Android Open Source Project (AOSP)

JUAN PABLO ARCHILA AMAYA
JUANA VALENTINA CELIS FORERO
LAURA VANESA FERNANDEZ BARRETO
JUAN SEBASTIAN MARTINEZ NOREÑA
NORBEY ESTEBAN ROJAS BECERRA

UNIVERSIDAD PEDAGOGICA Y TECNOLOGICA DE COLOMBIA
INGENIERÍA DE SISTEMAS
SISTEMAS OPERATIVOS
TUNJA BOYACÁ
2024

LINK VIDEO PRESENTACIÓN

https://youtu.be/96DMrCE0FYA

GUIÓN

CONSCIOUSNESS - JUANA VALENTINA

CYPHER/MRS SMITH - JUAN SEBASTIAN

NEO - JUAN PABLO

TRINITY -> LAURA VANESA

MORFEO -> NORBEY ESTEBAN

Scene 1: Pill Selection

1. **Consciousness (voice-over):** Neo, at this decisive moment, you must choose your destiny.
2. **Morpheus:** In front of you, you have two options. The blue pill will take you back to the world of proprietary systems, where everything is controlled and limited. The red pill will open your eyes to the world of AOSP: freedom, customization, and control over your own destiny.
3. **Neo:** AOSP? Why is it important?
4. **Consciousness (voice-over):** AOSP is the Android Open Source Project, the foundation of Android in its purest form, without the limitations of closed versions. It offers you the ability to customize your experience, decide what goes into your system, and provides the foundation on which various customized versions of Android are developed, giving a great door to adapting Android operating systems for the uses of each person or company.
5. **Morpheus:** Choosing AOSP means you decide how to use your device and who it's compatible with, from smartphones and tablets to specialized devices like TVs, cars, and wearables. But Neo, this is a path of responsibility and knowledge, and being aware of the dangers that lie ahead.
6. **Neo (sighs and takes the red pill):** I'm ready. I want to know the truth.
7. **Morpheus (smiles and nods):** Welcome to true freedom.

Scene 2: Induced Learning

8. **Cypher:** Neo, it's time to take you beyond the superficial, to the heart of AOSP knowledge. Prepare to receive a download of information.
9. **Neo (looking nervously):** What kind of information, Cypher?
10. **Cypher (smiles):** Information on how AOSP has evolved to make more agile and efficient development possible. It's a system that relies not only on its open source code, but on technological improvements that optimize its performance and flexibility.

11. **Neo (quietly):** I can see it... the file system... It's a type of file system that allows the execution of binaries before the entire package has been loaded!
12. **Trinity (interjects softly):** Keep going, Neo. You're getting the gist.
13. **Cypher:** Exactly, Neo. This is made possible by IncFS, Google's virtual file system. It's designed to streamline debugging and testing of applications. Previously, developers had to wait for the entire binary to be transmitted to the emulator or test device. With IncFS, this process is sped up by allowing the binary to be executed before the transmission is complete.
14. **Neo:** A layer on top of the real file system?
15. **Trinity (nods, eyes fixed on screen):** It's like a strategic cache, but deeper in the architecture.
16. **Cypher:** Right. IncFS is like a cache that works as a layer on top of the existing file system, integrating itself directly into the operating system kernel. This gives it low-level access, which makes sense, since its purpose is to serve as a bridge between actual storage and immediate execution of partial data.
17. **Neo (in amazement):** So, application file access is different depending on the type of file?
18. **Cypher:** Yes, Neo. For example, app-specific files, such as data or caches, are stored and accessed through methods like getFilesDir() or getCacheDir() for internal storage, and getExternalFilesDir() or getExternalCacheDir() for external.
19. **Neo:** And media content…
20. **Trinity (interjects confidently):** Access APIs are crucial for efficient management.
21. **Cypher:** It’s accessed via the MediaStore API, allowing images, audio, and video to be shared efficiently between apps. But downloaded documents and other files are handled using the Storage Access Framework, making it easier to control privacy and accessibility.
22. **Neo (trying to process everything):** You also mentioned something about preferences…
23. **Cypher:** App preferences, which are stored as key-value pairs, are handled using Jetpack’s Preferences library. And when we talk about more structured data, we enter the realm of databases. This is where the Room persistence library comes in, allowing apps to handle databases more efficiently.
24. **Neo:** This all sounds incredibly efficient. But how do you achieve storage emulation?
25. **Cypher (with a touch of humor):** Ah, that's where FUSE comes in. It was originally used to emulate storage and allow apps to transparently access both internal storage and external SD cards. But FUSE, while useful, has considerable overhead.
26. **Neo:** What kind of overhead are we talking about?
27. **Cypher:** Every userspace request follows a complicated path: from userspace to the VFS, then to the FUSE driver, through the FUSE daemon, and back to the VFS before reaching the ext4 system and page cache. This process can slow down access, but it's a solution that has enabled the storage flexibility we know.

28. **Trinity (looking at Neo):** That's why it's important to know when to use each tool, Neo. Each choice has a cost and a benefit.
29. **Neo (with a knowing look):** So every part of AOSP, from file access to databases to storage emulation, is designed to offer a balance between flexibility and efficiency.
30. **Cypher:** And that, Neo, is why understanding how AOSP works from the ground up is the key to getting the most out of it. It's not just an operating system, it's an ecosystem of possibilities.
31. **Consciousness (voiceover):** And so, Neo understood that true freedom was not only in knowing AOSP, but in contributing to its development and evolution. He delved into the General Architecture Memory of AOSP, which defines how memory is managed within the system to ensure optimal performance, security, and resource control for both applications and system processes.

    Neo discovered that the architecture featured efficient resource allocation, allowing dynamic memory assignment and reuse based on system demands. Each application operated in an isolated memory space, ensuring data security through sandboxing. The Android Runtime (ART) played a crucial role in managing memory at runtime, using garbage collection to allocate and free resources as needed.

    Scalability was another key element, enabling devices from budget to high-end models to manage resources effectively. AOSP also supported virtual memory, using techniques like paging and zram compression managed by the kernel to extend RAM capabilities and handle high demand.

    The architecture’s layers included User Applications interacting with Android APIs, followed by the Framework for secure memory management. ART provided an isolated runtime environment, while System Services coordinated resources. The HAL ensured hardware independence, and at the core was the Linux Kernel, managing both physical and virtual memory with features like zram and the Low Memory Killer.

Scene 3: Back to the Real World

32. **Morpheus:** Welcome to the real world, Neo. This is where true AOSP users take control. Each AOSP family brings something different to the table.
33. **Trinity:** LineageOS is known for its stability and simplicity, ideal for those who want customization and long-term support.
34. **Morpheus:** Then there's Pixel Experience, which offers an experience close to Google devices, but with more control for the user.

35. **Neo:** What about the others?
36. **Trinity:** Each has its own approach, from versions that prioritize privacy, like GrapheneOS, to those that focus on delivering the best possible performance.
37. **Neo:** So, AOSP isn't just a system. It's a community and a philosophy of freedom.
38. **Morpheus:** Exactly, Neo. And now, you know that it's up to you and all users to help keep it that way.
39. **Neo (with a smile):** I'm ready to join the AOSP community, together with cypher.
40. **Consciousness (voiceover):** On the other hand, AOSP families are versions that start from the same base, adapting to different uses and needs. 'Pure' or 'Stock' Android is the most essential form, without extra customizations, offering a clean experience. Android with Google Services (GMS) goes a step further, adding Google applications and services for a complete integration, improving the user experience.

    Then, there are custom ROMs, versions modified by independent developers or communities, which offer deep customization, advanced privacy control and performance optimization. On the other hand, there are AOSP Derived Operating Systems, where companies like Amazon develop unique systems based on AOSP, eliminating Google services and using their own ecosystems.

    Finally, Android Open Source for Specialized Devices adapts AOSP for specific cases: Android TV for televisions, Android Automotive for vehicles, and Android Go for devices with limited resources. Each version is designed with interfaces and features specific to its environment, showing the flexibility and adaptability of AOSP.

Scene 4: Explanation of the Matrix

41. **Morpheus:** Neo, this is the Matrix. What you see and feel is a facade created by software. But behind this reality is AOSP.
42. **Neo:** So, everything we experience is part of an open system?
43. **Morpheus:** Exactly. AOSP's features include its open source and adaptability. It has essential components such as ART, the Android Runtime, which runs applications, and native libraries that optimize performance. Before, we worked with Dalvik, which used registers to store data during the execution of an application. But we needed to be even more efficient and that's why now it's handled by ART.
44. **Trinity (entering the scene):** And the best thing about AOSP is that anyone can modify it. That's why there are versions like LineageS and Pixel Experience, which take control and adapt it to their needs. By having tools like Android SDK and debuggers like ADB it is easy

to create a different version of the existing ones, and improve it to our taste, convenience and capacity. But all this is still AOSP and its strong and fast system applications.

45. **Neo (looking around):** I understand... AOSP is the pillar on which these realities are built.
46. **Morpheus:** But remember, Neo, with freedom comes responsibility. Developers and manufacturers must keep it cohesive to avoid fragmentation, which is something that affected Cypher greatly. He was a developer who loved the working architecture of AOSP, but realizing that every day it was more and more difficult to create applications compatible with each different version of Android, he entered a state where he calls himself Mrs Smith and wishes to return to proprietary Android systems.

Scene 5: Bullet Dodging Fight

47. **Mrs. Smith:** You can have all the freedom of AOSP, but with it comes disadvantages, like fragmentation and lack of support. What will you do when a device gets stuck on an outdated version? What will you do when your apps become incompatible on some versions and you get blamed?
48. **Neo (while dodging a bullet):** Yes, I know. Fragmentation is a real problem and that's why people are doing their best to improve it every day, projects like Project Treble which is a Google initiative to modularize the Android operating system and make updates easier. Or that every day new standards are created so that within freedom there is common control.
49. **Trinity:** And look what AOSP has achieved. Developer communities work tirelessly on custom ROMs to maintain security and add new features.
50. **Mrs. Smith (sarcastic):** Ah, but the average user doesn't know how to do that, right? The reality is that most people are still trapped in the Matrix. People don't know all the families that Android has, and they're uncomfortable not having the Google services they've always had.
51. **Neo (stopping a bullet with his hand):** Maybe, but more and more people are learning to take control of their devices and escape the Matrix of limited versions. They're learning that there's a new way to innovate on mobile devices, they're creating communities to improve every day, and it's a very flexible tool. Come back to us and you'll learn how to help the community with fragmentation problems and continue learning about this architecture.
52. **Mrs Smith:** Okay, you are right, let me try again being part of your team. The family is the first thing!

REFERENCIAS

Cano, R. (2023, 25 de marzo). *AOSP: Así es el Android Open Source de Google que queda como opción para Huawei*. Xataka. Recuperado el 7 de noviembre de 2024, de https://www.xataka.com/aplicaciones/aosp-asi-al-android-open-source-google-que-queda-como-opcion-para-huawei

Esteban. (2019, 2 de junio). Qué es Android AOSP y por qué es tan importante para marcas como Huawei. El Español. Recuperado el 7 de noviembre de 2024 https://www.elespanol.com/elandroidelibre/tutoriales/20190602/android-aosp-importante-marcas-huawei/403210371_0.html

Google. (s.f.). *Acerca de la configuración del entorno de desarrollo de Android*. Recuperado el 7 de noviembre de 2024, de https://source.android.com/docs/setup/about?hl=es-419

Google. (s.f.). *Preguntas frecuentes sobre la configuración del entorno de desarrollo de Android*. Recuperado el 7 de noviembre de 2024, de https://source.android.com/docs/setup/about/faqs?hl=es-419

Marín, Á. (2021, 3 de junio). *¿Qué es el Android Open Source Project (AOSP)?*. Malavida. Recuperado el 7 de noviembre de 2024, de https://www.malavida.com/es/articulos/que-es-el-android-open-source-project-aosp