



## **Department Of Software Engineering**

**Operating System and System Programming**

**Section B Individual Assignment**

**Title : Solaris OS Installation**

**Name**

**Yaschilal Enyew**

**ID Number**

**BDU1602719**

**Submitted To Lec Wendimu B.**

**Submission date 16/08/17**

## TABLE CONTENT

### CONTENTS

<b>1. Introduction-----</b>	<b>3</b>
<b>2. Motivation -----</b>	<b>4</b>
<b>3. Objectives-----</b>	<b>5</b>
<b>4. Requirements for solaris Os-----</b>	<b>7</b>
<b>5. Installation steps of solaris Os-----</b>	<b>8</b>
<b>6.Issues/Problem faced-----</b>	<b>29</b>
<b>7. solution for the problem-----</b>	<b>32</b>
<b>8. File System Support-----</b>	<b>33</b>
<b>9. Advantage and Disadvantage-----</b>	<b>34</b>
<b>10. Conclusion-----</b>	<b>37</b>
<b>11. Future outlook /recommendation-----</b>	<b>37</b>
<b>12what is virtualization -----</b>	<b>39</b>
<b>13. System call implementations-----</b>	<b>40</b>
<b>14. References-----</b>	<b>43</b>

# Introduction

In today's world, operating systems are like the brain of our computers — they control everything, from how apps run to how we connect to the internet or save files. One interesting part of learning about operating systems is trying out different ones to see how they work. That's exactly what I've done in this assignment.

For this task, I installed Oracle Solaris 11.4, which is a powerful and advanced type of UNIX-based operating system. It's mostly used in big companies, data centers, and servers because it's super reliable and secure. Solaris might not be very common for everyday users, but it's great to learn about it because it's built for professional-level tasks and gives you a better idea of how enterprise systems work.

Since I didn't want to install it directly on my real computer (which could be risky), I used a virtual machine instead. Virtual machines let us run other operating systems inside our own computer safely, without affecting anything important. It's like having a computer inside a computer — which is perfect for testing and learning.

This introduction covers the goal of my assignment: to explore how Solaris works, practice installing it, and get real experience using a virtualization tool like VirtualBox. It was definitely a hands-on learning moment, and I faced a few issues along the way — but that's part of the learning too.

## Background

Solaris has actually been around for a long time. It was originally made by a company called Sun Microsystems, and later it was taken over by Oracle. Over the years, it's been used a lot in places where computers need to be stable, secure, and able to handle a lot of tasks — like in servers, data centers, and research labs.

What makes Solaris stand out is that it's not just a regular operating system like Windows or even Linux. It's more focused on the enterprise level — which means it's built for serious business. It comes with powerful tools like the ZFS file system, strong security features, and it supports things like virtualization and resource control right out of the box.

Even though most people don't use Solaris on their personal laptops, it's still important for us as software engineering students to get familiar with systems like this. It helps us understand how big tech companies manage their infrastructure and keep their systems running smoothly. Learning Solaris also gives us a chance to practice using command-line interfaces and understand how professional operating systems are built and managed.

## Motivation

The main reason I chose to install Solaris in a virtual machine was to get some real, hands-on experience without worrying about messing up my actual computer. Installing an operating system like Solaris directly on hardware can be risky — especially if you're still learning. So using a virtual machine made it way easier and safer to explore.

Another reason is that I wanted to challenge myself. Solaris isn't the easiest operating system to work with, but that's actually why I found it interesting. It's used in serious environments like servers and data centers, and learning how to install and interact with it felt like a step toward becoming more comfortable with professional-level systems.

I also wanted to get better at using virtualization tools like VirtualBox. These tools are really useful in real-world IT and development jobs. By going through the full process — downloading the ISO, setting up the VM, fixing issues, and actually logging into Solaris — I learned a lot more than I would have by just reading about it.

Plus, doing everything myself (and not just copying someone else) helped me understand how operating systems actually work behind the scenes. That's something I can definitely use in future courses, internships, or projects.

## Objectives

The main goal of this assignment was to actually go through the full process of installing an operating system — not just in theory, but step by step in a virtual machine. I chose Oracle Solaris 11.4 and used VirtualBox to install it, which gave me the chance to learn by doing.

Here's what I wanted to achieve with this task:

1. Understand the full installation process of Solaris, from creating a virtual machine to setting up network settings, choosing the right disk, and creating a user account.
2. Get comfortable using virtualization tools like Oracle VirtualBox, since they're important in both academic and real-world tech environments.
3. Learn how to deal with problems during installation — things like freezing screens, missing network options, or input not working. This helped improve my troubleshooting skills.
4. Explore what makes Solaris unique, especially its powerful ZFS filesystem and how it manages system resources differently from other operating systems.
5. Document the entire process with clear steps, explanations, and screenshots so that I (and possibly others) can refer back to it later.
6. Compare the pros and cons of Solaris in a virtual setup and understand where it fits in the world of operating systems.

By the end of this project, I hoped to walk away with more confidence in setting up advanced OS environments — and I definitely learned a lot in the process.

# Requirements

To install and run Solaris smoothly inside a virtual machine, I needed both the right hardware and software setup. Here's what I used for this assignment:

## i. Hardware Requirements

I used my personal laptop, which had enough power to handle a virtual machine without slowing things down too much. These were

- **Laptop:** HP EliteBook
- **Processor:** Intel Core i7, 8th Generation
- **RAM:** 16 GB
- **Storage:** 512 GB SSD
- **Architecture:** 64-bit
- **Other:** Virtualization enabled in BIOS (important for running VMs)

**Note:** You could technically run Solaris on a system with less RAM (like 4 or 8 GB), but for a smoother experience, especially in VirtualBox, having 16 GB made a big difference.

## ii. Software Requirements

Here's the software I installed and used for the whole process:

- **Oracle VM VirtualBox** – This is the tool I used to create and run the virtual machine. It's free and works well for testing OSes.
- **Oracle Solaris 11.4 ISO file** – I downloaded the official ISO from Oracle's website.
- **Host OS: Windows 11 (64-bit)** – This is the main operating system running on my laptop.

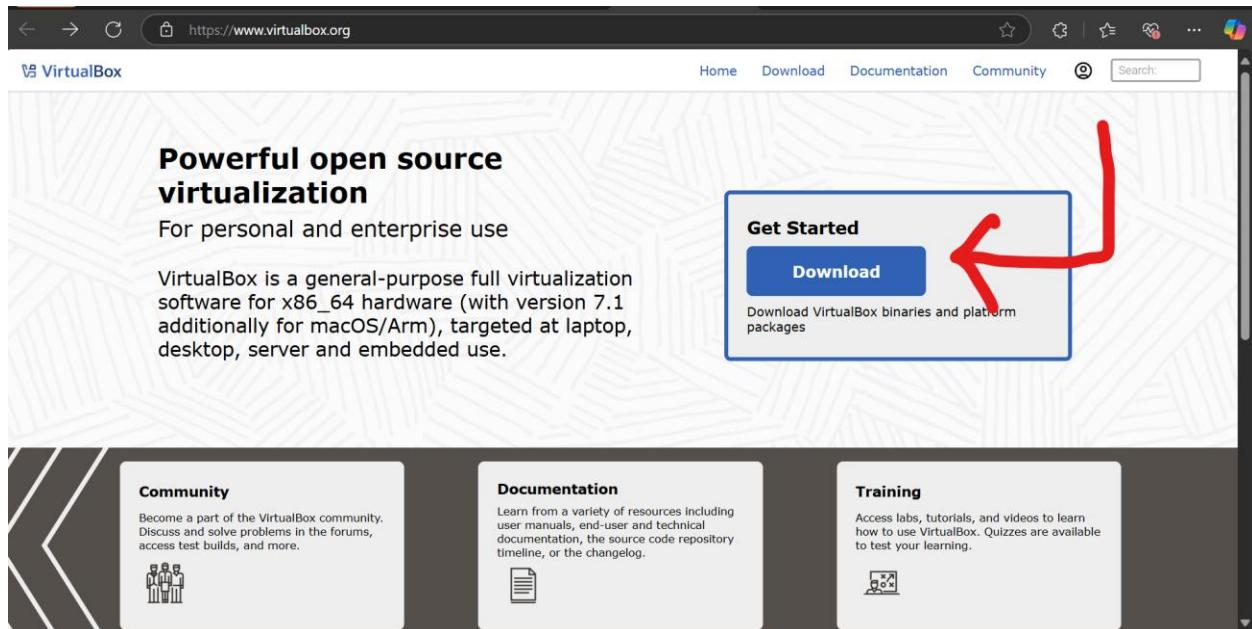
## Installation Steps

Here's how I installed Oracle Solaris 11.4 using Oracle VM VirtualBox on my laptop. I tried to make each step clear and easy to follow.

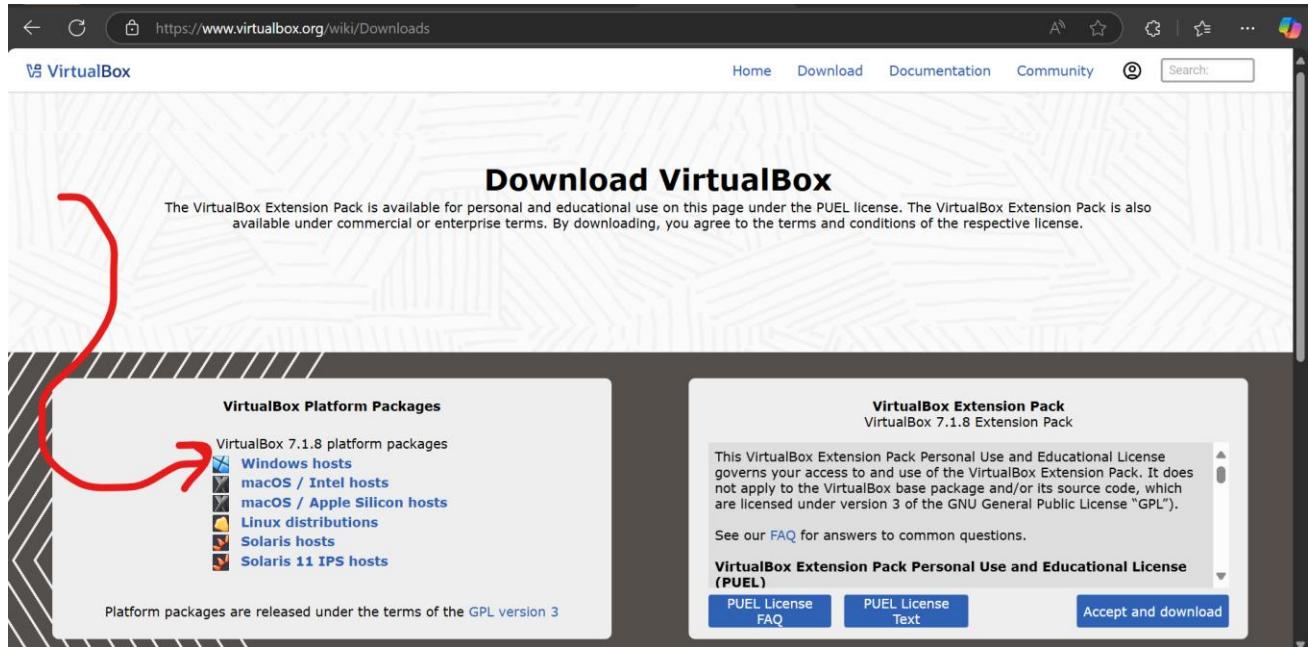
### Step 1: Download and Install VirtualBox

Before anything else, I had to install the tool that lets me create a virtual machine — in this case, **Oracle VM VirtualBox**.

1. I went to the official website: <https://www.virtualbox.org>



2. I clicked on the "**Downloads**" section.
3. because my laptop runs Windows, I chose the **Windows hosts** version.

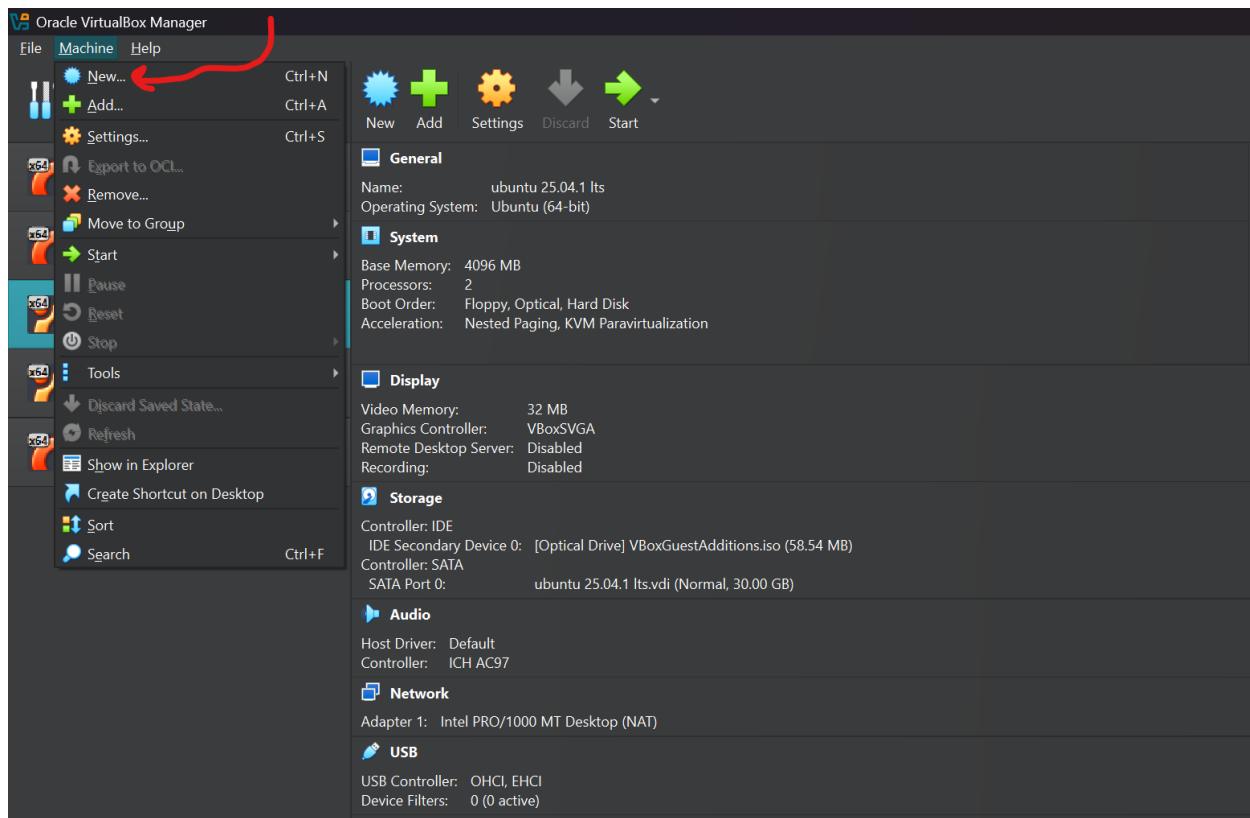


4. After downloading the installer, I opened it.
5. when the installation was done, I launched VirtualBox to make sure it was working properly.

## Step 2: Create a New Virtual Machine

Once VirtualBox was installed, I started by creating a new virtual machine (VM) for Solaris.

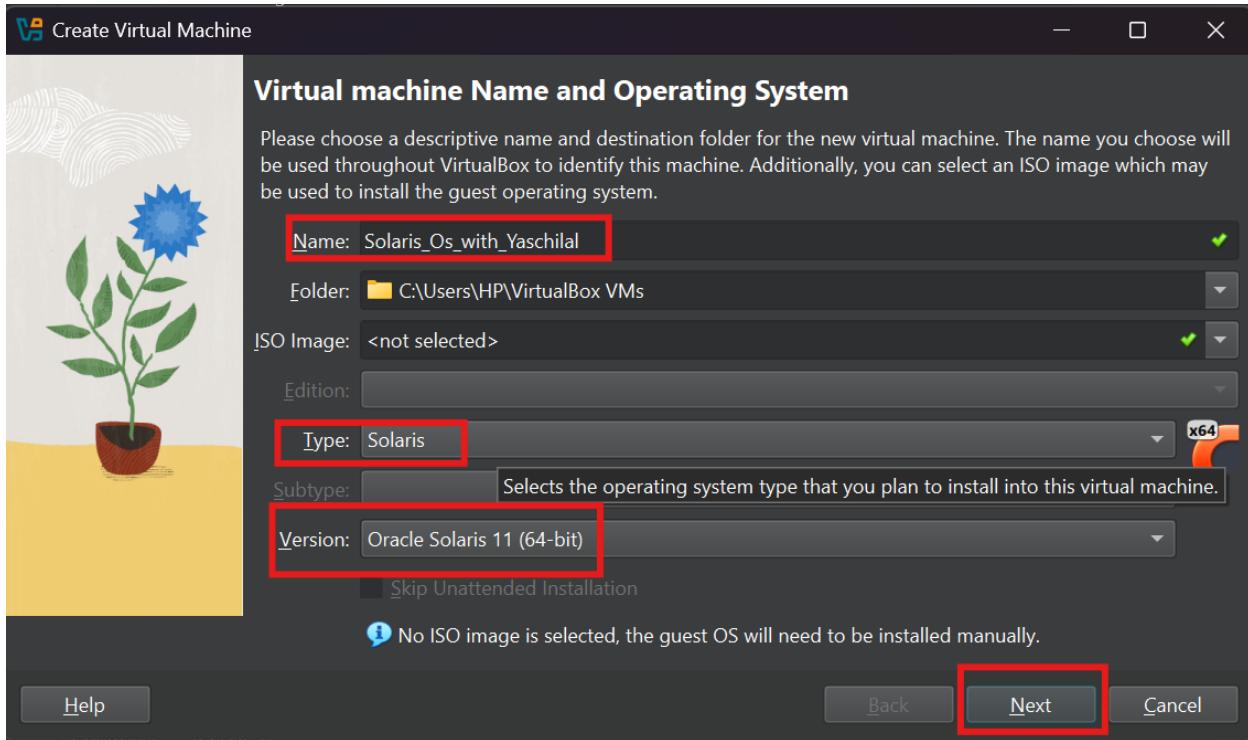
1. I opened VirtualBox and clicked "**New**".



2. For the name, I typed **Solaris\_Os\_with\_Yaschilal**.

3. I selected:

- **Type:** Solaris
- **Version:** Oracle (64-bit) or Solaris 11 (64-bit) — depending on what shows up

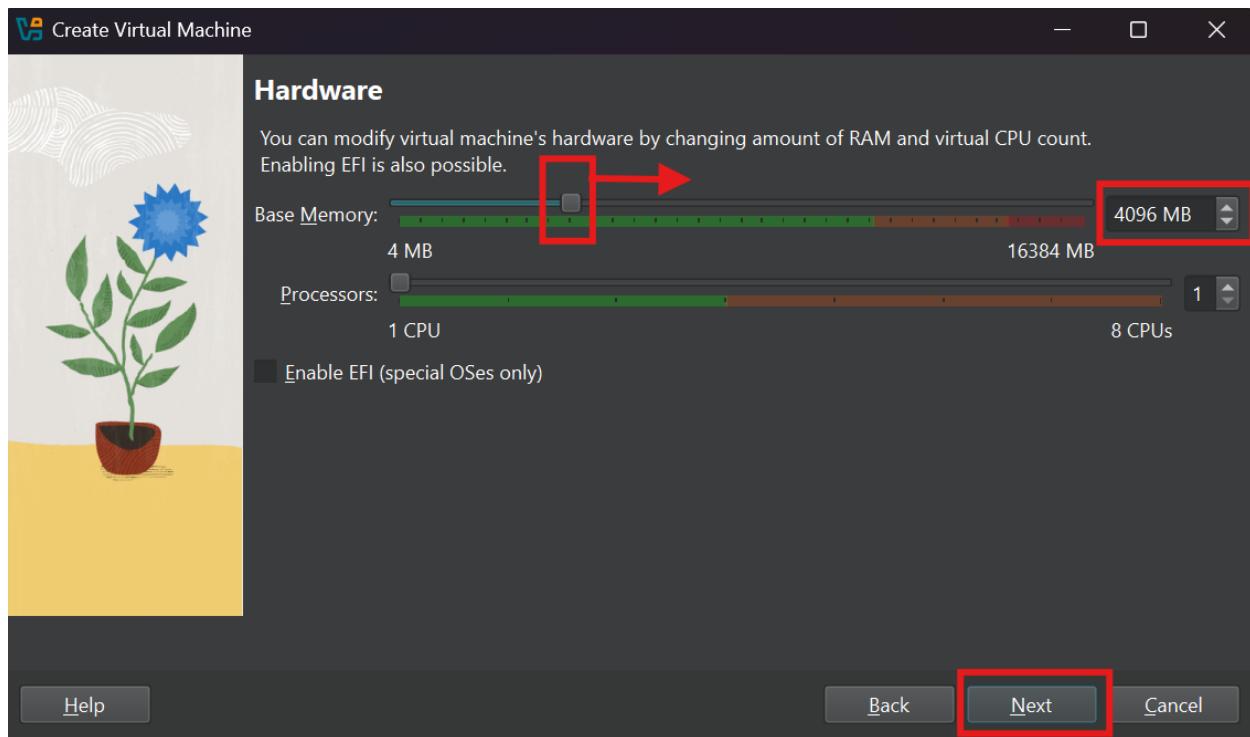


4. Clicked **Next** to continue.

### Step 3: Assign Memory (RAM)

Here I had to give some RAM to the virtual machine.

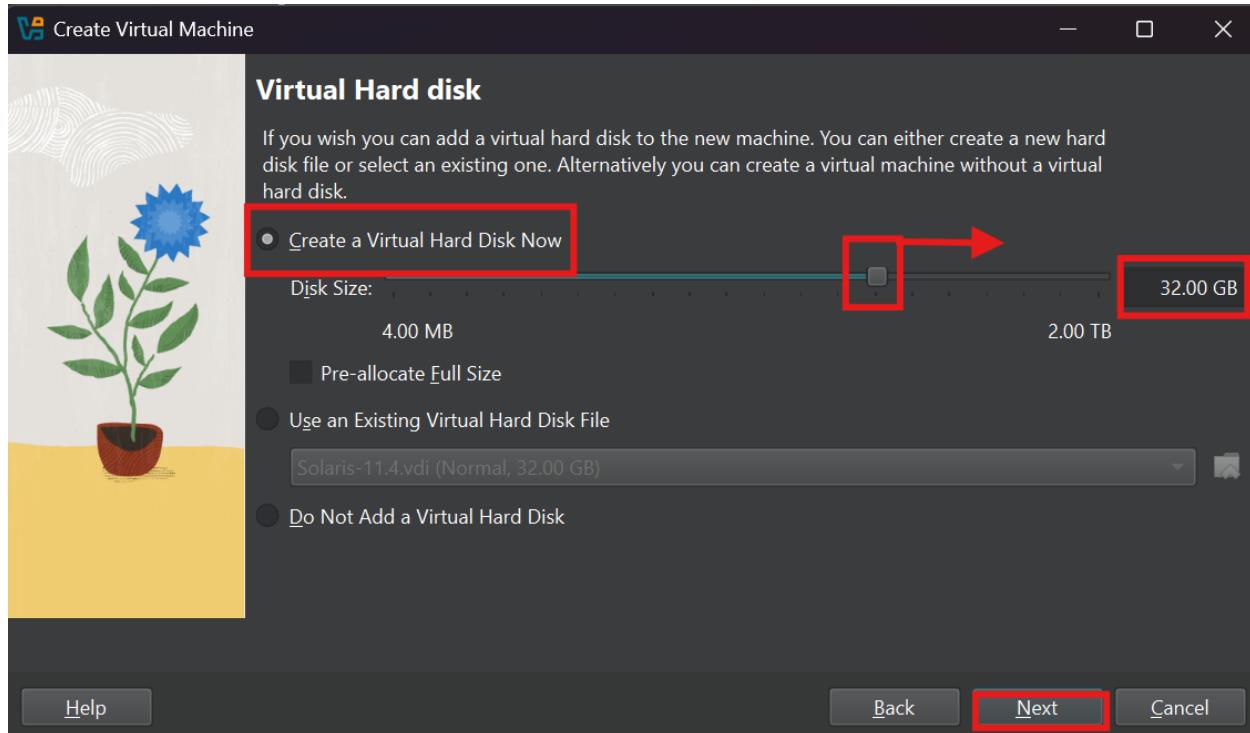
- I moved the slider to **at least 2048 MB (2 GB)**, but since my laptop has 16 GB, I gave it **4096 MB (4 GB)** for better performance.



- Clicked **Next**.

## Step 4: Create a Virtual Hard Disk

1. I chose “**Create a virtual hard disk now**” and clicked **Create**.

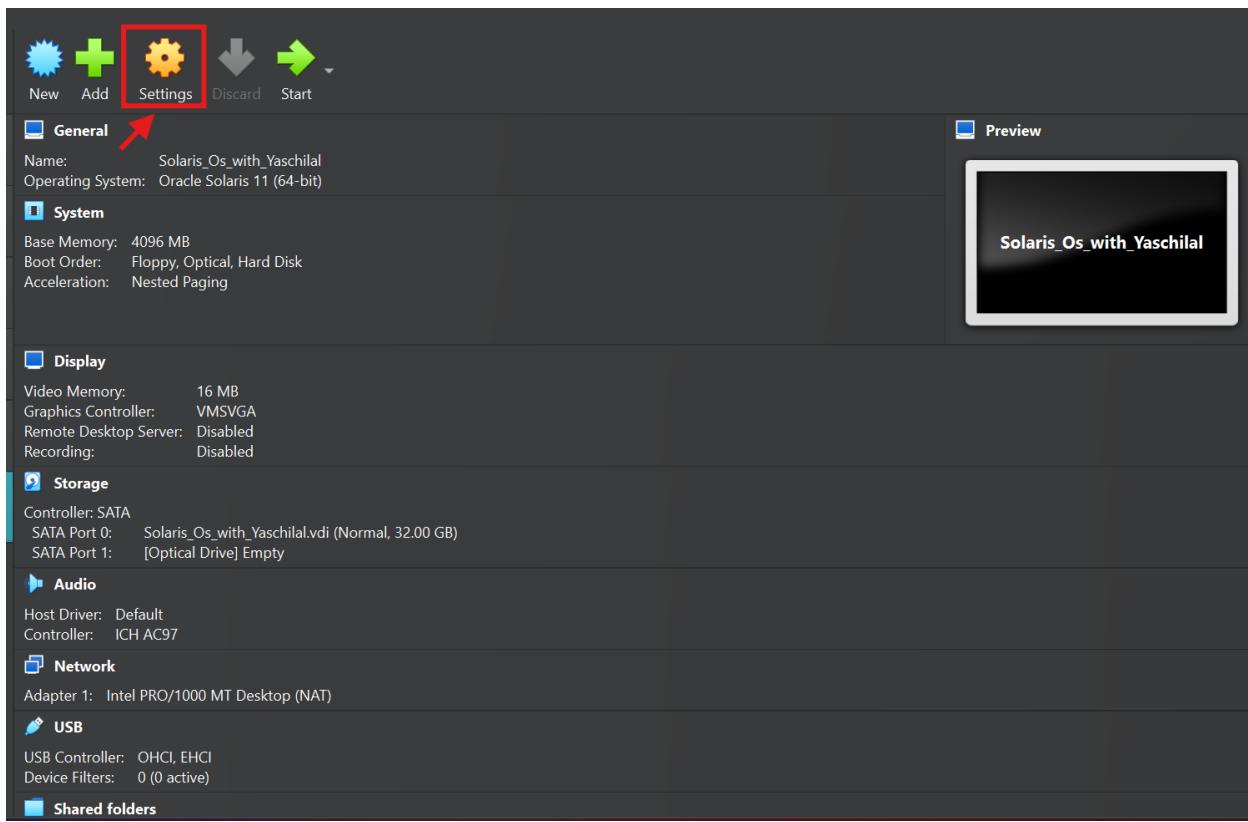


2. For hard disk file type, I picked **VDI (VirtualBox Disk Image)**.
3. Then I selected **Dynamically allocated** (so it grows as needed).
4. I set the size to **at least 20 GB** — I used **30 GB** to be safe.
5. Clicked **Next** to finish the setup.

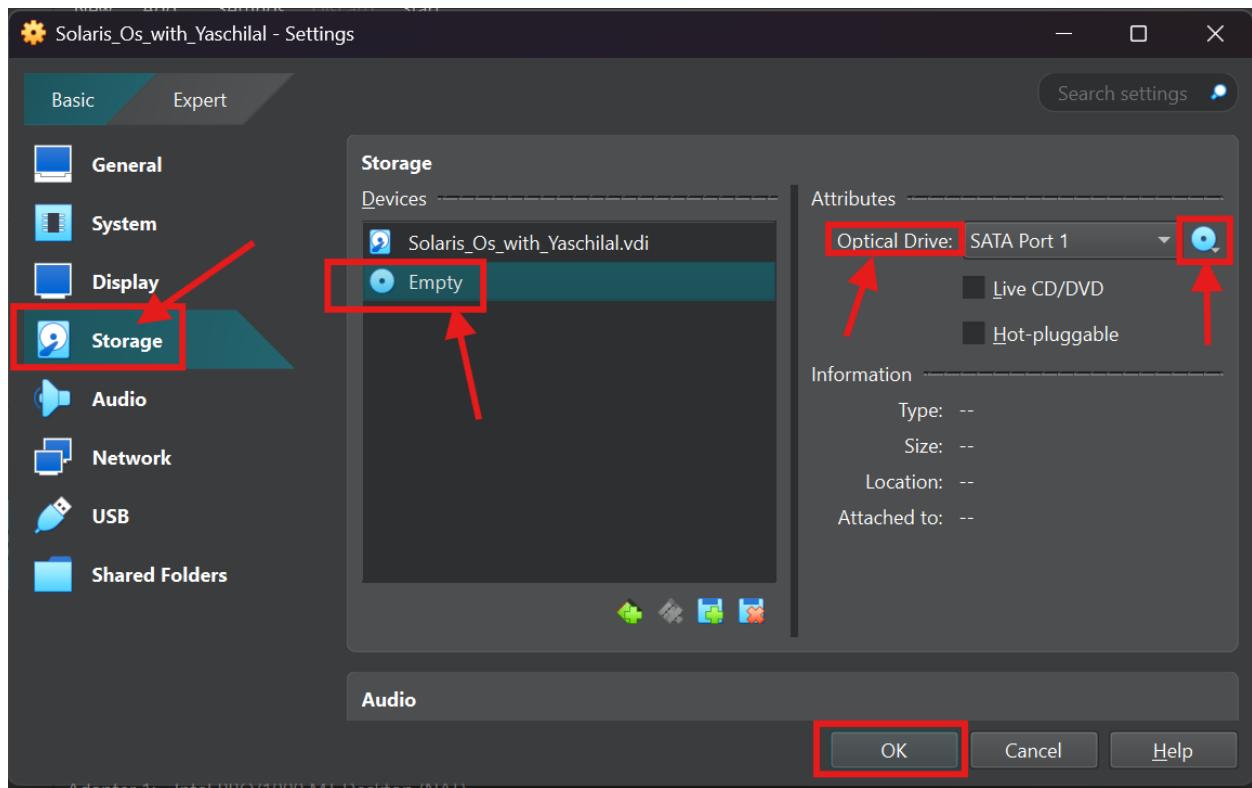
## Step 5: Mount the Solaris ISO File

Now I had to attach the Solaris installer to the VM so it could boot from it.

1. I selected the Solaris VM from the list and clicked **Settings**.



2. Went to **Storage** → Under **Controller: IDE**, I clicked the **Empty disk** icon.

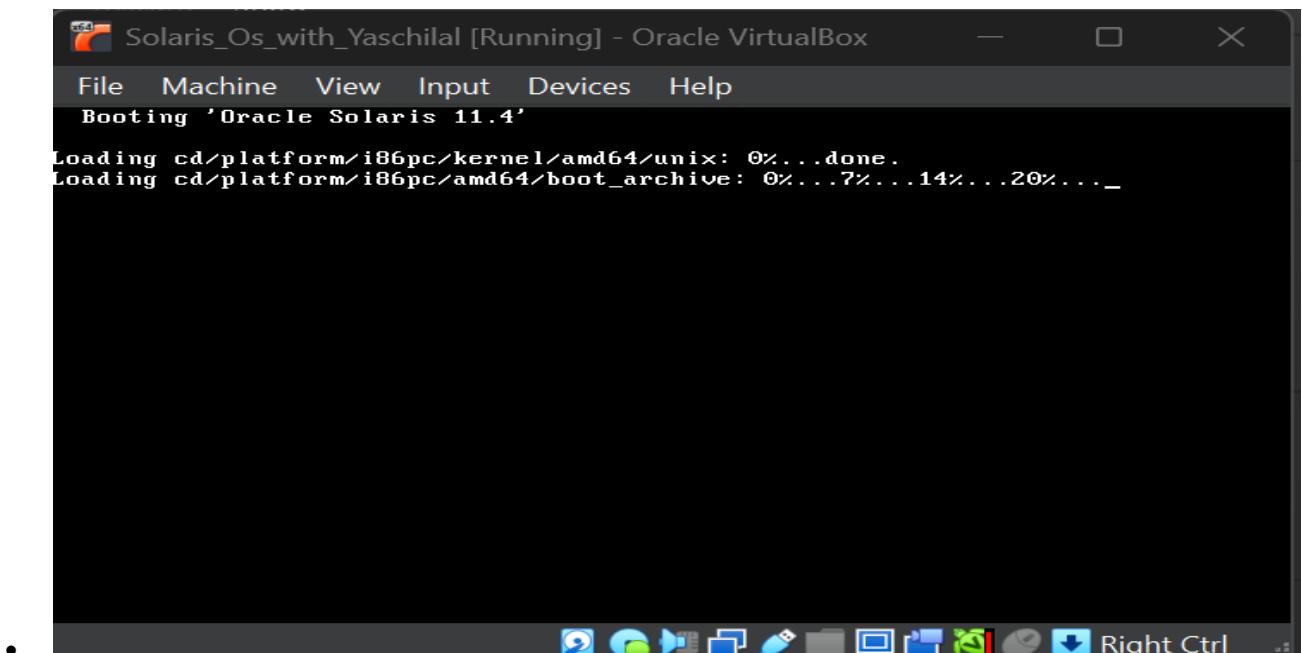


3. On the right, I clicked the small disk icon next to “Optical Drive” and chose “**Choose a disk file...**”.
4. I selected the **Solaris 11.4 ISO file** I downloaded earlier.
5. Clicked **OK** to save the changes.

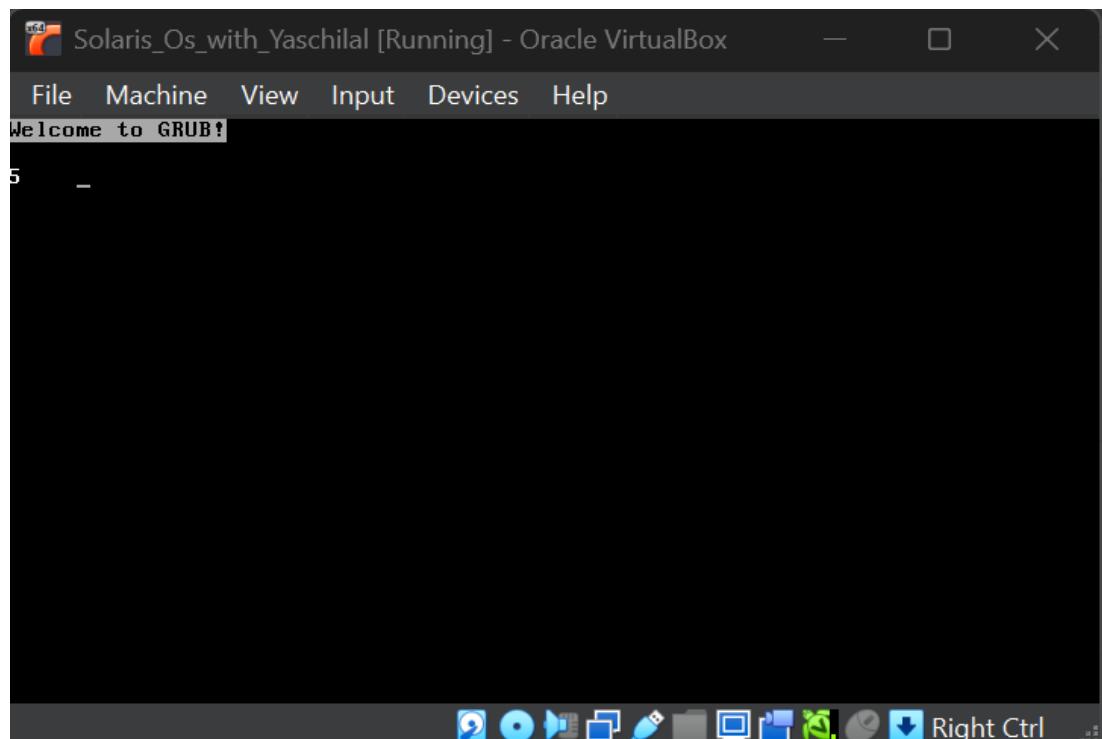
## Step 6: Start the Virtual Machine

I clicked **Start** on the VirtualBox menu to boot up the virtual machine.

- The system began loading the Solaris installer.

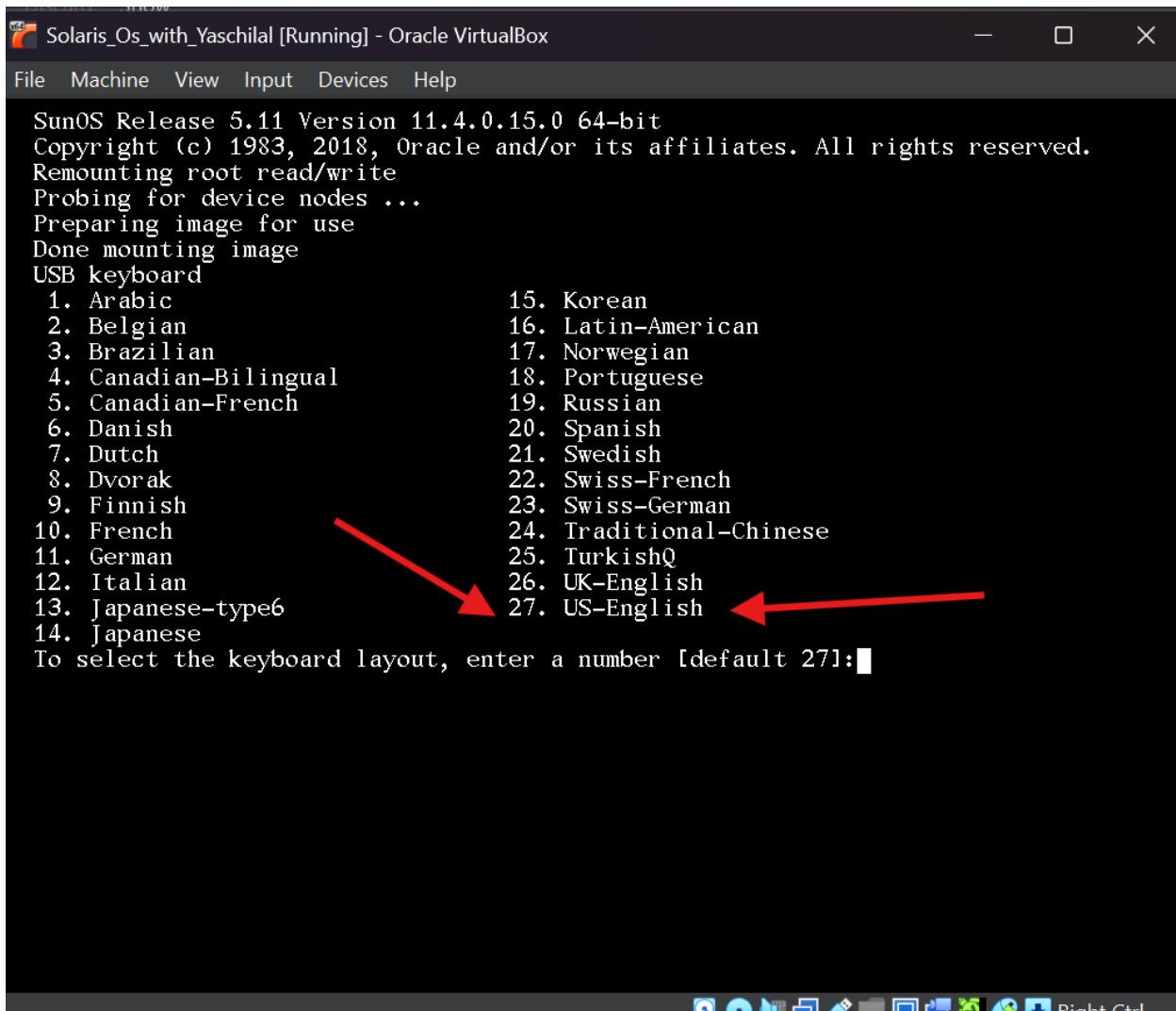


- After a few moments, I saw the welcome screen and installation menu.



## Step 7: Select Keyboard Layout

- The installer asked me to choose a keyboard layout.



- I went with the default: **US English** (usually option 27).
- Pressed **F2** to continue.

```
Welcome to the Oracle Solaris installation menu

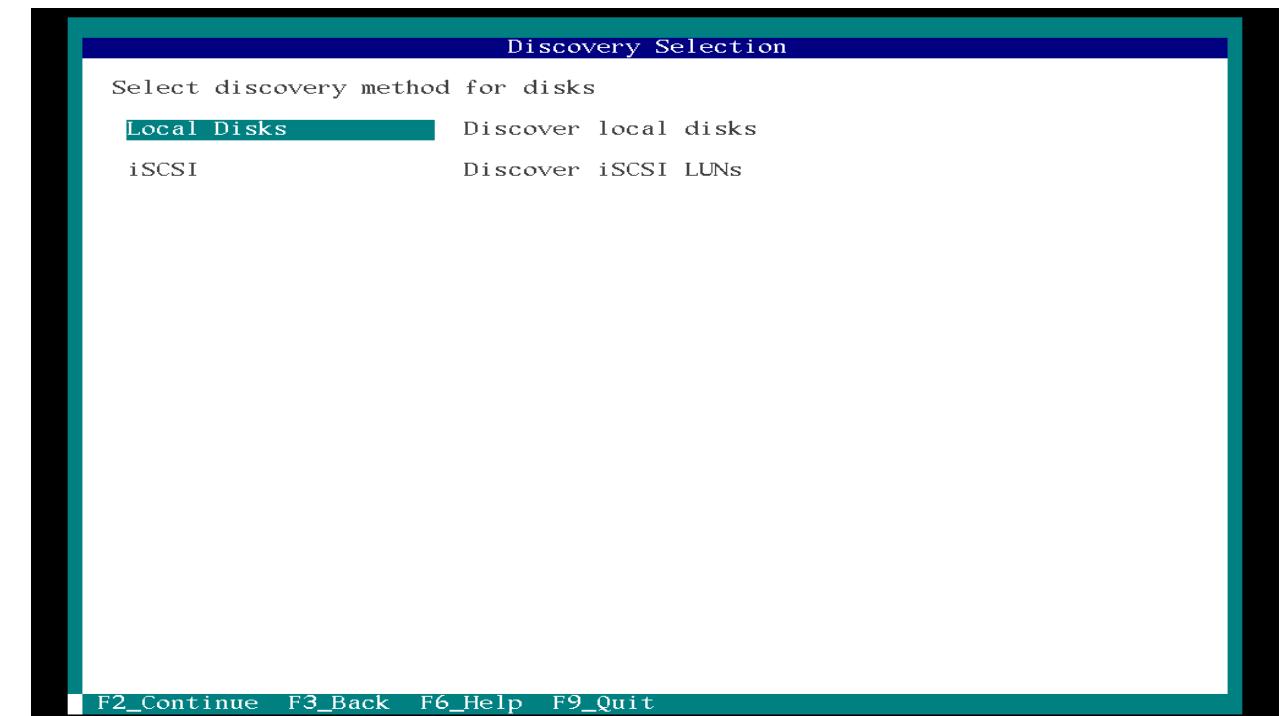
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot

Please enter a number [1]:
SUNW-MSG-ID: SUNOS-8000-LG, TYPE: Alert, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Apr 24 12:43:40 UTC 2025
PLATFORM: VirtualBox, CSN: VirtualBox-75c1a36a-bac2-491b-a037-2bfe710712c6, HOST
NAME: vbox
SOURCE: software-diagnosis, REV: 0.2
EVENT-ID: 3ef68554-9e16-4698-8060-c307375b1122
DESC:
AUTO-RESPONSE: No automated response available
IMPACT: Oracle Solaris is not running with Spectre Vulnerability Mitigation Enab
led
REC-ACTION: Update the CPU with Spectre capable microcode. Please refer to the a
ssociated reference document at http://support.oracle.com/msg/SUNOS-8000-LG for
the latest service procedures and policies regarding this diagnosis.
1
■
```

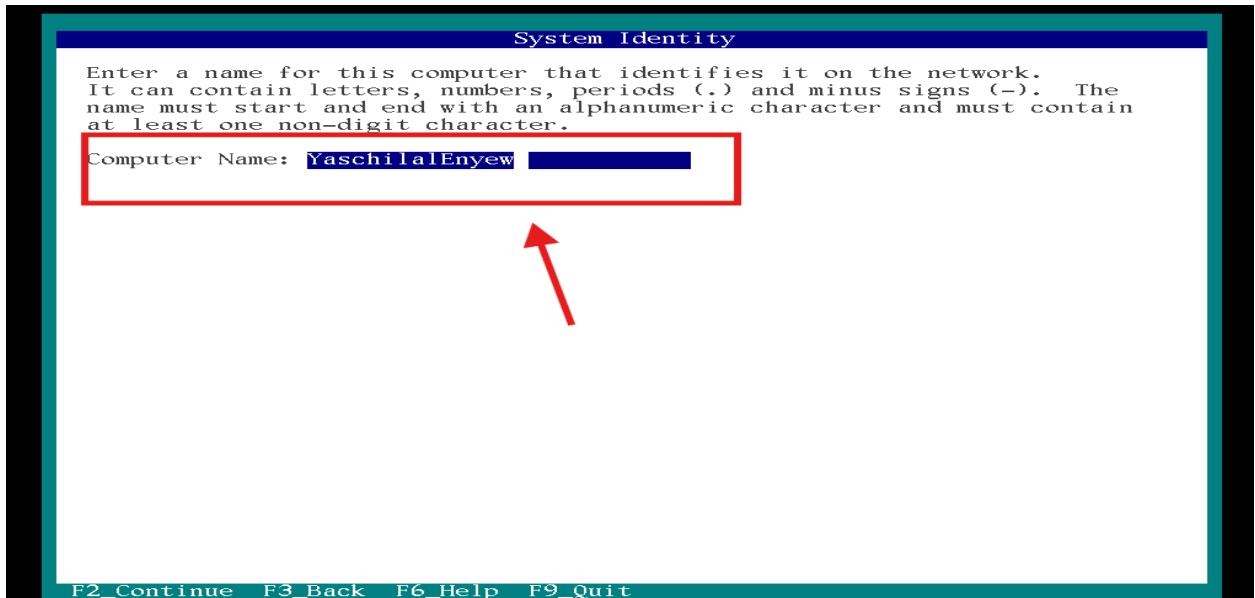
## Step 8: Select Disk Discovery Method

Next, the installer asked how to find available disks for installation.

1. I chose “**Local Disks**” (option 1) when asked how to discover disks.



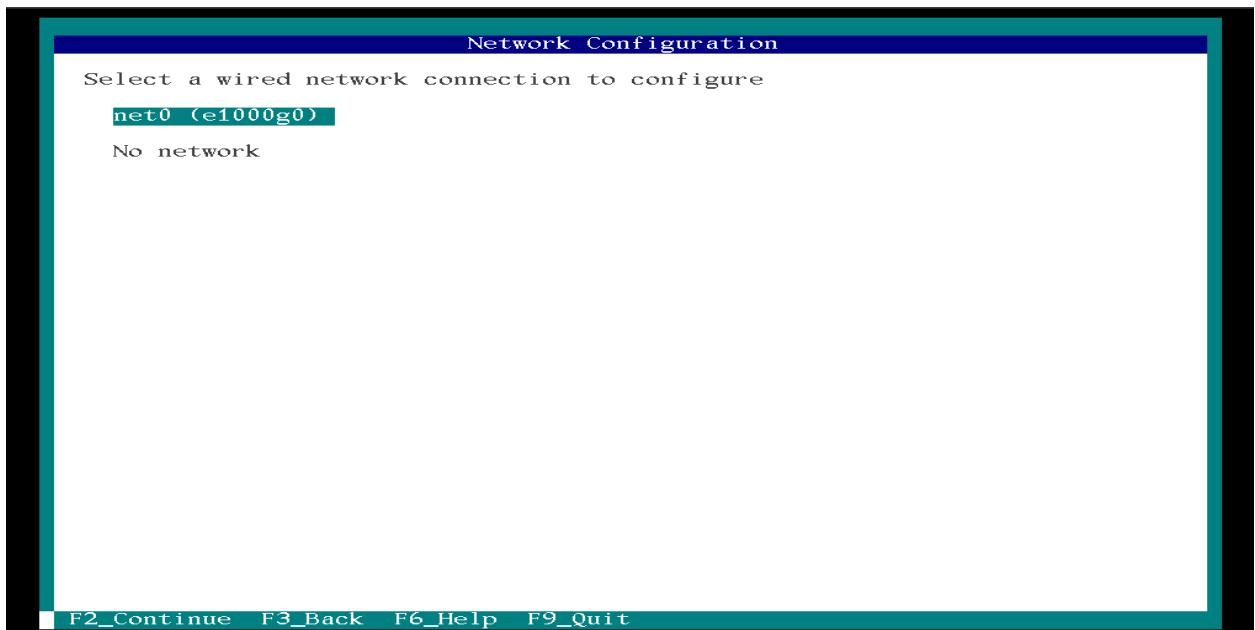
- 2.
3. Solaris scanned for available virtual disks and proposed using the whole disk.
4. pressed **F2** to continue.
5. Then I entered computer name like YaschilalEnyew



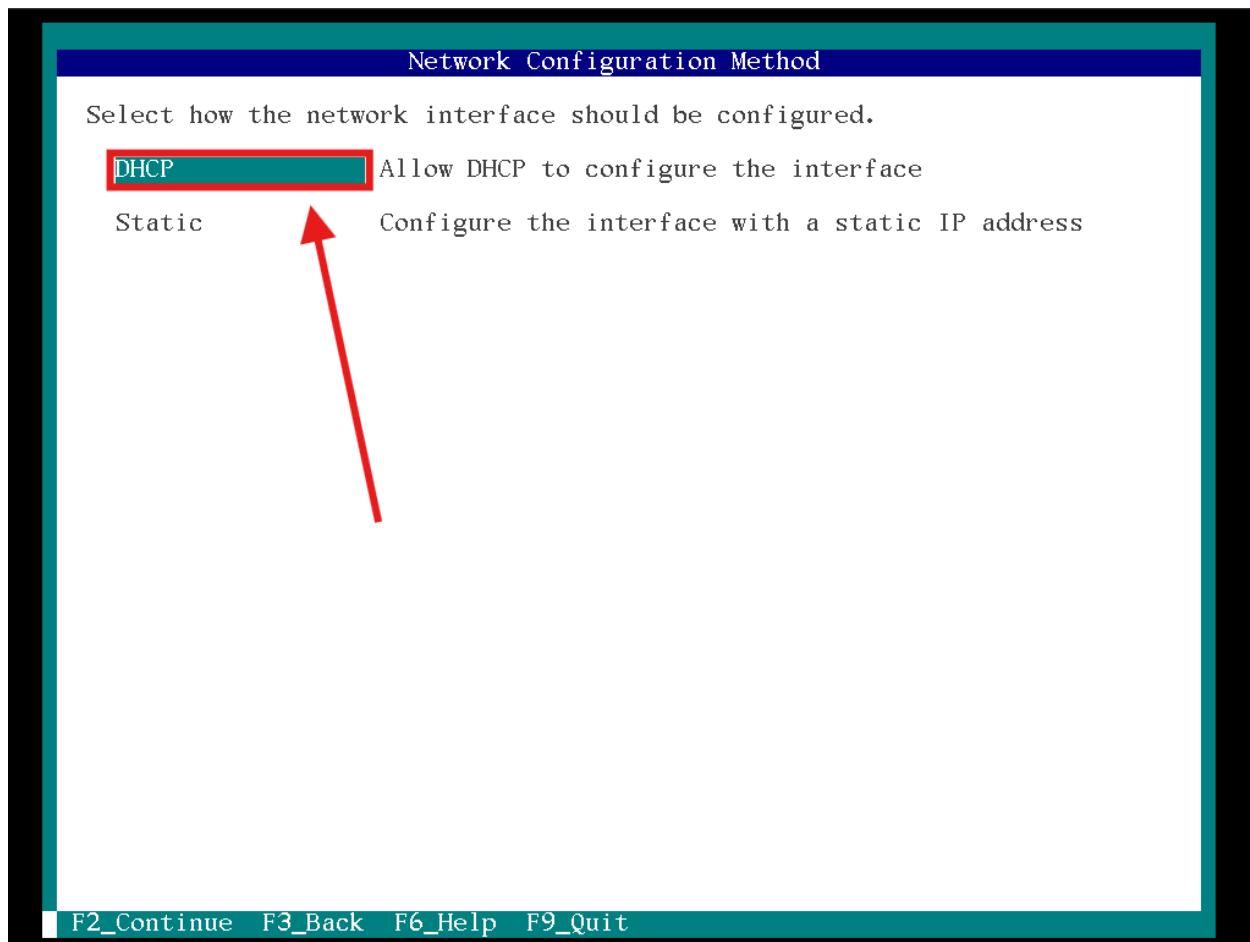
## Step 9: Hostname and Network Settings

Then I had to set up basic networking and give the system a name.

1. For the **hostname**, I typed a simple name like Yaschilal.
2. For **network interface**, I selected **net0 (e1000g0)**.



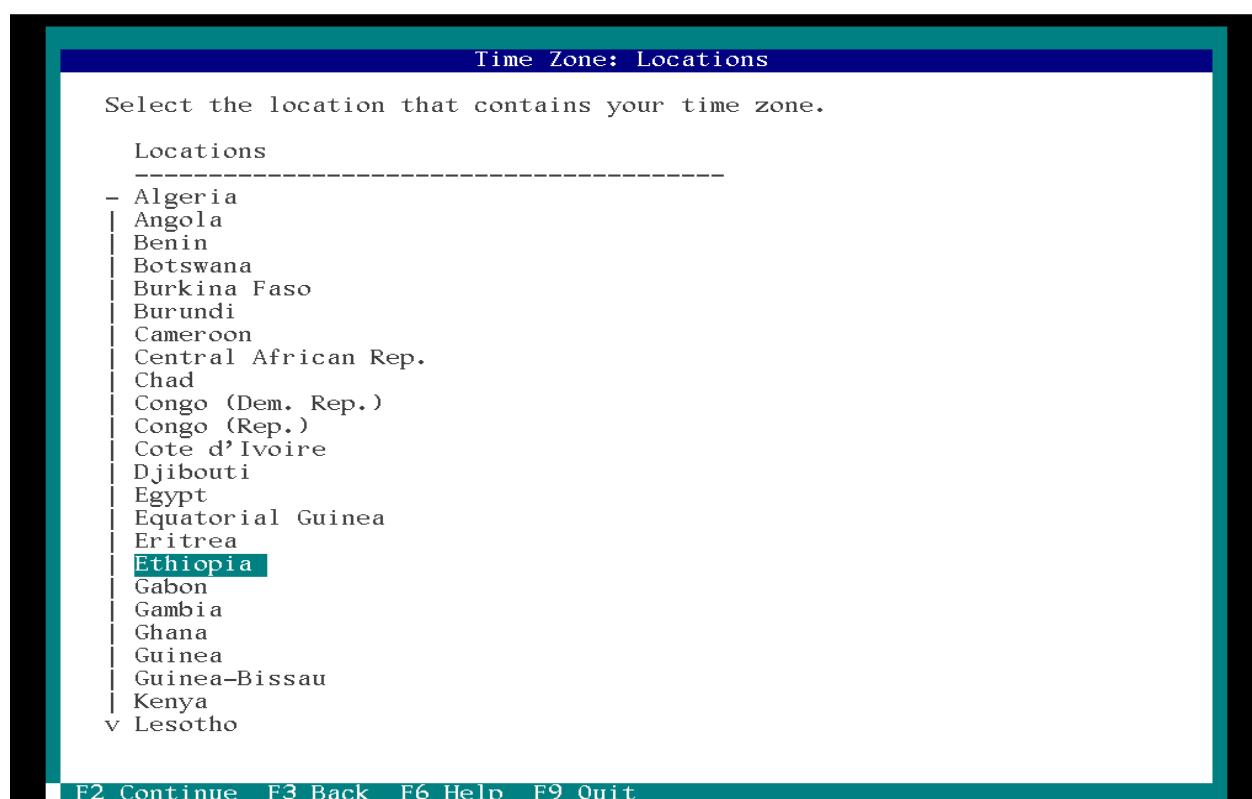
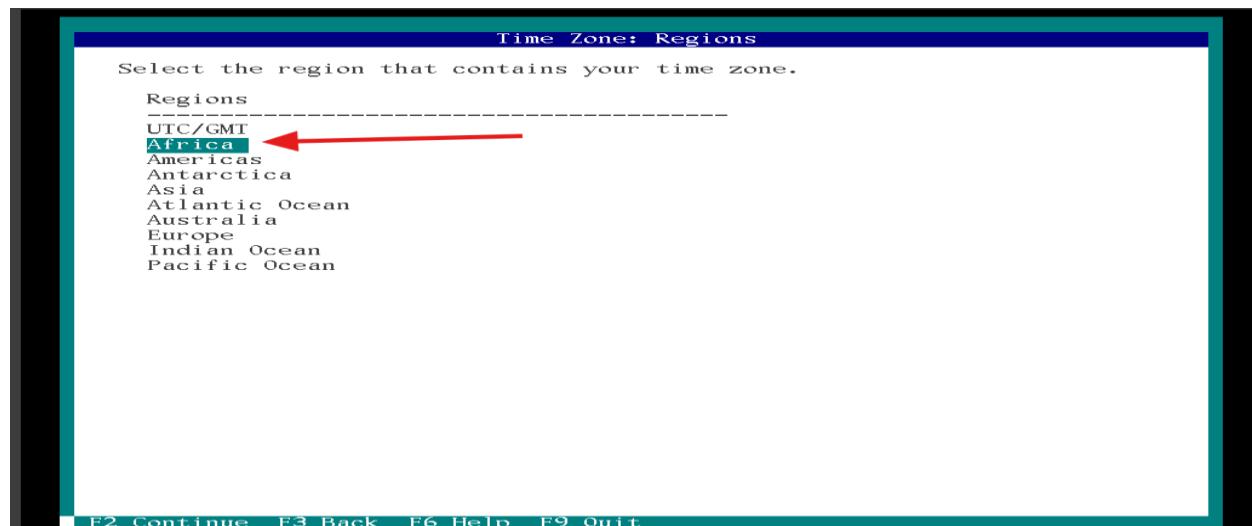
4. I chose **DHCP** so that Solaris would automatically get an IP address from my network.

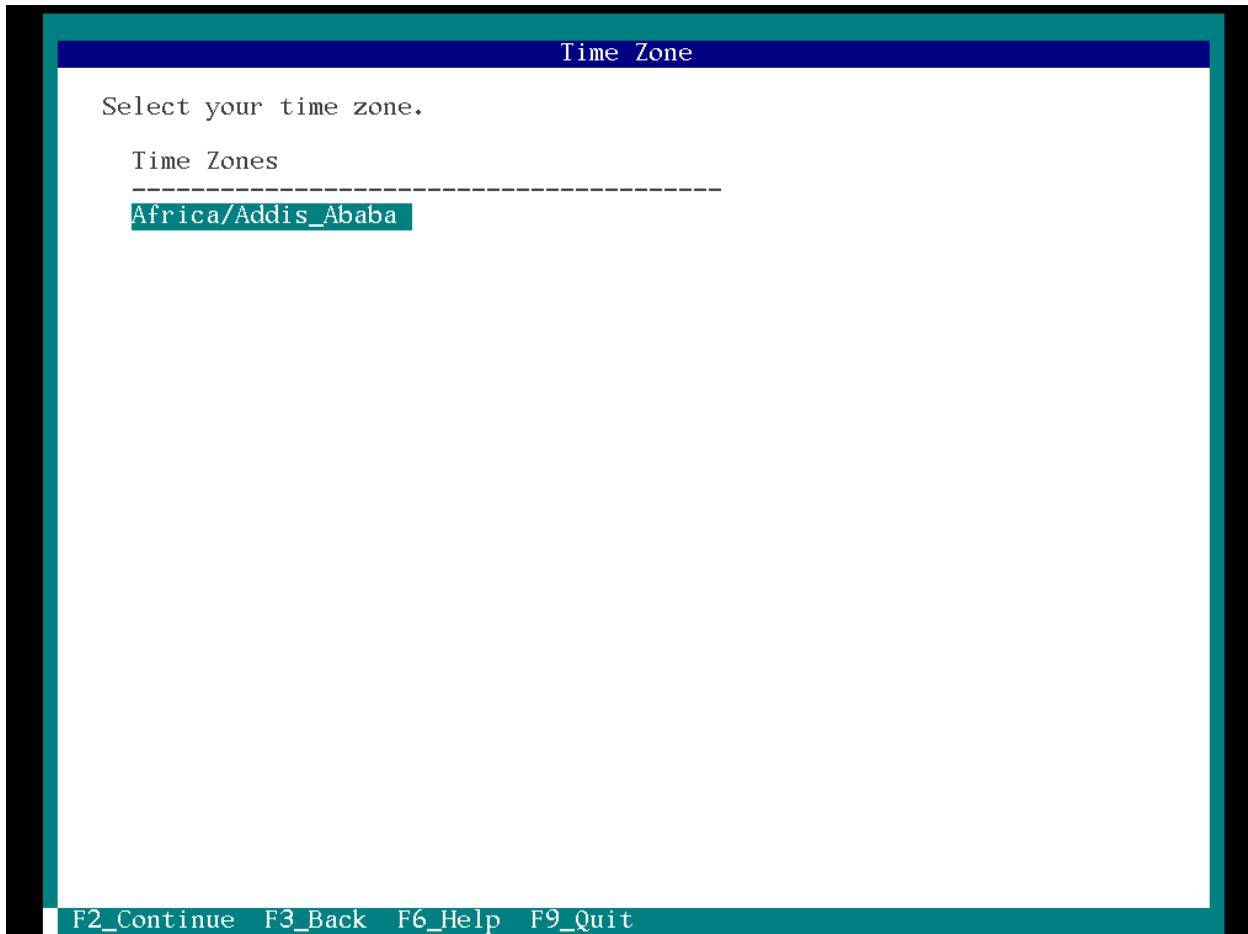


## Step 10: Set Time Zone and Language

Time zone and language settings came next.

- I selected **Africa/Addis\_Ababa** as my time zone.



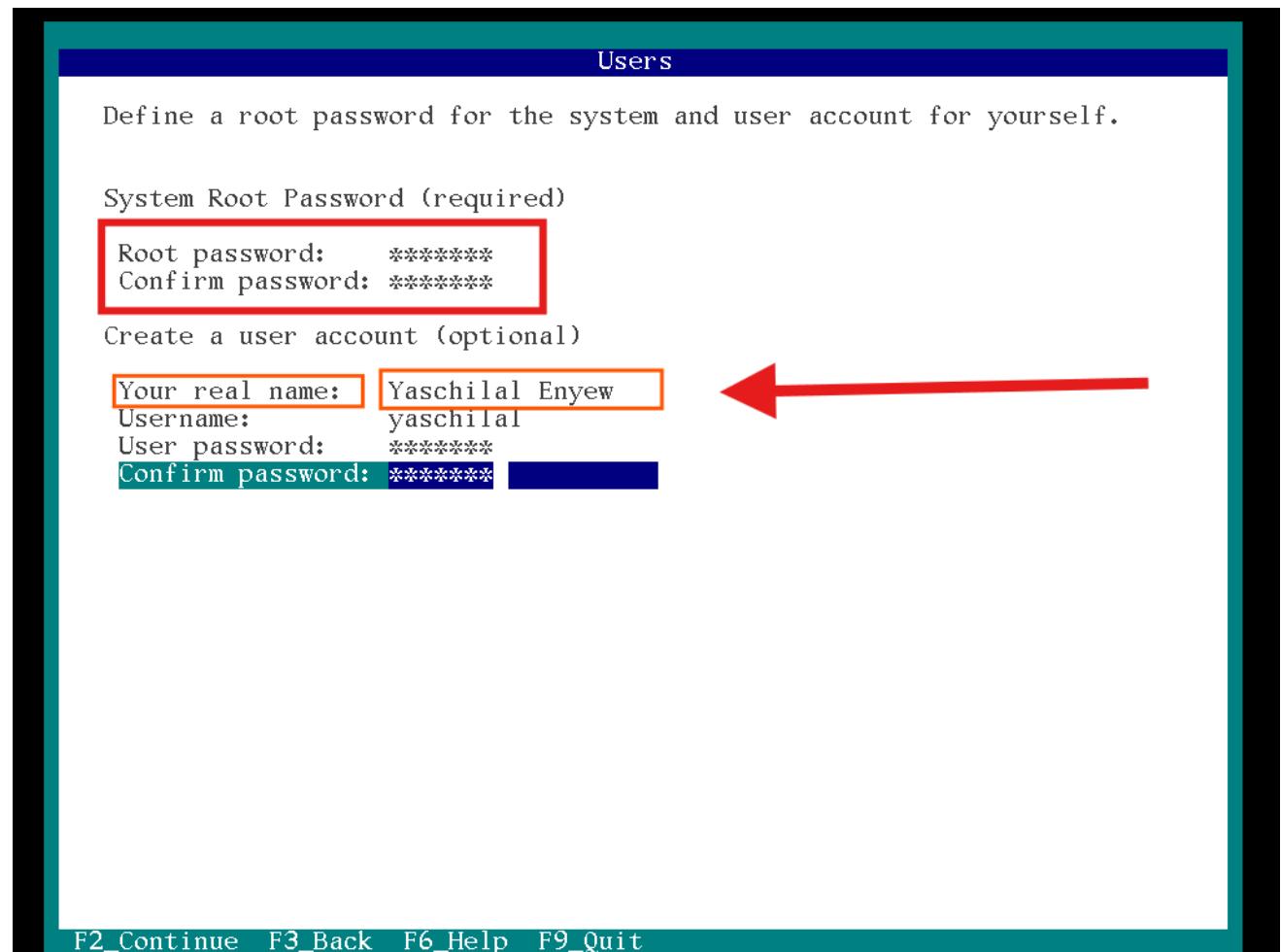


- For language, I went with **English (en\_US.UTF-8)**.
- Then pressed **F2** to proceed.

## Step 11: Set Root Password and Create Account

At this point, I had to create a password for the **root user** (which is like the main admin account in UNIX systems).

- ✓ I typed in a strong password and confirmed it.
- ✓ This **account** is super important because I'll use it to log in and manage the system after installation.



## Step 12: Oracle Support (Optional Step)

Solaris asked if I wanted to register with Oracle Support.

- I skipped this part since it wasn't required.

## Support - Registration

Provide your My Oracle Support credentials to be informed of security issues, enable Oracle Auto Service Requests.

See <http://www.oracle.com/goto/solarisautoreg> for details.

Email: [anonymous@oracle.com](mailto:anonymous@oracle.com)

Easier for you if you use your My Oracle Support email address/username.

Please enter your password if you wish to receive security updates via My Oracle Support.

My Oracle Support password:

- F2\_Continue F3\_Back F6\_Help F9\_Quit

## Step 13: Installation Summary

Before starting the actual install, Solaris showed me a summary of all the settings I chose.

- I reviewed everything — disk, network, language, and root password.
- If it all looked good, I pressed **F2** to begin the installation.

### Installation Summary

Review the settings below before installing. Go back (F3) to make changes.

Software: Oracle Solaris 11.4 X86

Root Pool Disk: 32.0GB sata

Computer name: YaschilalEnyew

Network:

DHCP Configuration: net0/v4

Time Zone: Africa/Addis\_Ababa

Locale:

Default Language: English

Language Support: English (United States)

Keyboard: US-English

Username: yaschilal

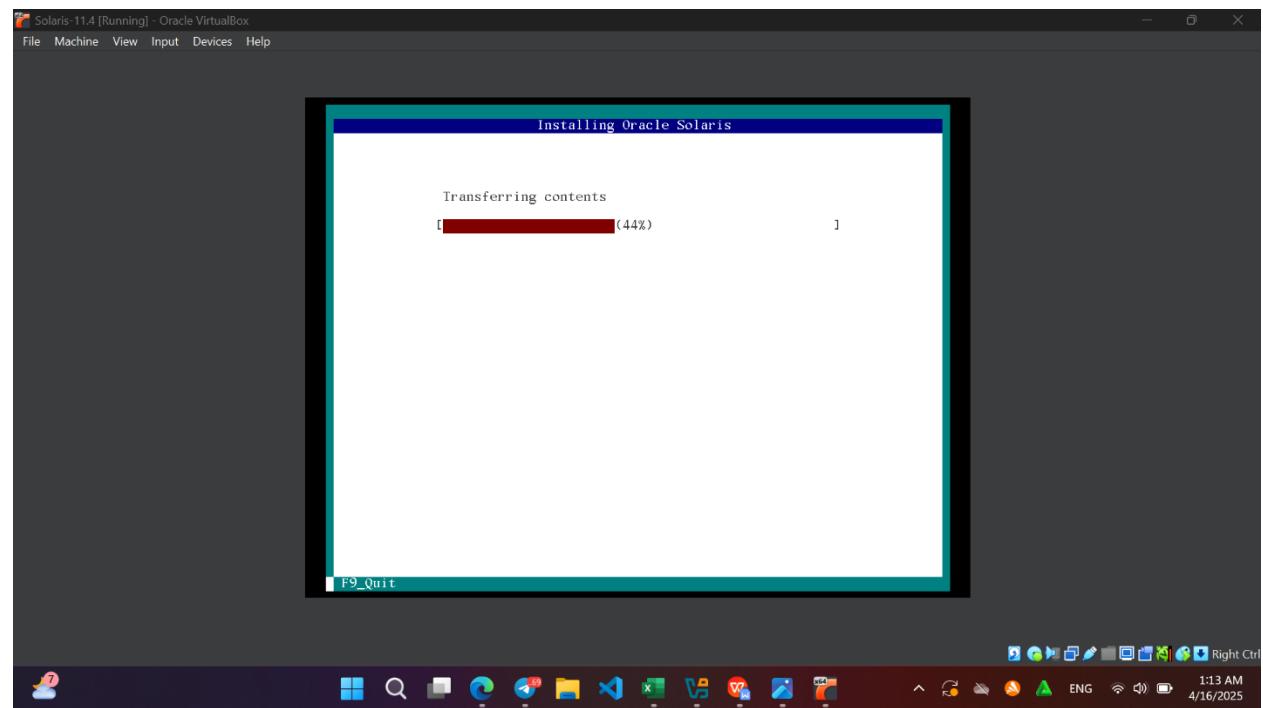
Support configuration:

No telemetry will be sent automatically

F2\_Install F3\_Back F6\_Help F9\_Quit

### Step 14: Wait for Installation to Complete

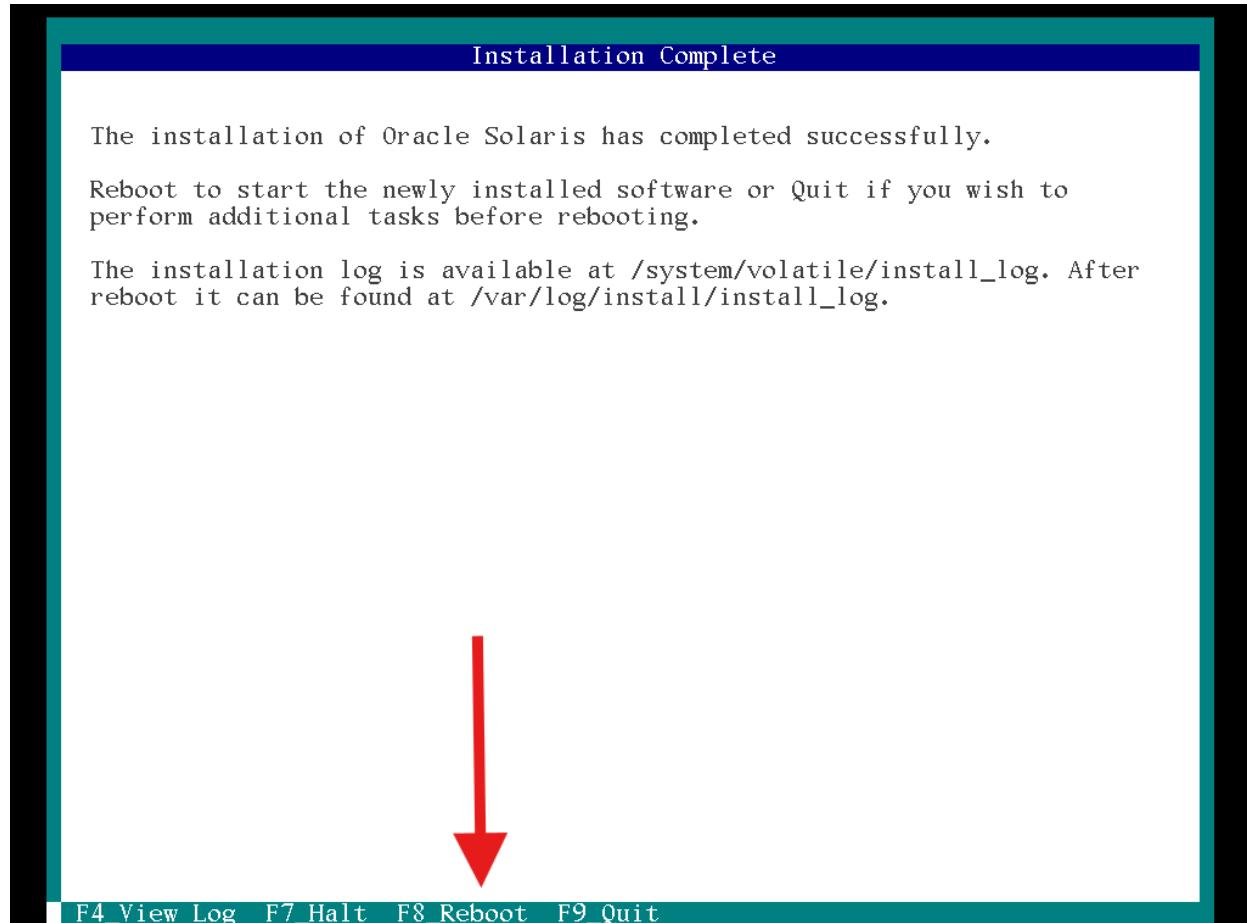
- The installation took a few minutes.
- I just waited while it copied and set up files inside the virtual machine.



## Step 15: Reboot and Login

Once the installation was done:

1. The installer asked me to **Reboot** — I clicked it.



2. After reboot, Solaris loaded up and asked me to **log in**.
3. I used the **root** username and the password I had set earlier.

```
Loading smf(7) service descriptions: 236/236
Booting to milestone "svc:/milestone/config:default".
Configuring devices.
Booting to milestone "all".
Hostname: vbox
Apr 24 16:50:55 vbox sendmail[1406]: My unqualified host name (vbox) unknown; sleeping for retry
Apr 24 16:50:55 vbox sendmail[1415]: My unqualified host name (vbox) unknown; sleeping for retry

vbox console login:
SUNW-MSG-ID: SMF-8000-YX, TYPE: Defect, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Apr 24 16:50:58 EAT 2025
PLATFORM: VirtualBox, CSN: VirtualBox-75c1a36a-bac2-491b-a037-2bfe710712c6, HOST NAME: vbox
SOURCE: software-diagnosis, REV: 0.2
EVENT-ID: 3327f50d-358d-431d-8142-d6321028a800
DESC: Service svc:/milestone/goals:default failed - goal service has dependencies that cannot be satisfied without administrative intervention.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/milestone/goals:default is unavailable.
REC-ACTION: Run 'svcs -xv svc:/milestone/goals:default' to determine the generic reason why the service failed, the location of any logfiles, and a list of other services impacted. Please refer to the associated reference document at http://support.oracle.com/msg/SMF-8000-YX for the latest service procedures and policies regarding this diagnosis.
Apr 24 16:51:55 vbox sendmail[1406]: unable to qualify my own domain name (vbox)
-- using short name
Apr 24 16:51:55 vbox sendmail[1415]: unable to qualify my own domain name (vbox)
-- using short name
yaschilal
Password:
vbox console login: 1
Password: ■
```

4. I was then logged into the **command-line interface (CLI)** of Oracle Solaris.

## Issues (problem I faced)

While installing Solaris in VirtualBox, I definitely didn't have a smooth ride the whole way. There were a few moments where I got stuck or confused, but I was able to figure things out with some patience and trial-and-error. Here are the main issues I ran into:

## **1. Problem Downloading the ISO File (Error 400 - Bad Request)**

When I tried to download the Solaris ISO file from the Oracle website, I kept getting this weird error:

*“400 Bad Request – Request Header Or Cookie Too Large”*

# **400 Bad Request**

**Request Header Or Cookie Too Large**

It wouldn't even let me start the download. I thought something was wrong with the site or my internet, but it turned out to be a browser issue.

## **2. Installer Freezing on Disk Selection Step**

During the installation process, when it reached the step to select the disk, the installer suddenly froze. The message showed up about using

the whole disk, but I couldn't press **F2** or even move around with the arrow keys. It just stayed stuck on that screen.

### 3. No Network Interface Showing

At the network setup step, there was a moment where it said:

*"No network interface found"*

Even though I had selected **net0** before, it just showed nothing, like the network wasn't available. I was worried I wouldn't be able to finish the installation without a network.

### 4. Keyboard Not Responding (Arrow Keys and F2)

At a few points during the installation, the keyboard just stopped working. The **arrow keys** and **F2** wouldn't respond at all — which made it hard to continue. I thought maybe something was wrong with the VM or VirtualBox.

These problems were kind of frustrating in the moment, but they actually helped me learn a lot about how to fix things and stay patient when something goes wrong during system installations.

# **Solution (How I Fixed the Problems)**

Even though I ran into a few annoying problems during the installation, I was able to fix each one with a bit of research, testing, and patience. Here's how I solved them:

## **1. Fix for Error 400 – Bad Request (While Downloading ISO)**

The first issue was getting the ISO file from Oracle. I kept seeing the "400 Bad Request – Request Header Or Cookie Too Large" error.

**How I fixed it:**

- I opened the download link in an **Incognito (Private) window** in Chrome.
- This cleared out all the cookies and extra headers that were causing the problem.
- The download started without any errors after that.

## **2. Installer Freezing on Disk Selection (F2 Not Working)**

When the installer asked me to use the whole disk, the screen froze and I couldn't press anything.

**Solution:**

- I pressed **Ctrl + C**, which kind of “reset” the screen.
- After that, the prompt came up again asking if I wanted to use the whole disk — this time it worked.
- I confirmed, and the installation moved forward.

## **3. No Network Interface Showing**

The network section didn't show any options at first, even though I expected to see **net0**.

## How I fixed it:

- I selected **net0** again manually and chose **DHCP** (so the system could automatically assign an IP).
- After a few seconds, the connection worked and I was able to continue.

## 4. Arrow Keys and F2 Not Responding

Sometimes during the install, my keyboard didn't respond — no arrow keys, no F2, nothing.

### What I did:

- I waited a few seconds (sometimes VirtualBox was just lagging).
- Pressed **Enter** or **Esc** gently a few times to “wake” the screen.
- Eventually, the installer caught up and the input started working again.

All of these problems were part of the learning experience — I didn't just install an OS, I also practiced troubleshooting like a real system admin!

## Filesystem Support

One of the things that makes Solaris a bit different from other operating systems is the kind of filesystems it supports — and the main one is **ZFS**.

## What Filesystems Does Solaris Support?

- ❖ **ZFS (Zettabyte File System)** – This is the **default** and most powerful filesystem in Solaris.
- ❖ **UFS (Unix File System)** – This is an older filesystem that Solaris still supports, but it's not commonly used anymore.

### ✓ Why ZFS?

ZFS is basically the **superhero of filesystems**. Here's why Solaris uses it as the default:

- **Self-healing**: It can detect and fix data corruption automatically — you don't even have to do anything.
- **Snapshots**: You can take a snapshot of the entire system and roll back if something goes wrong.
- **Data compression**: Saves space by automatically compressing data.
- **Scalability**: It can handle **massive** amounts of data — way more than ext4 or NTFS.
- **Built-in RAID**: ZFS can manage multiple drives and protect your data across them.

### Advantages and Disadvantages

After installing and using Solaris in a virtual environment, I noticed a few really strong points — but also some things that made it a bit challenging, especially for students or beginners like me.

# **Advantages**

## **1. Super Stable and Reliable**

- Solaris is built for serious systems like servers and data centers. Once it's running, it's rock solid — no crashes or random issues.

## **2. ZFS File System**

- As I mentioned earlier, ZFS is awesome. It gives strong data protection, lets you take snapshots, and can even fix issues by itself.

## **3. Great for Learning System Admin Stuff**

- It's not flashy like Windows or Ubuntu, but it helps you learn real-world things like command-line usage, disk management, and network setup.

## **4. Built-in Security**

- Solaris has advanced security features like Role-Based Access Control (RBAC) that are more common in enterprise environments.

## **5. Scalability**

- It works well on both small virtual setups (like mine) and huge servers in the cloud — so it's flexible for different needs.

- Solaris supports containers (called “zones”), which is cool if you want to learn about managing multiple isolated environments.

## Disadvantages

### 1. Not Beginner-Friendly

- ❖ Let’s be real — if you’ve never touched a terminal before, Solaris can feel a little scary. There’s no graphical installer in some versions, and it’s very command-line focused.

### 2. Limited Software

- ❖ It doesn’t have an app store or common software like you’d find in Windows or Ubuntu. You have to manually install most things — sometimes using older or custom methods.

### 3. Compatibility Issues

- ❖ It doesn’t support popular filesystems like NTFS or ext4, which can be annoying if you’re trying to share data between systems.

### 4. Some Hardware Isn’t Supported

- ❖ Since Solaris is designed more for servers, it doesn’t always support newer consumer hardware like fancy graphics cards or Wi-Fi chips.

### 5. Small Community

- ❖ Unlike Linux, there aren’t a ton of tutorials or forum answers for beginners. That means if you get stuck, it might take longer to find a solution.

# Conclusion

Installing Oracle Solaris 11.4 in a virtual machine was definitely a learning experience. It wasn't always easy — I ran into a few unexpected problems — but that's actually what made it valuable. I didn't just follow a tutorial; I had to figure things out and fix real issues, which helped me understand how operating systems really work.

I got hands-on practice with using VirtualBox, setting up a full OS environment, and working inside a command-line interface — something that's important in system admin roles or backend server work. I also learned about Solaris's special features like ZFS, and why it's used in serious enterprise setups.

Even though Solaris isn't the most beginner-friendly OS, I'm glad I got the chance to try it out. I now have a much better appreciation for how powerful and secure enterprise operating systems can be, and I feel more confident dealing with things outside of Windows or Linux.

## Future Outlook / Recommendations

After working with Solaris in a virtual machine, I would definitely recommend that other students give it a try — especially if they want to explore operating systems that are used in the real world, beyond just Linux and Windows.

Here are a few tips and recommendations I'd give to anyone trying this in the future:

## **For New Learners:**

- ❖ **Start with VirtualBox** — It's free, simple to use, and perfect for testing out operating systems like Solaris without breaking anything on your main computer.
- ❖ **Read official documentation** — The Solaris admin and install guides from Oracle can really help when you get stuck.
- ❖ **Learn some Linux basics first** — Since Solaris is UNIX-based, having basic Linux/terminal skills makes things easier.

## **For Continued Learning:**

- ❖ Try using **ZFS commands** to explore more about snapshots, compression, and storage management.
- ❖ Look into **Solaris zones (containers)** — they're like lightweight virtual machines within Solaris and are great for learning virtualization concepts.
- ❖ After Solaris, you could also explore other enterprise OS like **Red Hat Enterprise Linux (RHEL)** or **SUSE Linux** for comparison.

This assignment not only helped me improve my technical skills, but also boosted my confidence when it comes to tackling new systems that I'm not familiar with.

## 2. What, Why, and How of Virtualization in Modern Operating Systems

### What is Virtualization?

Virtualization is a technology that allows us to create virtual versions of physical hardware — such as operating systems, servers, storage devices, or networks. For example, using tools like **VirtualBox** or **VMware**, we can run multiple operating systems on a single physical computer.

### Why is Virtualization Important?

- ◆ **Cost-Efficiency:** Instead of buying multiple machines, we can run several virtual environments on one.
- ◆ **Testing & Development:** Developers and students can safely test apps or new OS versions without affecting the main system.
- ◆ **Resource Optimization:** It helps in using CPU, memory, and disk more efficiently.
- ◆ **Isolation:** Each virtual machine is isolated, so one VM's problem won't crash the entire system.
- ◆ **Portability:** Virtual machines can be copied, shared, or moved easily

### How Does It Work?

- ❖ A **hypervisor** (like VirtualBox or VMware) runs on a host system.
- ❖ It allows you to create and manage **virtual machines (VMs)**.

- ❖ Each VM acts like a real computer, with its own OS, memory, and virtual hardware.
- ❖ The hypervisor allocates resources (CPU, RAM, Disk) from the physical machine to these VMs.
- ❖ The user interacts with the VM as if it's a separate computer.

### **Question 3: Implement System Calls (sigsuspend() - Suspends the calling process until a signal is received.)**

In Solaris, like in other Unix-like systems, the `sigsuspend()` system call is used to pause a running process until it receives a specific signal. This is useful in situations where a process needs to wait for a certain event, like waiting for user input, or when it's part of a larger system that needs to respond to signals from other processes.

How does `sigsuspend()` work?

- When a process calls `sigsuspend()`, it pauses and goes into a state where it waits for a signal to be sent to it.
- This system call is often used in combination with a signal handler, which is a function that handles the signal once it's received.
- Once the signal arrives, the process wakes up and continues from where it was paused. If a signal handler is set up, it will run first before continuing the execution.

Why use `sigsuspend()`?

It's helpful in scenarios where a program needs to wait for an event or action to occur, like when a user sends a specific signal (for example,

pressing Ctrl+C or some custom action). This allows the process to remain inactive until the desired event takes place.

Example Code to Implement sigsuspend()

Here's a simple example to show how sigsuspend() can be used in a C program on Solaris:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

void signal_handler(int sig) {
    printf("Signal %d received! Process is now resuming... \n",
sig);
}

int main() {
    // Set up the signal handler for SIGUSR1
    signal(SIGUSR1, signal_handler);

    printf("Process suspended. Waiting for SIGUSR1... \n");

    // Suspend the process and wait for the signal
    if (sigsuspend(NULL) != -1) {
        // This part will execute after the signal is handled
        printf("Process resumed after receiving SIGUSR1!\n");
    }
}
```

```
    return 0;  
}
```

### Explanation of the Code:

#### 1. Setting up the Signal Handler:

- `signal(SIGUSR1, signal_handler);` — This line tells the program to call the `signal_handler` function whenever it receives the `SIGUSR1` signal.

#### 2. Suspending the Process:

- `sigsuspend(NULL);` — This line pauses the process. It will remain paused until it gets a signal (in this case, `SIGUSR1`).

#### 3. Handling the Signal:

- Once the program receives `SIGUSR1`, it calls the `signal_handler` function, which prints a message and then lets the program continue executing.

### How to Test:

- ❖ First => I Compile the Program using this code:

```
cc -o sigsuspend_example sigsuspend_example.c
```

- ❖ then second => I Run the Program

```
./sigsuspend_example
```

Thirdly => I was Send the Signal from Another Terminal: Find the process ID (PID) of the running program using:

```
ps aux | grep sigsuspend_example
```

Then, send the SIGUSR1 signal Using this code:

*kill -SIGUSR1 <PID>*

After sending the signal, the process will resume and print the message "Process resumed after receiving SIGUSR1!"

## Conclusion

The `sigsuspend()` system call is a powerful way to make a process pause and wait for specific signals. It is commonly used when a program needs to wait for an event, such as user input or other system events, before continuing execution. The combination of `sigsuspend()` with signal handlers makes it easy to control process flow in response to external actions or signals.

## REFERENCES

- ❖ . Solaris Operating System Documentation:
  - Oracle Solaris Documentation:  
<https://docs.oracle.com/en/solaris/>

- Oracle Solaris 11.4 Installation Guide:  
[https://docs.oracle.com/cd/E53394\\_01/html/E55603/index.htm](https://docs.oracle.com/cd/E53394_01/html/E55603/index.htm)  
!
  - Oracle Solaris 11.4 System Administration Guide:  
[https://docs.oracle.com/cd/E53394\\_01/html/E55603/index.htm](https://docs.oracle.com/cd/E53394_01/html/E55603/index.htm)  
!
- ❖ System Calls in Unix/Linux:
- The Linux Programming Interface (Book) by Michael Kerrisk:
    - A comprehensive reference for system calls in Unix-like systems. While focused on Linux, many system calls and principles are the same across Unix-like systems, including Solaris.
    - The Linux Programming Interface
  - POSIX System Calls (Manual):
    - POSIX Signal Handling
    - This standard documentation gives a detailed explanation of how signals are handled and includes system calls like `sigsuspend()`.
- ❖ . Oracle Solaris System Calls and Signals:
- `sigsuspend()` Manual Page in Solaris:  
[https://docs.oracle.com/cd/E53394\\_01/html/E55603/sigsuspend-2.html](https://docs.oracle.com/cd/E53394_01/html/E55603/sigsuspend-2.html)  
This page explains how `sigsuspend()` works, its usage, and related functions.

- ❖ Solaris File Systems (ZFS, UFS, etc.):
  - Oracle Solaris ZFS Documentation:  
[https://docs.oracle.com/cd/E53394\\_01/html/E55603/zfs-file-system-usage.html](https://docs.oracle.com/cd/E53394_01/html/E55603/zfs-file-system-usage.html)  
An in-depth look at ZFS, which is the default file system in Solaris and a key topic in your assignment.

- ❖ Oracle Solaris Network Configuration:
  - Solaris Networking:  
[https://docs.oracle.com/cd/E53394\\_01/html/E55603/solaris-networking-guide.html](https://docs.oracle.com/cd/E53394_01/html/E55603/solaris-networking-guide.html) This guide provides detailed steps on configuring networking settings in Solaris, which was part of your installation process.

- ❖ Oracle Documentation for Solaris Installation and Setup:
  - [Oracle Solaris 11.4 Installation and Configuration](#) Official guide for installing Solaris OS.

- ❖ Practical Resources on Signals and System Calls:
  - Unix Signal Programming:  
<https://beej.us/guide/bgipc/output/html/multipage/signals.htm> Beej's Guide is an excellent resource for understanding Unix signals and system calls like `sigsuspend()`.

- ❖ System Programming and Practical Examples:
- ❖ The Art of Unix Programming by Eric S. Raymond:  
<http://www.faqs.org/docs/artu/> This book is a great resource for understanding the philosophy and practical aspects of Unix system calls and how they work in practice.

