencia:** Los hilos permiten a una aplicación realizar múltiples tareas simultáneamente, como manejar varias conexiones de red o realizar cálculos en paralelo. Esto se traduce en una mayor eficiencia y mejor rendimiento, ya que se pueden realizar múltiples operaciones al mismo tiempo en lugar de una tras otra.

**17. ¿Qué es la sincronización de threads y por qué es importante?**

**Sincronización de threads:** Es el proceso de coordinar el acceso a recursos compartidos por múltiples hilos para evitar conflictos y asegurar que los datos no se corrompan. Esto se realiza utilizando mecanismos como bloqueos (locks), semáforos y monitores.

**Importancia:** La sincronización es crucial para evitar problemas como condiciones de carrera, donde varios hilos intentan modificar los mismos datos simultáneamente, lo que puede llevar a resultados inesperados o incorrectos. Sin una adecuada sincronización, los hilos pueden interferir entre sí, causando errores y comportamientos erráticos en la aplicación.

**18. Explica las diferencias entre procesos y threads.**

* Procesos:

- **Son instancias independientes de un programa en ejecución.**

**- Tienen su propio espacio de direcciones de memoria y recursos.**

**- La comunicación entre procesos (IPC) suele ser más costosa en términos de rendimiento.**

**- La creación y gestión de procesos tiene más sobrecarga comparada con los hilos.**

**- Threads (Hilos):**

**- Son unidades de ejecución dentro de un proceso.**

**- Comparten el mismo espacio de direcciones y recursos del proceso.**

**- La comunicación entre hilos (dentro del mismo proceso) es más rápida y sencilla.**

**- Los hilos tienen menos sobrecarga y son más ligeros que los procesos.**

**19. Describe una situación en la que utilizarías threads para mejorar el rendimiento de una aplicación en red.**

**Situación:** Imagina un servidor web que recibe solicitudes HTTP de múltiples clientes simultáneamente.

**Uso de Threads:** Utilizar hilos para manejar cada solicitud de cliente de forma concurrente permite que el servidor procese varias solicitudes al mismo tiempo. Esto mejora significativamente el rendimiento y la capacidad de respuesta del servidor, ya que cada solicitud puede ser manejada de manera independiente sin esperar a que se completen otras solicitudes.

**20. Investiga y describe brevemente un protocolo de red moderno (como HTTP/2, WebSockets, gRPC) y su importancia en la programación en red.**

* **Protocolo: HTTP/2**

Descripción: HTTP/2 es una versión mejorada del protocolo HTTP/1.1, diseñado para mejorar la eficiencia y el rendimiento en la transferencia de datos web. Introduce características como multiplexión de solicitudes (enviar múltiples solicitudes a través de una única conexión TCP), compresión de encabezados y un enfoque binario en lugar de textual.

**Importancia:** HTTP/2 reduce la latencia y mejora el rendimiento general de las aplicaciones web al permitir la transferencia paralela de múltiples archivos a través de una sola conexión, lo que resulta en tiempos de carga más rápidos y una mejor experiencia de usuario.

* **Protocolo: WebSockets**

**Descripción**: WebSockets es un protocolo que proporciona una comunicación bidireccional y persistente entre el cliente y el servidor sobre una única conexión TCP. Permite la transferencia de datos en tiempo real con menor sobrecarga en comparación con HTTP.

Importancia: WebSockets es ideal para aplicaciones que requieren actualizaciones en tiempo real, como chats en línea, juegos multijugador y aplicaciones de colaboración en tiempo real. Permite la transmisión continua de datos sin la necesidad de volver a establecer conexiones repetidamente.

* **Protocolo: gRPC**

**Descripción:** gRPC (gRPC Remote Procedure Calls) es un marco de trabajo para la comunicación entre servicios basado en HTTP/2 y Protocol Buffers. Facilita la creación de servicios distribuidos y la comunicación entre ellos utilizando RPC (llamadas a procedimientos remotos).

**Importancia:** gRPC permite la implementación eficiente de microservicios y la comunicación entre servicios con bajo retraso y alta eficiencia. Soporta múltiples lenguajes de programación y proporciona características como autenticación, compresión y control de versiones de API, lo que lo hace adecuado para sistemas distribuidos y aplicaciones en la nube.