

Práctica 2: Sockets

Laura Itzel Rodríguez Dimayuga

24 de agosto de 2025

1. Introducción

1. Instalar Sudo

```
1  # nos cambiamos al root
2  $ su --login
3  Password:
4  # instalamos sudo
5  $ apt install sudo
6  # anadimos el usuario laudima al grupo sudo
7  $ adduser laudima sudo
8
```

2. Ahora en ambas maquinas instale el paquete openssh-server y verificamos la ip

```
1  $ apt install openssh-server
2  $ ip a
3
```

3. En ambas máquinas habilite el ssh. Y cheque su estatus.

```
1  $ sudo systemctl enable ssh
2  $ sudo systemctl start ssh
3  $ sudo sys
4  temctl status ssh
5
```

4. Em ambas maquinas instale los paquetes para poder correr C, con

```
1  $ sudo apt install build-essential
2  $ gcc --version
3
```

5. En mi máquina de ubuntu escribí el archivo `server.c` y descargué el archivo `cipherworlds.txt` desde el repositorio.

6. En mi máquina de debian escribí el archivo `client.c`.

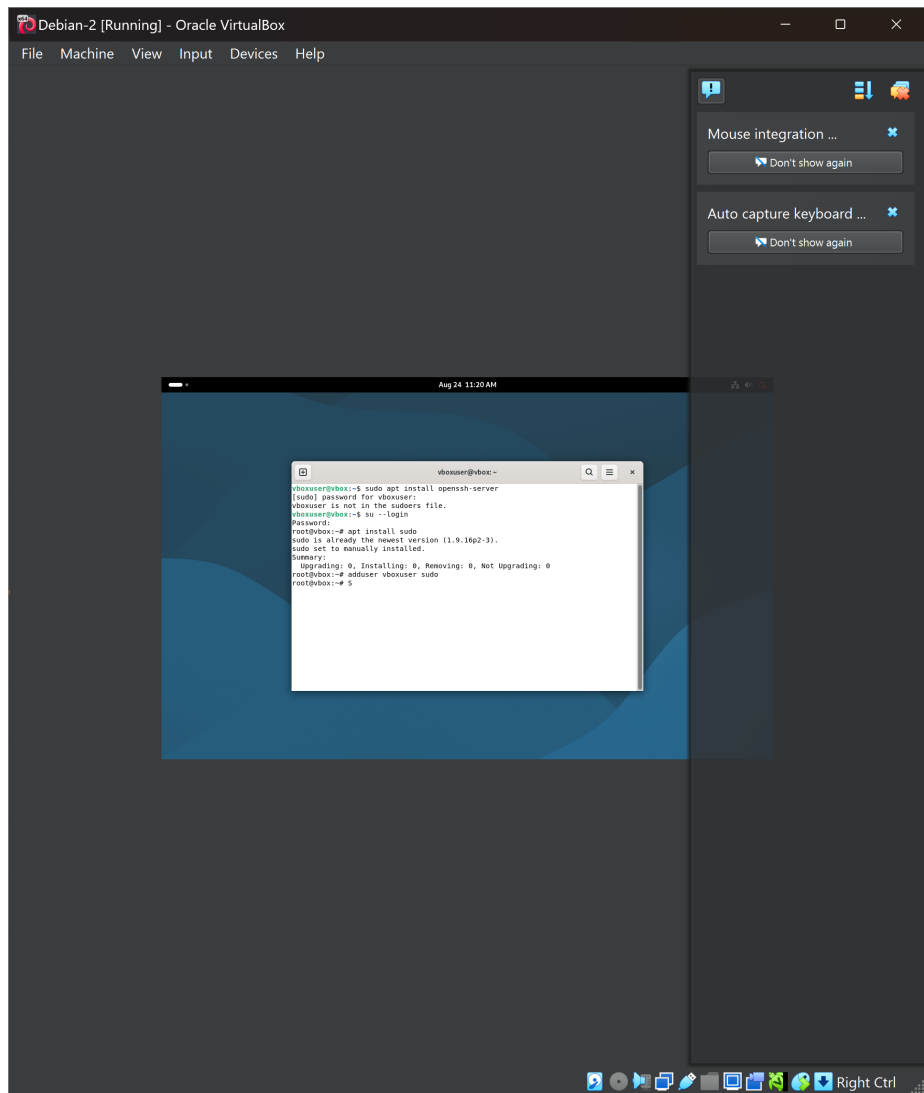


Figura 1: Instalación de Sudo

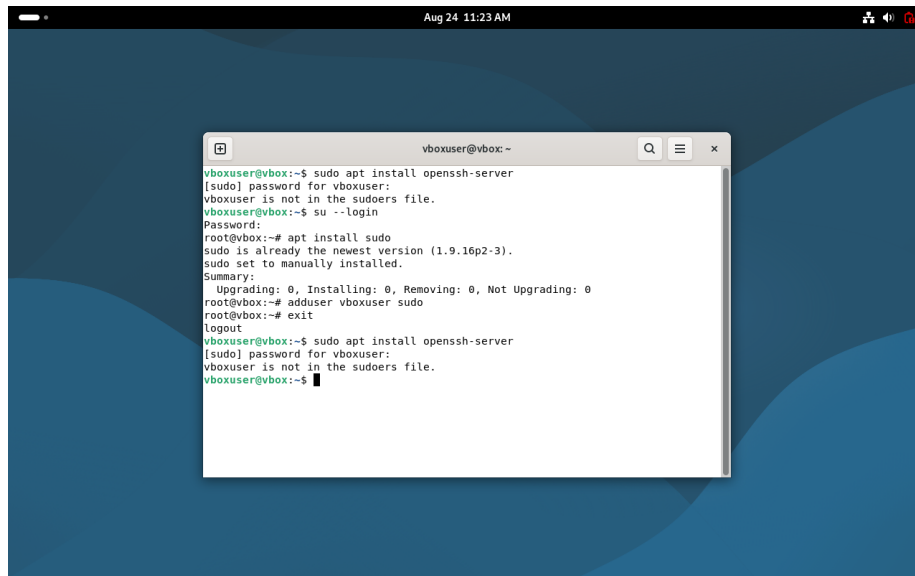


Figura 2: Instalación de OpenSSH

7. En ambas máquinas compilé los archivos con el siguiente comando:

```
1 $ gcc -o server server.c
2 $ gcc -o client client.c
3
```

8. Verifique que el puerto 7006 esté abierto.

9. En la máquina de debian corré el siguiente comando:

```
1 $ ./client <key> <shift>
2
```

Y pude lograr decifrar el mensaje

2. Teoría

- ¿Para qué sirven las bibliotecas `arpa-inet.h`, `sys-socket.h`, `netinet-in.h`, `unistd.h`, `ctype.h`?
 - `arpa-inet.h`: definiciones para operaciones de Internet
 - `sys-socket.h`: Contiene definiciones relacionadas con la creación y manejo de sockets.
 - `netinet-in.h`: Define las estructuras de datos y funciones específicas para el manejo de direcciones IP y protocolos de red.

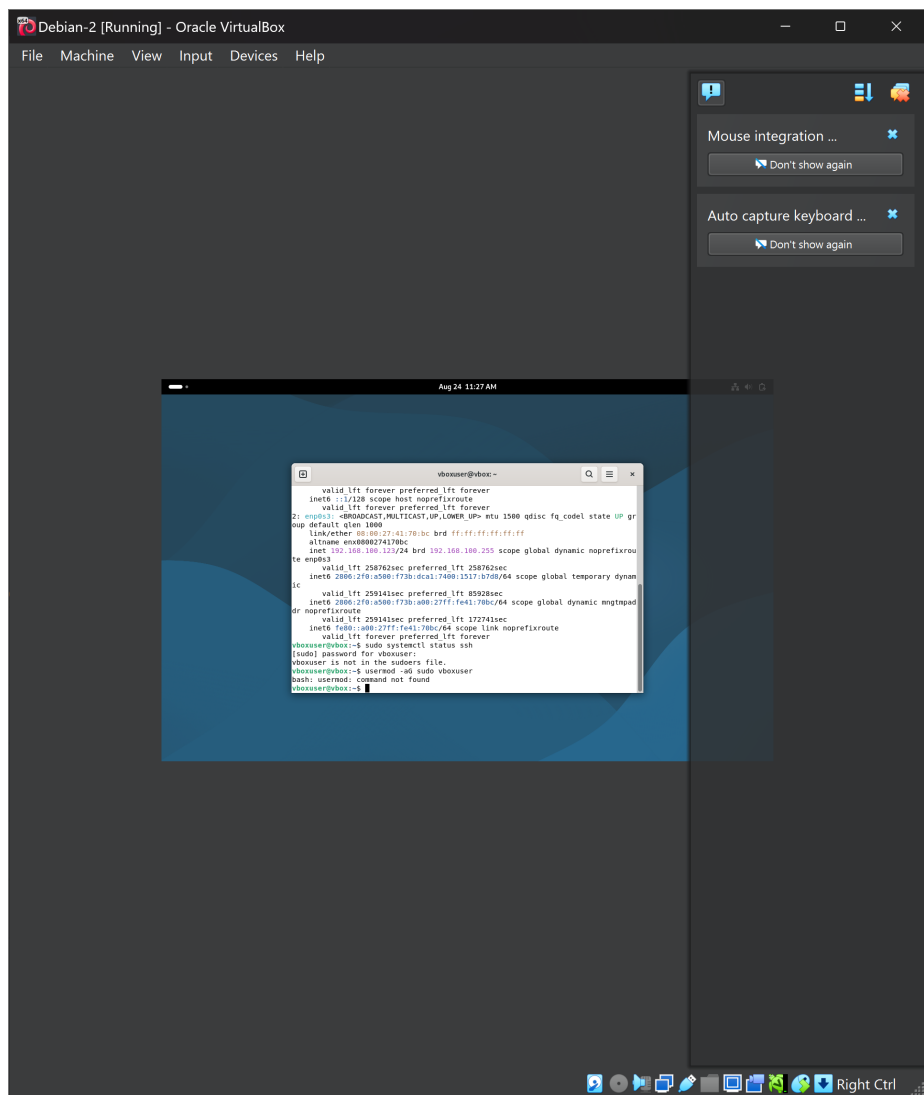


Figura 3: Verificación de IP

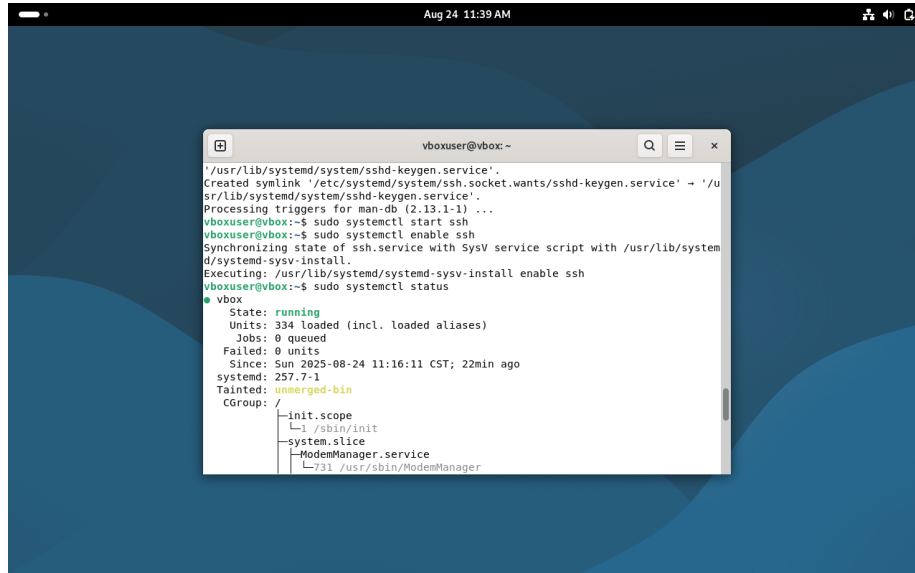


Figura 4: Estado del servicio SSH en Debian

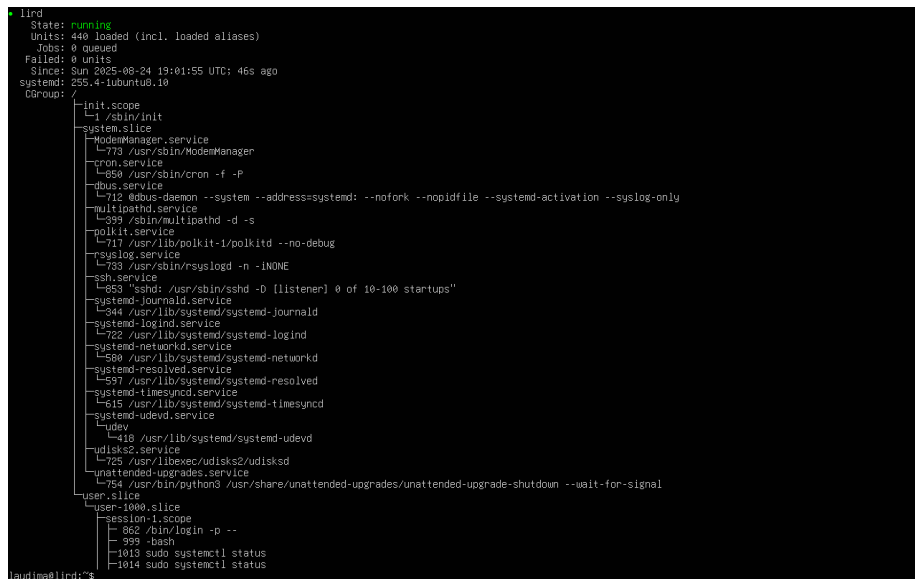


Figura 5: Estado del servicio SSH en Ubuntu

```
◀ ▶ ubunl × debi × host ▶ + ▼ - □ ×

To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug 22 21:32:11 2025 from 10.18.179.225
laudima@lird:~$ ls
sockets
laudima@lird:~$ cd sockets/
laudima@lird:~/sockets$ ls
client client.c server server.c
laudima@lird:~/sockets$ wget https://github.com/JohnKun136NVCP/
RedesDatosFC/blob/main/P2/cipherworlds.txt
--2025-08-24 17:51:07-- https://github.com/JohnKun136NVCP/Rede
sDatosFC/blob/main/P2/cipherworlds.txt
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'cipherworlds.txt'

cipherworlds.txt      [ <=>          ] 180.46K  712KB/s
                    in 0.3s

2025-08-24 17:51:07 (712 KB/s) - 'cipherworlds.txt' saved [1847
91]

laudima@lird:~/sockets$ ls
cipherworlds.txt client client.c server server.c
laudima@lird:~/sockets$ gcc server.c -o server
laudima@lird:~/sockets$ ./server
[+] Server listening port 7006...
```

Figura 6: Verificación de puerto en Ubuntu

```
debi x host x ubu x + v - □ x

Install the latest PowerShell for new features and improvements
! https://aka.ms/PSWindows

PS C:\Users\lr255029> ssh -l laudima 192.168.100.122
laudima@192.168.100.122's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Aug 24 06:00:01 PM UTC 2025

System load:            0.0
Usage of /:              39.7% of 14.66GB
Memory usage:           6%
Swap usage:             0%
Processes:              147
Users logged in:        1
IPv4 address for enp0s3: 192.168.100.122
IPv6 address for enp0s3: 2806:2f0:a500:f73b:a00:27ff:fef6:5308

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Aug 24 17:46:47 2025 from 192.168.100.103
laudima@lird:~$ netstat -tuln | grep "7006"
tcp        0      0 0.0.0.0:7006          0.0.0.0:*
           LISTEN
laudima@lird:~$ |
```

Figura 7: Verificación de puerto en ubuntu pero desde otra terminal

```

vboxuser@vbox:~/sockets x audima@bird:~/sockets x Windows PowerShell x + v x
command 'kservr' from deb freeem-kserver (1.1.1-a021cvs20130302-7build1)
Try: sudo apt install <deb name>
audima@bird:~/sockets$ ./server
[*] Server listening port 7000...
[*] Client connected
[*] [Server] Encrypted key obtained: lggnatqnn
[*] File opened successfully
[*] Word not found: vcfbdwds
[*] Word not found: scpfkf
[*] Word not found: nnczrffs
[*] Word not found: wylvmrgyqfmx
[*] Word not found: xauvhvhdxs
[*] Word not found: lajldq
[*] Word not found: gszirxmsrepmmq
[*] Word not found: qnlafpnlgyrgnl
[*] Word not found: yljngac
[*] Word not found: dragons
[*] Word not found: gswljgrk
[*] Word not found: oicl
[*] Word not found: rdasmuf
[*] Word not found: ashrrzllkxway
[*] Word not found: passia
[*] Word not found: fdqotnzs
[*] Word not found: eljita
[*] Word not found: xertqvlbuz
[*] Word not found: zwtqkibqov
[*] Word not found: alscpmqpc
[*] Word not found: jcwvpkpl
[*] Word not found: klccmacyn
[*] Word not found: lndecj
[*] Word not found: pilehivs
[*] Word not found: eljfygyafy
[*] Word not found: umpfmgqc
[*] Word not found: salubrious
[*] Word not found: lqzpyzqgr
[*] Word not found: noobe
[*] Word not found: end
[*] Word not found: ignrantrc
[*] Word not found: spskc
[*] Word not found: onskogmd
[*] Word not found: qmawrota
[*] Word not found: iqhyisyq
[*] Word not found: saylndq
[*] Word not found: acctophenridin
[*] Word not found: alieno
[*] Word not found: gysmndirv
[*] Word not found: dbmdklsvdi
[*] [Server] Key decrypted: jellyroll
[*] Sent file
audima@bird:~/sockets$

```

Figura 8: Servidor en ejecución

```

vboxuser@vbox:~/sockets x audima@bird:~/sockets x Windows PowerShell x + v x
vboxuser@vbox:~/sockets$ ./client
Type: ./client <key> <shift>
vboxuser@vbox:~/sockets$ ./client lggnatqnn 28
[*] Starting client...
[*] Key: lggnatqnn
[*] Shift: 28
[*] Client ready to send data...
[*] Socket created successfully
[*] Connected to server
[*] [Client] Key and shift was sent: lggnatqnn 28
[*] [Client] Server message: ACCESS GRANTED
[*] [Client] The file was save like 'info.txt'
vboxuser@vbox:~/sockets$ ls
cipherworlds.txt client client.c info.txt server.c
vboxuser@vbox:~/sockets$ cat info.txt
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp8s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f6:53:88 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.122/24 metric 100 brd 192.168.100.255 scope global dynamic enp8s3
        valid_lft 259085sec preferred_lft 259085sec
    inet6 2806:2f0:a500:f73b:a80:27ff:fe6:5308/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 259025sec preferred_lft 17625sec
    inet6 fe80:a80:27ff:fe6:5308/64 scope link
        valid_lft forever preferred_lft forever
vboxuser@vbox:~/sockets$

```

Figura 9: Cliente en ejecución

- `unistd.h`: Proporciona acceso a las llamadas al sistema POSIX, incluyendo funciones para manejar archivos y procesos.
 - `ctype.h`: Contiene funciones para la manipulación de caracteres, como la verificación de tipos de caracteres (por ejemplo, si es un dígito o una letra).
2. ¿Qué tipo de Socket se creó en esta práctica? ¿UDP, TCP? Si es alguno de esos dos o los dos explica por qué.
- Como ocupamos `SOCK_STREAM` se creó un socket TCP, ya que este tipo de socket proporciona una conexión orientada a la conexión y garantiza la entrega de datos en el orden correcto.
3. ¿Qué hacen `fgets` y `sizeof` ?
- `fgets`: Lee una línea de texto de un flujo de entrada (como un archivo o la entrada estándar) y la almacena en un buffer. Asegura que no se lea más allá del final del buffer, lo que ayuda a prevenir desbordamientos de buffer.
 - `sizeof`: Devuelve el tamaño en bytes de un tipo de dato o de una variable. Es útil para determinar la cantidad de memoria que se necesita para almacenar un tipo de dato específico.
4. ¿Cuál es la diferencia entre `sscanf` y `scanf`?
- `scanf`: Lee datos de la entrada estándar (teclado) y los almacena en variables. Es útil para leer datos de forma interactiva.
 - `sscanf`: Lee datos de una cadena de caracteres en lugar de la entrada estándar. Permite analizar datos que ya están en memoria, lo que puede ser útil para procesar cadenas formateadas. En el caso de la práctica, se utilizó `sscanf` para extraer los valores de clave y desplazamiento de la cadena de entrada.

5. Intenta explicar lo que está pasando en el cliente.

Primero valida que se le proporcionen exactamente dos argumentos (clave y desplazamiento), luego crea un socket TCP y se conecta al servidor Ubuntu en el puerto 7006. Una vez establecida la conexión, manda la clave y el desplazamiento al servidor, y espera recibir una respuesta de autorización.

Si el servidor responde con `ACCESS GRANTED` (porque la clave descifrada coincide con alguna palabra en el archivo `cipherworlds.txt`), el cliente procede a recibir un archivo completo con información de red del servidor y lo guarda localmente como `info.txt`; si recibe `ACCESS DENIED`, simplemente termina la conexión. En resumen, el cliente debe demostrar conocer una palabra secreta cifrada para acceder a datos del sistema servidor.

6. ¿Cuál fue tu palabra secreta?

jellyroll