Table of Contents

1.Introduction

1.1background

HarmonyOS (HongmengOS) was conceived by Huawei as a pre-emptive response to new challenges in the global technology market. Reports shows that the operating system has been around since 2012, when Huawei started researching alternative operating systems as part of its long-term contingency planning. The development speeded up considerably after May 2019, when the U. S. government placed Huawei on the Entity List which effectively vetoed access to Google Mobile Services (GMS) for Huawei's Android smartphones. This event led to loss of all of Huawei's smartphone business outside China because most popular apps in Western markets were built on Google's ecosystem. Huawei executive Yu Chengodong described an in-house platform as a "plan B" in case it is prevented from using Android on future smartphones product due to the sanction.

On 24 May 2019 Huawei registered HarmonyOs as a trade mark in china. HarmonyOS 1.0 was first released on August 15, 2019, with Huawei's Honor Vision smart TVs on Huawei developer's conference. It ran on LiteOS and Linux, and included a layer to support Android apps. HarmonyOS 2.0 came out on June 2, 2021. It ran on a Linux base with Android support and used the OpenHarmony L3-L5 source code. It was first released in China, quickly reaching 10 million users in one week, and later rolled out to tablets, watches, and fitness bands by mid-2022. In July 2022, HarmonyOS 3.0 introduced new features like declarative programming and better device connectivity. HarmonyOS 3.1 followed on March 30, 2023, with support for Snapdragon chipsets and came preinstalled on Huawei P60 phones. The next major version, HarmonyOS 4.0, launched on August 4, 2023. It reduced Android code in some devices and expanded the OpenHarmony base. A later update, OpenHarmony 4.0 API 10, arrived on October 26, 2023, aligning with HarmonyOS NEXT. This version moved toward a single modular system, blending LiteOS, Linux, and a custom HarmonyOS kernel. On January 18, 2024, Huawei released the first HarmonyOS NEXT Developer Preview to developers in China. This version will lead into HarmonyOS 5, planned for release on November 26, 2024, and will drop support for older Android and LiteOS layers in favor of a unified, HarmonyOS-only platform.

1.2 Motivation

Huawei developed HarmonyOS with several clear goals in mind, all connected by a desire for greater control, flexibility, and performance across its technology. At the heart of the project was the idea of building one operating system that could power many types of devices. Instead of relying on separate systems for phones, TVs, smartwatches, or home appliances, Huawei wanted a unified platform that could bring all these devices together and allow them to work as one.

The motivation for HarmonyOS also came at a time when Huawei faced strict trade restrictions from the United States. These sanctions limited the company's access to Android and Google services, which were central to its smartphones. HarmonyOS became a way for Huawei to stay in the market without depending on U.S. technology. It allowed the company to move forward with its hardware while developing its own software to fill the gap left by Android.

The intention of HarmonyOS was to make the user experience better. The system was designed to let different types of devices communicate easily. For example, a user could share a screen between a phone and a TV, or start a task on one device and finish it on another. HarmonyOS supports this kind of interaction through built-in features, without needing extra apps or complex setups.

Huawei built HarmonyOS to be fast and efficient. The system is lightweight and manages resources carefully, which helps devices perform well even with limited hardware. This makes it suitable for both high-end phones and simpler devices like sensors or wearables. In all of this, Huawei focused on creating a system that not only met current needs but also gave it the freedom to grow on its own terms.

2.Objective

The objective is to install and evaluate HarmonyOS in a virtual environment using appropriate tools. The focus is on understanding the installation process, identifying system compatibility, and documenting each step clearly. It also involves analyzing software and hardware requirements, addressing installation issues, exploring supported filesystems, and assessing the overall performance of the operating system in a virtual setup.

3.Requirements

i. Hardware Requirements

To run HarmonyOS or its open-source version, HarmonyOS, in a virtual environment, the system should meet the following hardware specifications:

- Processor:
  A modern multi-core CPU such as Intel Core i5 (10th generation or newer) or AMD Ryzen 5 3500U or higher. The processor must support hardware virtualization extensions (VT-x for Intel or AMD-V for AMD).
- Memory (RAM):
  A minimum of 8 GB is required. However, 16 GB or more is recommended for smoother performance, especially when running virtual machines alongside development tools.

- Storage:
  At least 256 GB of SSD storage is necessary, with NVMe drives preferred for faster data access and improved overall performance during compilation and system boot.

- Graphics:
  A GPU capable of supporting OpenGL ES 3.0 or higher is recommended. This ensures proper rendering of graphical interfaces and improved hardware acceleration support during testing.

ii. Software Requirements

- DevEco Studio:
  The official tool from Huawei for HarmonyOS development. It can be downloaded from the Huawei Developer site.

- HarmonyOS SDK:
  Comes with DevEco Studio. It includes tools, libraries, and emulators for building and testing apps.

- QEMU (version 6.2 or newer):
  Used to run HarmonyOS in a virtual environment. VMware and VirtualBox don't support ARM systems, so QEMU is needed.

- OpenHarmony Image:
  This is the open version of HarmonyOS. It's used for testing and development. It works like HarmonyOS but does not include all Huawei features.

4.Installation

4.1Using decEco studio

Unlike traditional desktop or server operating systems like Ubuntu, Windows, or CentOS, HarmonyOS (OpenHarmony) is not distributed as a standard ISO or disk image file suitable for direct installation on virtual machine platforms such as VMware Workstation or Oracle VM VirtualBox. Instead, HarmonyOS is designed primarily for embedded and IoT devices (like smart TVs, watches, and mobile devices). As a result there is no direct .iso or .vmdk installation image for HarmonyOS. Instead, Huawei provides development environment through DevEco Studio, which includes built-in device emulators for testing and running HarmonyOS apps and components.

Part 1: DevEco Studio Installation

Huawei's DevEco Studio is the official integrated development environment (IDE) for HarmonyOS application and system development. However, during this work, the IDE was not accessible through Huawei's official developer portal. As a result, a stable version of DevEco Studio 5.0.13 was obtained from a third-party distribution source.

The installation completed without requiring a Huawei developer account or user authentication. The default development environment was successfully launched after setup.

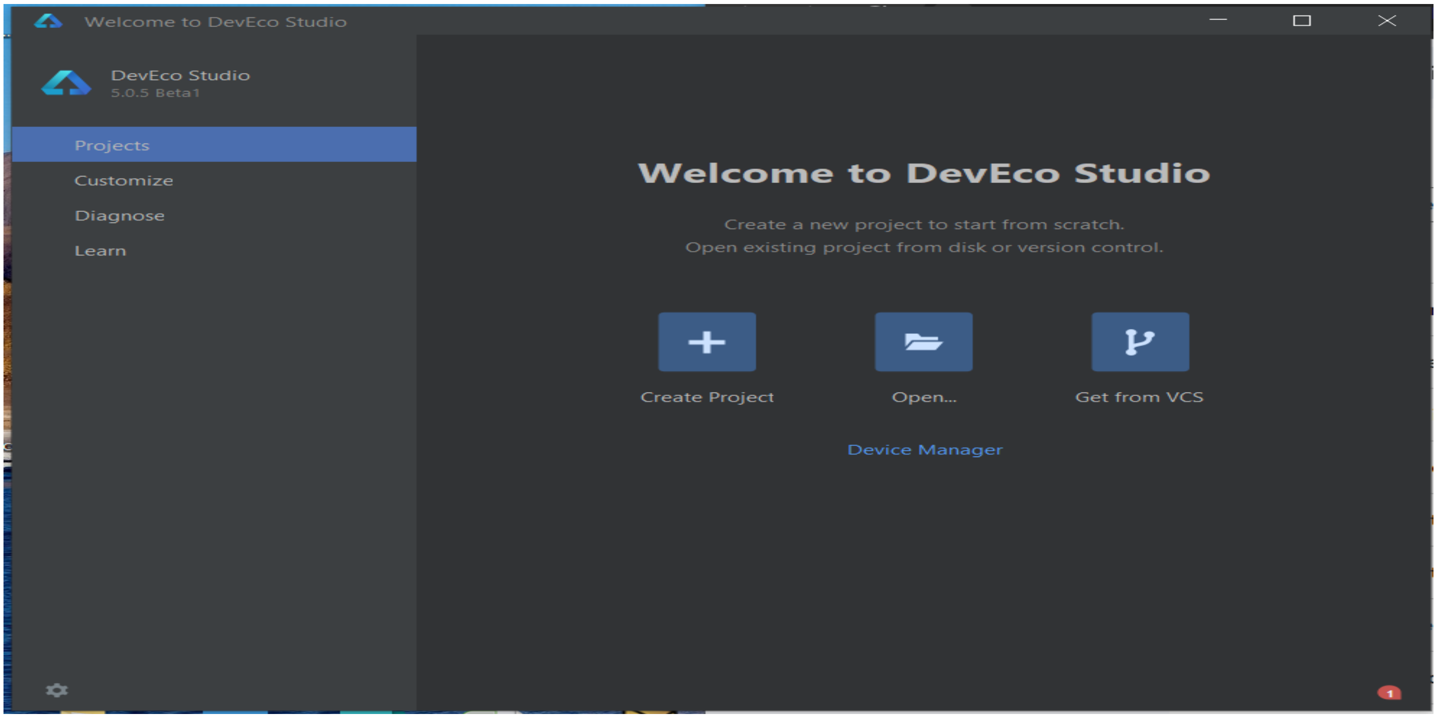


Figure 1 deveco studio stetup

Part 2: HarmonyOS Emulator Configuration Attempt

Once the IDE was installed, the next step was to emulate a HarmonyOS device using the built-in emulator.

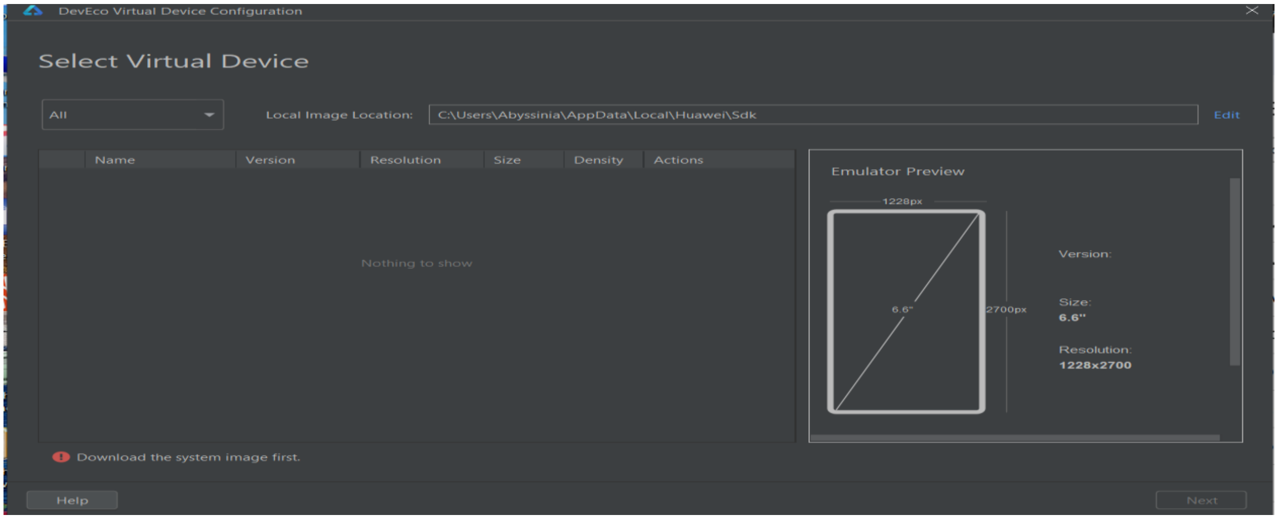
The procedure followed:

1. Open the Device Manager
   After launching DevEco Studio, the welcome screen appeared. Below the usual options like Create Project and Open Project, there was a Device Manager option. Clicking this opened the window for managing and setting up emulators.

2. Attempt to Create a New Emulator
   By selecting "+ New Emulator," the system opened a virtual device configuration window

intended for setting up: Device type, OS version, Device model, RAM, resolution, and storage parameters.

3. Setup Blocked – Missing System Image
   At this point, no devices or system images were available to select. The interface displayed a message stating: “Download the system image first.”
   As shown in the figures below, the emulator list was empty, and no configuration options were enabled. The process could not proceed without the required HarmonyOS system image, which was neither preloaded nor downloadable from within the IDE.



Local Image Location: C:\Users\Abyssinia\AppData\Local\Huawei\Sdk

Nothing to show

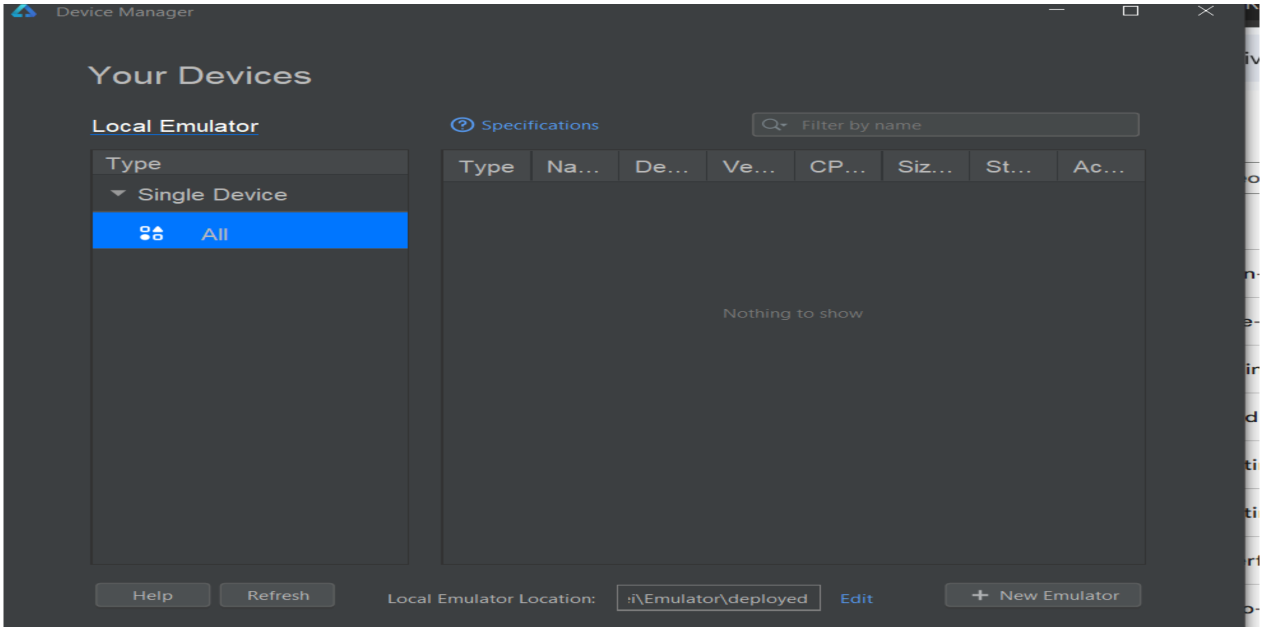
Version:

Size: 6.6"

Resolution: 1228x2700

Download the system image first.

Figure 2 Emulator configuration interface with missing system image



Nothing to show

Local Emulator Location: :i\Emulator\deployed

Figure 3Device Manager showing no available emulators

If the system image had been available and properly linked, I would have been possible to proceed with the emulator setup. Device template and hardware parameters like RAM, internal storage, and screen resolution would have selected. After creating the virtual device, it would have appeared in the list of available emulators. Launching it would have initiated the HarmonyOS boot sequence, culminating in the display of the home interface, effectively simulating a fully functional device environment.

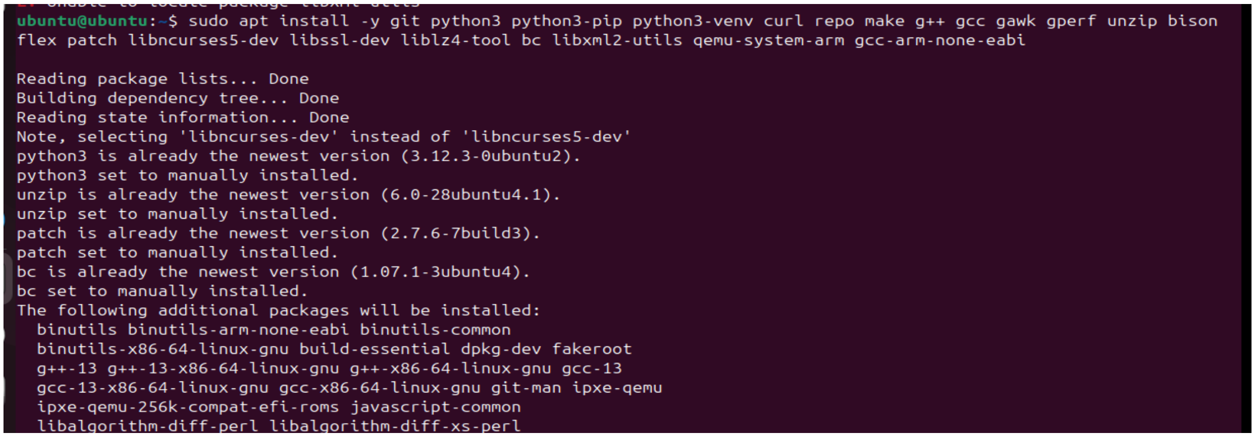
4.2Using openharmony and qemu emulator

OpenHarmony is an open-source operating system that originated from Huawei's HarmonyOS. Huawei donated the source code of HarmonyOS to the OpenAtom Foundation, leading to the development of OpenHarmony as a community-driven project. It is designed to provide a unified and secure operating system for a wide range of devices, from smartphones and tablets to IoT devices and industrial equipment.

- The emulation attempt of Openharmony using qemu emulator is executed in the following procedure:

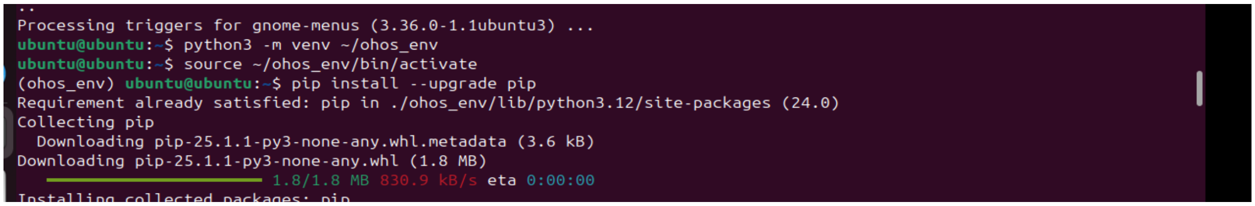
The system packages of ubuntu were updated to ensure that all software was current. his was achieved by executing the following command in the terminal: “sudo apt update && sudo apt upgrade -y”. This command updated the list of available packages and their versions, then upgraded the installed packages to the latest versions.

then, essential tools and dependencies required for building and running OpenHarmony were installed. These included compilers, libraries, and utilities necessary for compiling software and managing source code.



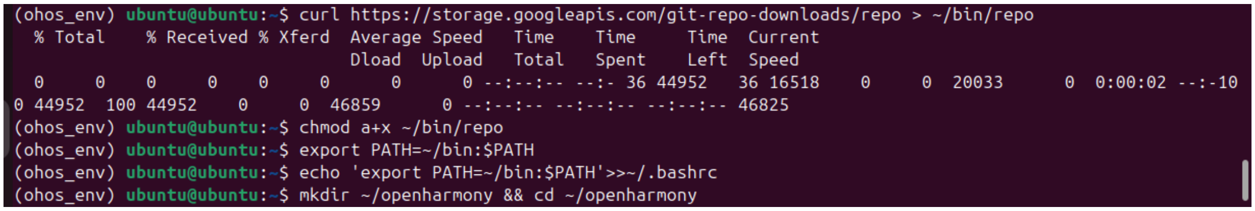
```
ubuntu@ubuntu:~$ sudo apt install -y git python3 python3-pip python3-venv curl repo make g++ gcc gawk gperf unzip bison
flex patch libncurses5-dev libssl-dev liblz4-tool bc libxml2-utils qemu-system-arm gcc-arm-none-eabi

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libncurses-dev' instead of 'libncurses5-dev'
python3 is already the newest version (3.12.3-0ubuntu2).
python3 set to manually installed.
unzip is already the newest version (6.0-28ubuntu4.1).
unzip set to manually installed.
patch is already the newest version (2.7.6-7build3).
patch set to manually installed.
bc is already the newest version (1.07.1-3ubuntu4).
bc set to manually installed.
The following additional packages will be installed:
  binutils binutils-arm-none-eabi binutils-common
  binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot
  g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc-13
  gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu git-man ipxe-qemu
  ipxe-qemu-256k-compat-efi-roms javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl
```

A Python virtual environment was then set up to manage project-specific Python packages without affecting the system-wide Python installation.



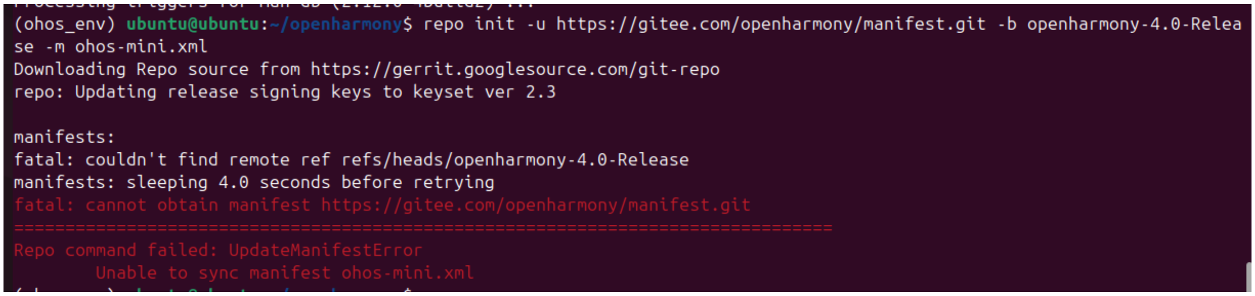
```
..
Processing triggers for gnome-menus (3.36.0-1.1ubuntu3) ...
ubuntu@ubuntu:~$ python3 -m venv ~/ohos_env
ubuntu@ubuntu:~$ source ~/ohos_env/bin/activate
(ohos_env) ubuntu@ubuntu:~$ pip install --upgrade pip
Requirement already satisfied: pip in ./ohos_env/lib/python3.12/site-packages (24.0)
Collecting pip
  Downloading pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
Downloading pip-25.1.1-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 830.9 kB/s eta 0:00:00
Installing collected packages: pip
```

The repo tool, which helps manage multiple Git repositories, was installed next. This tool is common in large projects like OpenHarmony.



```
(ohos_env) ubuntu@ubuntu:~$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:- 36 44952   36 16518    0     0  20033      0  0:00:02 --:-10
0 44952  100 44952    0     0  46859      0 --:--:-- --:--:-- --:--:-- 46825
(ohos_env) ubuntu@ubuntu:~$ chmod a+x ~/bin/repo
(ohos_env) ubuntu@ubuntu:~$ export PATH=~/bin:$PATH
(ohos_env) ubuntu@ubuntu:~$ echo 'export PATH=~/bin:$PATH'>>~/.bashrc
(ohos_env) ubuntu@ubuntu:~$ mkdir ~/openharmony && cd ~/openharmony
```

the next step was to initialize the OpenHarmony repository. A directory for the source code was created, and the repository was initialized using the repo tool, error occurred. As a result, the source code could not be retrieved, and the process was halted at this point.



```
(ohos_env) ubuntu@ubuntu:~/openharmony$ repo init -u https://gitee.com/openharmony/manifest.git -b openharmony-4.0-Relea
se -m ohos-mini.xml
Downloading Repo source from https://gerrit.googlesource.com/git-repo
repo: Updating release signing keys to keyset ver 2.3

manifests:
fatal: couldn't find remote ref refs/heads/openharmony-4.0-Release
manifests: sleeping 4.0 seconds before retrying
fatal: cannot obtain manifest https://gitee.com/openharmony/manifest.git
================================================================================
Repo command failed: UpdateManifestError
        Unable to sync manifest ohos-mini.xml
```

If the cloning had succeeded, the next step would have been to build the OpenHarmony image using the hb tool. This command-line utility simplifies the build process by managing configurations and compiling the source code.

First, the build environment would be set up with hb set. This command configures the product, board, and kernel to be used. After setting up, the build process would be initiated with hb build. This command reads the configuration, runs the GN tool to generate Ninja build files, and then uses Ninja to compile the code. The final output would be a set of images ready for deployment or emulation.

Once the image is built, it can be emulated using QEMU, a tool that emulates hardware platforms. QEMU can simulate the scenario where a kernel runs on different boards, so that the kernel no longer depends on physical development boards. This allows for testing and exploration of OpenHarmony's features in a virtual environment.

5.Problem faced

I encountered several challenges while attempting to set up the emulation environment. Initially, I couldn't locate DevEco Studio on Huawei's official website. After installation, I attempted to create a new emulator, but the system asked me to download a system image, which was unavailable, halting the setup process. Subsequently, I tried to emulate OpenHarmony using QEMU. However, I was not able to clone the OpenHarmony source code using the provided commands. Attempting to download the source manually was not possible due to its large size, approximately 31.6GB. Additionally, Additionally, I was unfamiliar with the command-line utilities involved, which made the process more challenging.

6.Solution

Though DevEco Studio wasn't available on Huawei's official site, I was able to get it from a third-party source. It installed without issues and didn't require a Huawei account. That allowed me to at least open the tool and explore its features. Despite my best efforts, I couldn't solve the remaining problems.

7.Filesystem support

HarmonyOs officially supports the following filesystems:

1.EROFS (Enhanced Read-Only File System)

EROFS is a read-only file system developed by Huawei. It's designed to save storage space and improve read speeds. This makes it useful for system files that don't change, like those in smartphones and IoT devices.

2.HMDFS (Harmony Distributed File System)

HMDFS is HarmonyOS's built-in distributed file system. It lets devices on the same network share and access files as if they were local. This is helpful for syncing data between devices like phones, tablets, and smart TVs.

3.exFAT (Extended File Allocation Table)

Huawei added exFAT support to HarmonyOS in October 2024 to enhance compatibility with external storage devices, such as USB drives and SD cards, that use the exFAT file system.

8.Advantages and Disadvantages of HarmonyOs

7.1 Advantages of HarmonyOS

HarmonyOS offers several notable benefits that enhance user experience and device interoperability.

One of its key strengths is cross-device integration. The "Super Device" feature enables seamless interaction across various devices, including smartphones, tablets, smart TVs, and wearables. This allows users to share files, media, and even app states between devices effortlessly.

Additionally, HarmonyOS boasts a microkernel architecture, which enhances security and performance by isolating critical components, reducing the risk of system-wide failures. The system is also designed to provide smooth performance across a range of devices, from smartphones to IoT devices, ensuring efficient resource utilization.

Huawei has embraced open-source principles with HarmonyOS, allowing developers to contribute to its growth and ensuring transparency in its development process. Furthermore, the operating system incorporates advanced security measures, including the "Star Shield" architecture, to protect user data and privacy.

7.2 Disadvantages of HarmonyOS

Despite its advantages, HarmonyOS faces certain challenges that may affect its adoption and usability.

The app ecosystem for HarmonyOS is smaller compared to Android and iOS, which may limit the availability of certain applications. Some applications may experience issues such as lag or crashes, particularly those not optimized for HarmonyOS.

Developers accustomed to Android may face challenges in adapting their applications to HarmonyOS, requiring additional time and resources. Huawei's position as a Chinese company has led to scrutiny and restrictions in certain markets, potentially limiting the global reach of HarmonyOS.

9. Conclusion

Huawei's independent operating system HarmonyOS (formerly Hongmeng), which began as a response towards U.S. sanctions; it started as an alternative for Android, but it has grown becoming its own independent operating system, currently deployed on over a billion Chinese devices such as phones, tablets, home appliances and more. The newest release of HarmonyOS, HarmonyOS NEXT, has removed any support for Android entirely, as it will now move forward with native apps. They are also aiming for more growth abroad as they look to pitch HarmonyOS as a third major mobile platform next to Android and iOS. While there are challenges we face as Huawei also looks to tackle some of the biggest ones such as growing a large enough app ecosystem, the expansion towards a full OS shows Huawei are serious about their technological independence and working on long term, gradual expansion.

Regarding the installation process, HarmonyOS is hard to install unless you already know a lot about development tools and systems. It’s not beginner-friendly and doesn’t support common virtual machine platforms.

10. Recommendation

For Huawei to compete globally, HarmonyOS must become more useful and user-friendly for everyone. Huawei should ensure their developer tools are accessible and easy to understand for people around the world. This means removing download restrictions and not having instructions only in Chinese.

The limited availability of native apps remains a challenge. Huawei should continue investing in developer support and incentives to increase the number of essential applications, ensuring users have access to the tools they need.

HarmonyOS's adoption outside China is hindered by the absence of popular global apps and services. Collaborating with international developers and ensuring compatibility with widely used applications can help bridge this gap.

Developers face challenges adapting their apps to HarmonyOS. Simplifying development tools and providing comprehensive documentation can ease this transition, encouraging more developers to support the platform.

Virtualization

1.What Is Virtualization?

virtualization is the technology that makes it possible for a virtual(simulated)to be established instead of having to use only physical equipment. By abstracting from the underlying physical system, virtualization makes it possible for virtual machines (VMs) to be executed in concurrent manner on a single physical machine. Each virtual system act as if it were one physical machine even though they are all sharing the same hardware resources. This abstraction is facilitated by a software layer that allocates physical resources (CPU, memory, storage, etc.) to each VM, so that it appears that each virtual machine has its own hardware.

Why Is Virtualization Used?

Virtualization has become an important process in modern computing by a number of reasons:

1. Resource Optimization and Efficiency: When a system is not virtualized, hardware components are often underutilized. Virtualization allows the maximum utilization of resources by enabling a physical machine to run many operating systems and applications at the same time,
2. Isolation and Enhanced Security: Each VM runs in its own isolated environment. Therefore, problems like a crash, breach, or bug only impact one VM, not others, on the same hardware. Therefore, in critical environments, if a VM must run potentially vulnerable or experimental software, systems can be secured and maintained more easily by keeping that software isolated in its own VM.

3. Flexibility and Scalability: There is a high degree of flexibility in deployment when using virtualization. Systems can be scaled toward the up or down, quickly and easily, by spinning up new VMs, enabling organizations to better react to changing business needs.

4. Less cost: Virtualization enables the ability to run multiple VMs n a single host and limits the need to deploy additional equipment, which only further drives up capital hardware budgets, and continuing operating costs on power, cooling, and maintenance.

3.How Virtualization Works

Virtualization makes one computer act like many computers. Here's how it accomplishes this step by step:

1.Hypervisors
This is the software that makes virtualization possible.

Type 1 (Bare-metal): This installs directly on the hardware of the computer. It doesn't need an operating system beneath it. That makes it more efficient and ideal for servers. Examples: VMware ESXi, Microsoft Hyper-V.

Type 2 (Hosted): This runs on top of an already installed OS, like Windows or Linux. It's easier to use at home. Examples: VirtualBox, VMware Workstation.

2.Hardware_Assistance
New CPUs have special features built-in "like" Intel VT-x or AMD-V. They help the hypervisor with its job more effectively and at a higher speed. It'd be a lot slower without them.

3.Shared Resources
The hypervisor splits the computer's resources (CPU, memory, and storage) and gives each

virtual machine a portion. Each VM thinks it's on its own computer, even though they're all running on the same real one.

4.Virtual Devices
Each VM needs things like a network card, hard drive, or USB ports. The hypervisor presents virtual versions of these devices, so the VM thinks it's using real hardware. It's all pretend, but it works.

5.Management Tools
Other tools enable you to control your virtual machines. You can start or stop them, back them up, and move them around if you have to. Examples: VMware vCenter and Proxmox.