



---

# **BAHIRDAR UNIVERSITY**

## **Department Of Software Engineering Operating System and System Programming Section B Solaris OS\_Individual Assignment**

**Name**  
**Yaschilal Enyew**

**ID Number**  
**BDU1602719**

**Submitted To LecWendimu B.**  
**Submission date 16/08/17**

# Introduction

In the modern computing landscape, operating systems are essential software components that manage hardware and software resources, offering a stable environment for applications to run. One of the key advances in recent years is the **use of virtualization**, which allows multiple operating systems to run on a single physical machine through virtual environments such as **Oracle VirtualBox** and **VMware Workstation**.

This assignment explores the installation of **Oracle Solaris 11.4**, a powerful UNIX-based operating system developed by Oracle Corporation. Solaris is known for its enterprise-level features, such as **ZFS filesystem**, **advanced security**, **scalability**, and **support for cloud and data center applications**.

## Background

The Oracle Solaris operating system has a long history, originally developed by Sun Microsystems before being acquired by Oracle. It has become a prominent choice for critical environments requiring performance, security, and reliability. As a UNIX-based OS, Solaris is widely used in data centers, server environments, and academic research due to its stability and feature-rich architecture.

Motivation

The motivation behind performing this installation in a virtual environment is to gain hands-on experience without the risk of damaging physical hardware. Virtualization also enables:

- ✧ **Efficient resource usage** (RAM, CPU, storage),
- ✧ **Snapshot and recovery support,**
- ✧ **Testing without affecting host systems,** and
- ✧ **Learning enterprise operating systems in a flexible and safe environment.**

By installing Solaris in a virtual machine, students and professionals can explore its capabilities, learn system administration commands, and understand how modern operating systems manage resources and services.

## b. Objectives

The main goal of this assignment is to get some real, hands-on experience installing and working with an operating system in a virtual environment. Specifically, I worked with **Oracle Solaris 11.4** and installed it using **Oracle VM VirtualBox**.

This kind of task is important because it helps build practical skills that go beyond just theory—skills that are useful in the real world, especially in fields like system administration, networking, and enterprise computing.

Here's what I aimed to accomplish with this assignment:

1. **Learn how to set up and install an operating system** (Solaris 11.4) inside a virtual machine.
2. **Understand the steps involved** in the installation process, like selecting the right disk, setting up the network, and creating user accounts.
3. **Get comfortable using virtualization tools** like VirtualBox.
4. **Run into problems (and solve them!)** so I can learn from real-world troubleshooting.
5. **Explore the types of filesystems** supported by Solaris and understand why they matter.
6. **Compare the pros and cons** of using Solaris and how it performs in a virtual setup.
7. **Document everything** with screenshots and explanations, so I have a full record of what I did and what I learned.

To successfully install and run Oracle Solaris 11.4 in a virtual environment, certain hardware and software components are needed. Below are the specific requirements I used for this assignment.

---

## i. Hardware Requirements

The following hardware setup was used to support the virtual machine environment:

- **Laptop Model:** HP EliteBook
- **Processor:** Intel Core i7, 8th Generation
- **RAM:** 16 GB
- **Storage:** 512 GB SSD
- **Architecture:** 64-bit
- **Other:** Touchscreen display (optional), Virtualization enabled in BIOS

*Note: While Solaris can run on systems with lower specs, at least 8 GB RAM and a modern processor are recommended for smooth performance in virtual environments.*

---

## ii. Software Requirements

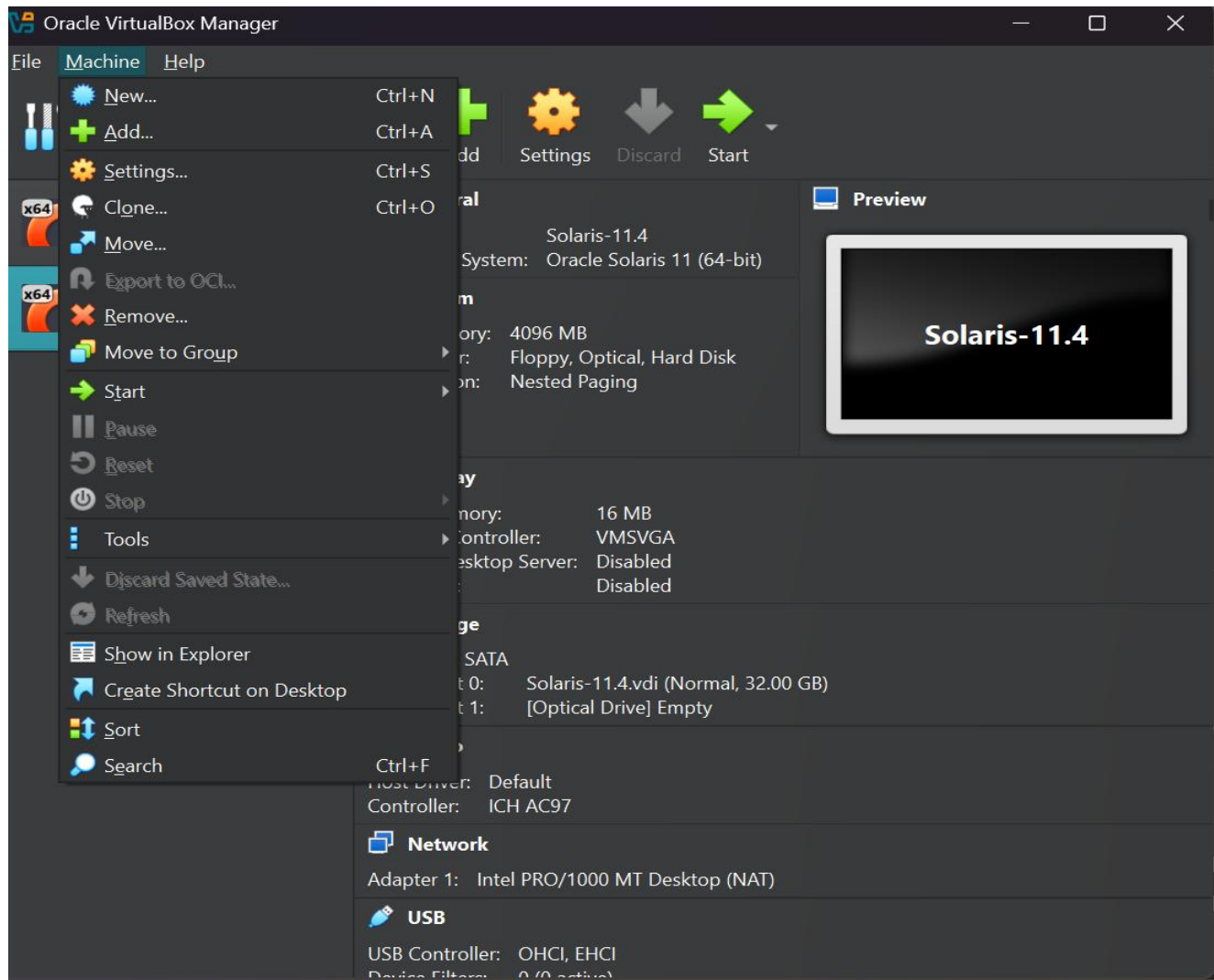
To install and run Oracle Solaris in a virtual machine, the following software was used:

- **Oracle VM VirtualBox** (version X.X.X) – a free and open-source virtualization tool used to create and manage virtual machines
- **Oracle Solaris 11.4 Text Installer ISO** – official ISO image downloaded from the Oracle website
- **Host Operating System:** Windows 11 (64-bit)

*VirtualBox Extensions (Optional): May be installed for better performance and additional features like USB support.*

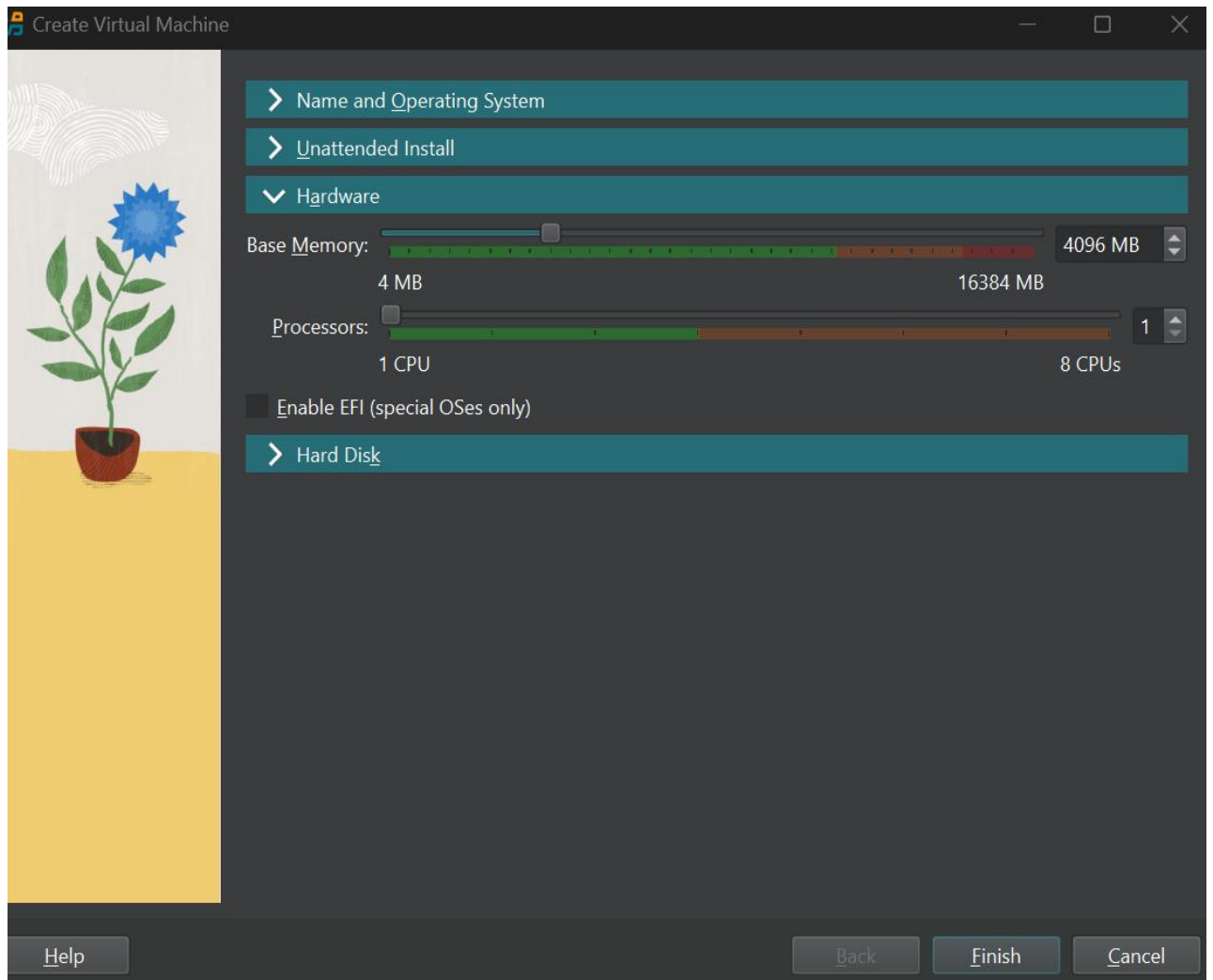
## d. Installation steps

Step 1: Open VirtualBox and Create a New Virtual Machine



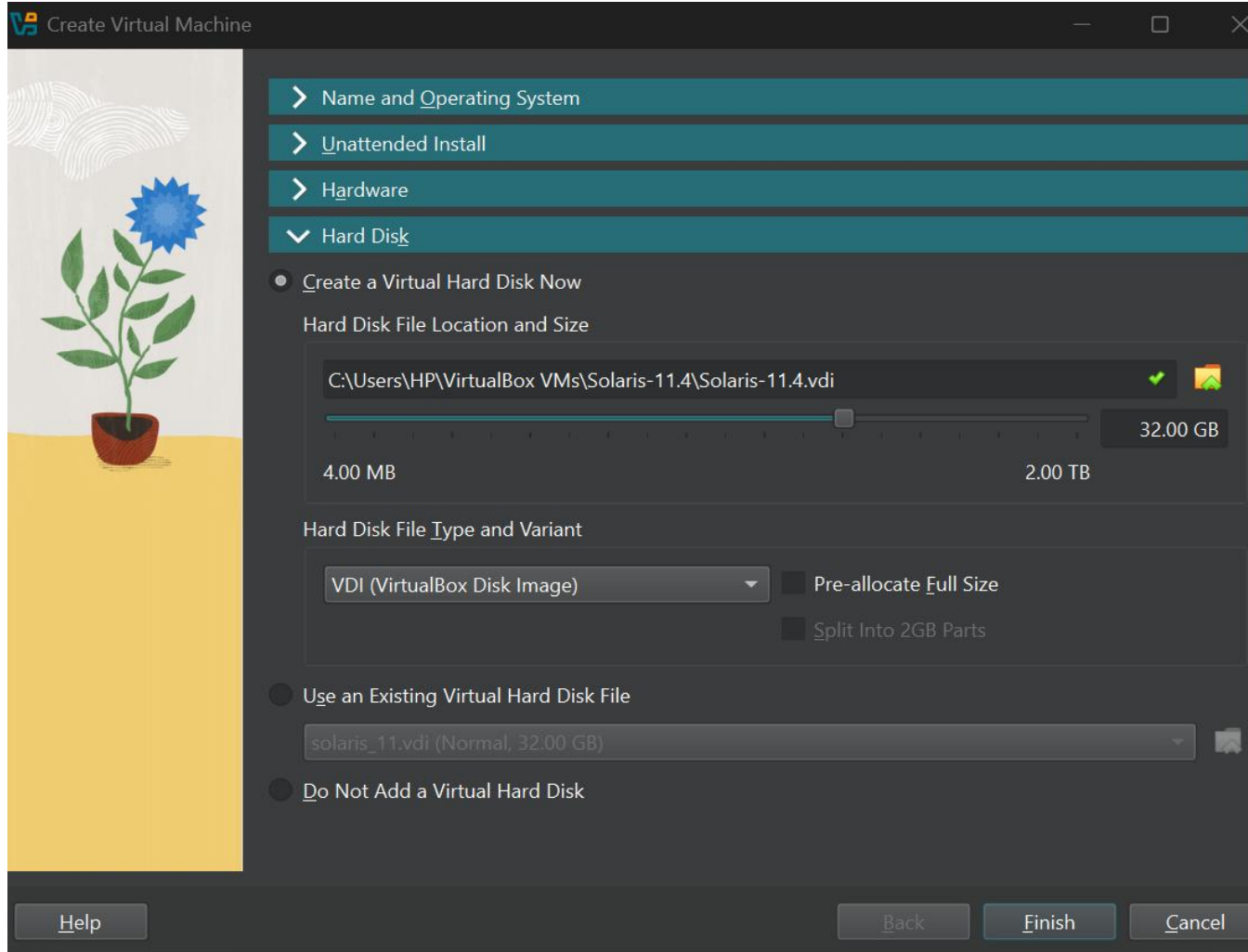
## Step 2: Assign Memory Size

Set memory (RAM) to **at least 2048 MB (2 GB)** — or higher for better performance.



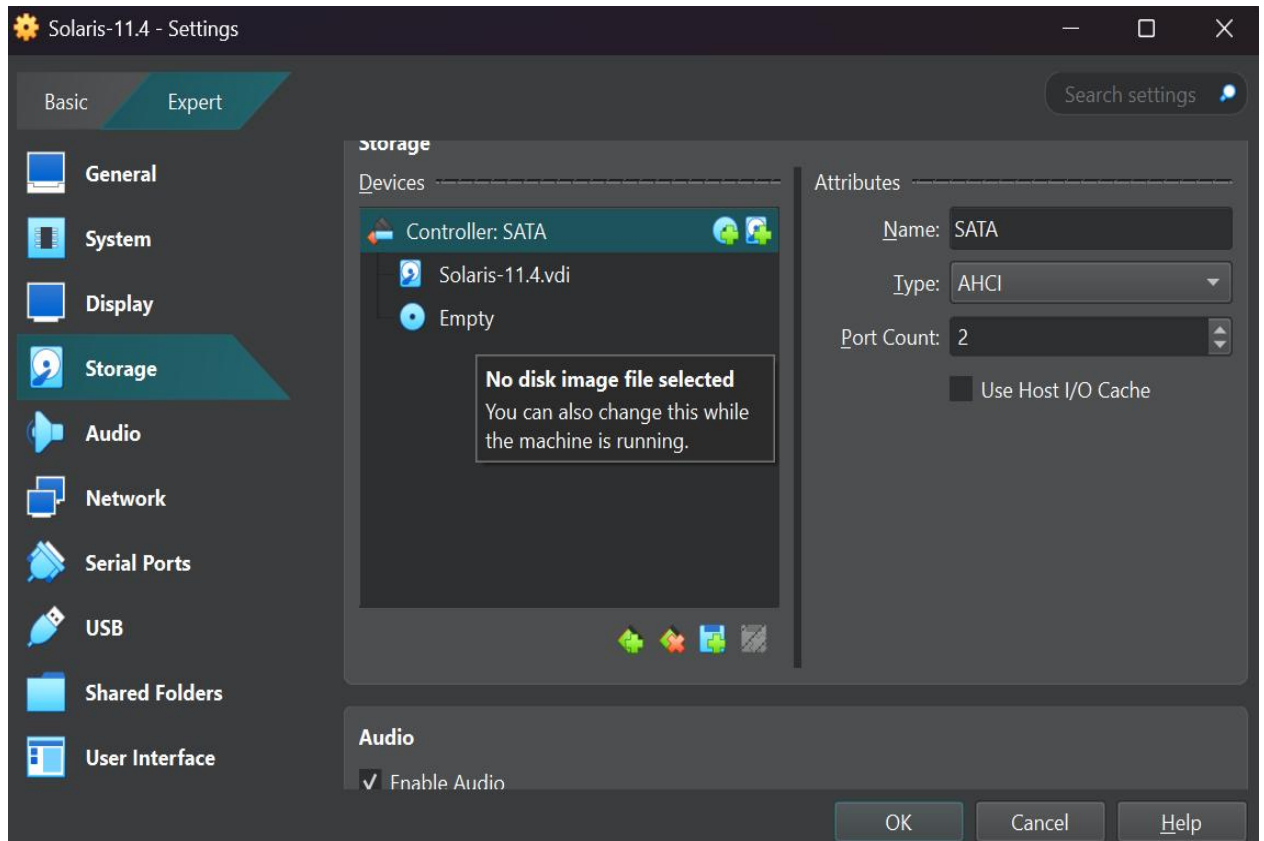
### Step 3: Create a Virtual Hard Disk

- Choose "Create a virtual hard disk now" → Click **Create**.
- Select **VDI (VirtualBox Disk Image)** → **Dynamically allocated**.
- Set size to **at least 20 GB**.



## Step 4: Mount the Solaris ISO

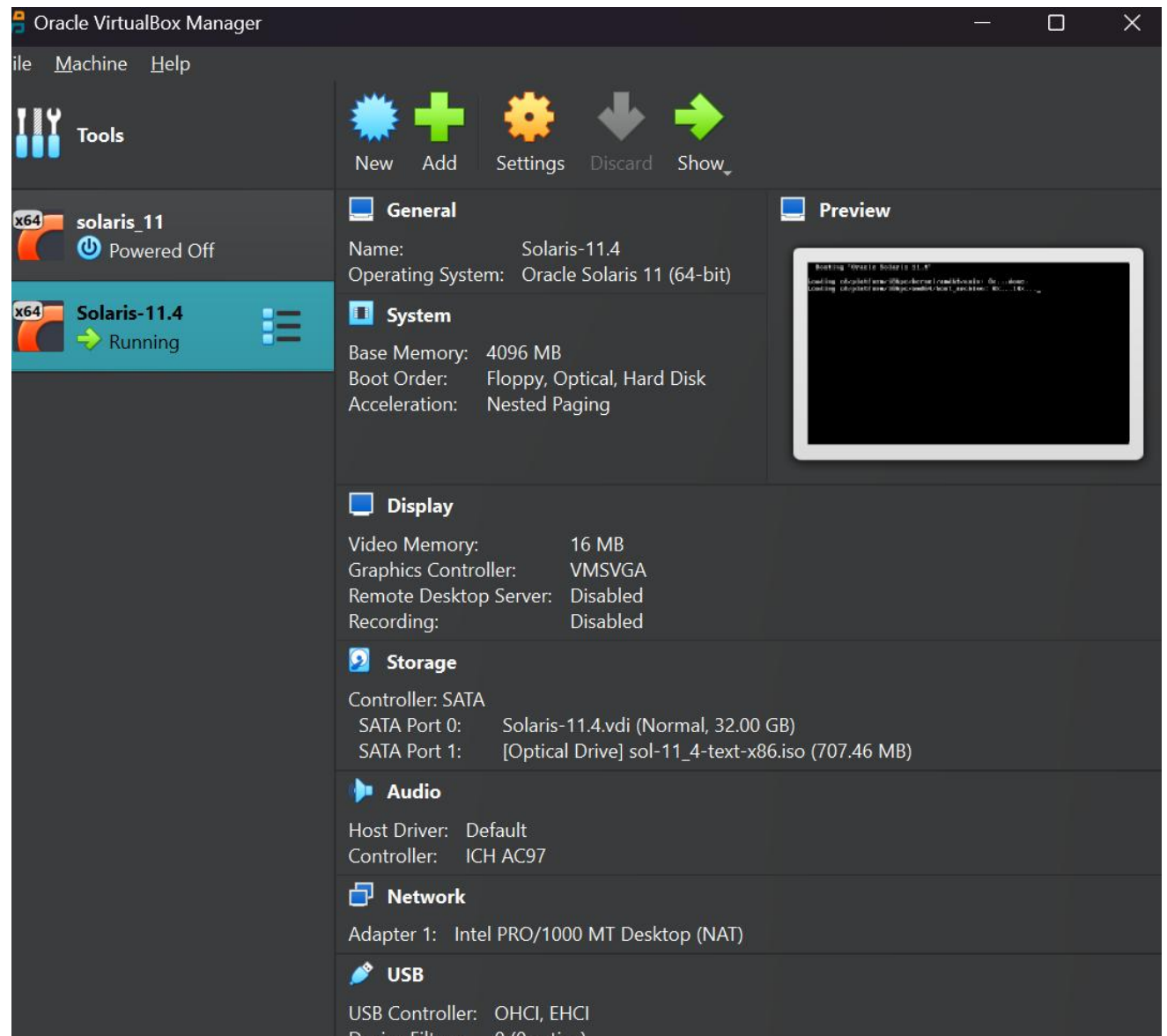
- Go to the **Settings** of the created VM → **Storage**.
- Under the **Controller: IDE**, click the empty disk icon → choose the Solaris 11.4 ISO you downloaded.





## Step 5: Start the Virtual Machine

- Click **Start** to boot the virtual machine.
- The system will begin loading Oracle Solaris.



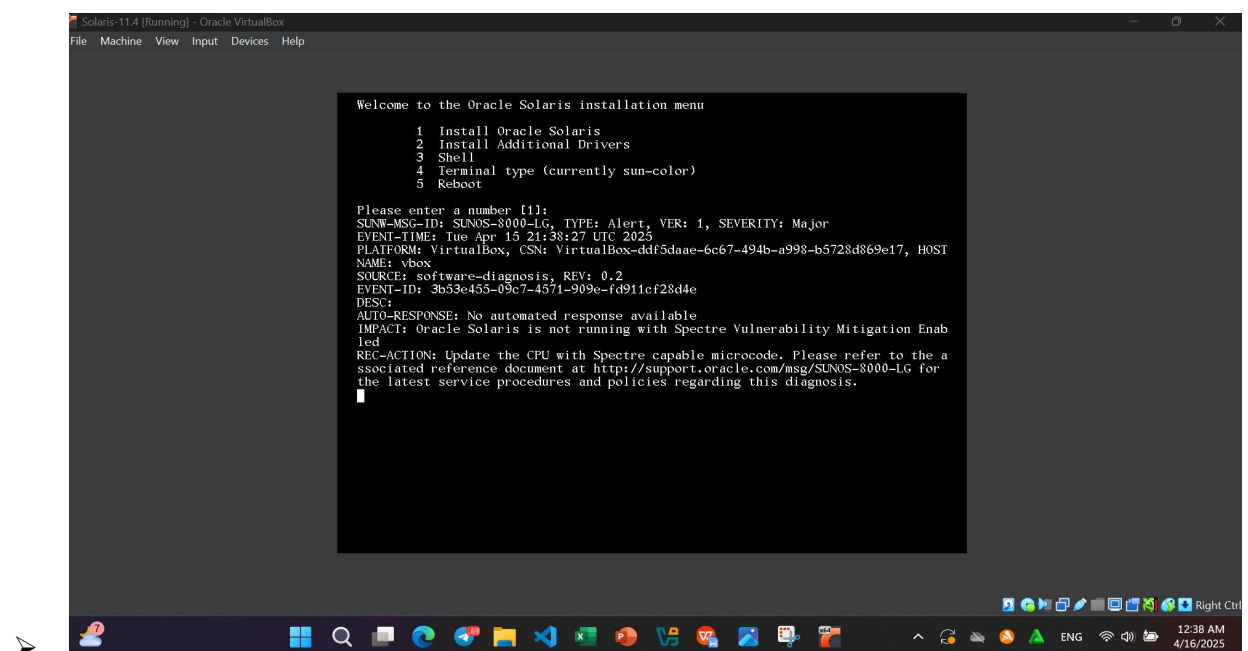
## Step 6: Select Keyboard Layout

- Choose your keyboard layout. I selected the default: **US English** (Option 27).



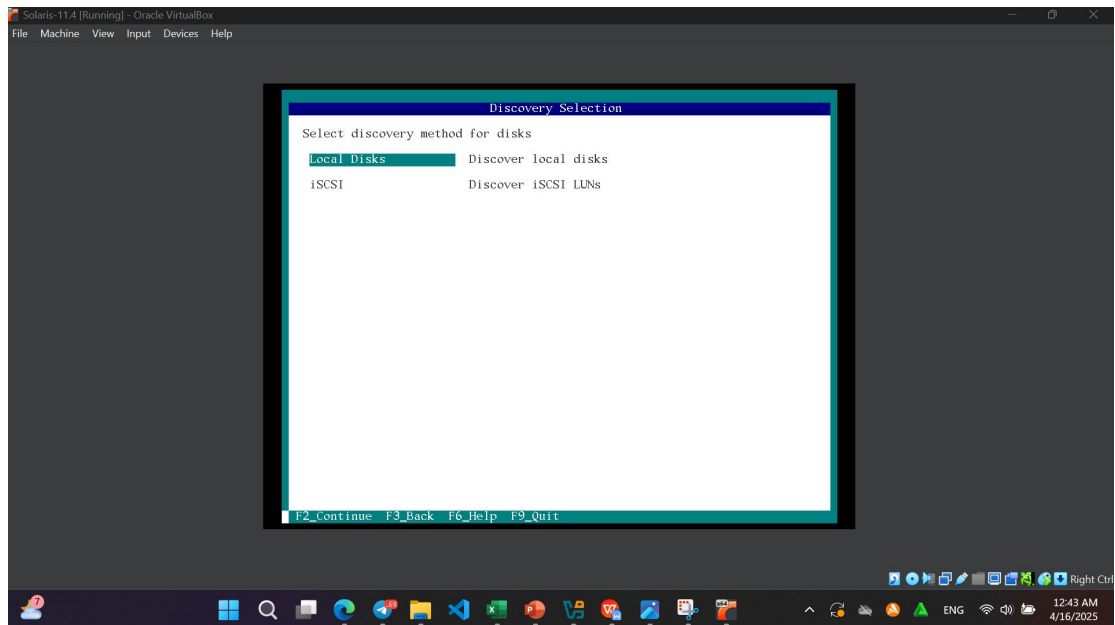
## Step 7: Installation Menu

- The Solaris installer will launch.
- Use **arrow keys** to navigate and **F2** to continue at each screen.



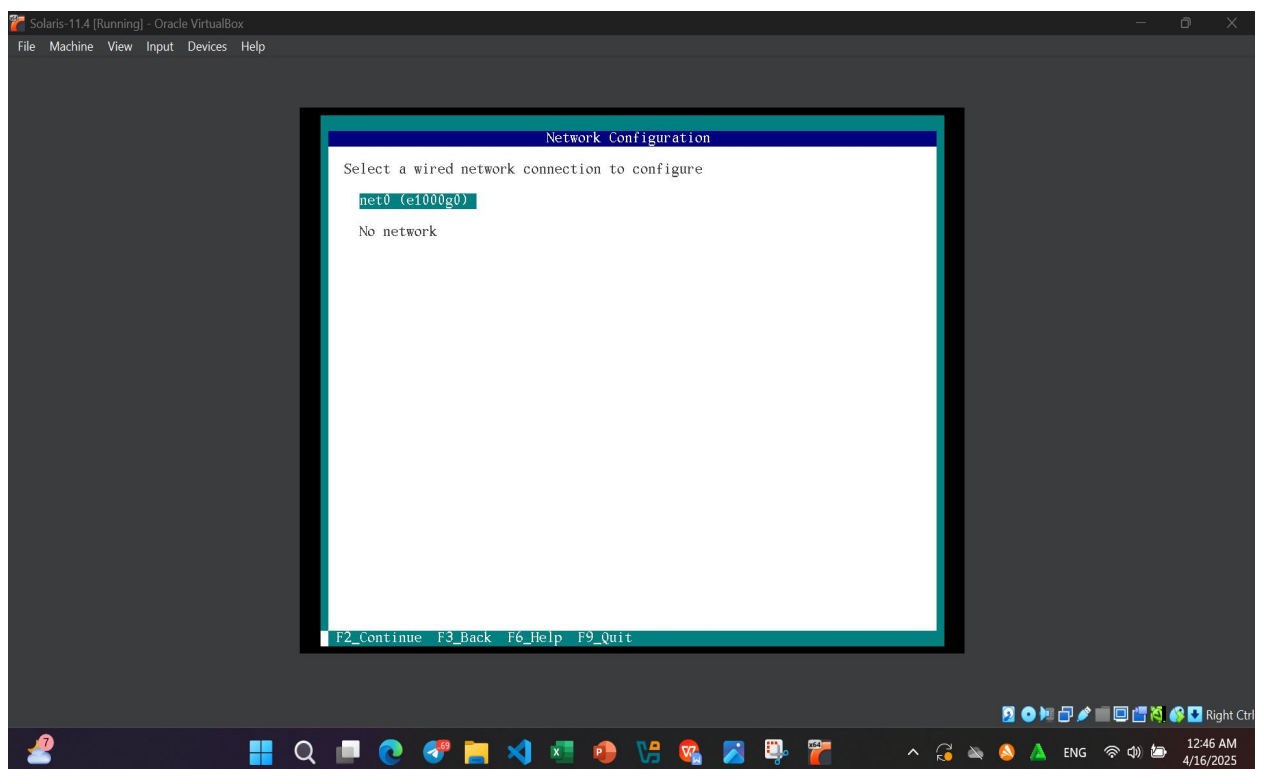
## Step 8: Select Disk Discovery Method

- Choose 1. Local Disks when asked how to discover disks.
- The installer may propose to use the whole disk — accept it.



## Step 9: Hostname & Network Settings

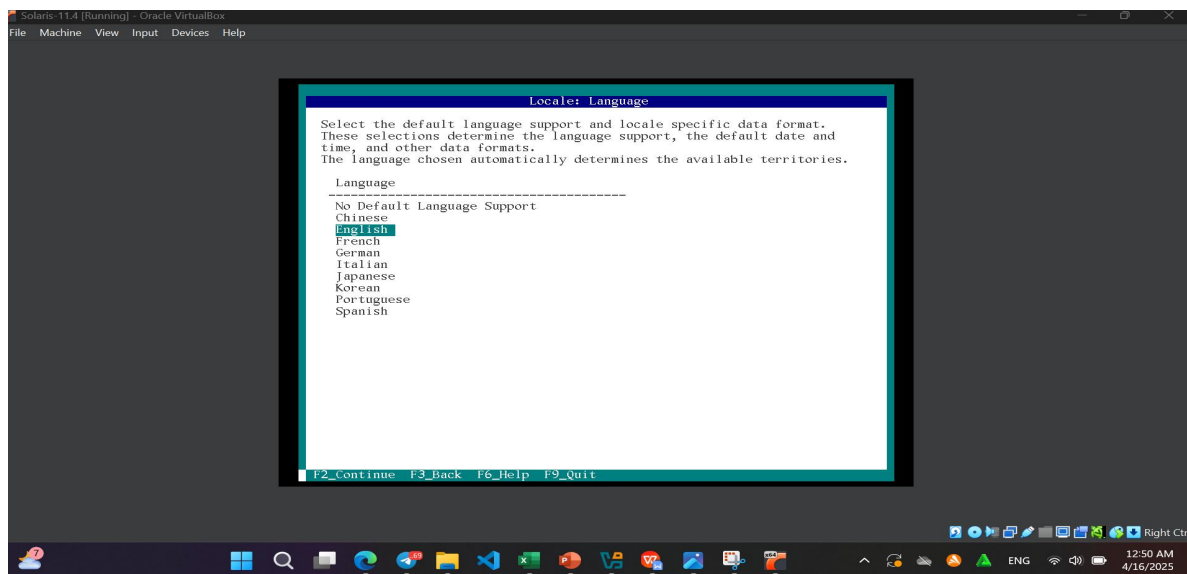
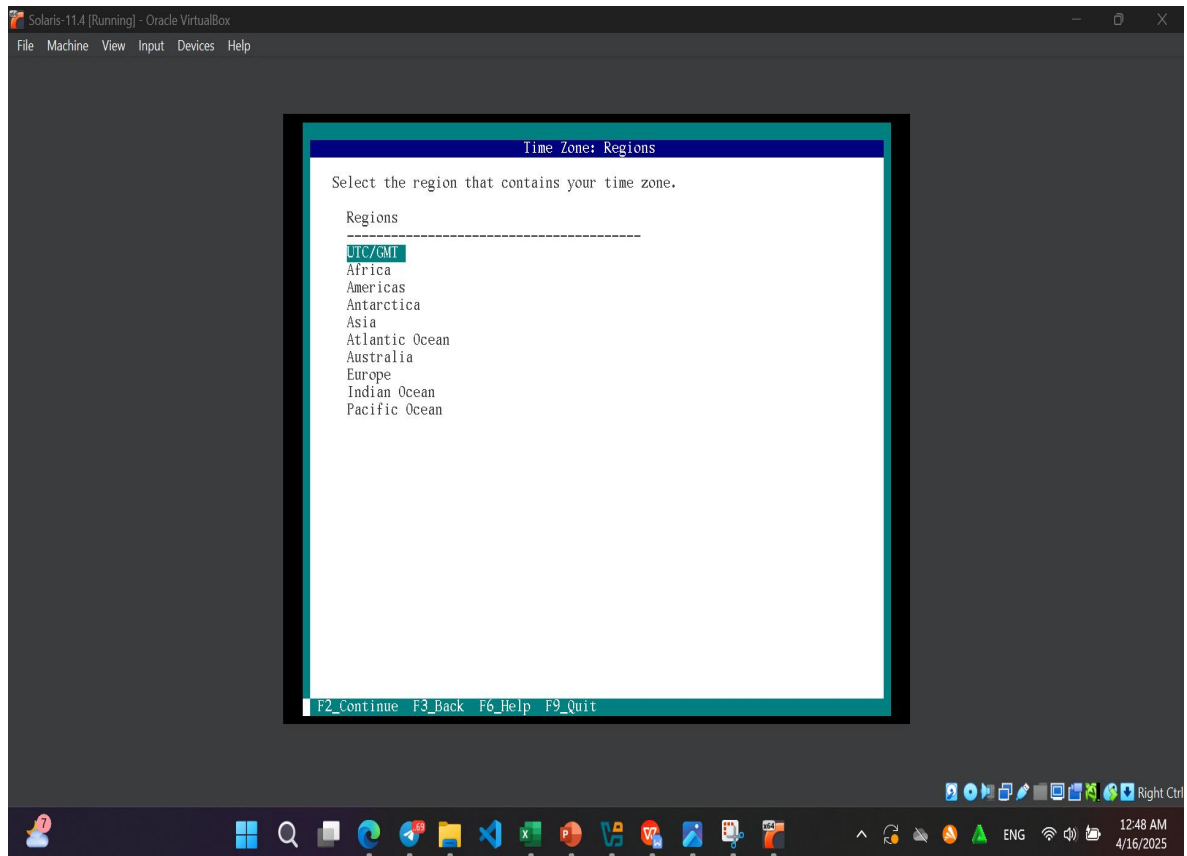
- Enter a **computer name** of your choice.
- Select network interface: I chose `net0 (e1000g0)`.
- Choose **DHCP** for automatic IP assignment.



## Step 10: Set Time Zone and Language

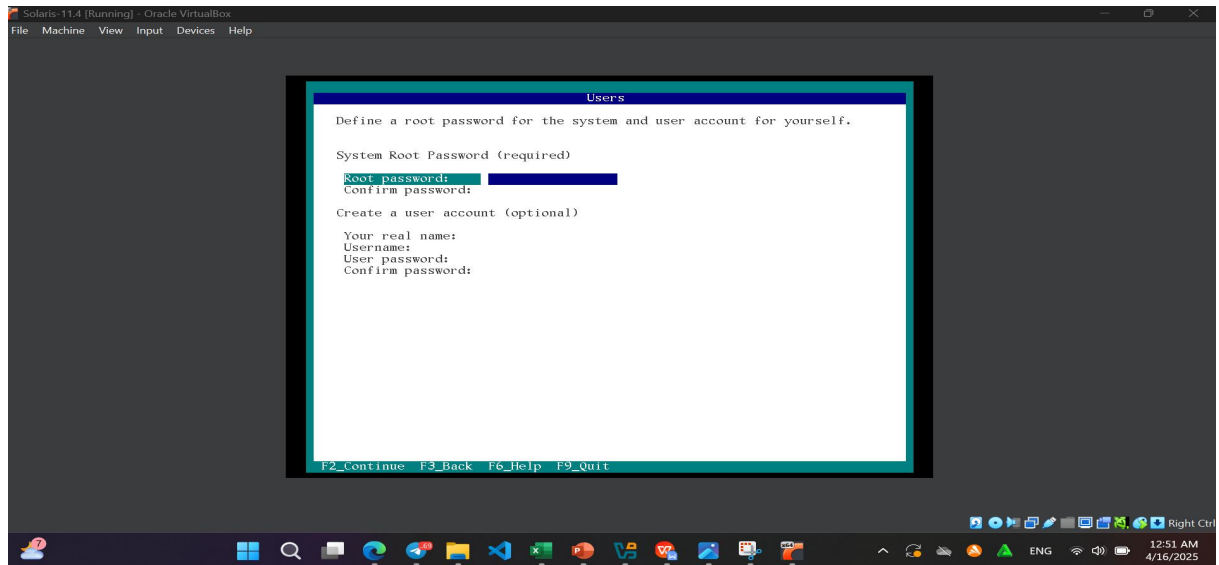
- Choose your region/time zone and language. I selected:
- Time Zone: e.g., Africa/Addis\_Ababa Language: **English (en\_US.UTF-8)**

○



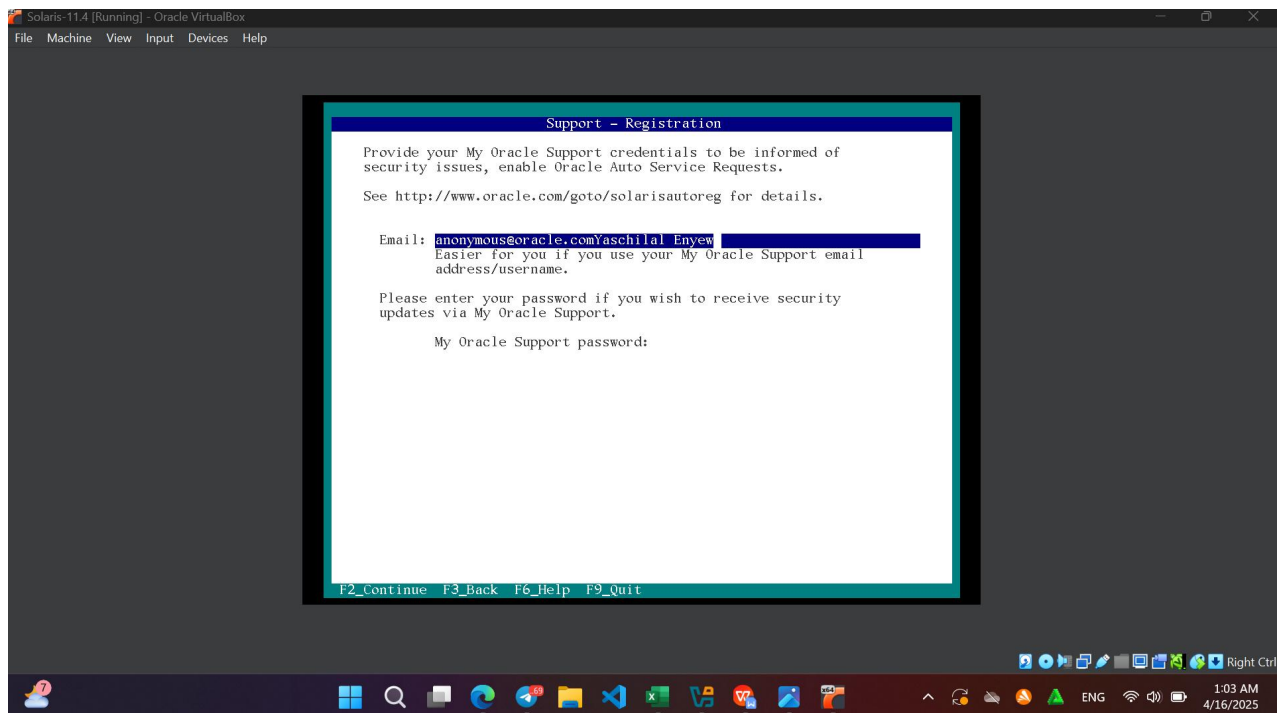
## Step 11: Set Root Password

- Create a **strong root password**. This will be used to log in as the system administrator.



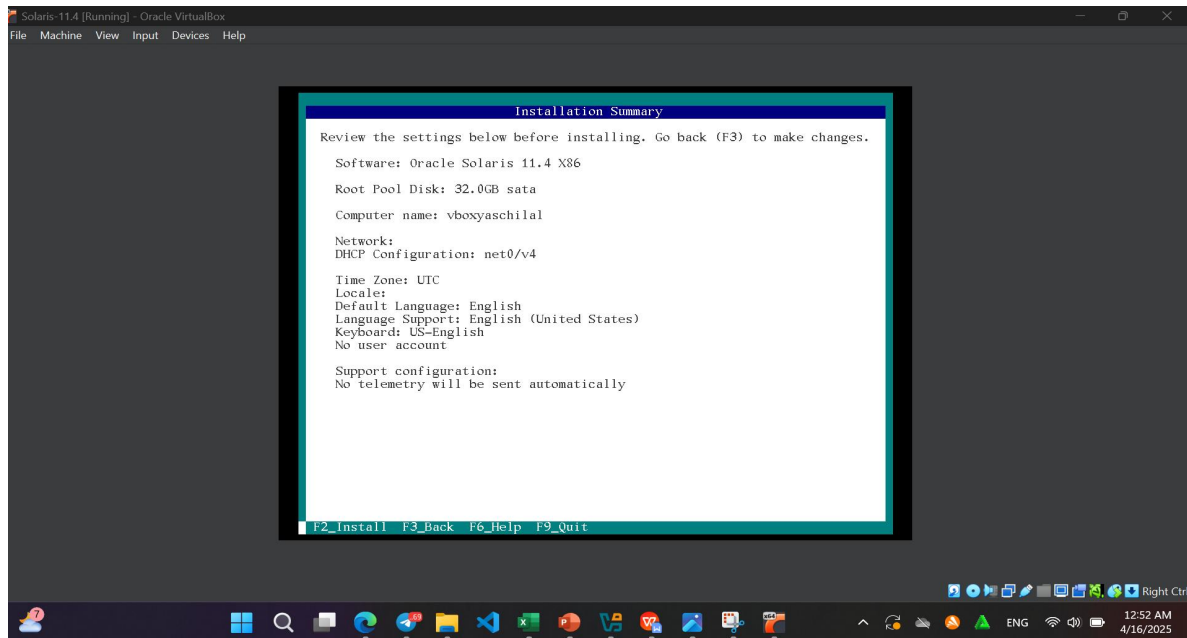
## Step 12: Oracle Support (Optional)

- You may skip entering Oracle Support credentials unless required for updates.



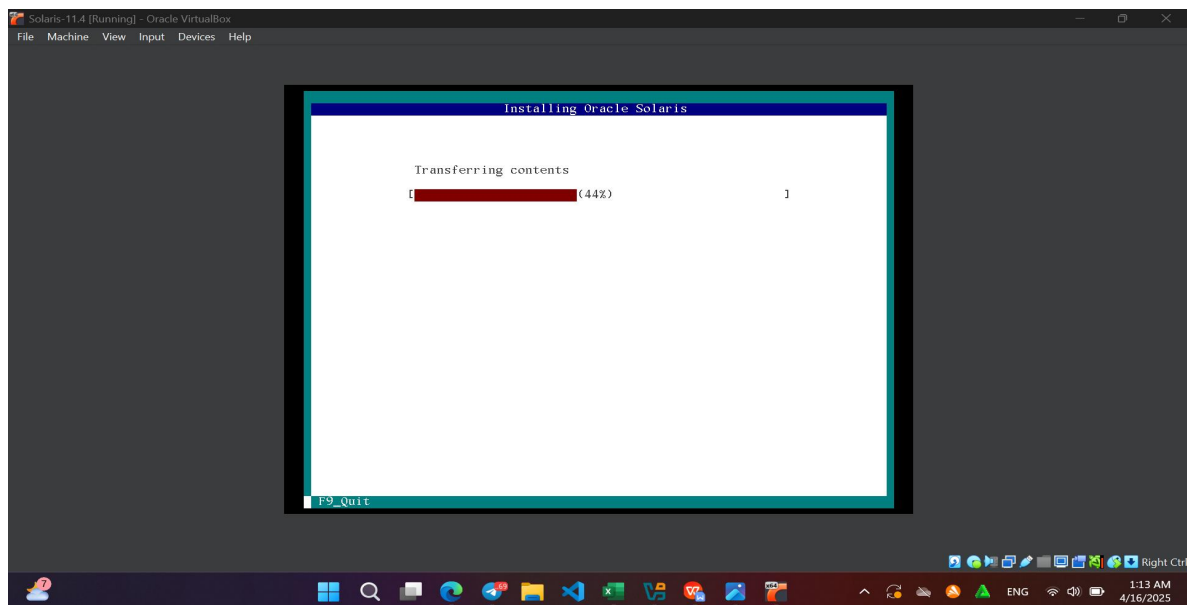
## Step 13: Installation Summary

- ✓ Review all the settings in the summary.
- ✓ If everything is correct, press **F2** to begin installation.



## Step 14: Wait for Installation to Complete

- ✓ The process may take a few minutes.
- ✓ Once complete, select **Reboot**.



## Step 15: Log in to Solaris

- ✓ After reboot, log in using the **root** account and the password you created.
- ✓ You'll be taken to the **CLI (command-line interface)**.

## e. Issues (Problem Faced)

During the installation of Oracle Solaris 11.4 in VirtualBox, I encountered a few problems that required extra attention and troubleshooting. Here's a list of those issues, along with brief descriptions and where possible, screenshots or snips can be added.

### 1. Downloading the ISO File - Error 400 Bad Request

**Issue:** When trying to download the Solaris ISO from the Oracle website using a normal browser window, I received an error: 400 Bad Request - Request Header Or Cookie Too Large

### 2. Installation UI Freezing on Disk Selection (F2 Not Working)

**Issue:** When the installer reached the disk discovery step, it displayed a message:

*"A GPT-labeled disk was not found. The following is proposed..."* But pressing F2 or using arrow keys did nothing.

### 3. No Network Option Showing

**Issue:** When selecting the network interface, I initially saw: *"No network"* even though an adapter (e.g., net0 (e1000g0)) was listed.

### 4. Keyboard or Arrow Keys Not Responding

**Issue:** At one point, arrow keys and F2 didn't respond in the terminal.

## f. Solution

**1.Solution for bandrequist error** : I opened the Oracle download link in **Incognito Mode** (private browsing window), which cleared cookies and headers, allowing the download to start successfully.

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

**Solution:** I pressed **Ctrl + C**, which interrupted the freeze and brought up the prompt:

*“Use the entire disk?”*

Confirming this allowed the installation to continue.

## 3. No Network Option Showing

**Solution:** I selected `net0` again and chose **DHCP** (Dynamic Host Configuration Protocol). After a few seconds, the interface was configured correctly.

## 4. Arrow Keys Not Responding

**Solution:** I waited patiently, pressed **Enter** or **Esc** to refresh the screen, and input control returned after a moment. It’s possible that VirtualBox temporarily lagged during a UI transition

# g. Filesystem Support

## i. Which filesystem is supported and why?

When it comes to filesystems, **Oracle Solaris** mainly uses something called **ZFS** — and it’s actually one of the best out there. Let me explain it in a simple way:

✓ What does Solaris support?

- **ZFS (Zettabyte File System)**

This is the **default and most powerful filesystem** in Solaris. It’s like the brain of storage — it manages your files, keeps them safe, and even fixes problems on its own if anything goes wrong.

- **UFS (Unix File System)**

This one is older. Solaris still understands it, but it’s not the main choice anymore because ZFS is way more advanced.

In conclusion:

Solaris supports **ZFS** because it’s **super smart, safe, and efficient**. It’s built for reliability and is perfect for big, professional systems. Other filesystems like NTFS or



ext4 are made for different operating systems and aren't supported by default.

## h. Advantages and Disadvantages

### Advantages

- i. **Enterprise-Grade Performance**  
Oracle Solaris is built for serious business and server environments. It's stable, fast, and reliable for critical tasks.
- ii. **ZFS File System**  
As mentioned earlier, ZFS is a big win. It gives you strong data protection, snapshots, auto-repair, and great performance.
- iii. **Security**  
Solaris comes with strong built-in security features like Role-Based Access Control (RBAC), process rights management, and secure user privileges.
- iv. **Scalability**  
It works well on both small setups and large data centers. You can scale up easily.
- v. **Virtualization Support**  
Solaris supports virtualization technologies like zones (containers) natively, which helps with efficient resource management.
- vi. **Resource Monitoring**  
It provides good tools to watch system performance and resource usage — helpful for system administrators.

### Disadvantages

1. **Not Beginner-Friendly**  
Solaris is more suited for system admins and professionals. If you're just getting started with OS concepts, it can feel a bit complicated.
2. **Limited Software Support**  
Unlike Windows or Linux, you don't get a wide range of apps or developer tools out of the box.
3. **Compatibility Issues**  
It doesn't support many common filesystems like NTFS or ext4, and installing certain packages might require workarounds.
4. **Hardware Driver Limitations**  
Since Solaris is specialized, it may not support all kinds of modern hardware, especially consumer-grade devices.
5. **Less Community Support**  
Compared to Linux or Windows, there are fewer tutorials or forums out there, so finding help can be a bit harder.

# i. Conclusion

Installing Oracle Solaris 11.4 in a virtual environment was a valuable learning experience. It gave me a chance to work with an enterprise-level operating system that's known for its reliability, security, and advanced features like the ZFS file system.

Throughout the process, I explored how to set up a virtual machine, handle installation issues, and interact with the Solaris interface. While some parts were challenging — especially around disk setup and network configuration — I was able to complete the installation successfully and understand how Solaris functions.

This hands-on experience also improved my confidence in dealing with operating systems that are typically used in server environments, and not just for everyday desktop use.

## J. Future Outlook / Recommendations

For future learners or anyone interested in system administration or enterprise OS environments, I highly recommend trying out Oracle Solaris in a virtual machine. Here are a few suggestions:

- ◆ **Use VirtualBox** – It's user-friendly and works well for learning OS installations.
- ◆ **Read official Solaris docs** – Oracle provides useful guides and help.
- ◆ **Start with basics** – If you're new, it's good to first understand Linux or simpler Unix systems.
- ◆ **Join forums and groups** – Since Solaris has a smaller community, getting support from experts or online forums can help a lot.
- ◆ **Try other advanced OS too** – After Solaris, explore other enterprise OS like Red Hat or SUSE to compare and learn more.

## 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.)

### What Are System Calls?

System calls are how a **program (user space)** interacts with the **operating system kernel (system space)**.

They are the "bridge" between your code and the OS.

### What is sigsuspend() ?

`sigsuspend()` is a system call that suspends (pauses) the calling process until it receives a signal. It's typically used to wait for signals like `SIGINT` (Ctrl+C) or other custom signals.

### How does sigsuspend() work?

1. The process is **suspended** (paused) and goes into a sleep state.
2. It remains suspended until a **signal** (like `SIGUSR1`, `SIGTERM`, or `SIGINT`) is received.
3. When the signal is received, the process **wakes up** and continues execution after the signal handler completes its task.

## Why use `sigsuspend()`?

This call is useful when you need to have a process pause and wait for certain events or signals to occur. For instance, you might want to wait for a specific user input or a message from another process.

### Example Usage in Solaris (C program):

Here's a simple example demonstrating how `sigsuspend()` works.

```
#include <stdio.h>

#include <signal.h>

#include <unistd.h>

void signal_handler(int sig) {

    printf("Received signal %d, resuming process...\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 a signal

    sigsuspend(NULL); // Pauses the process until a signal is received

    // This part is executed once the signal is received
```

```
    printf("Process resumed after signal received.\n");

    return 0;

}
```

### Explanation of Code:

- `signal(SIGUSR1, signal_handler)`: Registers a signal handler for `SIGUSR1`.
- `sigsuspend(NULL)`: Pauses the process and waits for any signal. The `NULL` argument indicates no specific signal mask is provided, so it waits for all signals.
- When `SIGUSR1` is sent to the process, it wakes up and executes the signal handler.

### How to Test:

1. Compile the program: using this command

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

2. Run the program:using this command

```
./sigsuspend_example
```

3. In another terminal, send the signal:

```
kill -SIGUSR1 <pid_of_program>
```

The program will pause at `sigsuspend()` and resume when the signal is received.

This is a straightforward way to use `sigsuspend()` in a Solaris or UNIX-like system to pause the process until a signal is receive.

## Thank you for Reading My Assignment!

