## INFORMÁTICA II - 2023

## TRABAJO PRÁCTICO sobre SISTEMAS OPERATIVOS Y REDES

Para cada consigna, Ud. debe resolverla en la terminal de Ubuntu de las máquinas del laboratorio LIE.

Debe copiar y pegar los comandos y su resultado como solución a cada ejercicio. También puede adjuntar captura de pantalla.

### EJERCICIO 1

1.a) Dibuje o estructure un árbol de directorios de al menos tres (3) niveles de profundidad. Dentro de cada directorio debe haber al menos dos directorios y dos archivos.

1.b) Crear dicho árbol utilizando la terminal en el directorio home del usuario que esté utilizando. Use los comandos mkdir, cd y touch.

Copie y pegue todos los comandos y salida de la terminal

### EJERCICIO 2

Desde el directorio home, diríjase a modificar los atributos de uno de los archivos a sólo lectura. Verificar lo que sucede si se intenta realizar cambios. Restablecer los atributos anteriores al mismo archivo. Liste abajo el listado de comandos utilizados y su salida (copiar el texto desde la consola).

### EJERCICIO 3

Copiar el siguiente código fuente a un archivo llamado client.c en la carpeta Descargas. Use gedit.

#include <netinet/in.h> //structure for storing address information

#include <stdio.h>

#include <stdlib.h>

#include <sys/socket.h> //for socket APIs

#include <sys/types.h>

int main(int argc, char const\* argv[])

{

int sockD = socket(AF\_INET, SOCK\_STREAM, 0);

struct sockaddr\_in servAddr;

servAddr.sin\_family = AF\_INET;

servAddr.sin\_port

= htons(9001); // use some unused port number

servAddr.sin\_addr.s\_addr = INADDR\_ANY;

int connectStatus

= connect(sockD, (struct sockaddr\*)&servAddr,

sizeof(servAddr));

if (connectStatus == -1) {

printf("Error...\n");

}

else {

char strData[255];

recv(sockD, strData, sizeof(strData), 0);

printf("Message: %s\n", strData);

}

return 0;

}

Copiar el siguiente código fuente a un archivo llamado server.c en la carpeta Descargas. Use gedit.

#include <netinet/in.h> //structure for storing address information

#include <stdio.h>

#include <stdlib.h>

#include <sys/socket.h> //for socket APIs

#include <sys/types.h>

int main(int argc, char const\* argv[])

{

// create server socket similar to what was done in

// client program

int servSockD = socket(AF\_INET, SOCK\_STREAM, 0);

// string store data to send to client

char serMsg[255] = "Message from the server to the "

"client \'Hello Client\' ";

// define server address

struct sockaddr\_in servAddr;

servAddr.sin\_family = AF\_INET;

servAddr.sin\_port = htons(9001);

servAddr.sin\_addr.s\_addr = INADDR\_ANY;

// bind socket to the specified IP and port

bind(servSockD, (struct sockaddr\*)&servAddr,

sizeof(servAddr));

// listen for connections

listen(servSockD, 1);

// integer to hold client socket.

int clientSocket = accept(servSockD, NULL, NULL);

// send's messages to client socket

send(clientSocket, serMsg, sizeof(serMsg), 0);

return 0;

}

Origen del código fuente: <https://www.geeksforgeeks.org/simple-client-server-application-in-c/>

Mueva ambos archivos .c a alguna carpeta en el tercer nivel de directorios que creó en el punto 1. Use en consola el comando mv. Copie y pegue todos los comandos y salida de la terminal

### EJERCICIO 4

Compila el código fuente que has descargado en el ejercicio anterior usando Zinjai en Ubuntu o gcc en la consola. Sólo debes compilarlo, no ejecutarlo.

Si te surge algún error de compilación, indica cuál es. Comenta por qué crees que ha surgido. Lista tu solución.

### EJERCICIO 5

Muestra los detalles de la PC linux que estás utilizando usando el comando uname y copiando la Configuración.

### EJERCICIO 6

Iniciar el programa de calculadora desde consola, obtener el número de PID (Process IDentification) y con éste terminar (matar) el proceso. Abajo indique la secuencia de comandos utilizados.

### EJERCICIO 7

Determinar cuál es la actual configuración de Red para la Conexión Ethernet y de la Red Inalámbrica (en caso de tener conectividad WiFI).  
Determinar: MAC Address, Dirección IP, máscara subred, puerta de enlace. Especificar como se hizo y poner los resultados (Texto y/o Imagen)

### EJERCICIO 8

Comparar el tiempo de respuesta de un ping a:

* IP local (PC). (127.0.0.1 o localhost)
* [www.](http://www./)utn.edu.ar
* www.google.com

Indique abajo tu respuesta al ejercicio

### EJERCICIO 9

Obtener la IP de “www.google.com”. Ingresar la IP obtenida en un navegador de internet, ¿cuál es el resultado?

### 

### EJERCICIO 10

**Realizar una conexión de red entre dos computadoras. Verificar conexión entre ambas realizando comandos ping desde ambos extremos. Probar con un programa conectividad entre ambas PCs.**

**En linux:**

**Server: nc -l -p 1234,**

**Cliente: nc localhost 1234, deberá cambiar IP del Server (allí usa localhost) por la correcta para probar entre distintas PCs.**

**Pruebe el programa cliente servidor, el código fuente descargado en el ejercicio 3. ¿Haría alguna modificación o agregado al código? Indique y agregue su modificación.**

### EJERCICIO 11