LABORATORY 01 - BASE PLATFORM

Laboratory 1 corresponding to the first term

Santiago Amaya Zapata

Julian Camilo Tinjacá Corredor

Escuela Colombiana de Ingeniería Julio Garavito

NETWORK ARCHITECTURE AND SERVICES

GROUP 3

Bogotá DC, Colombia

2025 - 2

Lab No.01 - Base Platform

Objective

- Install and configure different distributions of Unix and Windows operating systems as part of platform setup.

- Become familiar with the use of virtualization software.

Tools to be used

- Items provided by the Computer Laboratory:
  - Computers
  - Internet access
  - Virtualization software
- Items students must bring:
  - Some operating system images.
  - A USB flash drive or an external hard drive with approximately 128 GB of storage (per group).

Introduction

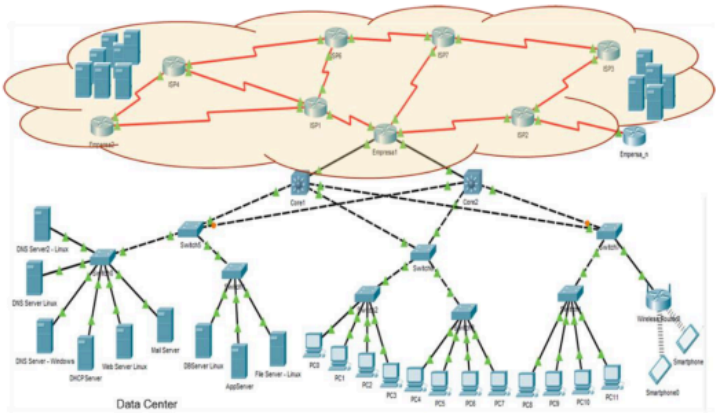
This lab report covers various operating system configurations and installations, including Slackware, Windows Server 2025, Android, and Solaris, all using virtualization software in VMware Workstation Pro. The purpose of this laboratory is not only to practice the installation process of different operating systems, but also to analyze their structure, management tools, and network behavior within a controlled virtualized environment.

The activities carried out throughout the lab provide a practical approach to essential concepts in system administration, such as the use of hypervisors, the configuration of IP addresses, gateways, and DNS services, as well as the comparison between bridge and NAT network modes. Additionally, the exercise emphasizes user and group management, file system structures, permission handling, and system logging, which are fundamental for understanding how modern operating systems ensure security, organization, and resource control.

Another important component of this laboratory is the comparison between Unix-based systems and Windows Server environments. By working with both families of operating

systems, students gain insight into their similarities and differences regarding directory structures, configuration files, registry mechanisms, and permission models. This comparative perspective is complemented by the exploration of containers, cloud computing, and virtualization costs, which connects traditional system administration with modern computing paradigms.

Finally, the lab also includes the execution of basic shell commands across different environments, reinforcing the importance of the command line interface as a powerful tool for system configuration, troubleshooting, and automation. Together, these activities contribute to a deeper understanding of operating systems in practice, bridging the gap between theoretical concepts and real-world administration scenarios.



Theoretical Framework

The foundation of this laboratory lies in the study of operating systems and virtualization, two core concepts in modern computing. An operating system (OS) is the fundamental software that manages hardware resources and provides services for applications, ensuring multitasking, process scheduling, memory allocation, and user interaction. Different operating systems, such as Linux-based distributions, Solaris, Android, and Windows Server, implement these functions in distinct ways, which makes their comparison relevant for both academic and professional contexts.

Virtualization plays a key role in this laboratory, as it allows multiple operating systems to run concurrently on a single physical machine through the use of hypervisors. Hypervisors are software or hardware platforms that abstract and allocate resources to virtual machines, enabling efficient testing and deployment of diverse systems. They are generally classified into two types: Type 1 (bare-metal), which runs directly on the hardware, and Type 2 (hosted), which runs on top of another operating system. VMware Workstation Pro, used in

this lab, is an example of a Type 2 hypervisor. Virtualization not only reduces costs and hardware needs, but also facilitates isolation, portability, and scalability, features that are also fundamental to cloud computing.

Networking concepts are another essential theoretical component. For an operating system to communicate effectively within a network, it requires correct configuration of its IP address, subnet mask, gateway, and DNS server. Modes such as Bridge and NAT in virtualization define how virtual machines interact with the external network, either by acting as independent devices or by sharing the host's IP address. Proper network configuration is necessary to ensure connectivity, name resolution, and communication between systems in the laboratory.

In addition, system administration relies heavily on user and group management, file system structures, and permission models. Unix-based systems (Linux and Solaris) employ a hierarchical file system, where configuration files, binaries, and log files are organized into standard directories. Permissions in these systems are managed through a combination of user, group, and others, with both symbolic and numeric representations. Windows Server, by contrast, employs the NTFS file system, the Windows Registry, and a graphical interface for managing users and permissions, offering a different perspective on access control and system configuration.

Finally, the use of the command line interface (CLI) across different operating systems is a central theoretical concept. The shell provides direct interaction with the system, enabling tasks such as file manipulation, process monitoring, and text processing. While Unix-based systems rely on shells like Bash or KornShell, Windows provides PowerShell and the Command Prompt. Understanding the similarities and differences between these environments reinforces not only technical proficiency but also adaptability when working across heterogeneous systems.

Experiments

---

1. Virtualization Software

> Record a video of up to 5 minutes addressing this topic. All team members submitting the report must participate. You may use a presentation, document, or animation to explain the subject.
> - What are hypervisors?
> - How are they classified?
> - What are their characteristics?
> - Explain their architecture.

- What is cloud computing?
- Are hypervisors used in cloud computing? Justify your answer.
- What is the cost difference between a physical server and a cloud-based server?
- What are containers? Explain their architecture.
- What are the similarities and differences between virtual machines and containers?

Visit the following link to watch the video:
https://youtu.be/zk4LI5QbA_Y

And you can see the presentation here
VirtualizationSoftware

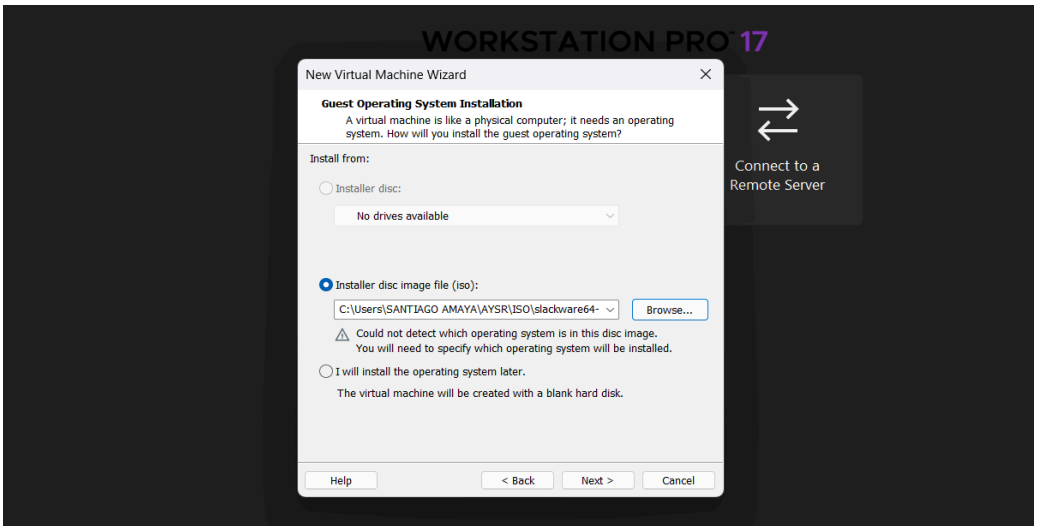
2. Unix-Based Server Setup

a) Server installation and configuration

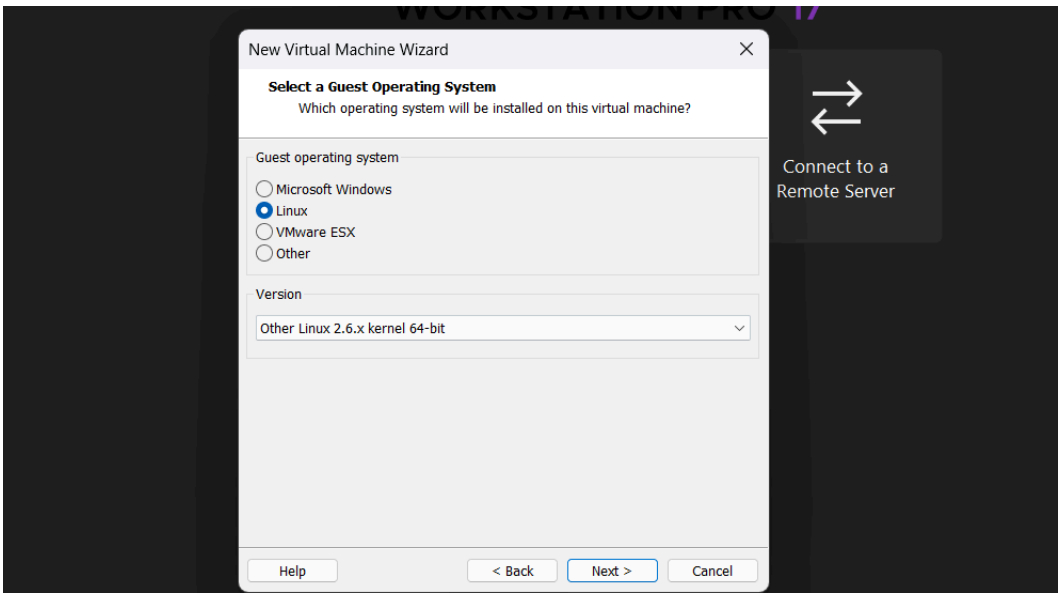
Using VMware, create new virtual machines and install Linux Slackware.
Note: Use expert mode for installation, only installing the necessary packages for basic system operation and network connectivity. Do not install a graphical environment.

Linux Slackware installation log

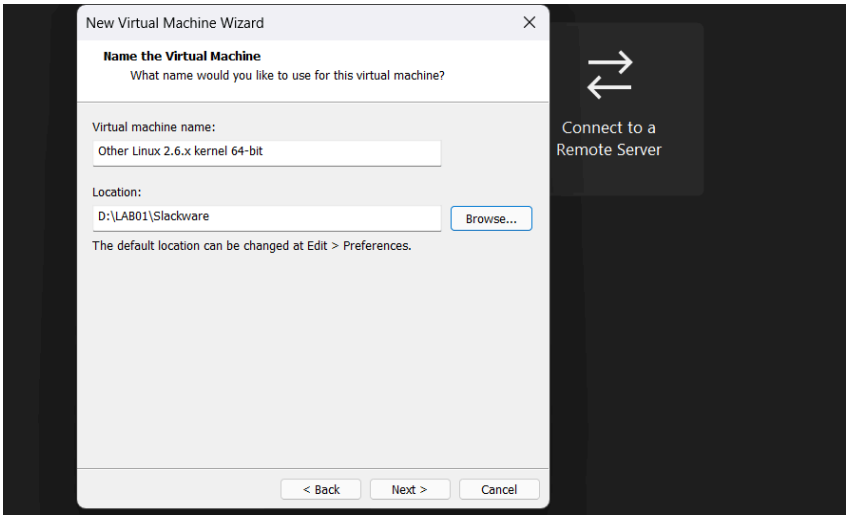
We assign the Slackware ISO for installation



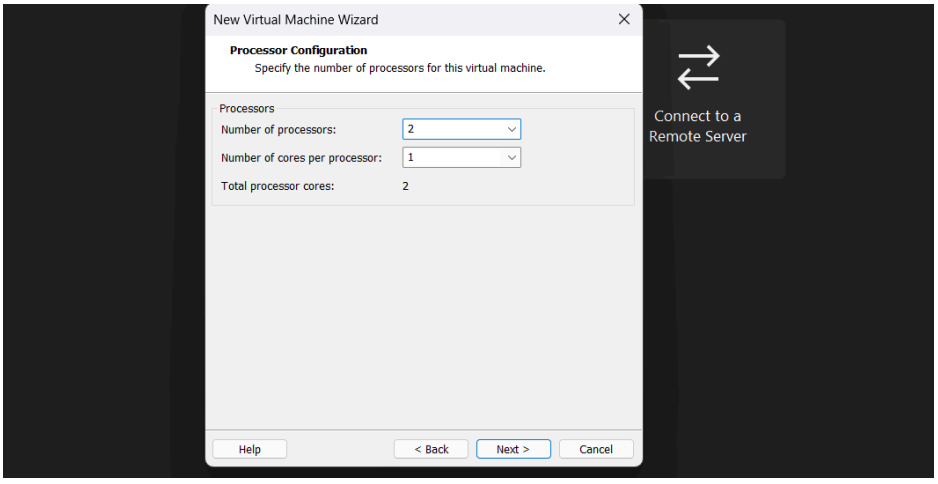
We chose the Linux OS type and the Other Linux 2.6 version.



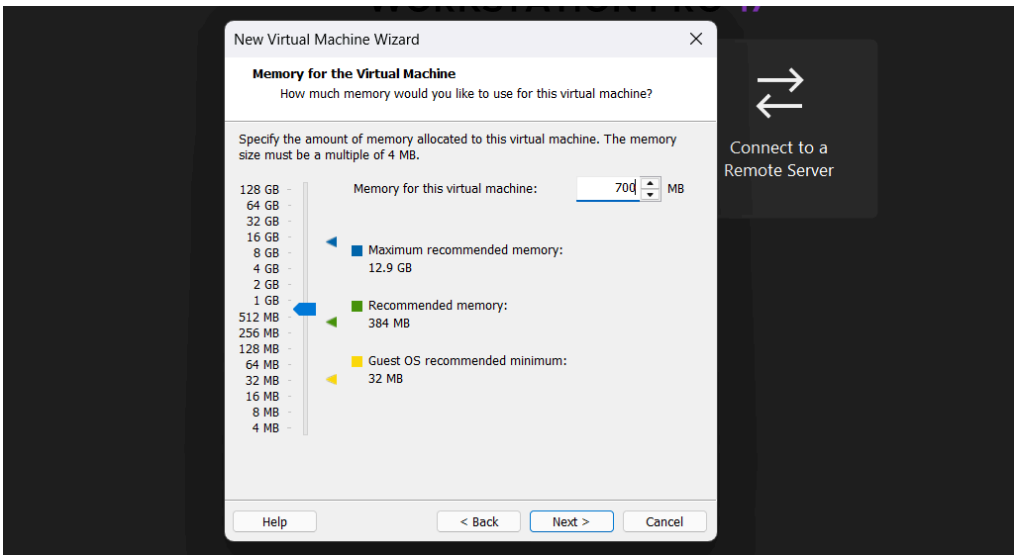
We choose the external hard drive where the virtual machine is saved



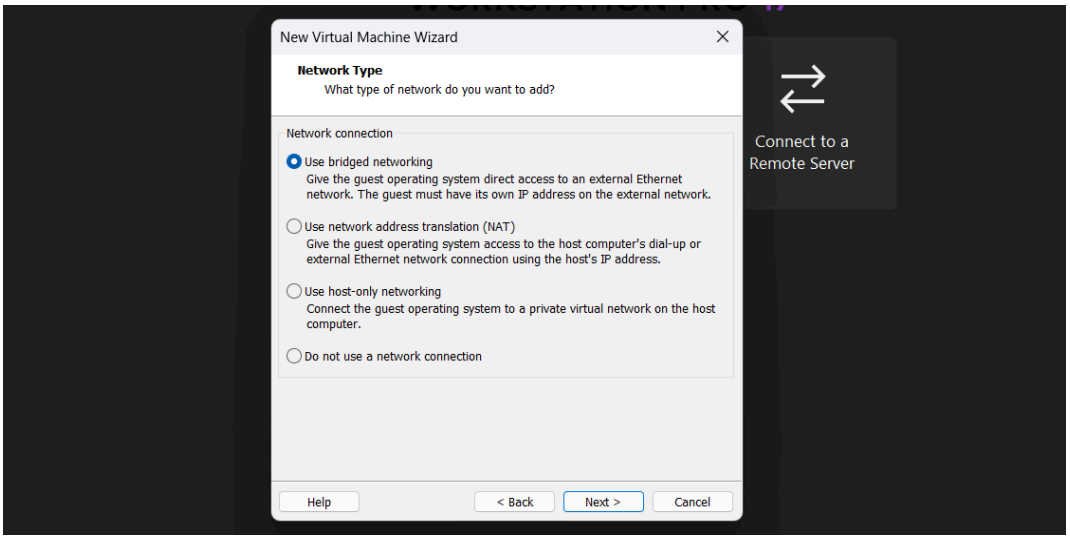
We assign two processors to the VM.



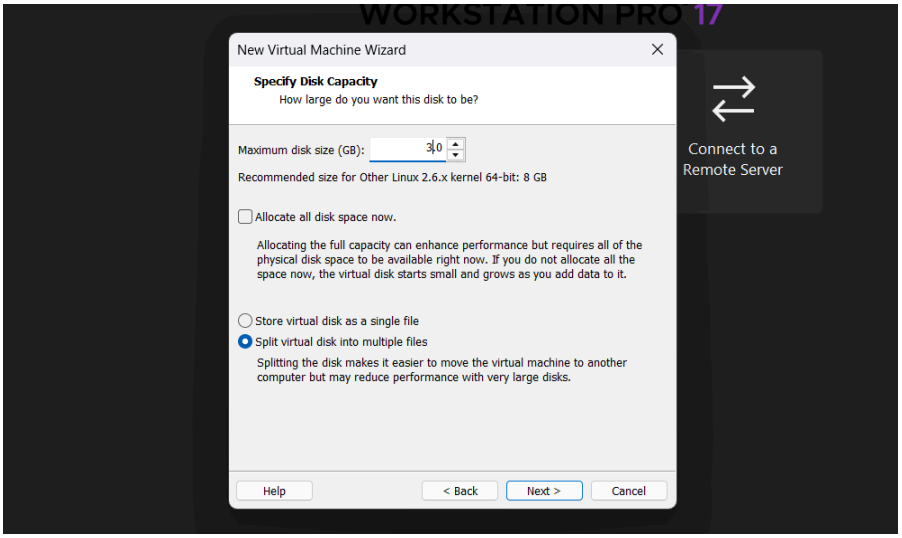
We assign 700MB for the memory.



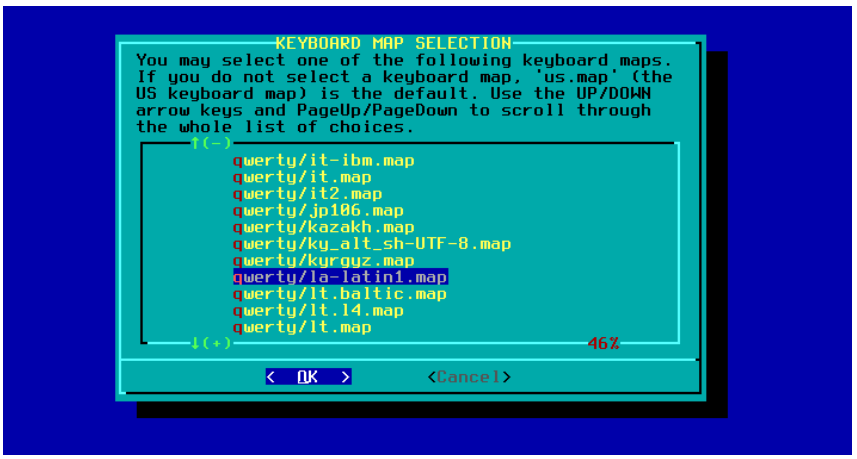
As a network connection we chose bridged networking.



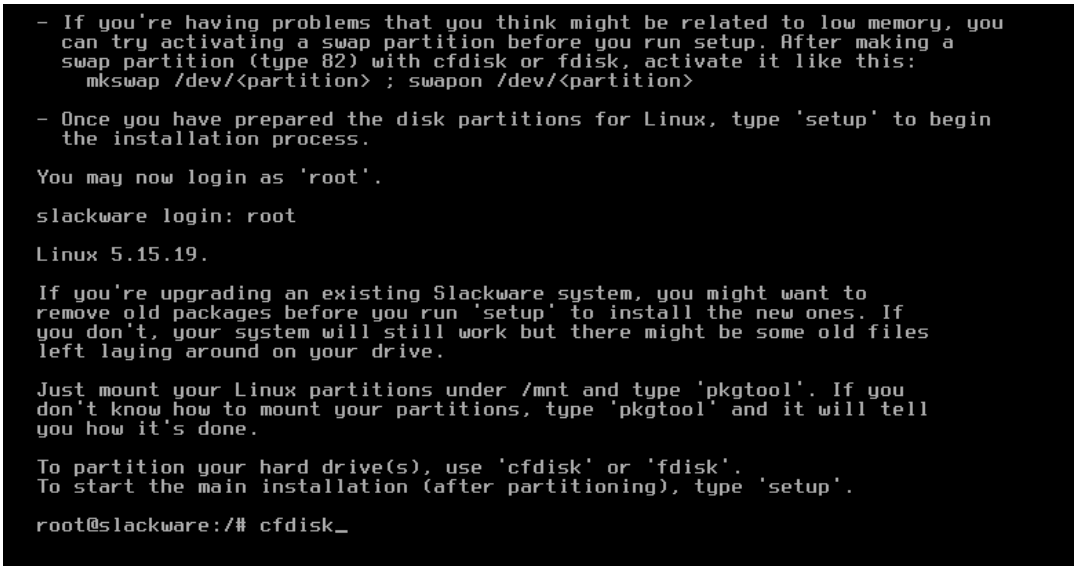
We assign a 3GB virtual disk



We select "querty/la-latin1.map" as the keyboard type.

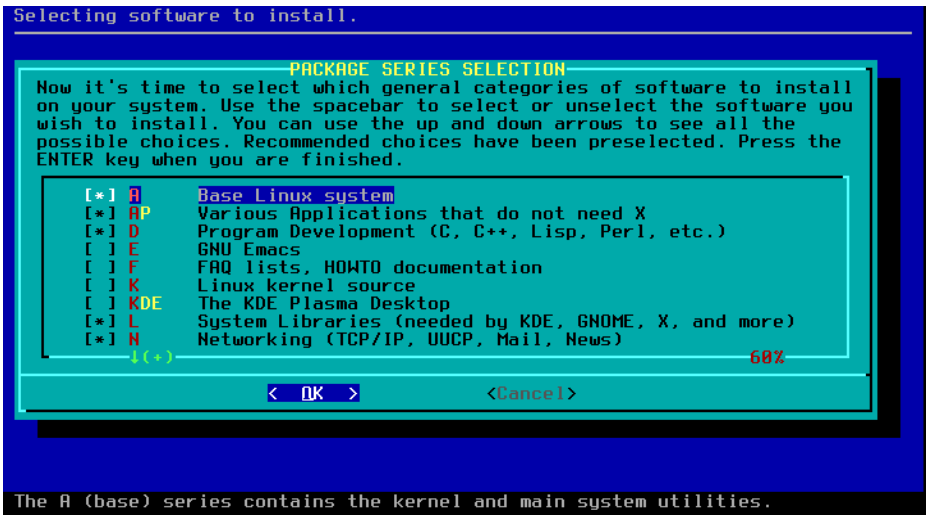


We start as root and type "cfdisk" for the disk partition.

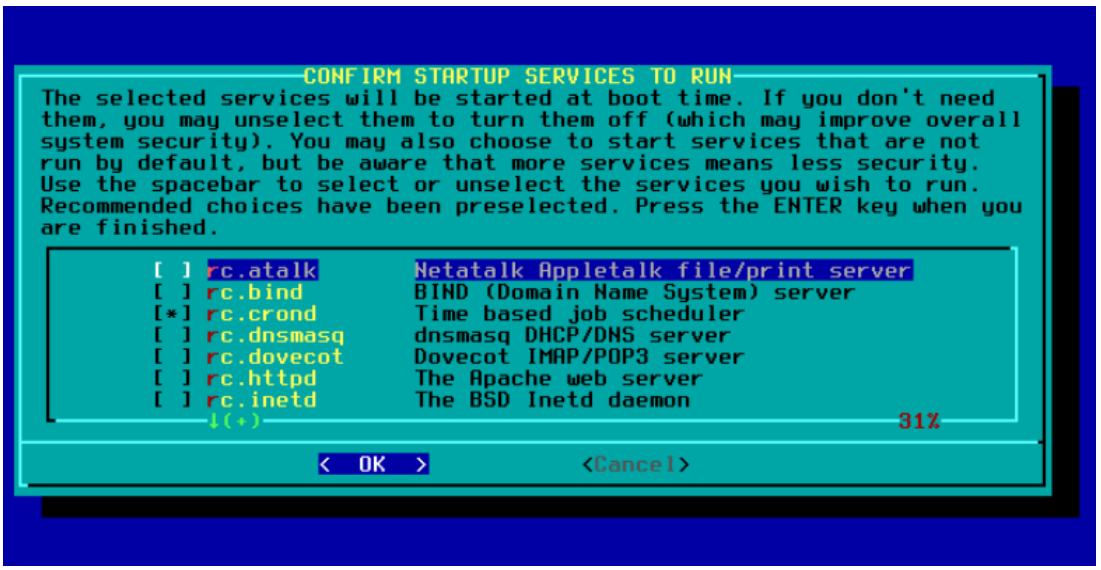


After creating the disk partition, we type "setup" to start the installation.

Select only the package series needed for installation.



Then we select the boot services.

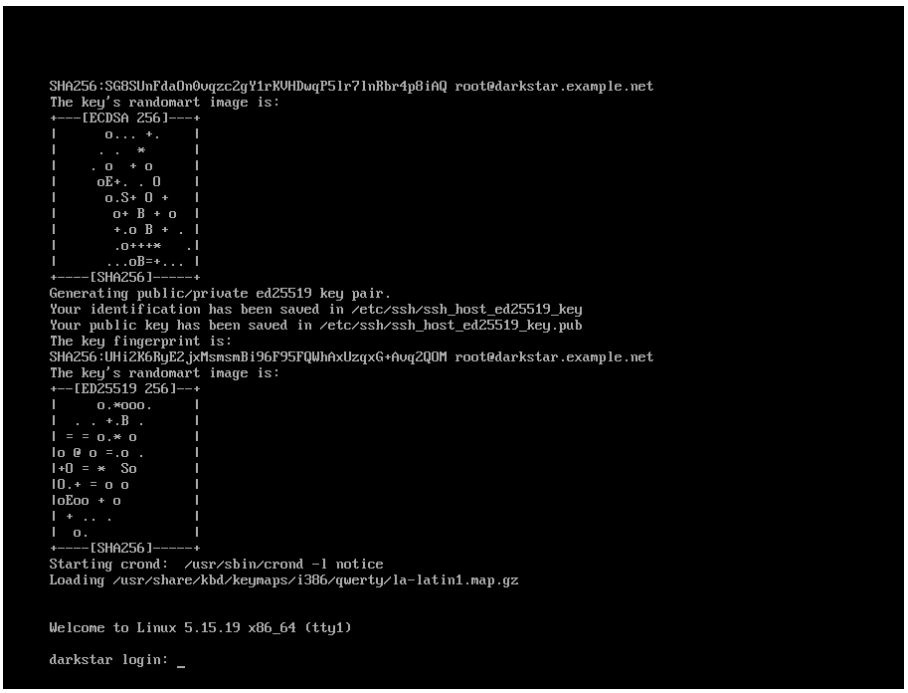


Now set time zone.

After some additional steps, we see that the configuration has been completed.

Now we reboot the system.

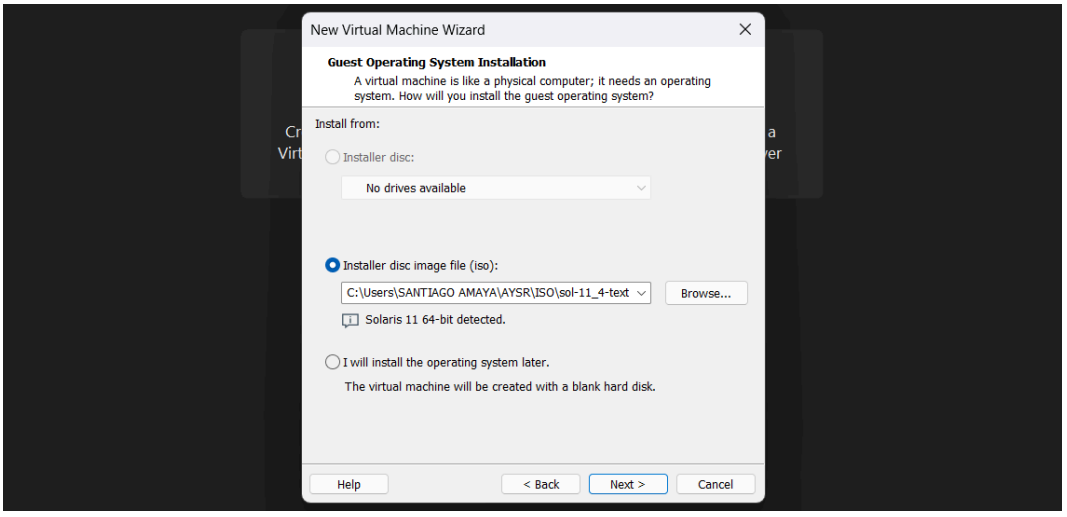
Finally, we see that Slackware starts correctly.



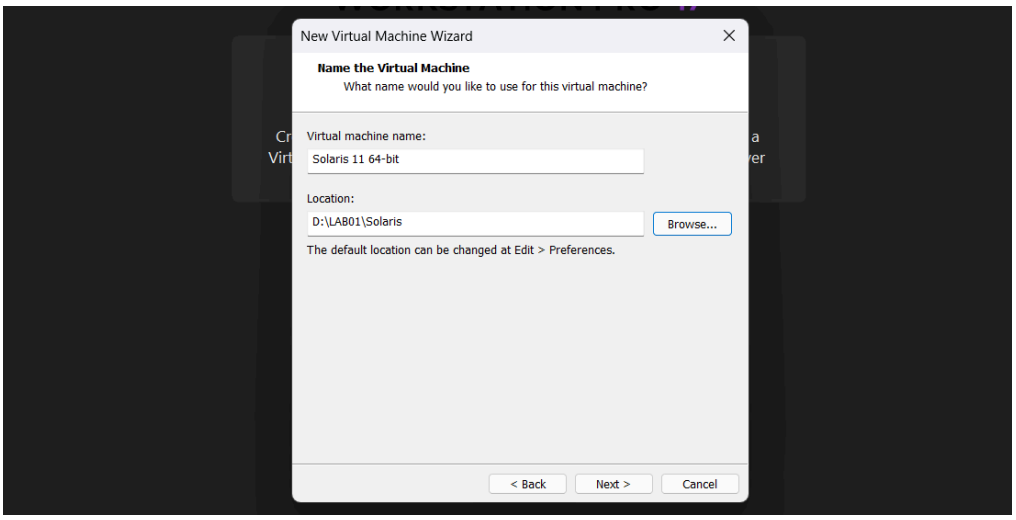
Using VMware, create a new virtual machine and install Solaris. Note: Use expert mode for installation, only installing the required packages for basic system operation and network connectivity. Do not install a graphical environment.

Oracle Solaris installation log

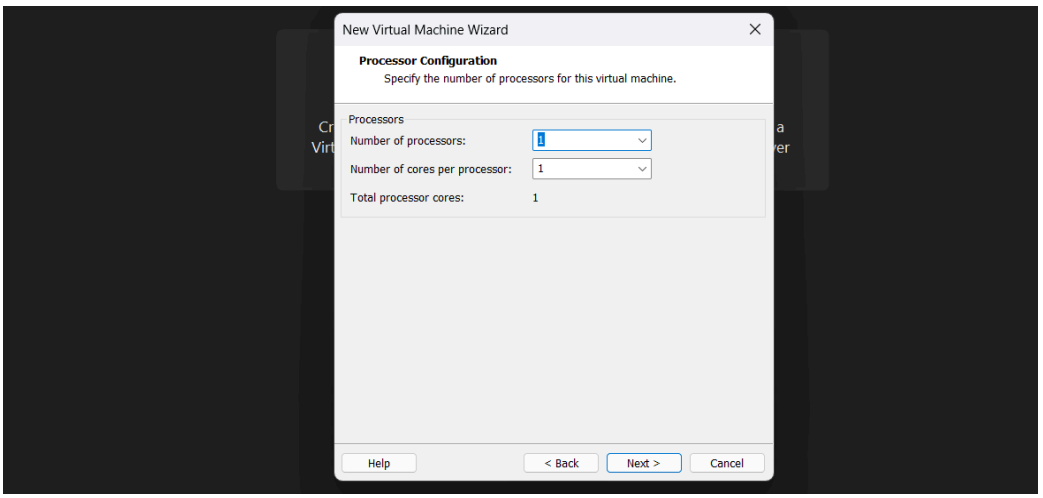
We select the Solaris ISO.



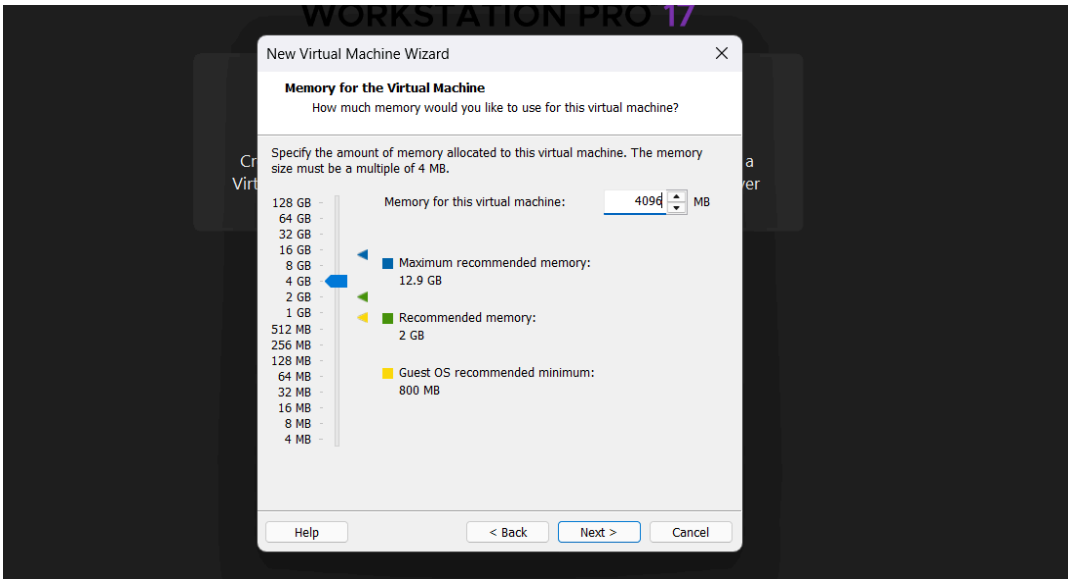
We save the machine on the external hard drive.



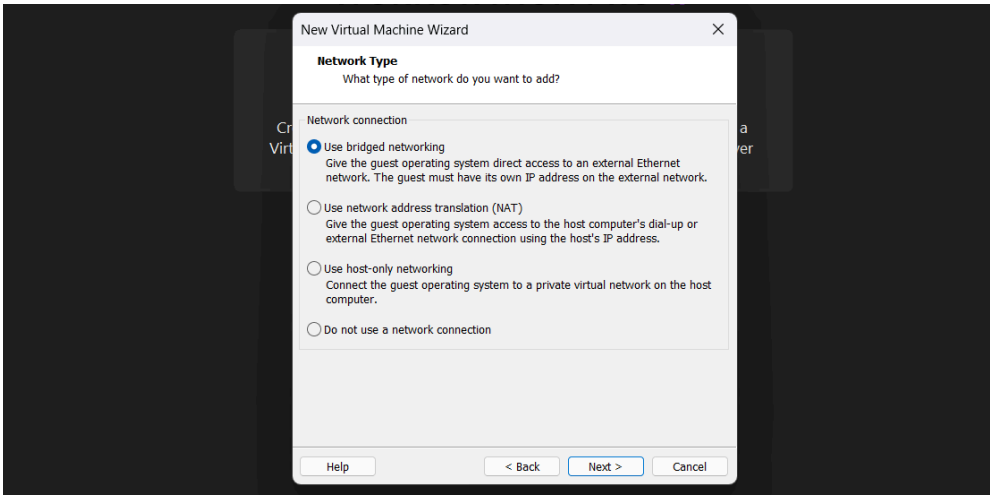
We select a processor and a core for the VM.



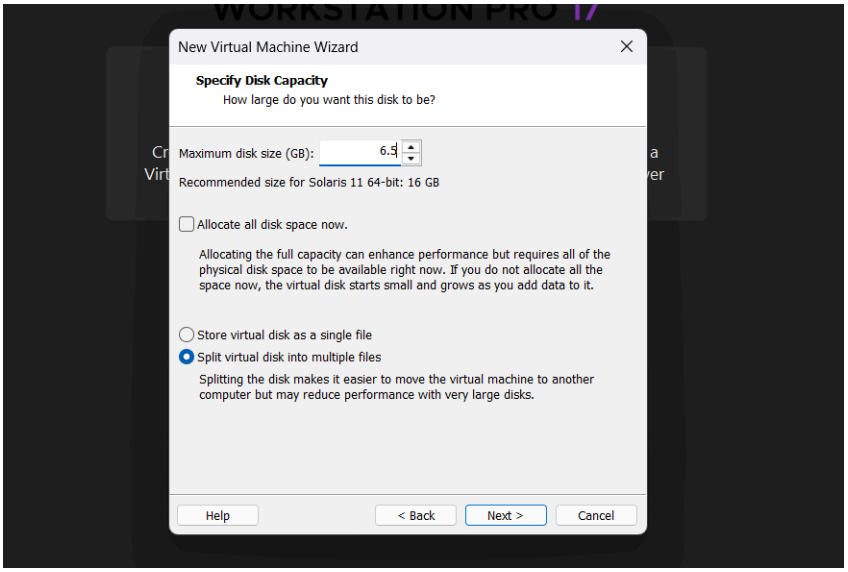
We give the VM a memory size of 4GB.



We chose the Bridge Adapter option for network configuration issues.



We select the creation of a virtual disk to which we assign 6.5GB.

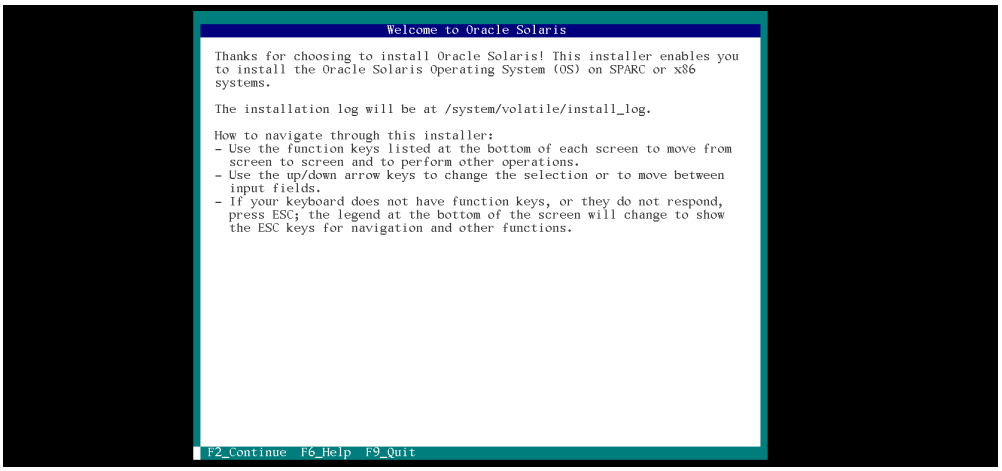


Then we wait for the machine to start and configure the keyboard language.

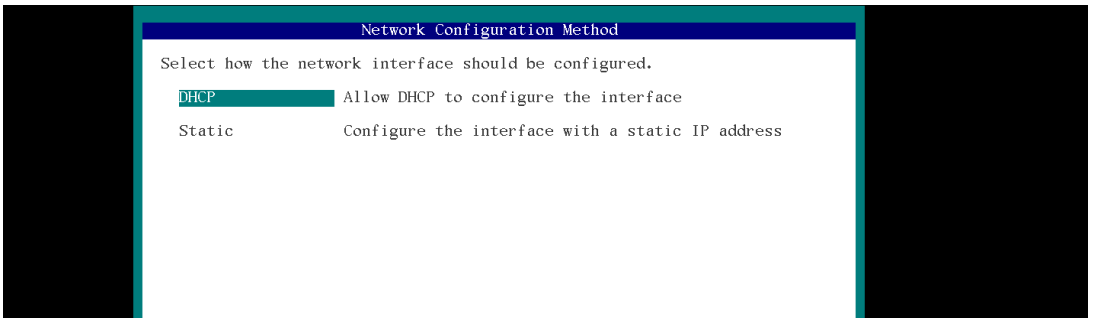
```
SunOS Release 5.11 Version 11.4.0.15.0 64-bit
Copyright (c) 1983, 2018, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing image for use
Done mounting image
USB keyboard
 1. Arabic                      15. Korean
 2. Belgian                     16. Latin-American
 3. Brazilian                   17. Norwegian
 4. Canadian-Bilingual          18. Portuguese
 5. Canadian-French             19. Russian
 6. Danish                      20. Spanish
 7. Dutch                       21. Swedish
 8. Dvorak                      22. Swiss-French
 9. Finnish                     23. Swiss-German
10. French                      24. Traditional-Chinese
11. German                      25. TurkishQ
12. Italian                     26. UK-English
13. Japanese-type6              27. US-English
14. Japanese
To select the keyboard layout, enter a number [default 27]:16
```

```
To select the keyboard layout, enter a number [default 27]:16

 1. Chinese - Simplified
 2. Chinese - Traditional
 3. English
 4. French
 5. German
 6. Italian
 7. Japanese
 8. Korean
 9. Portuguese - Brazil
10. Spanish
To select the language you wish to use, enter a number [default is 3]: 10
```

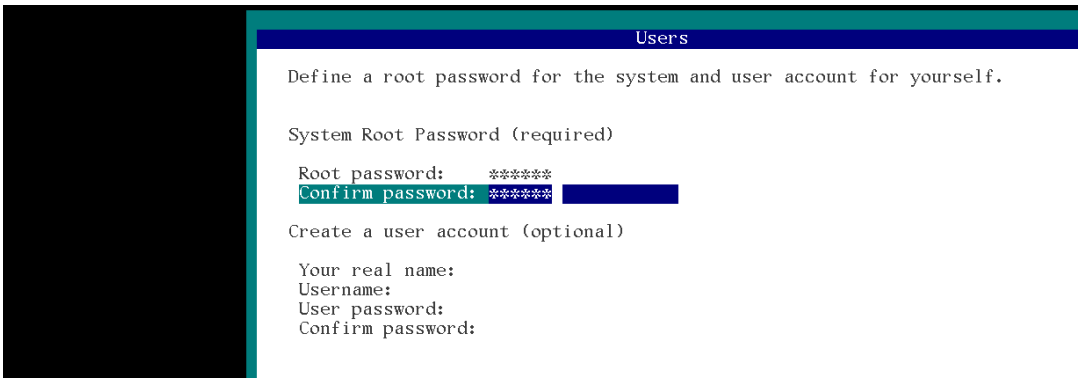
We choose option 1 to start the installation. We continue with f2.



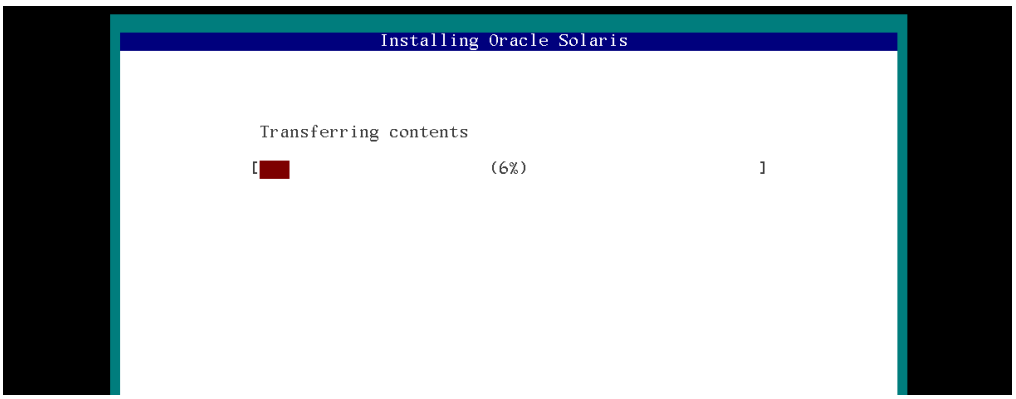
Initially we chose DHCP for automatic IP assignment.



After setting the time zone and so on, we assign a password to log in as root.



We wait for it to install.

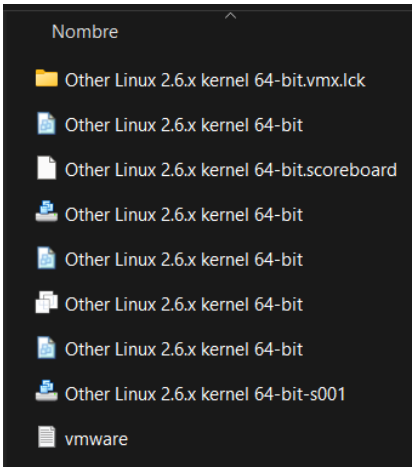


When finished, press f8 - Reboot.

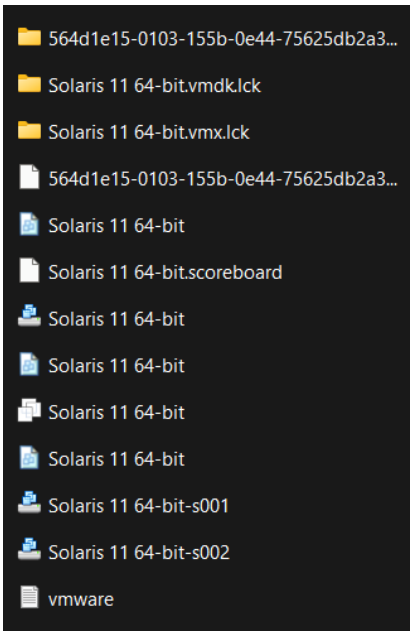
We observed that Solaris started correctly.

What files are generated during installation in each virtualization software, and what are their purposes?

These are the files generated when installing Linux Slackware:



And these are the files generated when installing Oracle Solaris:



The general purpose of virtual machine files is to **replicate the structure of a physical computer** using files that a virtualization program (like VMware or VirtualBox) can understand and manage. Each file plays a vital role in ensuring the virtual machine functions correctly.

Is it possible to convert a VMware virtual machine to VirtualBox and vice versa?
Yes, it is possible to convert a VMware virtual machine to VirtualBox and vice versa. This process is called **migration** and is a common practice in virtualization. The most reliable method involves using a universal disk format called **Open Virtualization Format (OVF)**.

Create four users in each operating system and ensure the following:

– Assign meaningful names. Hint: You may use team members' names (e.g., Claudia, John, Fabian, Diego).

– Provide each user with a meaningful description. Example: "User with the professor's first name" for a user named Claudia.

– Each user should have a home directory matching their username, located in the /usuarios directory at the root of the main file system.

– What is the file system? Which one did you use during installation? What are its characteristics?
A file system is the method and data structure that an operating system uses to control how data is stored and retrieved on a storage device. It defines how files are named, stored in directories, accessed, and managed, ensuring data integrity and organization.

– Create two groups: "Accounting" and "IT."

– The first two users should belong exclusively to the "Accounting" group, while the other two should be part of the "IT" group.

Creating Users in Slackware:

First we create the directory /usuarios

```
root@darkstar:~# mkdir /usuarios
root@darkstar:~# chmod 755 /usuarios
```

We create two groups with the command "groupadd [group´s name]"

```
root@darkstar:~# groupadd accounting
root@darkstar:~# groupadd it
root@darkstar:~#
```

We create the four users with the command "adduser", this is an interactive command in Slackware.

```
New account will be created as follows:

---------------------------------------
Login name.......:  claudia
UID..............:  [ Next available ]
Initial group....:  users
Additional groups:  accounting
Home directory...:  /usuarios/claudia
Shell............:  /bin/bash
Expiry date......:  [ Never ]

This is it... if you want to bail out, hit Control-C.  Otherwise, press
ENTER to go ahead and make the account.
-
```

```
Account setup complete.
root@darkstar:~#
```

After finishing we repeat the same process with the other users. Verify that the users have been created with the command:

cat /etc/passwd | grep -E ‘claudia|john|fabian|diego’

```
root@darkstar:~# cat /etc/passwd | grep -E 'claudia|john|fabian|diego'
claudia:x:1000:100:User with the professor's first name:/usuarios/claudia:/bin/bash
diego:x:1001:100:User with team member's first name Diego:/usuarios/diego:/bin/bash
john:x:1002:100:user with team member's first name john:/usuarios/john:/bin/bash
fabian:x:1003:100:user with member's first name fabian:/usuarios/fabian:/bin/bash
root@darkstar:~#
```

We also verified that the groups were created correctly.

```
root@darkstar:~# cat /etc/group | grep -E 'accounting|it'
polkitd:x:87:
accounting:x:1000:claudia,john
it:x:1001:diego,fabian
root@darkstar:~#
```

And the directory /usuarios.

```
root@darkstar: # ls -l /usuarios
total 16
drwx--x--x 2 claudia users 4096 Aug 16 21:05 claudia/
drwx--x--x 2 diego   users 4096 Aug 16 21:12 diego/
drwx--x--x 2 fabian  users 4096 Aug 16 21:17 fabian/
drwx--x--x 2 john    users 4096 Aug 16 21:15 john/
root@darkstar:~# _
```

Creating users in Solaris:

First we create the directory /usuarios

```
Oracle Corporation      SunOS 5.11      11.4    Aug 2018
root@solaris:~# mkdir /usuarios
root@solaris:~# chmod 755 /usuarios
root@solaris:~#
```

Now we create the groups

```
root@solaris:~# groupadd accounting
root@solaris:~# groupadd it
root@solaris:~#
```

We create the users and assign the password

```
root@solaris:~# useradd -d /usuarios/claudia -m -g accounting -s /bin/bash -c "
user with the professors first name" claudia
80 blocks
root@solaris:~# useradd -d /usuarios/john -m -g accounting -s /bin/bash -c "use
r with the team members first name" john
80 blocks
root@solaris:~# useradd -d /usuarios/fabian -m -g accounting -s /bin/bash -c "u
ser with the team members first name" fabian
80 blocks
root@solaris:~# usermod -g it fabian
root@solaris:~# id fabian
uid=102(fabian) gid=101(it)
root@solaris:~# useradd -d /usuarios/diego -m -g it -s /bin/bash -c "user with
the team members first name" diego
80 blocks
root@solaris:~#
```

```
passwd: password successfully changed for claudia
root@solaris:~# passwd john
New Password:
Re-enter new Password:
passwd: password successfully changed for john
root@solaris:~# passwd diego
New Password:
Re-enter new Password:
passwd: password successfully changed for diego
root@solaris:~# passwd fabian
New Password:
Re-enter new Password:
passwd: password successfully changed for fabian
root@solaris:~#
```

We verify

```
root@solaris:~# ls -l /usuarios
total 12
drwxr-xr-x   2 claudia  accounting       7 ago. 16 17:18 claudia
drwxr-xr-x   2 diego    it             7 ago. 16 17:25 diego
drwxr-xr-x   2 fabian   it             7 ago. 16 17:22 fabian
drwxr-xr-x   2 john     accounting       7 ago. 16 17:22 john
root@solaris:~#
```

Initially, set up network configurations automatically using DHCP and configure the machines in bridge mode.

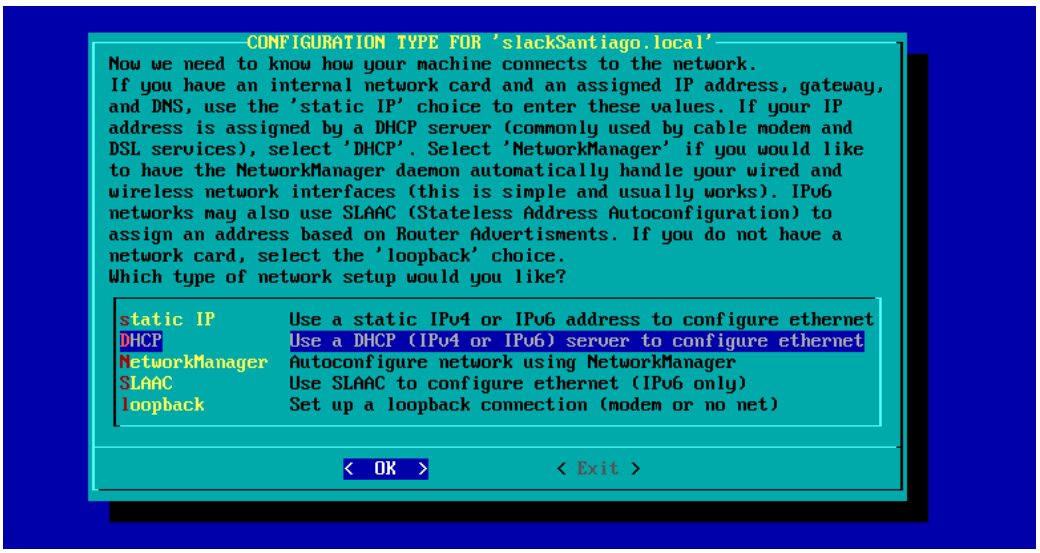
What do ”Bridge Mode” and ”NAT Mode” mean?

Bridge Mode: The virtual machine connects directly to the physical network through the host’s network card. It behaves like another device on the same LAN, getting its own IP from the network.

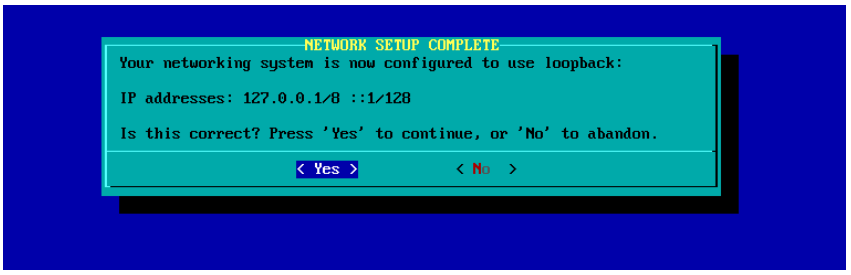
NAT Mode: The virtual machine shares the host’s IP through Network Address Translation. The VM can access external networks (like the internet), but devices on the LAN cannot directly access the VM.

Network DHCP configuration in slackware.

First we run the netconfig command to configure the network with DHCP

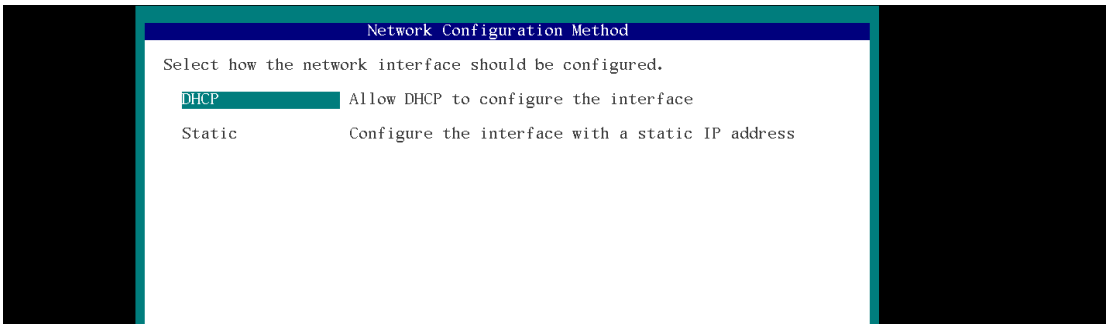


What IP address was assigned to the machine?



Network DHCP configuration in Solaris

During the installation process, the DHCP option was chosen for interface configuration.



What IP address was assigned to the machine?

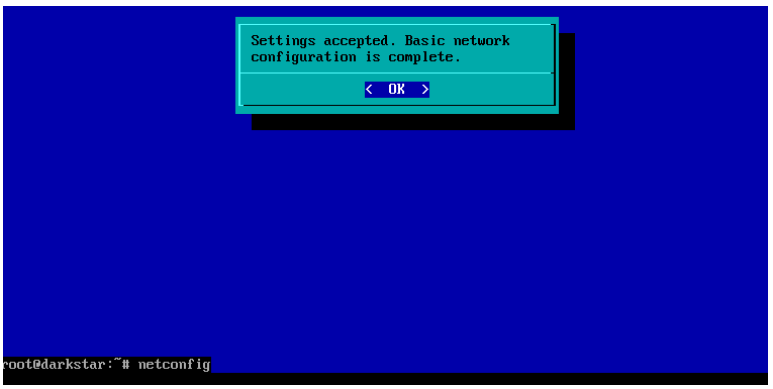
We observe the IP assigned with the command ifconfig -a

```
root@solaris:~# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index
 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,PHYSRUNNING> m
tu 1500 index 2
        inet 0.0.0.0 netmask ff000000
        ether 0:c:29:b2:a3:d1
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index
 1
        inet6 ::1/128
net0: flags=120002004841<UP,RUNNING,MULTICAST,DHCP,IPv6,PHYSRUNNING> mtu 1500 in
dex 2
        inet6 fe80::20c:29ff:feb2:a3d1/10
        ether 0:c:29:b2:a3:d1
root@solaris:~#
```

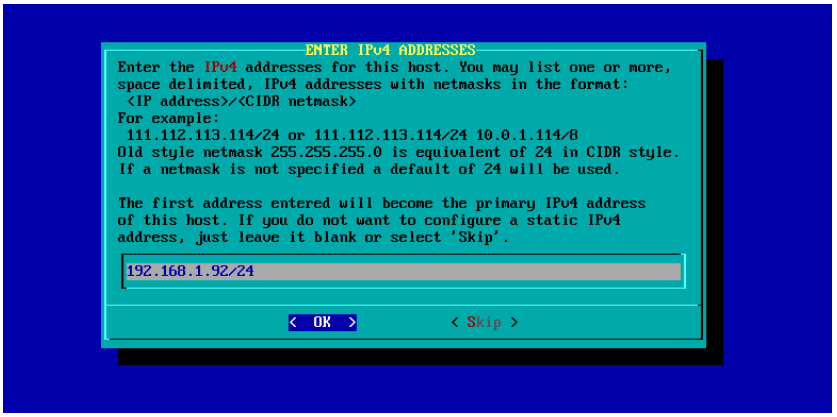
Manually configure the virtual machines' IP addresses using the details provided in the initial clarification.

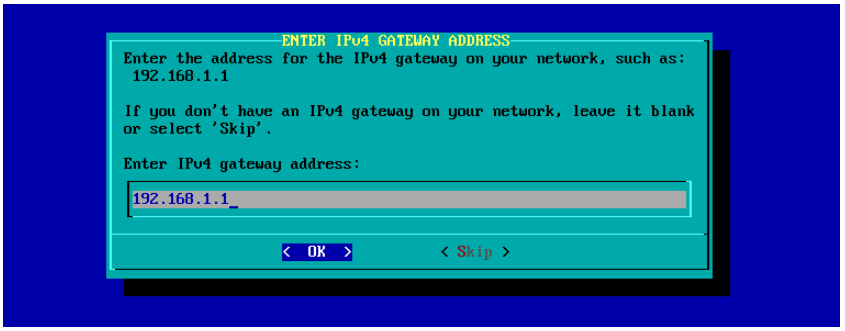
Slackware Static IP configuration

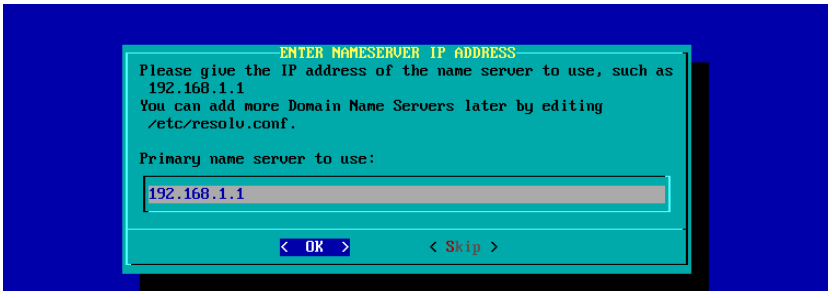
We use the netconfig command again



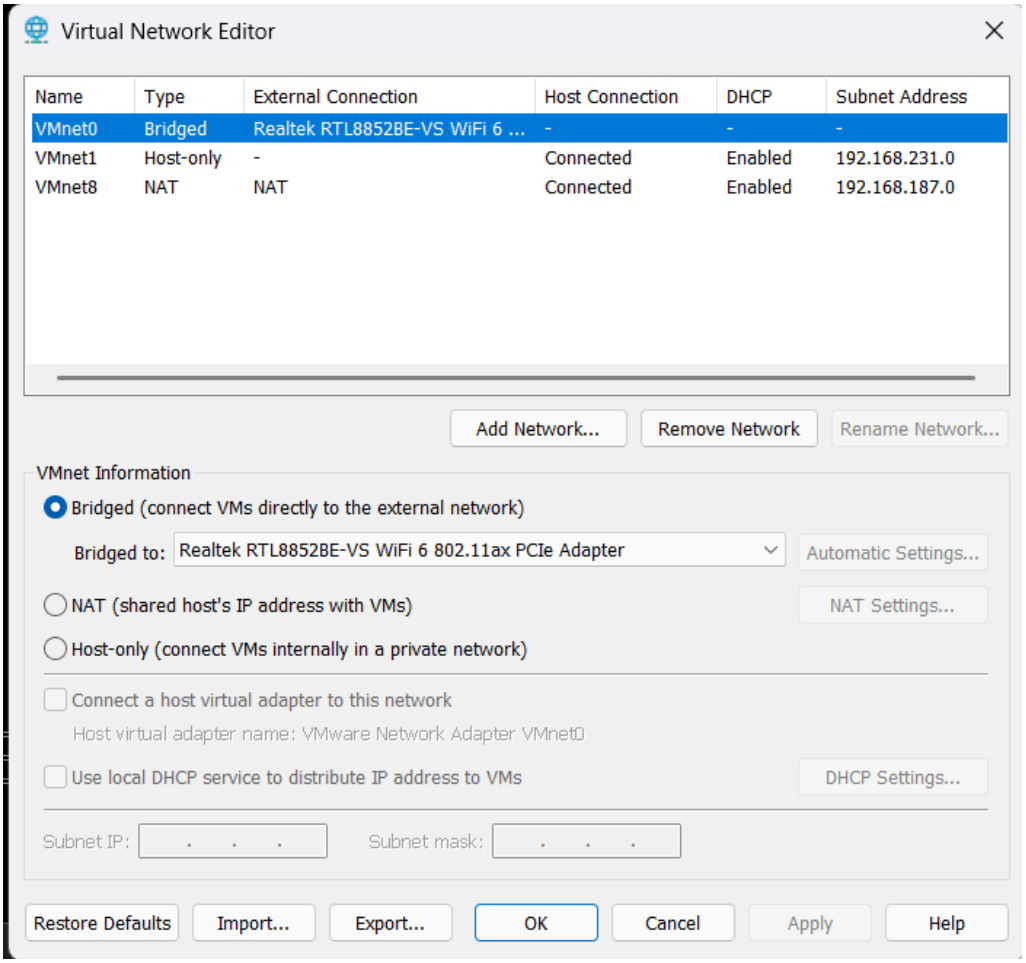
Now we choose the Static IP option to configure the network according to the given data.







Finally, in VMWare configure Edit -> Virtual Network Editor, in VMnet0, select the physical network card to bridge to.



Solaris Static IP configuration

To start, we remove the DHCP configuration and then assign the static IP address

```
root@solaris:~# ipadm delete-addr net0/v4
root@solaris:~# ipadm create-addr -T static -a local=192.168.1.82/24 net0/v4
root@solaris:~#
```

Now we configure the gateway by editing the /etc/defaultrouter file

```
root@solaris:~# echo "192.168.1.1" > /etc/defaultrouter
root@solaris:~#
```

And editing /etc/resolv.conf, then we check the changes

```
root@solaris:~# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           192.168.1.82/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf ok           fe80::20c:29ff:feb2:a3d1/10
root@solaris:~#
```

Now we must activate the reading of host names by DNS, since we cannot edit the /etc/nsswitch.conf file, we must use the following commands to activate this search, and thus be able to do, for example, ping www.google.com

```
root@solaris:~# svccfg -s name-service/switch setprop config/host = astring: "f
iles dns"
root@solaris:~# svcadm refresh name-service/switch
root@solaris:~# Aug 18 05:51:02 solaris sendmail[1163]: My unqualified host name
 (solaris) unknown; sleeping for retry
^C
root@solaris:~# svcadm restart name-service/switch
root@solaris:~# Aug 18 05:51:14 solaris sendmail[1196]: My unqualified host name
 (solaris) unknown; sleeping for retry
^C
root@solaris:~# svcprop -p config/host Aug 18 05:52:14 solaris sendmail[1196]: u
nable to qualify my own domain name (solaris) -- using short name
Aug 18 05:52:14 solaris sendmail[1196]: [ID 702911 mail.alert] unable to qualify
 my own domain name (solaris) -- using short name
name-service/switch
files\ dns
root@solaris:~# grep hosts /etc/nsswitch.conf
hosts:  files dns
root@solaris:~#
```

Test the operating system's network functionality by performing the following checks:

– Check your computer's IP address (host computer). Use the ipconfig command in Windows or ifconfig (or equivalent) in Linux.

```
Adaptador de LAN inalámbrica Wi-Fi:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::6941:a3c2:79c7:b43d%9
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.5
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

– Use the ping command to test connectivity:

- *** ping 10.2.77.n (the machine being configured)**
- *** ping 10.2.65.1**
- *** ping 8.8.8.8**
- *** ping 10.2.77.m (another machine in your group or another group)**
- *** ping www.google.com**

— IMPORTANT —

Since this configuration wasn't possible in the lab, we had to change the specifications so that the tests could be performed at home. The static IP, gateway, and DNS were adjusted according to the home's local network. However, it was still done correctly.

Testing on Linux Slackware

```
root@slackSantiago:~# ping 192.168.1.90
PING 192.168.1.90 (192.168.1.90) 56(84) bytes of data.
64 bytes from 192.168.1.90: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 192.168.1.90: icmp_seq=2 ttl=64 time=0.041 ms
^C
--- 192.168.1.90 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 0.031/0.036/0.041/0.005 ms
root@slackSantiago:~#
```

```
root@slackSantiago:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=12.9 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=4.16 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=3.27 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 3.267/6.766/12.877/4.336 ms
root@slackSantiago:~# _
```

```
root@slackSantiago:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=11.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=6.78 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 6.784/9.350/11.917/2.566 ms
root@slackSantiago:~# _
```

We created another Linux Slackware machine to test with that IP

```
root@slackSantiago:~# ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=64 time=1.19 ms
^C
--- 192.168.1.20 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.190/1.254/1.318/0.064 ms
root@slackSantiago:~# _
```

```
root@slackSantiago:~# ping www.google.com
PING www.google.com (172.217.29.4) 56(84) bytes of data.
64 bytes from eze03s06-in-f4.1e100.net (172.217.29.4): icmp_seq=1 ttl=119 time=5.45 ms
64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=2 ttl=119 time=5.57 ms
64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=3 ttl=119 time=5.84 ms
S64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=4 ttl=119 time=5.91 ms
64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=5 ttl=119 time=7.24 ms
64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=6 ttl=119 time=5.51 ms
64 bytes from 4.29.217.172.in-addr.arpa (172.217.29.4): icmp_seq=7 ttl=119 time=6.16 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6011ms
rtt min/avg/max/mdev = 5.450/5.954/7.240/0.574 ms
root@slackSantiago:~# _
```

Testing on Oracle Solaris

```
root@solaris:~# ping 192.168.1.82
192.168.1.82 is alive
root@solaris:~# ping 192.168.1.1
192.168.1.1 is alive
root@solaris:~# ping 8.8.8.8
8.8.8.8 is alive
root@solaris:~# ping 192.168.1.90
^C
root@solaris:~# ping 192.168.1.20
192.168.1.20 is alive
```

```
root@solaris:~# ping www.google.com
www.google.com is alive
root@solaris:~#
```

Compare the installation experiences of the operating systems.

From the experience of having configured both machines, Slackware allows greater customization in terms of which packages and libraries to install, Solaris is more automatic and with a defined configuration, in terms of network configuration, I think

they are similar, however Solaris requires a greater number of commands to fully configure the network.

b) Understanding and Managing Operating Systems

What is the directory structure of the installed operating systems? List the directories, describe their content, and compare Slackware and Solaris.

- Slackware (Linux)
    - / – Root directory
    - /bin – Essential user binaries (basic commands like ls, cp, mv)
    - /sbin – System binaries for administrative tasks
    - /etc – System configuration files
    - /home – Users' home directories
    - /usr – User programs, libraries, documentation
    - /var – Log files, mail, spool directories
    - /tmp – Temporary files
    - /boot – Boot loader files and kernel
    - /dev – Device files (disks, terminals, etc.)
    - /proc – Virtual filesystem with process and system info
    - /mnt or /media – Mount points for external devices

- Solaris (Unix)

    - Similar to Linux but with some differences:
    - /etc – Configuration files
    - /usr – Applications and libraries
    - /var – Logs and variable data
    - /export/home – Default home directory for users
    - /devices – Hardware devices (Solaris-specific)
    - /kernel – Kernel and modules
    - /platform – Platform-specific drivers

Comparison: Slackware uses /home for users, while Solaris often uses /export/home.

Solaris has directories like /devices and /platform that are not standard in Linux.

Where are the system configuration files located?

- In **Slackware (Linux)** → /etc/ (e.g., /etc/passwd, /etc/fstab, /etc/network/interfaces)

- In **Solaris** → /etc/ as well, but some configurations are also managed with Service Management Facility

Where are the system executables stored? Why multiple locations?

- **Locations:** /bin, /sbin, /usr/bin, /usr/sbin
- **Reason:**
    - /bin and /sbin contain essential programs needed to boot and repair the system
    - /usr/bin and /usr/sbin contain additional user and administrative programs not required during boot.

    - Multiple locations separate *core system binaries* from *user applications*.

Where are system log files stored? What are their purposes?

- **Slackware (Linux):** /var/log/
- **Solaris (Unix):** /var/adm/ and /var/log/
- **Purpose:** Log files record events such as system startup, authentication, errors, warnings, kernel messages, and application logs. They are essential for troubleshooting and auditing.

Where are external storage devices mounted? Connect a USB drive and configure it to be visible in the virtual machine. What commands did you use?

- **Slackware (Linux):** Usually /media/ or /mnt/
- **Solaris:** Typically under /mnt/ or manually mounted to a directory

Differences in filesystem structure

- **Slackware/Linux** is based on the Linux Filesystem Hierarchy Standard
- **Solaris** has a Unix System V style with unique directories like /devices, /platform, /export
- **Main difference:** Linux is more standardized across distributions, while Solaris adds platform-specific directories for hardware and kernel

What are system log files?

System log files are text files that record events and messages generated by the operating system and applications, they can help administrators monitor, debug, and secure the system

What is syslog?

- **Syslog** is a standard protocol used to collect and store log messages.

- **Main files (Linux/Slackware):** /var/log/syslog, /var/log/messages, /var/log/auth.log
- **Information recorded:** authentication attempts, service starts/stops, errors, warnings, kernel events, network activity.
- **Example events:**
    1. Failed SSH login attempt
    2. User login success
    3. Network interface going up/down
    4. Application error (e.g., Apache crash)
    5. Kernel boot messages

Syslog is supported in **both Slackware and Solaris**, although Solaris integrates it with **SMF** for service management.

How do permissions work in the installed operating systems?

- **Permissions:** Read (r), Write (w), Execute (x)
- **Assigned to:** Owner, Group, Others
- **Representations:**
    - **Character:** rwxr-xr--
    - **Numeric (octal):** 755
        - Owner: rwx (7)
        - Group: r-x (5)
        - Others: r-- (4)
- **Commands:**

  Change permissions:

  chmod 755 file.txt

  chmod u+x script.sh

  Change ownership:

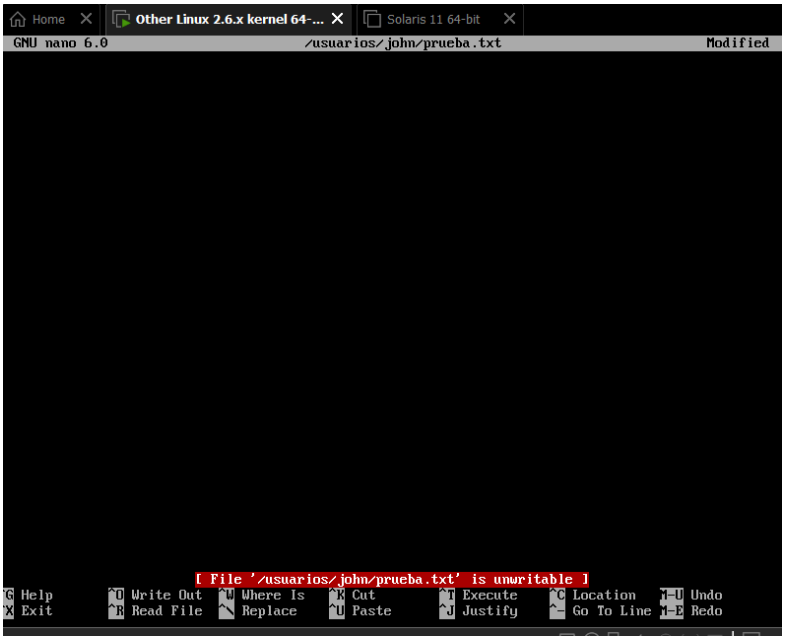
  chown user:group file.txt

Change user and group permissions to observe their impact. For example, allow or restrict users from creating files in other users' directories, executing programs, opening files, or modifying files.

As root, we create a file prueba.txt inside the folder /users/john, and we give only this user permissions to be able to write to it.

```
Welcome to Linux 5.15.19 x86_64 (tty1)

slackSantiago login: root
Password:
Last login: Sun Aug 17 12:44:56 on tty1
Linux 5.15.19.
root@slackSantiago:~# touch /usuarios/john/prueba.txt
root@slackSantiago:~# chown john /usuarios/john/prueba.txt
root@slackSantiago:~# chmod 744 /usuarios/john/prueba.txt
root@slackSantiago:~# ls -l /usuarios/john/prueba.txt
-rwxr--r-- 1 john root 0 Aug 17 12:49 /usuarios/john/prueba.txt*
root@slackSantiago:~#
```

Then we enter with another user (Claudia) and verify that it is not possible to write

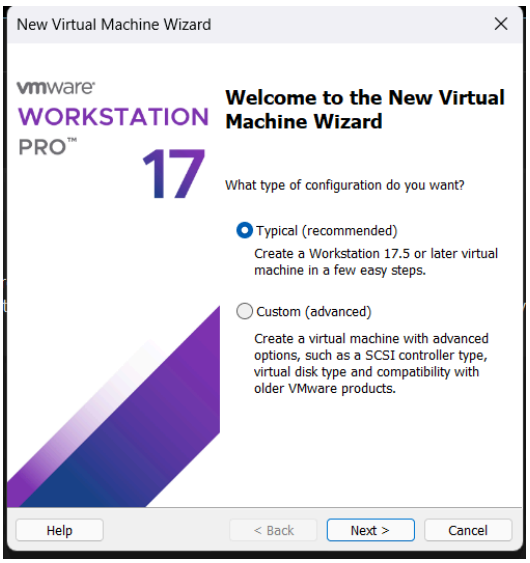


3. Windows Server Installation and Configuration – Phase 1

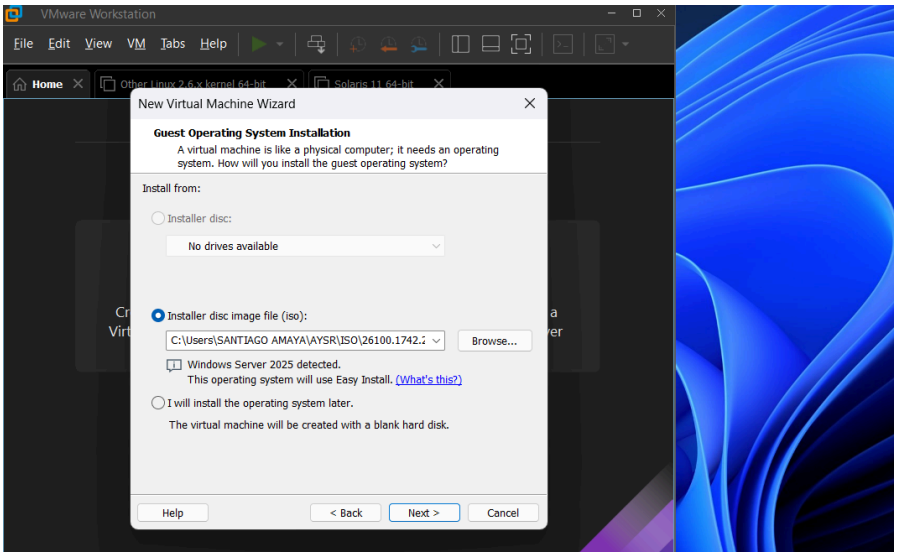
Create a new virtual machine using VMware and install Windows Server without a graphical interface. Note: Do not configure Active Directory.

Windows Server installation log (without GUI)

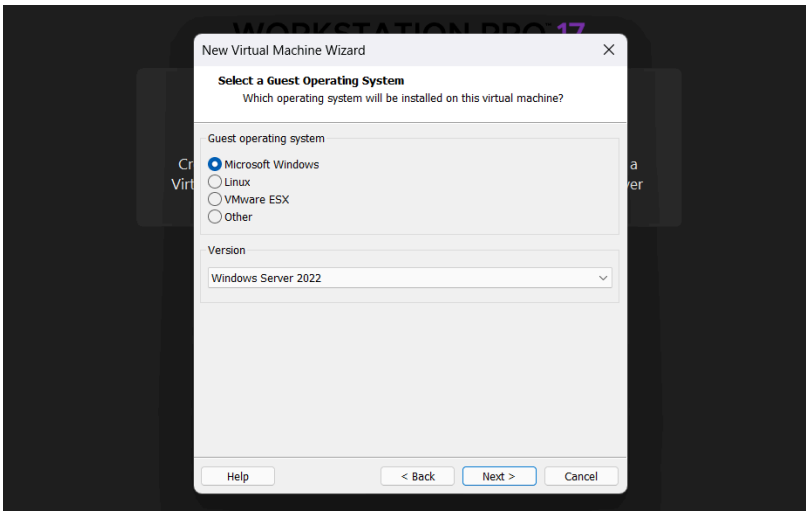
On this occasion, we chose the typical configuration.



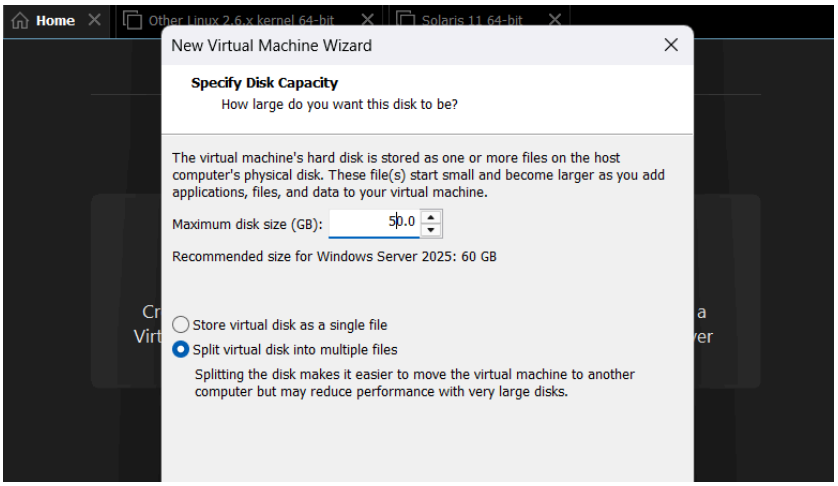
We select the option to install the OS later



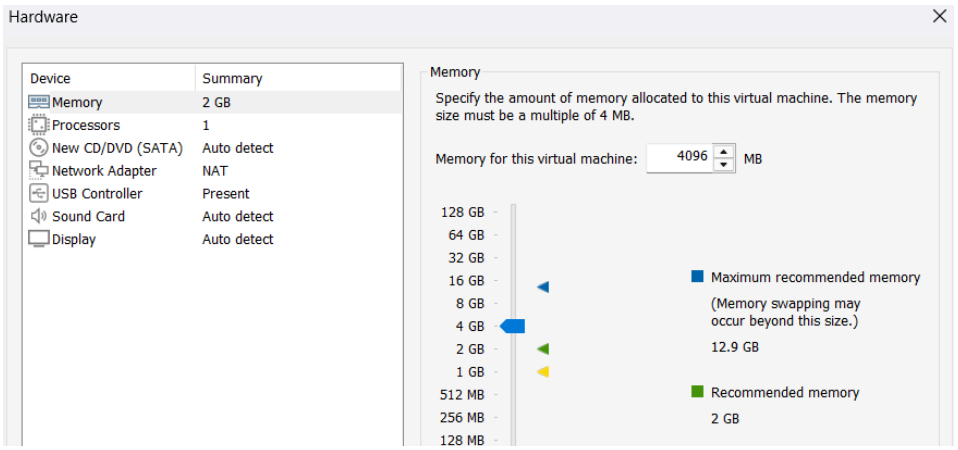
We select the guest operating system and type



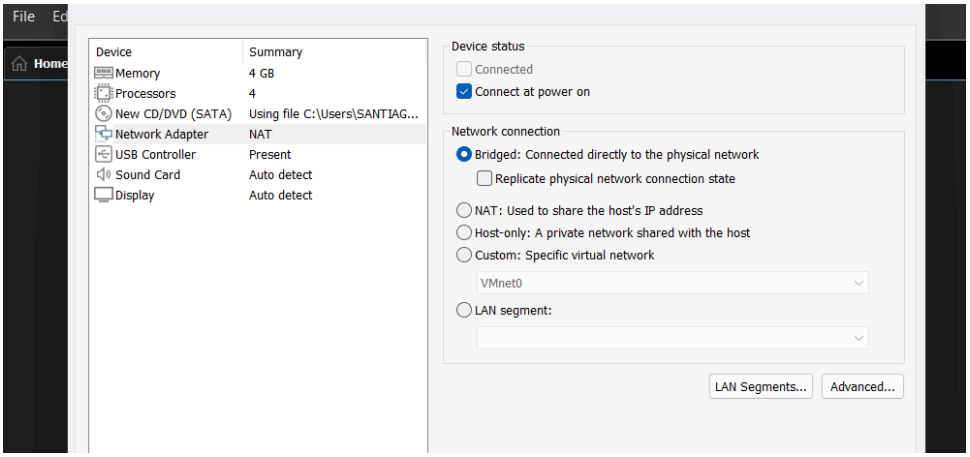
We select 50GB for the virtual disk size



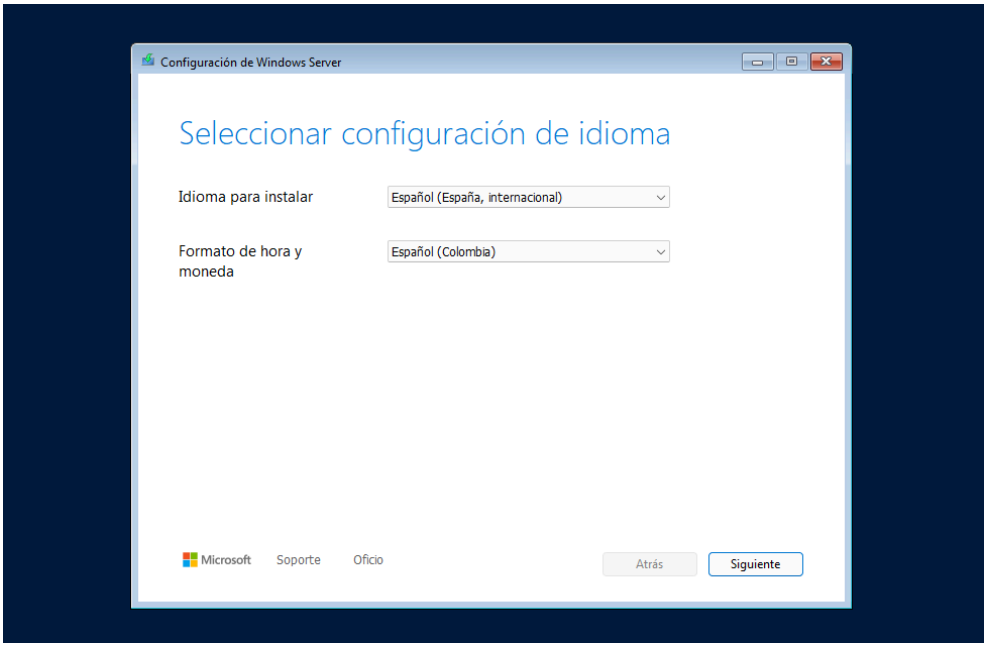
Then chose 4GB for the memory



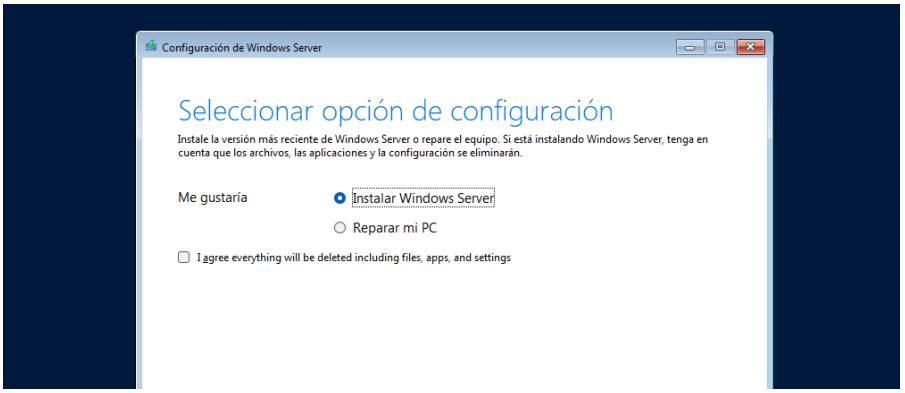
We select 4 processors and configure the network in bridge adapter



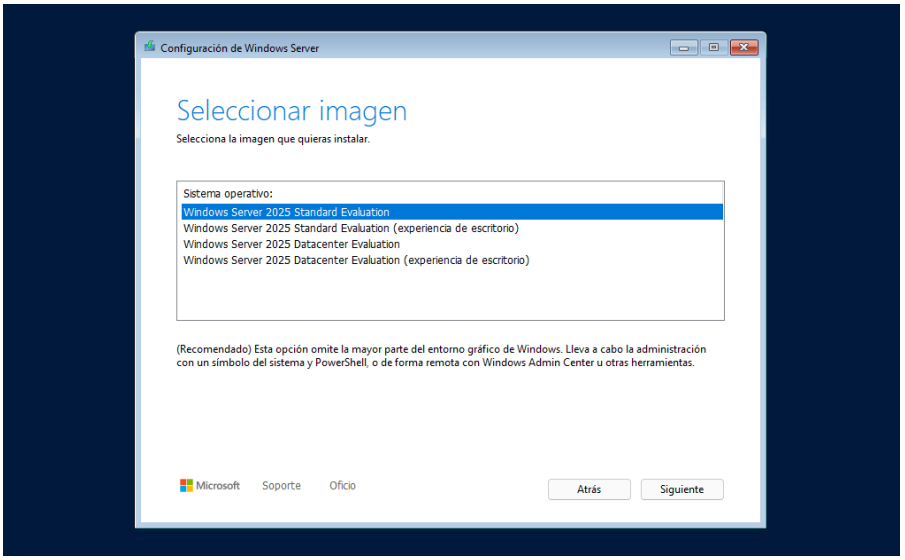
We select the installation language and keyboard layout.



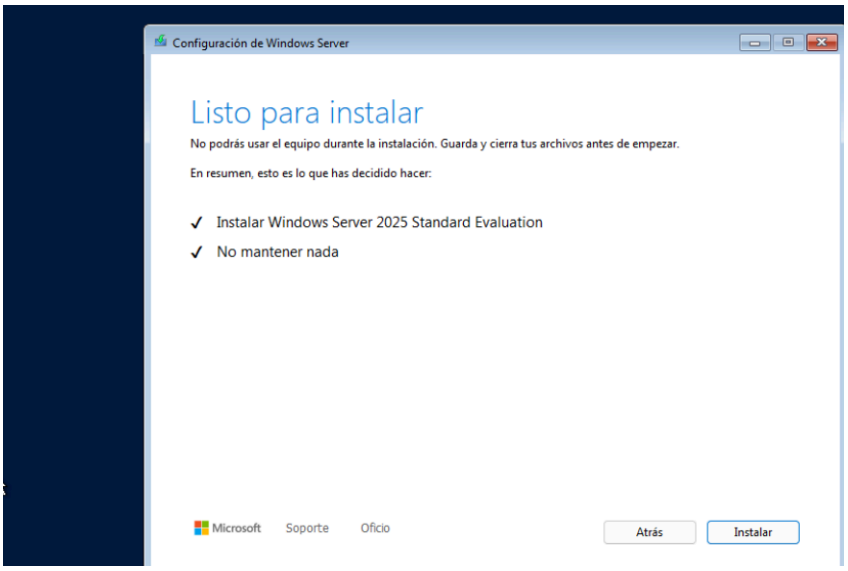
We choose the option to install Windows Server.



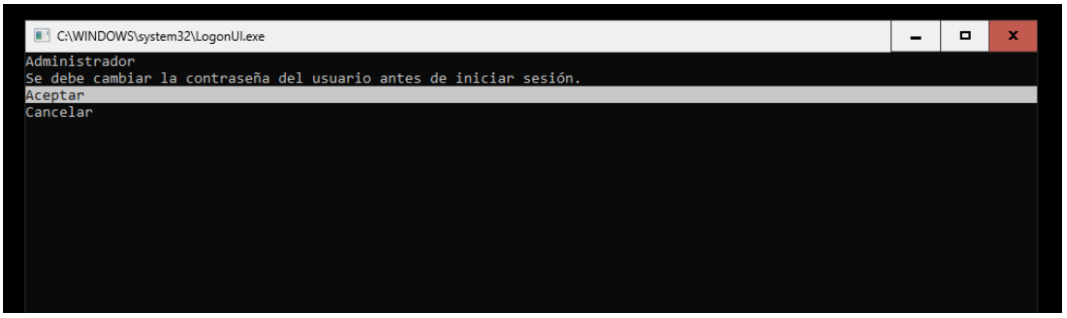
We choose the version without interface



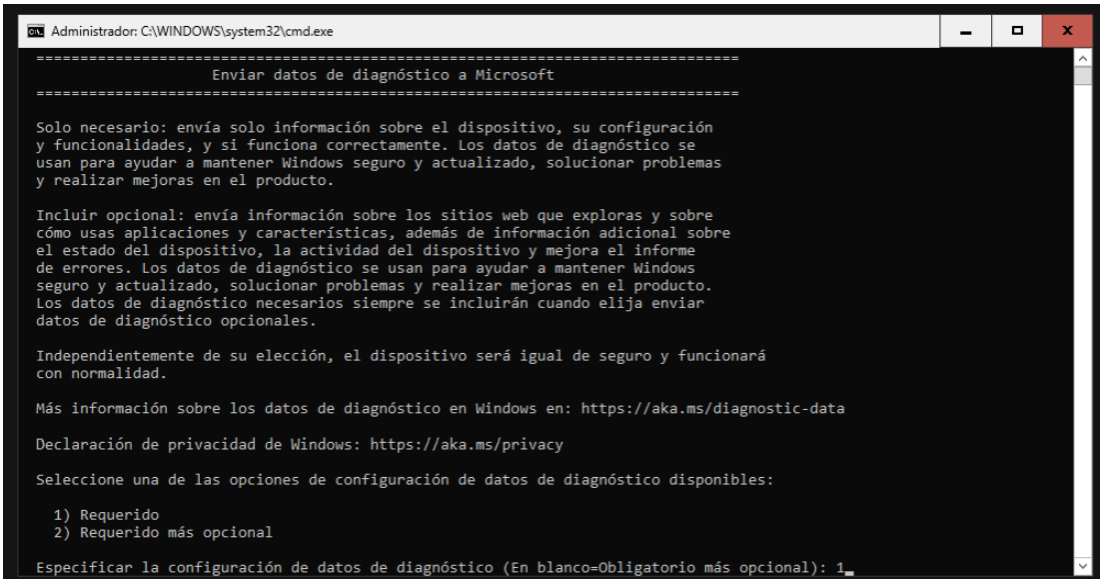
We install



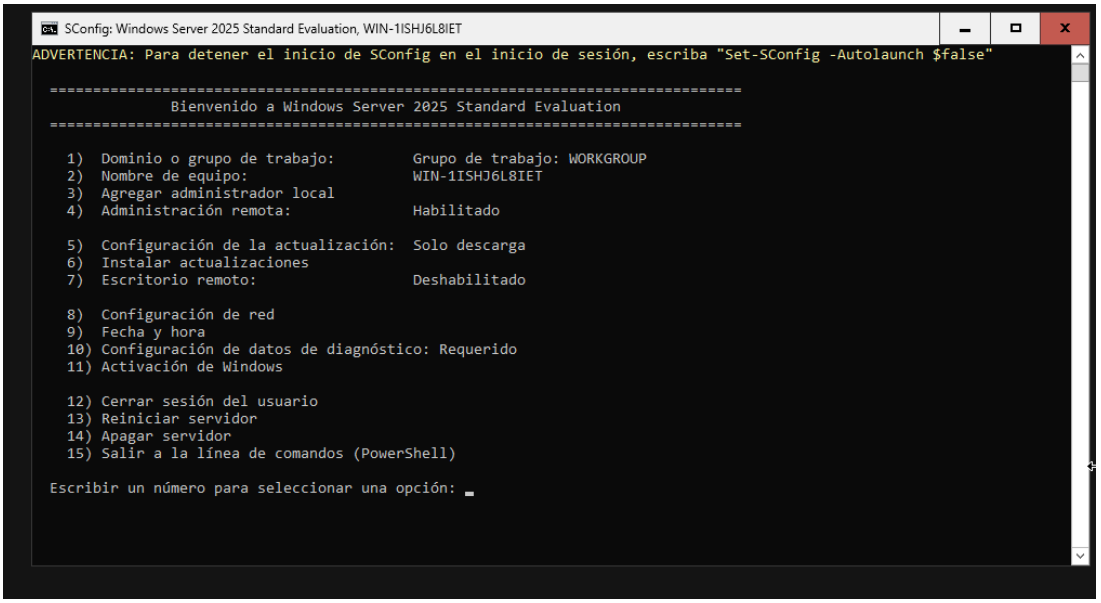
Now we click on “Accept” to change the password



Now we select option “1” for the diagnostic data required by Microsoft.

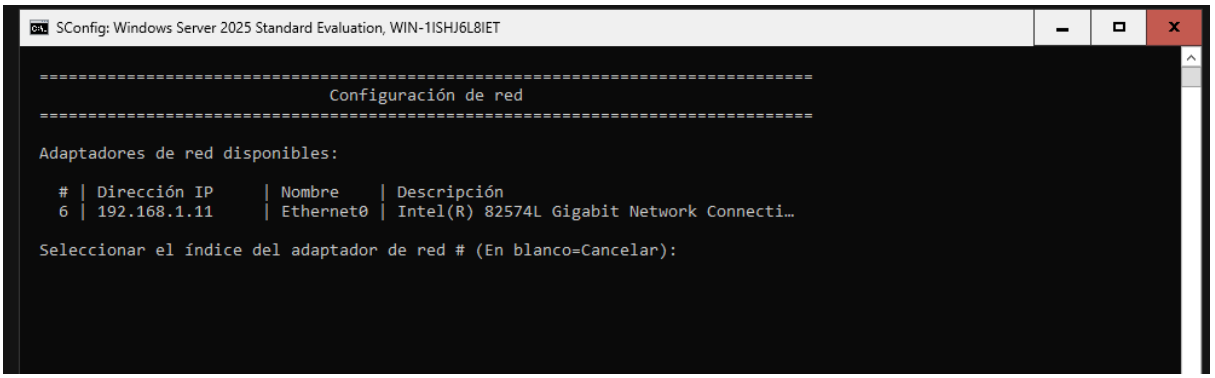


Now we select option 9 to change the date and time settings

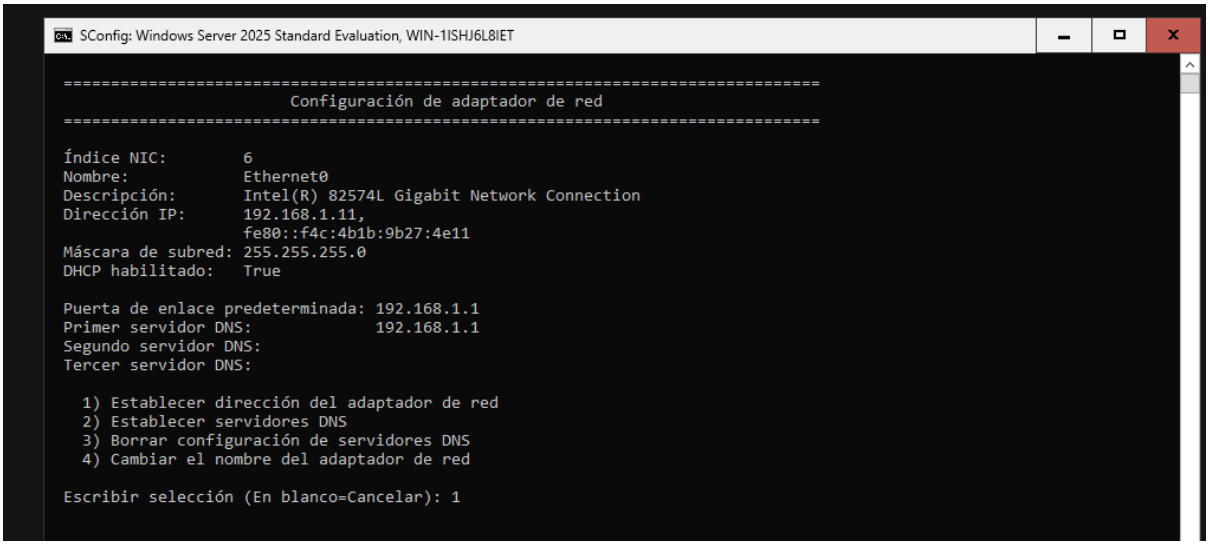


Set up network operation in Bridge Mode and configure the network according to the initial clarification.

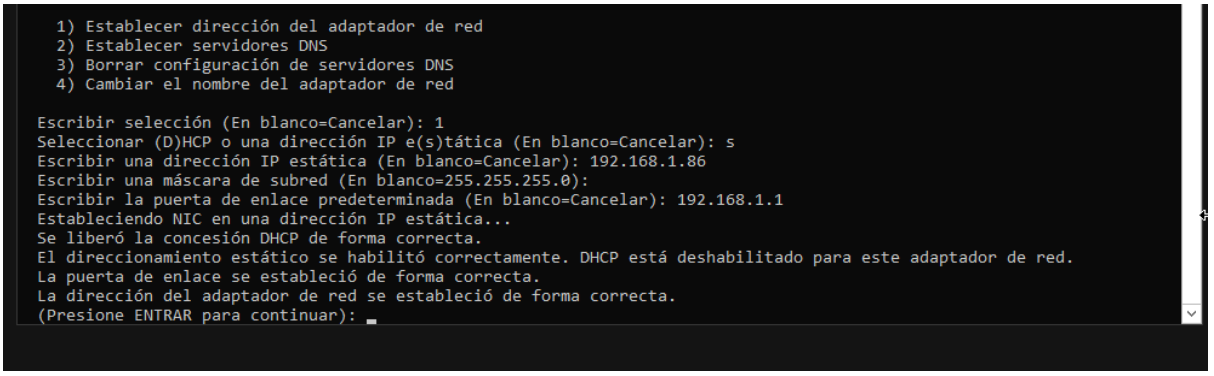
Now we select 8 to configure the machine's network and chose the network adapter 6



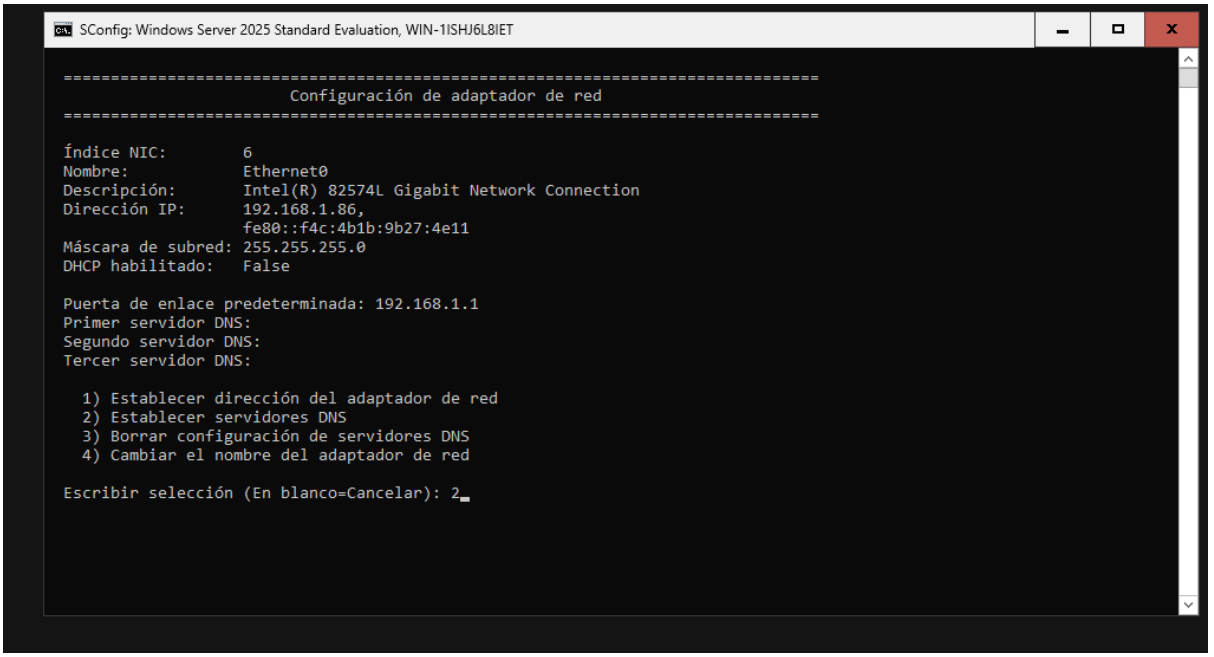
Now we select option 1, to configure the network adapter address



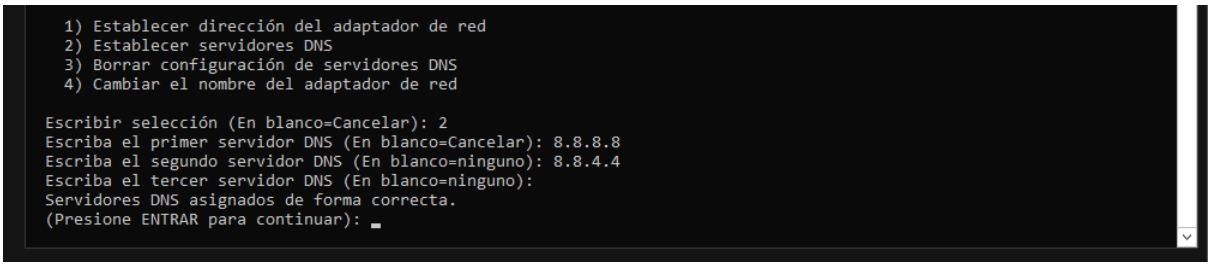
We mark “s” to configure a Static IP. We write the IPv4, mask and Gateway, then press enter to continue



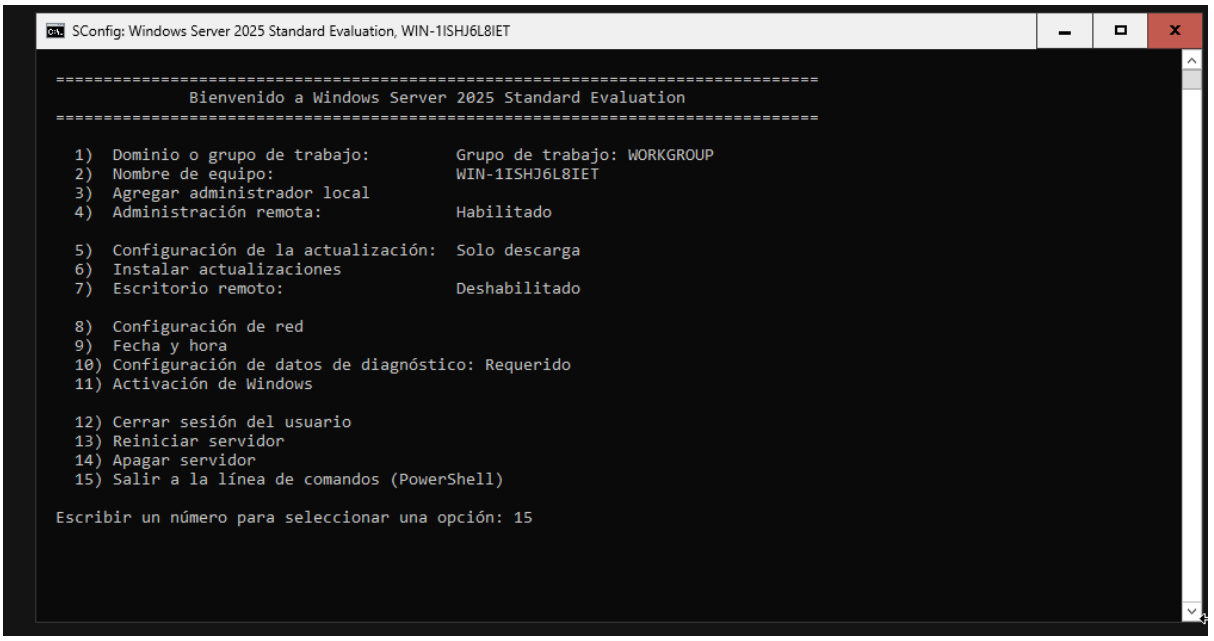
Now we select option 2 to configure the DNS.



We write the DNS servers, then press enter.

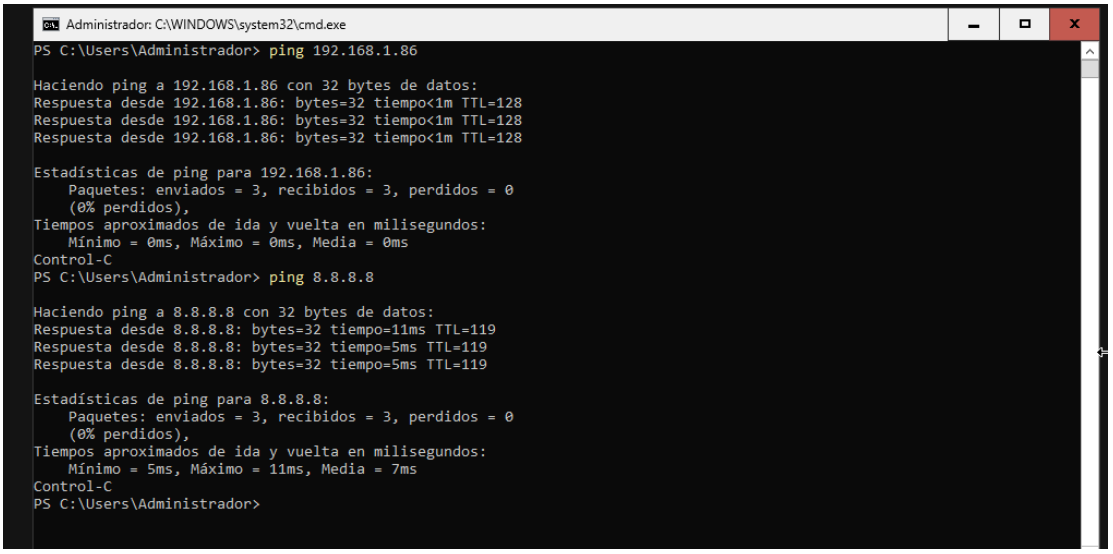


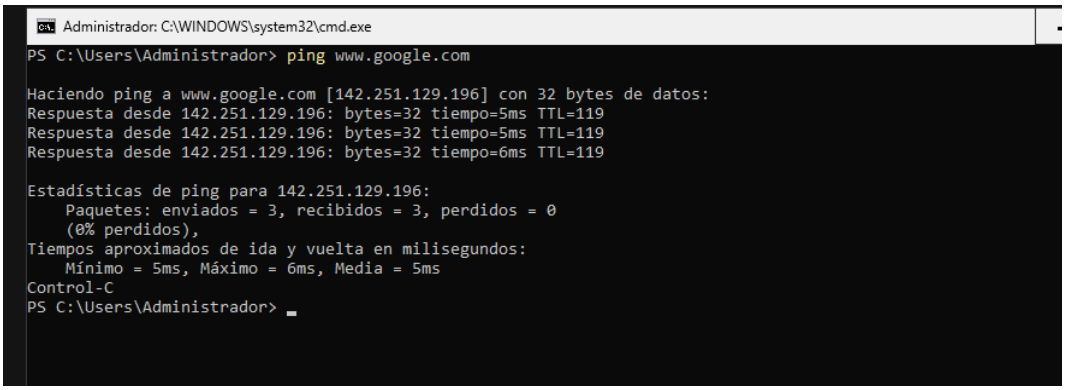
Ahora seleccionamos la opción 15, para salir hacia la línea de comandos (PowerShell)



Test system functionality by executing the following commands:

- **ping host computer ip**
- **ping 8.8.8.8**
- **ping www.google.com**

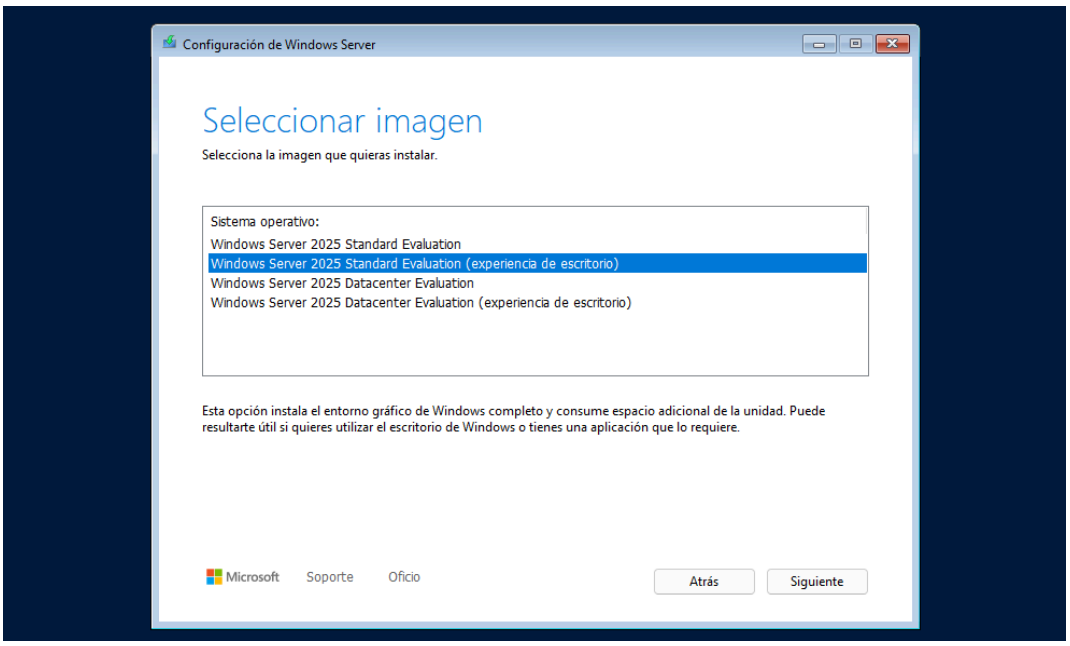




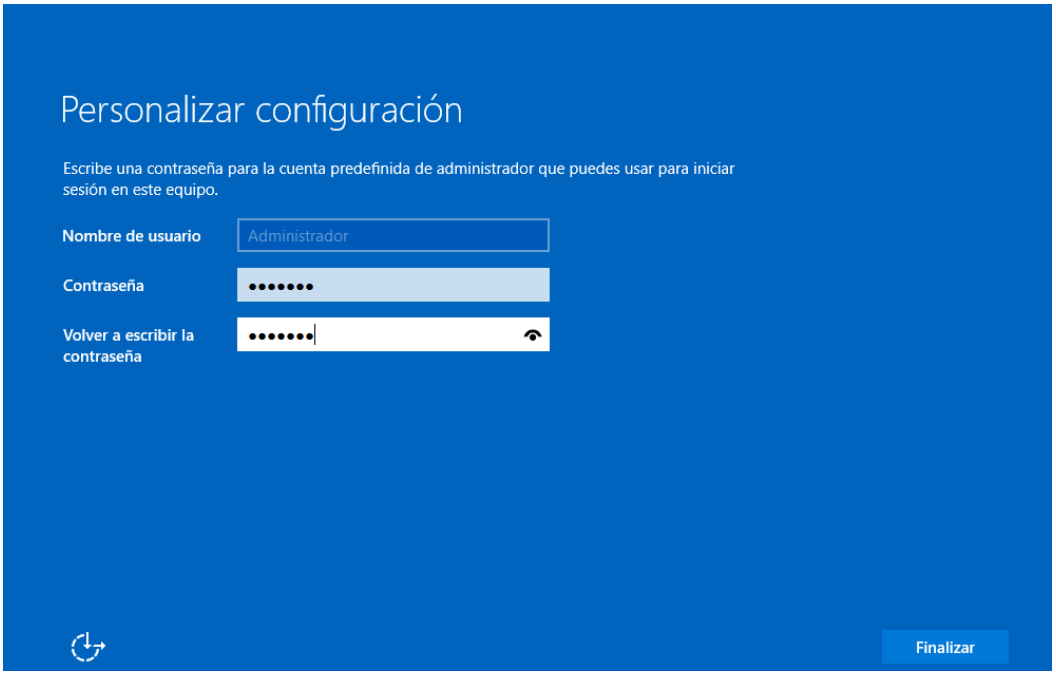
4. Windows Server Installation and Configuration – Phase 2

Install the graphical version of Windows Server using VMware.

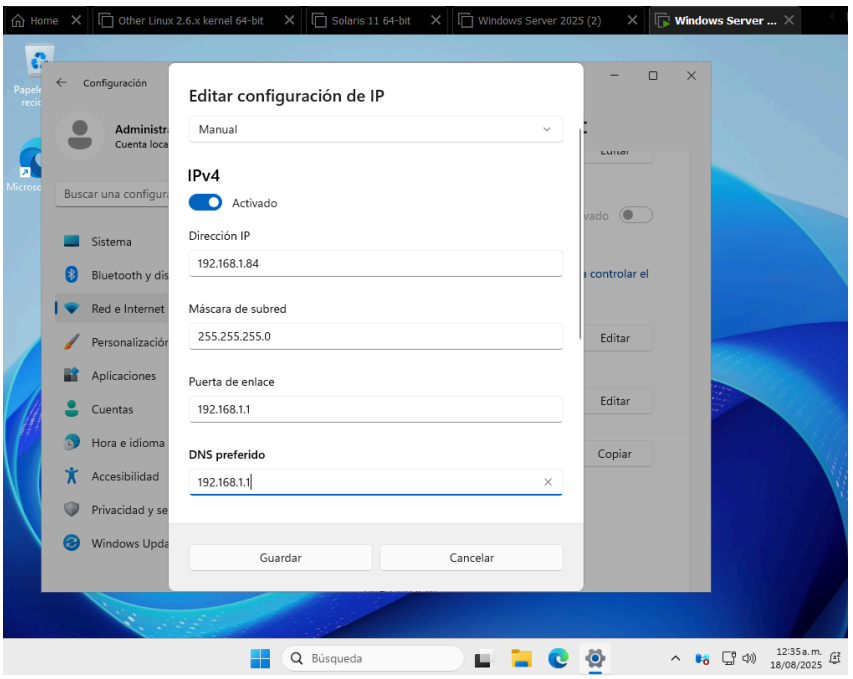
We follow the same steps as the previous machine until this step. Now we select the Windows Server 2025 Standard Evaluation option (desktop experience).



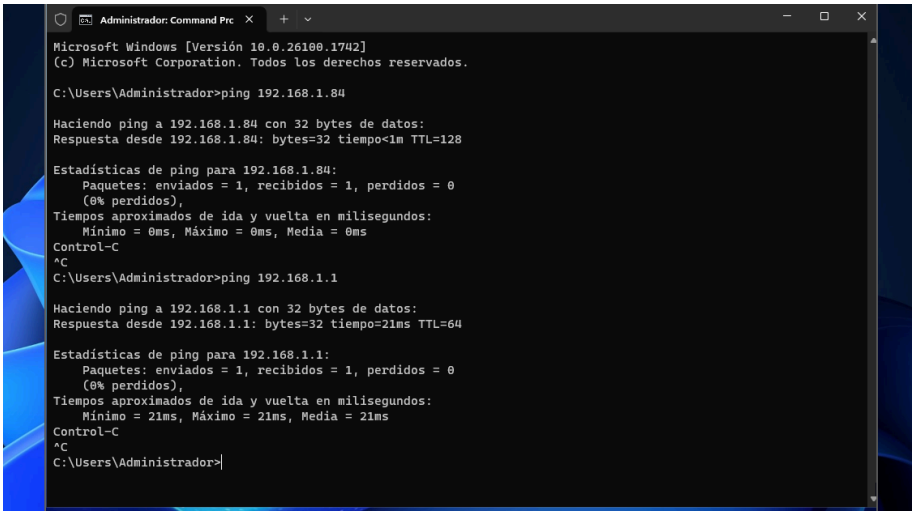
Now we add the administrator password



We configure the network

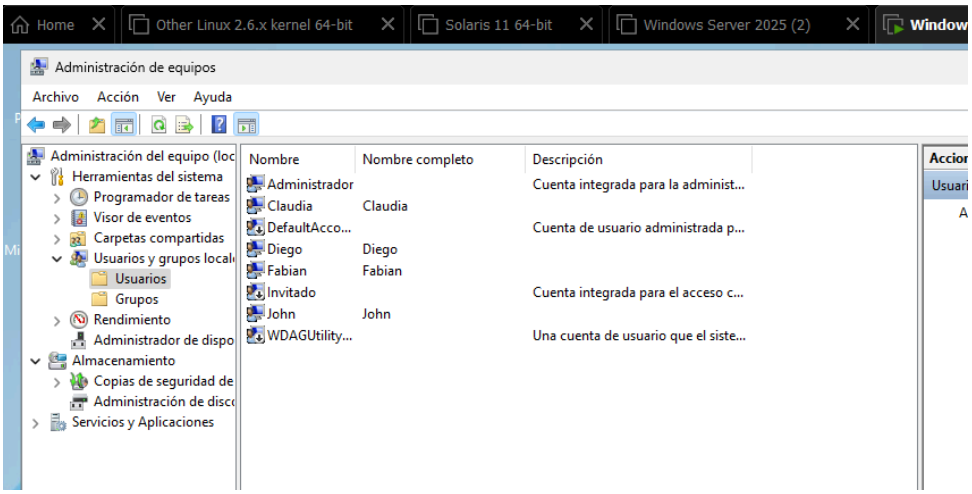


And test




```
C:\Users\Administrador>ping www.google.com

Haciendo ping a www.google.com [142.251.129.196] con 32 bytes de datos:
Respuesta desde 142.251.129.196: bytes=32 tiempo=6ms TTL=119

Estadísticas de ping para 142.251.129.196:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 6ms, Máximo = 6ms, Media = 6ms
Control-C
^C
C:\Users\Administrador>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=15ms TTL=119
Respuesta desde 8.8.8.8: bytes=32 tiempo=5ms TTL=119

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 5ms, Máximo = 15ms, Media = 10ms
Control-C
^C
C:\Users\Administrador>
```

Create four users.



How are permissions managed in Windows Server?

In Windows Server, permissions are managed through **NTFS permissions** and **security policies**. They can be applied to files, folders, users, and groups.

NTFS Permissions (for Files and Folders)

Permissions are configured in the **"Security"** tab of a file's or folder's properties. Some common permissions include:

- **Full Control:** Allows all actions (modify, delete, change permissions).
- **Modify:** Allows changing and deleting files.
- **Read & execute:** Allows opening files and executables.
- **List folder contents:** Only allows viewing the files within a folder.
- **Read:** Allows viewing and opening files without modifying them.
- **Write:** Allows creating and modifying files without deleting them.

Permissions can be inherited from parent folders or applied explicitly.

Folder Sharing (Network Permissions)

If a folder is shared over the network, both types of permissions must be configured:

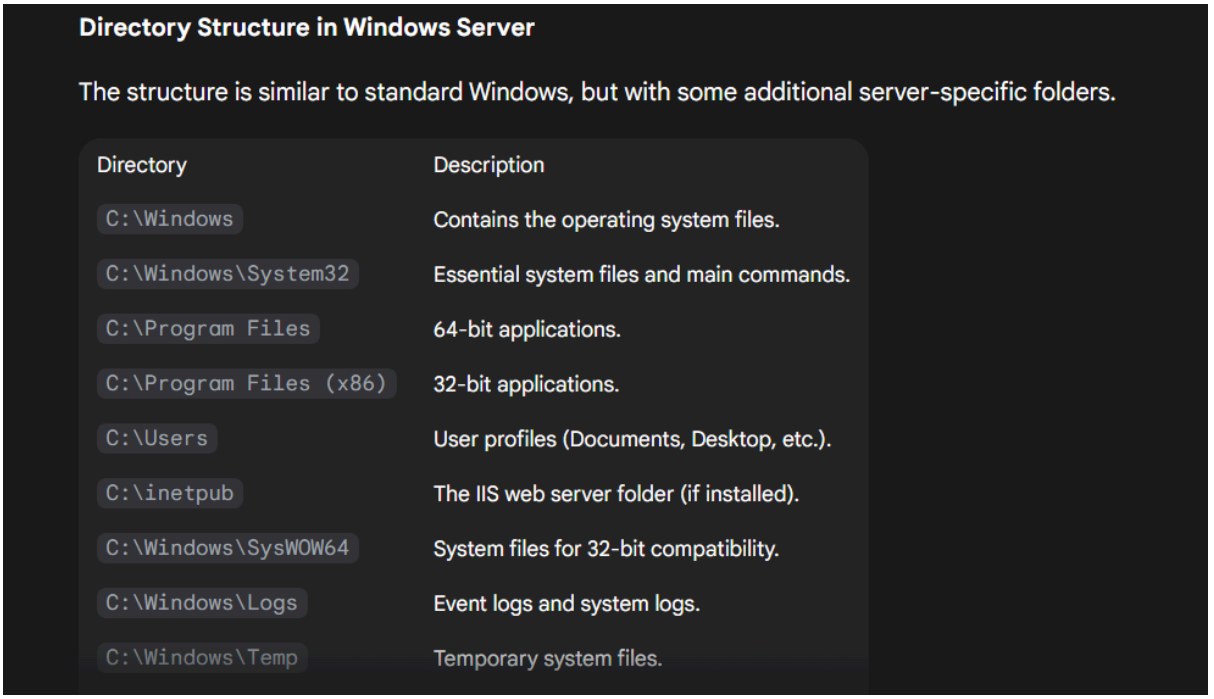
- **NTFS Permissions:** Control access at both the local and network levels.
- **Share Permissions:** Configured in the **"Sharing"** tab and can be:
    - **Full Control**
    - **Change**
    - **Read** It is recommended to set NTFS permissions to be more restrictive and share permissions more general.

Security Policies and Active Directory

To manage permissions in an enterprise environment, Active Directory is used in conjunction with **Group Policies (GPOs)** to apply settings to multiple users or groups. Examples of permissions managed with GPOs include:

- Blocking the execution of certain programs.
- Restricting access to specific drives (C:, D:).
- Applying restrictions on workstations.

What is the directory structure of Windows Server?



Directory Structure in Windows Server

The structure is similar to standard Windows, but with some additional server-specific folders.

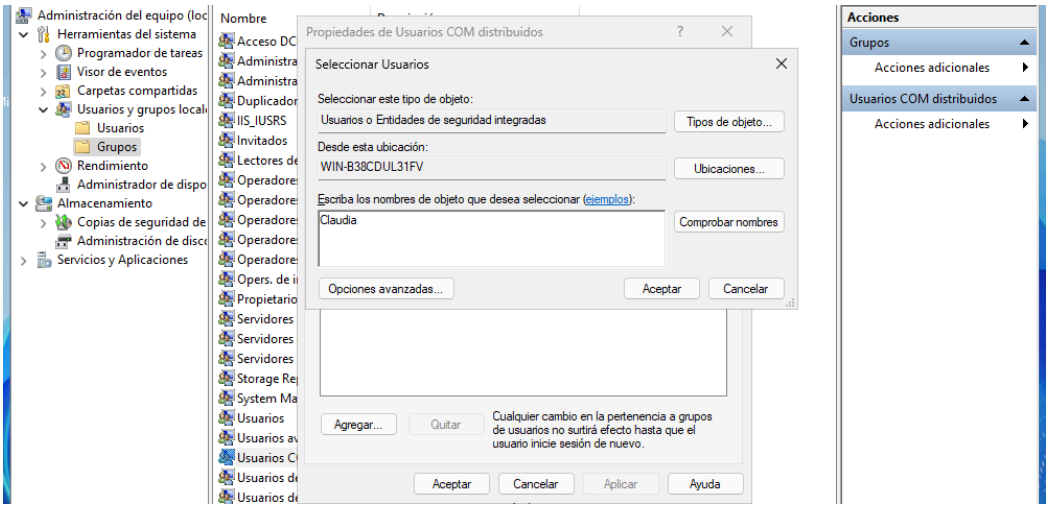
Directory	Description
`C:\Windows`	Contains the operating system files.
`C:\Windows\System32`	Essential system files and main commands.
`C:\Program Files`	64-bit applications.
`C:\Program Files (x86)`	32-bit applications.
`C:\Users`	User profiles (Documents, Desktop, etc.).
`C:\inetpub`	The IIS web server folder (if installed).
`C:\Windows\SysWOW64`	System files for 32-bit compatibility.
`C:\Windows\Logs`	Event logs and system logs.
`C:\Windows\Temp`	Temporary system files.

Source: Google Gemini

What is the Windows Registry? What is its purpose? How is it edited? What type of information does it store?

It is a hierarchical database that stores configuration settings and options in Windows operating systems. It contains the configuration of the low-level components of the operating system, as well as the applications running on the platform. The kernel, device drivers, services, SAM, the user interface, and third-party applications use the registry. The registry also provides access to counters to profile system performance. It is used to manage and edit settings related to user preferences and system configuration. To edit the Windows registry, you need to use a tool called "regedit." This program will allow you to access all entries and modify them according to your needs. To do this, simply open the Windows 10 search engine and type "REGEDIT" in it, followed by pressing Enter.

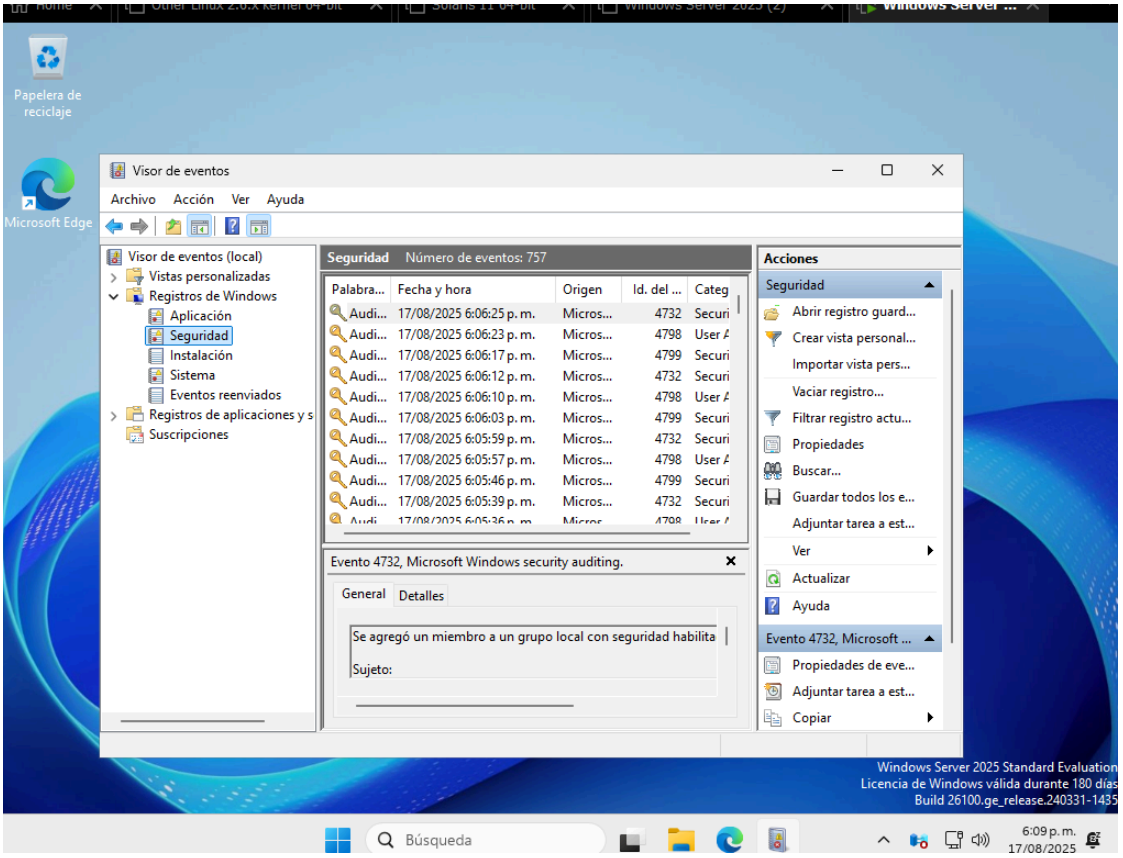
Assign different permission levels to the created users.



How are Windows Server logs accessed?

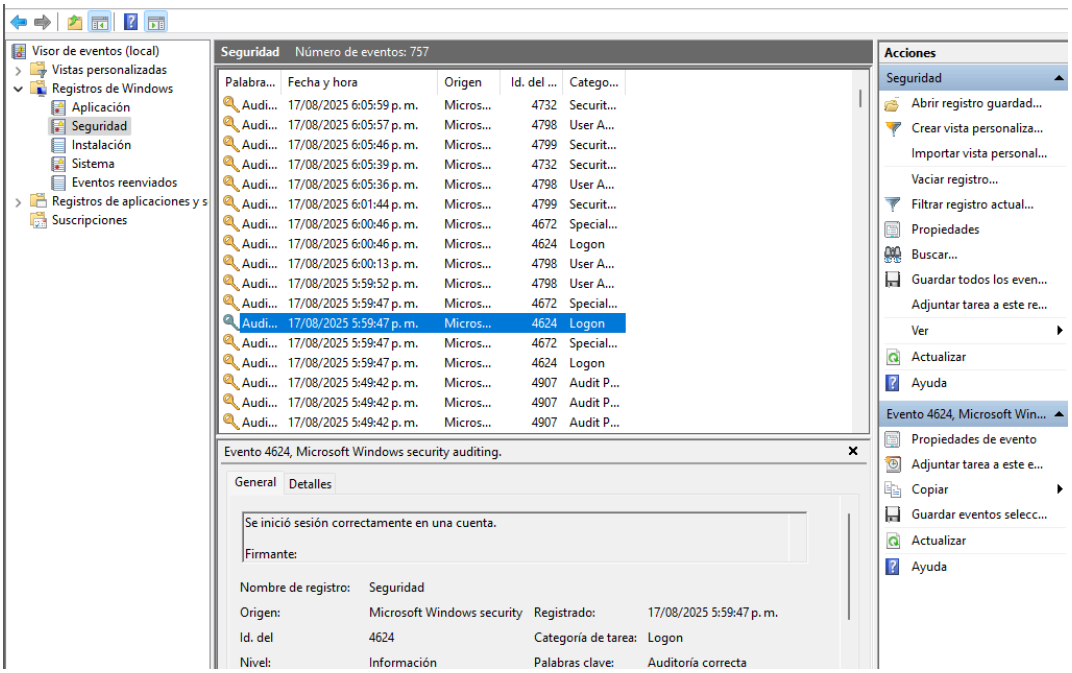
To review the logs in Windows Server, type Event Viewer in the search bar.

There you will find the Windows logs, where you can review various features such as application logs, security logs, system logs, and more.



Identify server log events such as failed login attempts, user access, and unauthorized actions (e.g., attempting to delete a file without permission).

We identify the time at which the login was made

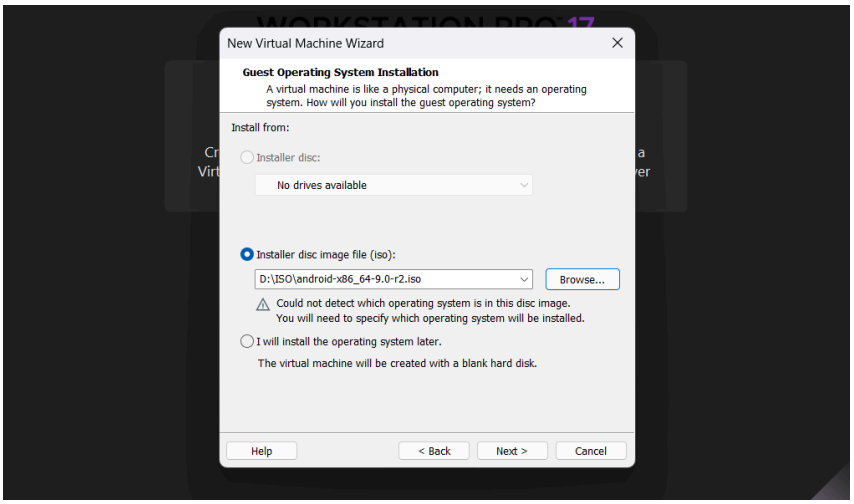


5. Android Installation

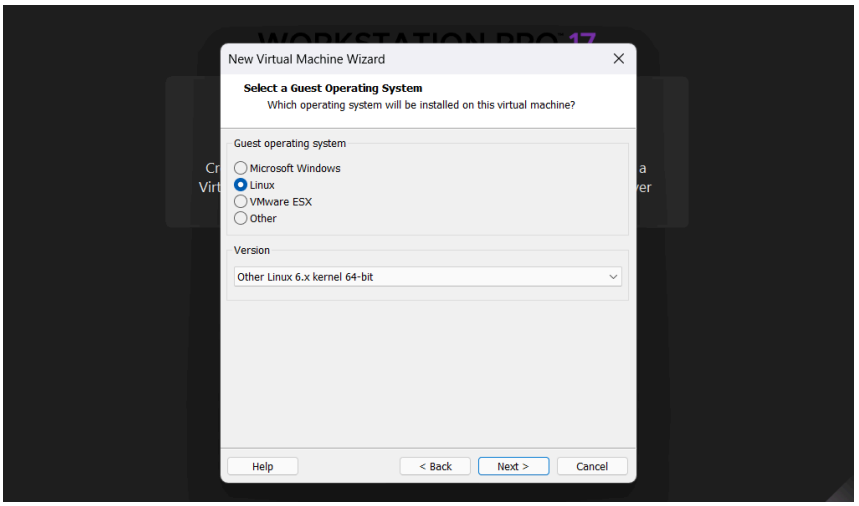
Create a new virtual machine using VMware and install Android.

We chose the typical installation

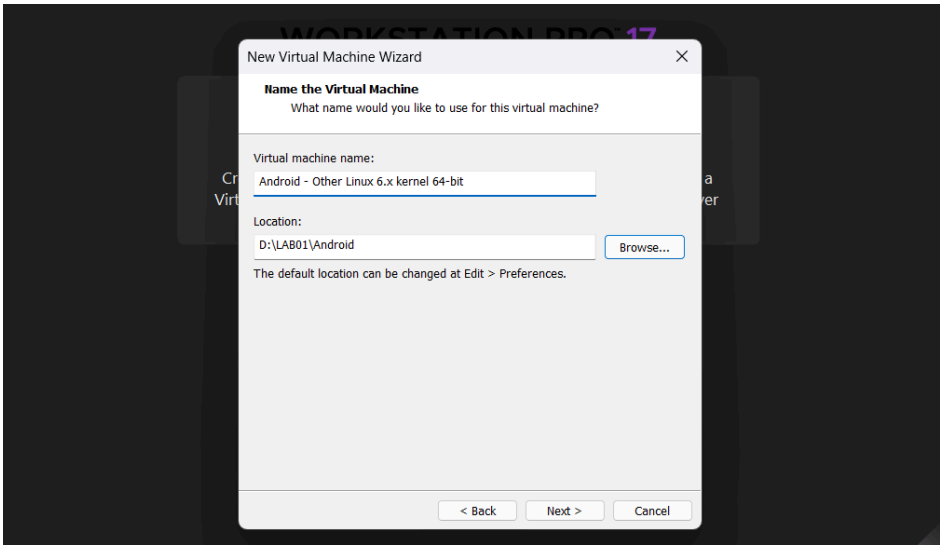
We choose the Android ISO



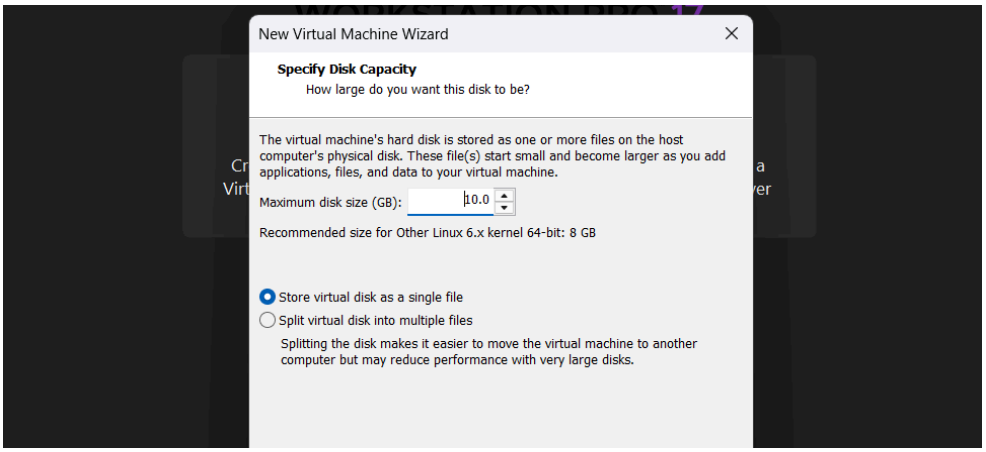
We created a virtual machine of type Linux and its version Other Linux 6.x (64-bit).



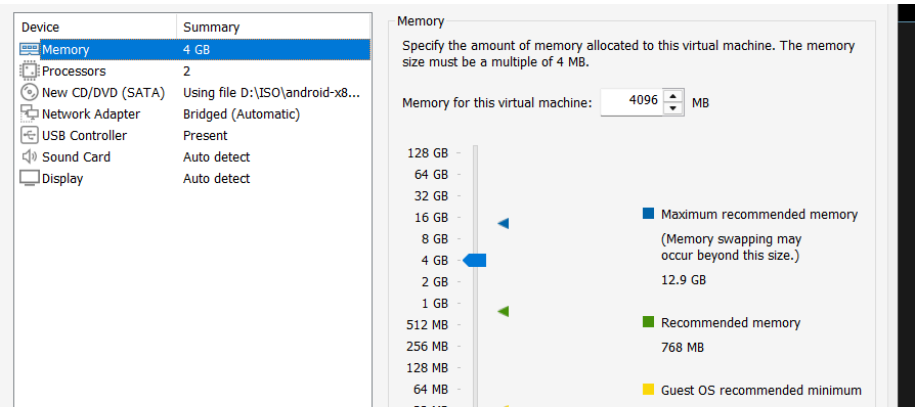
We name the machine and save it on the external hard drive.



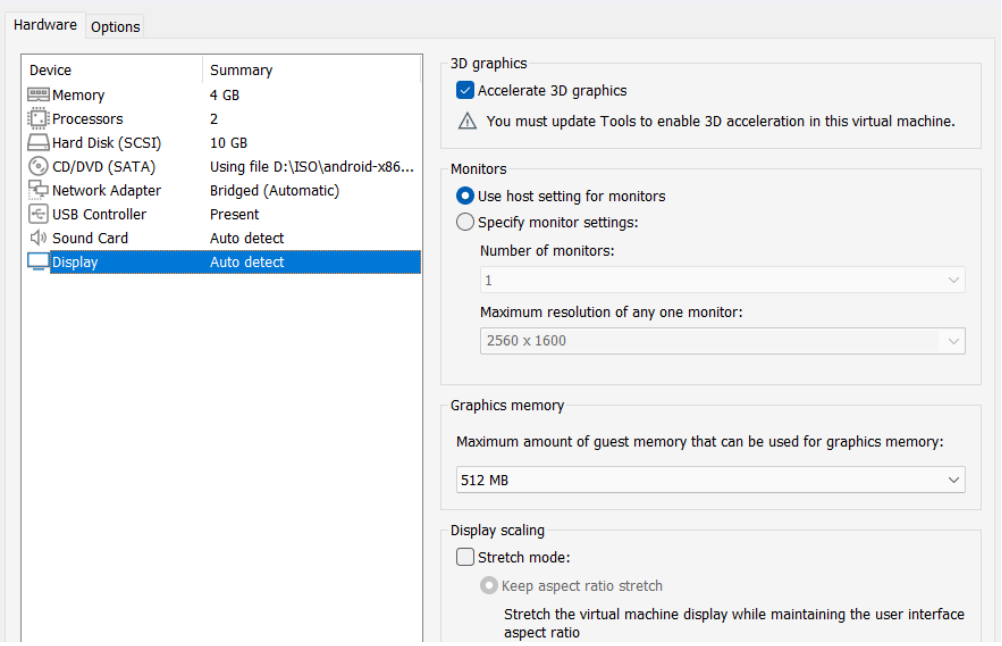
We select 10GB for memory and store virtual disk as a single file



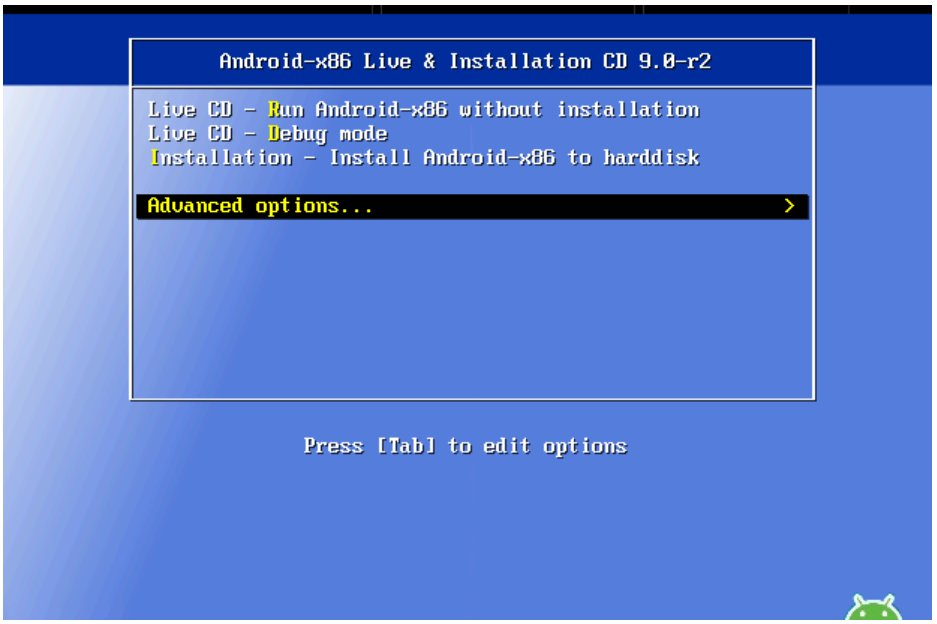
We assign 4GB of RAM, 2 processors and a bridge adapter



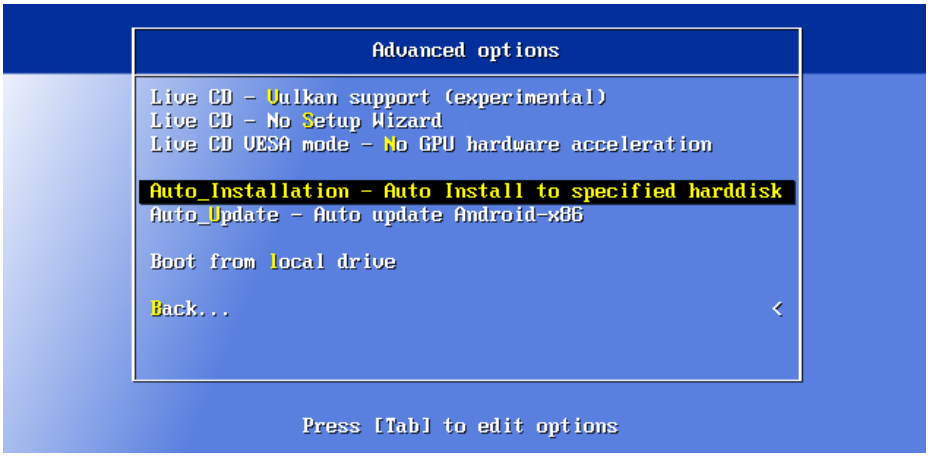
We activate accelerate 3D graphics and assign 512 MB of video memory



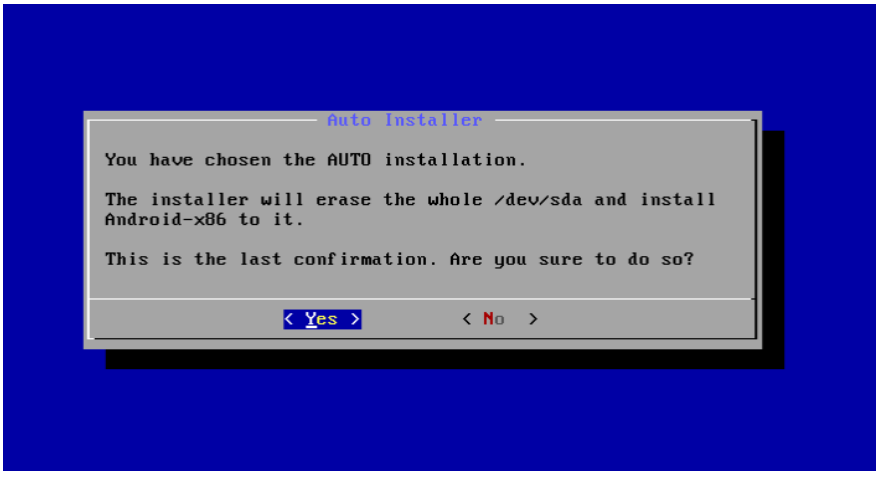
Once the machine is turned on, we go to advanced options



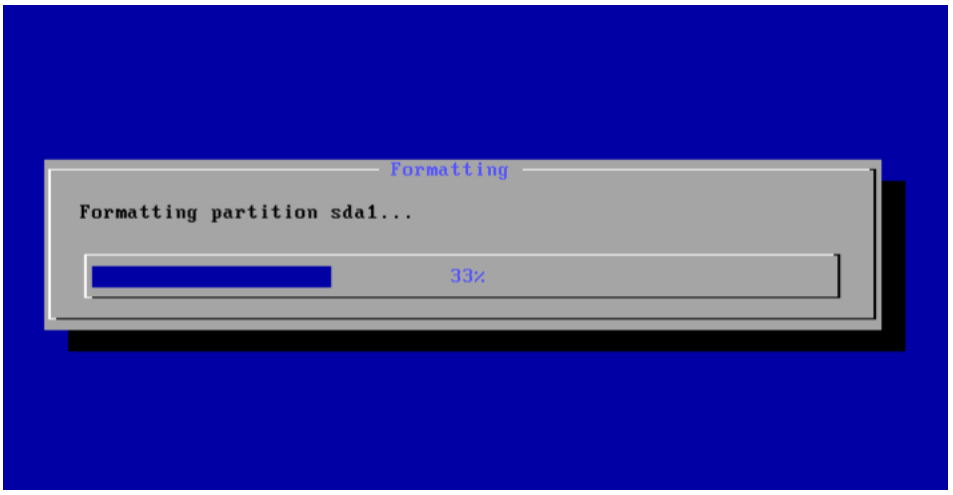
We will use the self-installation of the system.

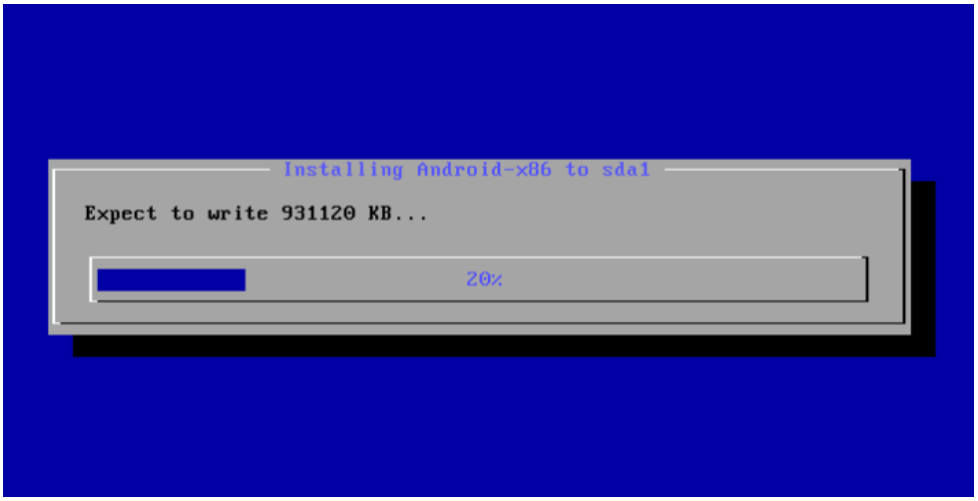


It will ask us for a confirmation which we accept.



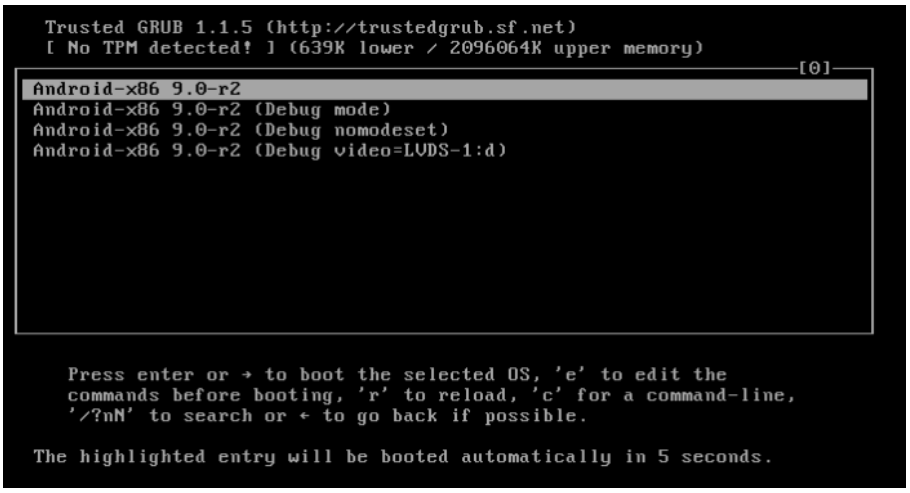
The disk is formatted and partitioned. After that, we'll see how to install the operating system.





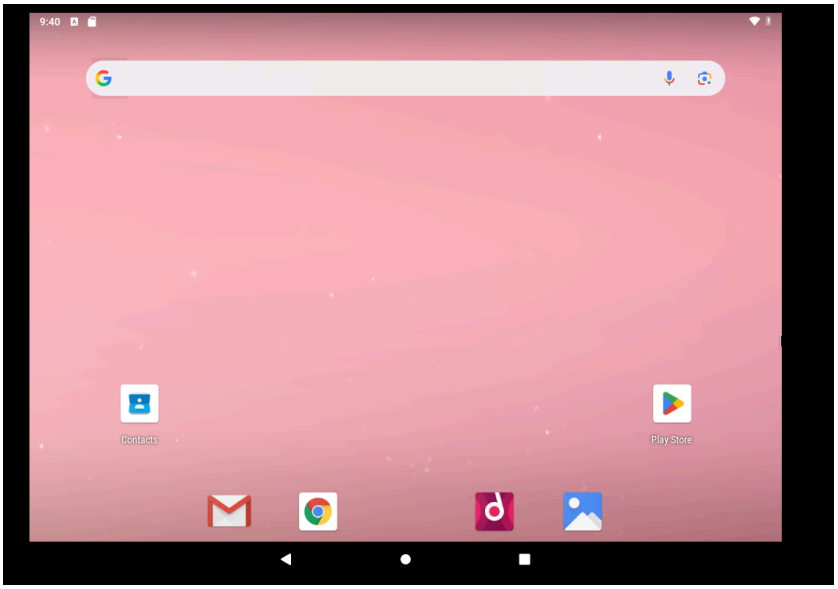
We restart the virtual machine by extracting the ISO file from the CD-ROM drive.

We select the installed operating system. Select Debug nomdeset due to vmware performance issues



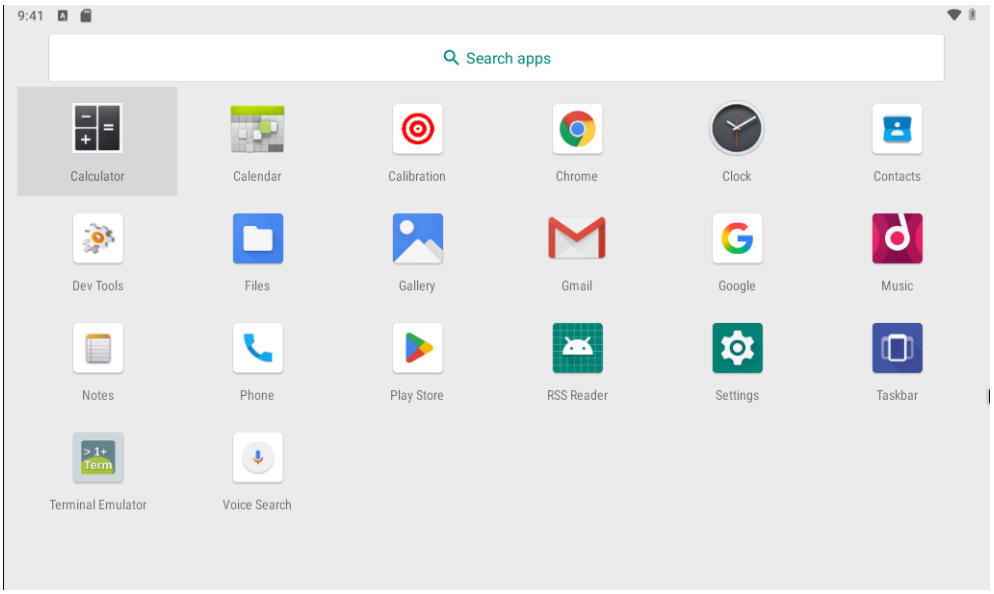
We wait for it to load

As we can see, Android has started successfully.

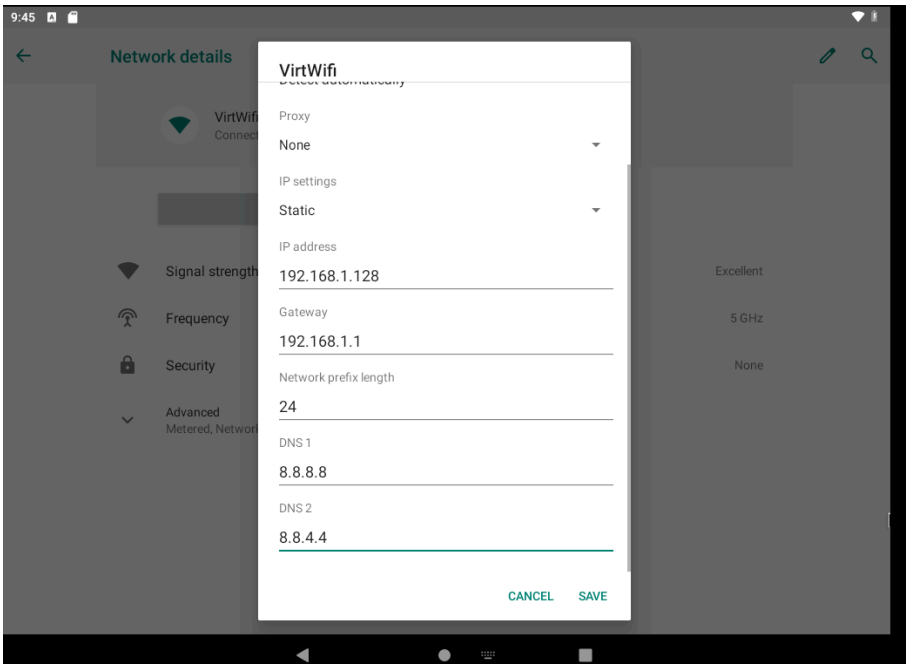


Configure network connectivity and test its operation as done with Linux.

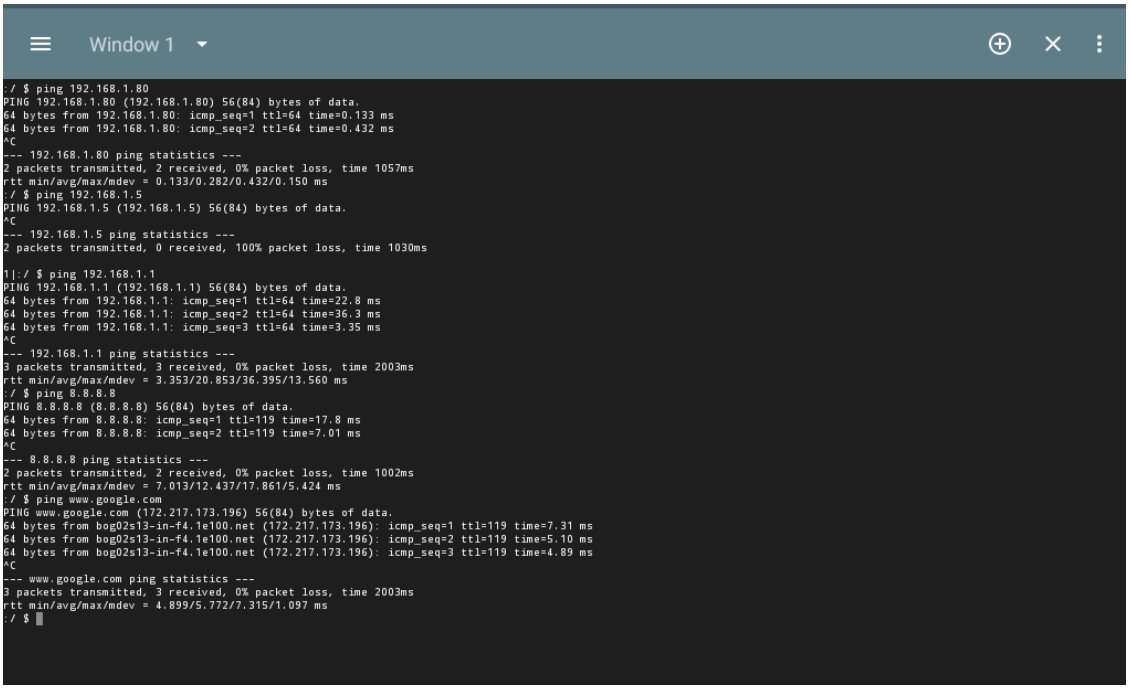
We access the configuration to configure the network.



We modify the network parameters as indicated in the laboratory guide.



Testing Network Configuration



6. Command Line Knowledge

- **What is the shell?**

  The **shell** is a program that provides an interface between the user and the operating system.

  It interprets commands entered by the user and sends them to the kernel for execution.

  It can be interactive (user types commands) or used for scripting (running command files automatically).

- **Which shells are supported by Slackware, Solaris, and Windows?**

  Slackware (Linux):

  - Bash (Bourne Again Shell) – default
  - Sh (Bourne Shell)
  - Tcsh, Zsh, Dash (optional)

  Solaris (Unix):

  - Bourne Shell (sh)
  - C Shell (csh)
  - Korn Shell (ksh, default in many Solaris systems)

Windows:

- Command Prompt (cmd.exe)
- PowerShell (modern, supports scripting with .NET)

- **What are their differences?**

Unix/Linux shells (Bash, sh, ksh, csh, zsh):

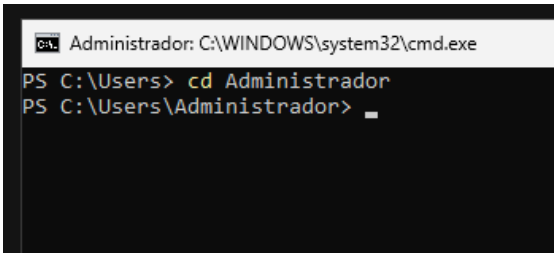
- Support scripting with loops, conditions, and variables.
- Strong integration with Unix tools (grep, awk, sed).
- Follow POSIX standards for portability.

- **Windows shells:**
  - **Command Prompt (cmd.exe):** Limited scripting, simple commands.
  - **PowerShell:** More advanced, object-oriented, integrates with Windows system administration

    In short: Unix/Linux shells are text-stream oriented, Windows PowerShell is object-oriented.

- **Commands in Linux, Unix, and Windows**
  - **Change directories:**

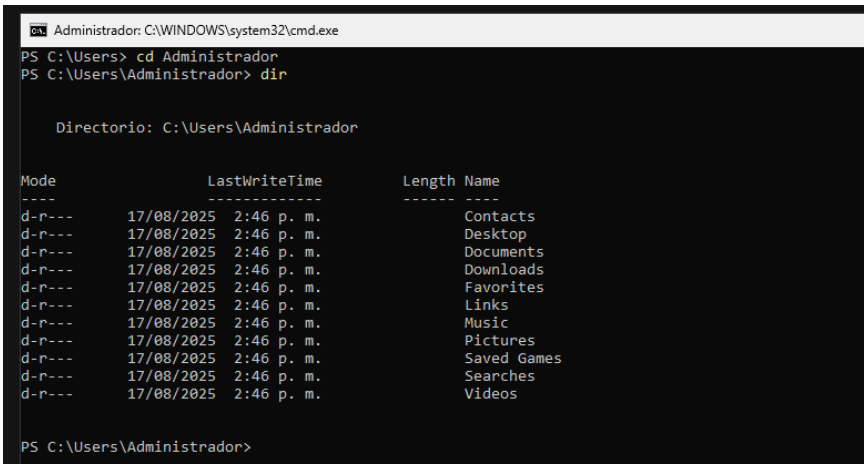
    Windows/Linux/Unix:
    cd [folder]



  - **List directory structures and files**

    Linux/Unix:
    ls [options] [route]

Windows:
dir [options] [route]

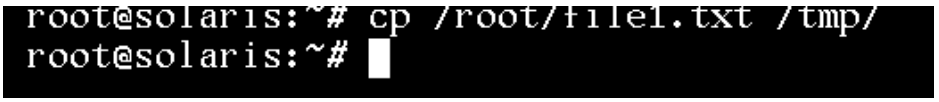


- **Copy or move a file**

  Linux/Unix:

  cp [file origin] [destination] # Copy

  mv [file origin] [destination] # Move



  Windows:

  copy [origin] [destination] # Copy

  move [origin] [destination] # Move

- **View file content without editing**

  Linux/Unix:

  cat [file]

  less [file]

  more [file]

```
root@solaris:~# cat /etc/resolv.conf

#
# _AUTOGENERATED_FROM_SMF_V1_
#
# WARNING: THIS FILE GENERATED FROM SMF DATA.
#   DO NOT EDIT THIS FILE.  EDITS WILL BE LOST.
# See resolv.conf(5) for details.

domain  localdomain
nameserver      8.8.8.8
nameserver      8.8.4.4
root@solaris:~#
```

Windows:

type [file]

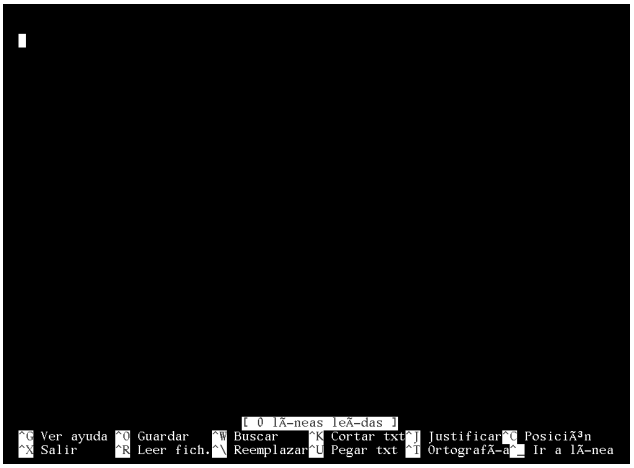
more [file]

```
PS C:\Users\Administrador> type C:\Windows\System32\drivers\etc\hosts
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
PS C:\Users\Administrador>
```

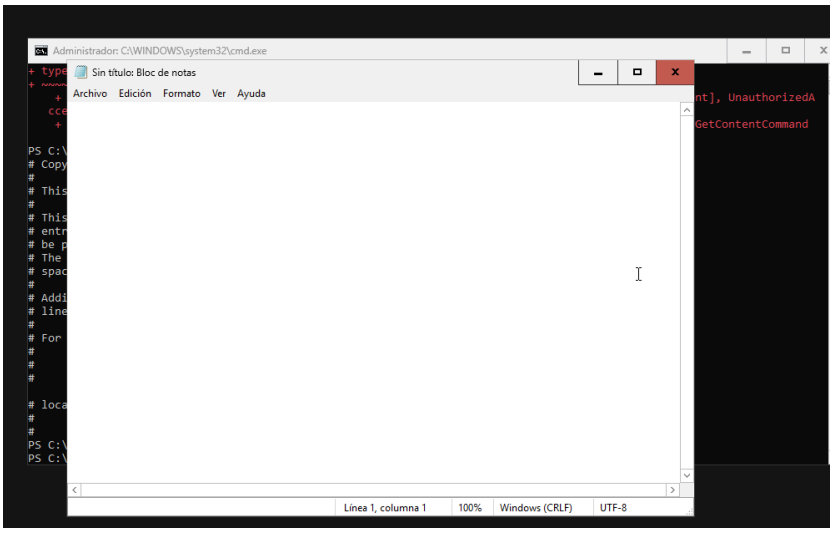
- **Edit a file**

Linux/Unix:
nano [file]

vi [file]



Windows:
notepad [file]



- **View the first and last lines of a file**

Linux/Unix:

head -n 10 [file] # First 10 lines

tail -n 10 [file] # Last 10 lines

```
root@solaris:~# head -n 10 /etc/resolv.conf

#
# _AUTOGENERATED_FROM_SMF_V1_
#
# WARNING: THIS FILE GENERATED FROM SMF DATA.
#   DO NOT EDIT THIS FILE.  EDITS WILL BE LOST.
# See resolv.conf(5) for details.

domain  localdomain
nameserver      8.8.8.8
root@solaris:~#
```

Windows:

Get-Content [file] -Head 10

Get-Content [file] -Tail 10

```
PS C:\Users\Administrador> Get-Content C:\Windows\System32\drivers\etc\hosts -Head 10
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
PS C:\Users\Administrador>
```

- **Search for a word in a line**

Linux/Unix:
grep "keyword" [file]

```
root@solaris:~# grep nameserver /etc/resolv.conf
nameserver      8.8.8.8
nameserver      8.8.4.4
root@solaris:~#
```

Windows:
findstr "keyword" [file]
Select-String "keyword" [file]

```
PS C:\Users\Administrador> findstr "localhost" C:\Windows\System32\drivers\etc\hosts
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
PS C:\Users\Administrador>
```

- **Locate a file in the system**

Linux/Unix:

find / -name [file]

locate [file] # if `mlocate` database is available

```
root@solaris:~# find /etc -name "passwd"
/etc/pam.d/passwd
/etc/passwd
/etc/default/passwd
root@solaris:~#
```

Windows:

dir /s /p file.txt

(or PowerShell)
Get-ChildItem -Recurse -Filter file.txt

```
PS C:\Users\Administrador> Get-ChildItem C:\Windows -Recurse -Filter notepad.exe


    Directorio: C:\Windows


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----       5/09/2024  11:11 p. m.         360448 notepad.exe
```