

ROYOS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

**MASTER OF DESIGN
IN
VISUAL COMMUNICATION**

Submitted by:

**PUROO KUMAR ROY
(2K22/MDVC/06)**

Under the supervision of
PROF PARTHA PRATIM DAS



DEPARTMENT OF DESIGN
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042
MAY, 2024

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CANDIDATE'S DECLARATION

I, PUROO KUMAR ROY, Roll No. 2K22/MDVC/06 of M.Des. (Visual Communication), hereby declare that the project Dissertation titled RoyOS which is submitted by me to the Department of Design, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Design, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

PUROO KUMAR ROY

Date: 14 May 2024

DEPARTMENT OF DESIGN
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “RoyOS” which is submitted by Puroo Kumar Roy, Roll No. 2K22/MDVC/06, Department of Design, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Design, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

PARTHA PRATIM DAS

Date: 14 May 2024

DEPARTMENT OF DESIGN
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Professor Partha Pratim Das for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve my target by explaining all the tasks to be done and explaining to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi

PURROO KUMAR ROY

Date: 14 May 2024

ABSTRACT

Mobile phones are presently one of, if not the most indispensable electronic devices in the world. One is expected to always have a device that can send and receive calls and messages, listen to music, watch videos, play games, and in most cases, have a flashlight. Currently, in the mobile operating system market, there are two major key players: Android and iOS. However, this was not always the case.

Twenty years ago, every cellphone had a unique operating system. Each device had a distinct approach to how to interact with the user and deliver the experience. This was because each company owned its own operating system, which it used to showcase its creativity in creating an environment that matches the design of the mobile phone. With a rise in the usage of Android as the primary operating system choice for the majority of the companies in the market, concerns regarding power consumption, control and cost implications need to be addressed.

The thesis aims to analyse the problems that arise from using an operating system owned by a multinational tech giant and develop an alternative for it. The primary aim here is to dive into the specifics of what makes an Android phone consume so much space, whether these *system apps* are as essential to the performance of the mobile phone, and how much space can be done away with to make the smartphone take up less space by default and operate faster. The secondary aim is to make an operating system distro that is attractive to look at and easy to adapt to. And finally, the tertiary aim is to see how many other features can be added to the distro which Android conventionally does not offer.

To achieve this aim, there needs to be a clear understanding of several topics. Primarily the research has to be on Android, from its origin, its concept, its rise in popularity, when and under what circumstances it was acquired by Google and later on, what major changes it has shown in terms of the features that it offered. Other than Android, other

Linux-based operating systems owned by mobile phone companies that do not use the Android operating system in their phones, as well as mobile phones that do not use Linux-based operating systems must be explored to get a better understanding of how different the distribution of an operating system can be.

While this shall be the general approach towards the secondary research of the thesis, the primary research will be done to get the perception of the end user and potential market for any mobile phones using a new Linux-based operating system distribution. It must be made clear over here that the project here is regarding an operating system distribution, and not the operating system itself. The difference between the two terms shall also be made clearer as the research progresses.

In the end, the outcome has to be the operating system distribution with a workflow from the point of view of the user, as they are going to witness their first interaction with a device using the distribution.

CONTENTS

Candidate's Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Contents	vi
List of Figures	vizi
List of Tables	viii
List of Symbols, Abbreviations	ix
CHAPTER 1 INTRODUCTION	1
1.1 An Introduction to Unix	1
1.2 Unix-like	3
1.3 GNU and the Rise of Linux	4
1.4 Kernel	5
1.5 An Introduction to Linux	6
1.6 Android	9
1.7 Android as an Open-Source	10

CHAPTER 2	LITERATURE REVIEW	12
2.1	AOSP documentation	12
2.2	Apache License Version 2.0	16
CHAPTER 3	PROBLEM IDENTIFICATION	20
3.1	Privacy	20
3.2	Storage	21
3.3	Customisation	22
CHAPTER 4	USER RESEARCH AND METHODOLOGY	23
4.1	Secondary Research	23
4.2	Primary Research	23
CHAPTER 5	CONCEPT DEVELOPMENT RESULTS	28
5.1	Introduction to RoyOS	28
5.2	Solving the Problem of Privacy	29
5.3	Solving the Problem of Storage	30
5.4	Solving the Problem of Customisation	31
CHAPTER 6	CONCLUSION	35
	References	37
	Appendix	43

LIST OF TABLES

All the Unix-like operating systems.	4
An oversimplified definition of a kernel	5
The difference between a monolithic kernel, a microkernel and a hybrid kernel's structure	6
The decline of Unix and the rise of Linux	8
All the previous and current releases of AOSP	14
Terms and definitions in the Apache License Version 2.0	16-17

LIST OF FIGURES

Screenshot from a Unix terminal	1
The desktop screen of Ubuntu, the most well-known Linux distro	7
All the Ubuntu flavours	8
Android's logo since 2023	9
A demo version of what the home screen of Android was supposed to look like at the time of being pitched	10
The first slide of the presentation that was shown by the Android Inc. team when they were pitching the operating system to Google	10
A summary of the difference between Android and AOSP	11
The landing page of the AOSP website, on which the documentation can be found	12
The logo of Apache License Version 2.0	16
The booting of RoyOS	28
The first screen of the homepage	29
The second screen of the homepage	29
A step-by-step tutorial on how to delete the GUI of the OS	30
After deleting the GUI	30
The option to select primary and secondary colours in the Display settings	31
GUI theme before and after applying the changes	32
Before all the changes are applied; Bottom: After the changes are applied	33
The icons on the home screen resized	34
The option to resize and rotate an app appears when the user long-presses the app	34

LIST OF ABBREVIATIONS

Abbreviation	Meaning
FOSS	Free and Open-Source Software
OS	Operating System
MacOS	Macintosh Operating System
OHA	Open Handset Alliance
AOSP	Android Open-Source Project
AT&T	American Telephone and Telegraph Company
GNU	GNU's Not Unix!
GPL	General Public License
BSD	Berkeley Software Distribution
HP-UX	Hewlett Packard Unix
MIT	Massachusetts Institute of Technology
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
App	Application software
etc	Et Cetera

CHAPTER 1

INTRODUCTION

1.1 An Introduction to Unix

To have a better understanding of what Linux is, one needs to learn about Unix, an operating system created by the telecommunication company American Telephone and Telegraph Company (AT&T) in the year 1969^[1] for their Bell System computers that were used for the telecommunication industry.^[2]

```
$ simh-pdp11 boot.ini

PDP-11 simulator V3.10-0
Disabling XQ
#0=unix

UNIX/3.0.1: unixhptm
real mem = 262144 bytes
avail mem = 195776 bytes
unix
single-user
# init 2
# process accounting started
errdemon started
cron started
multi-user
type ctrl-d

login: root
UNIX Release 3.0
# uname -a
unix unix 3.0.1 hptm
# █
```

Screenshot from a Unix terminal. Image by Rwoodsmall, CC BY-SA 4.0

Along with Unix, another well-known software that AT&T had developed at the time was the programming language C,^[3] which the company distributed to government and academic institutions. This allowed both Unix and C to be used on both mainframes and personal computers.^[4] By 1990, almost 90% of all supercomputers in the world were

using Unix as their operating system.^[5] Apple's operating system, MacOS, is a derivative of Unix.^[6]

There were several reasons behind the rise of Unix. Some of them are listed below.

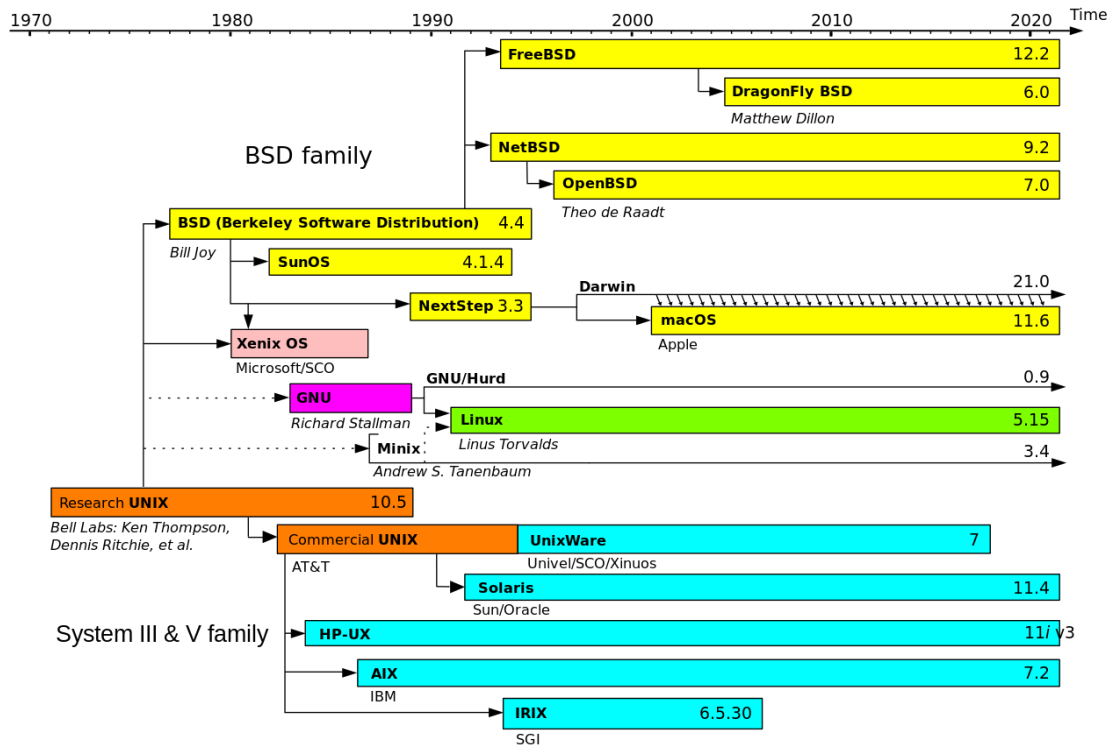
1. **Marketing** - Unix was marketed by The Open Group (then known as X/Open Company) in October 1993,^[7] and then sold by the Santa Cruz Operation in 1995.^[8] The operating system was provided at a low cost for educational use,^[9] and was capable of running on low-end computers,^[10] thus it could be marketed towards anyone with a computer.
2. **Long-term support** - Although originally written in assembly language, Unix was then rewritten in C, which was a high-level programming language.^[11] This helped it adapt to new hardware while at the same time also existing for old machines that were yet to be replaced.
3. **More features** - Unlike the present-day operating systems that have a well-defined structure, this was not the case back when Unix began to gain popularity. It was the first major operating system that had allocated disk drives,^[12] a command for printing^[13] and an unlimited number of levels to which the hierarchy of the folders could be created.^[14] Such features attracted more companies to use Unix as an operating system to create their operating systems, something that will be further discussed in the next section.
4. **C** - While Unix itself was popular enough, as mentioned earlier, it was sold along with C, a high-level programming language that was in high demand at the time. It was to become the source code of several software we use today, such as Microsoft Windows,^[15] Google Chrome,^[16] Mozilla Firefox^[17] and most importantly, another programming language that is widely considered its successor, C++. Being an operating system that was sold as a tie-in with a programming language so much in demand, Unix was even sold to clients who were only interested in learning and using C.

1.2 Unix-like

Witnessing the potential and rise of Unix, AT&T decided to allow the sub-licensing of their operating system in 1979.^[18] Consequently, this gave rise to several Unix-based operating systems, many of them having proprietary licences. A proprietary licence is when the software has a copyright, and anyone who has to use the software must pay for it. These operating systems are called Unix-like operating systems or by other names such as Un*x, *nix and *N?X.

American software developers Eric S Raymond and Rob Landley classified Unix-like operating systems into three categories.^[19]

- 1. Genetic Unix** - Operating systems that are direct descendants of Unix. These operating systems run on a source code tracing back to Unix. Eg. BSD, HP-UX, SunOS, NextStep etc.
- 2. Trademark/Branded Unix** - Commercially sold operating systems that may not have their source code tracing back to Unix, but they are qualified to use the Unix name. Eg. MacOS, Linux derivatives such as EulerOS and Inspur K-UX, IBM z/OS, System V etc.
- 3. Functional Unix** - Operating systems that were derivatives of Unix and even use the same consistency as the original Unix operating system, but their source codes are no longer tracing back to Unix. These operating systems are not sold commercially but rather are FOSS. Eg. Linux, Minix, Meego, KaiOS etc.

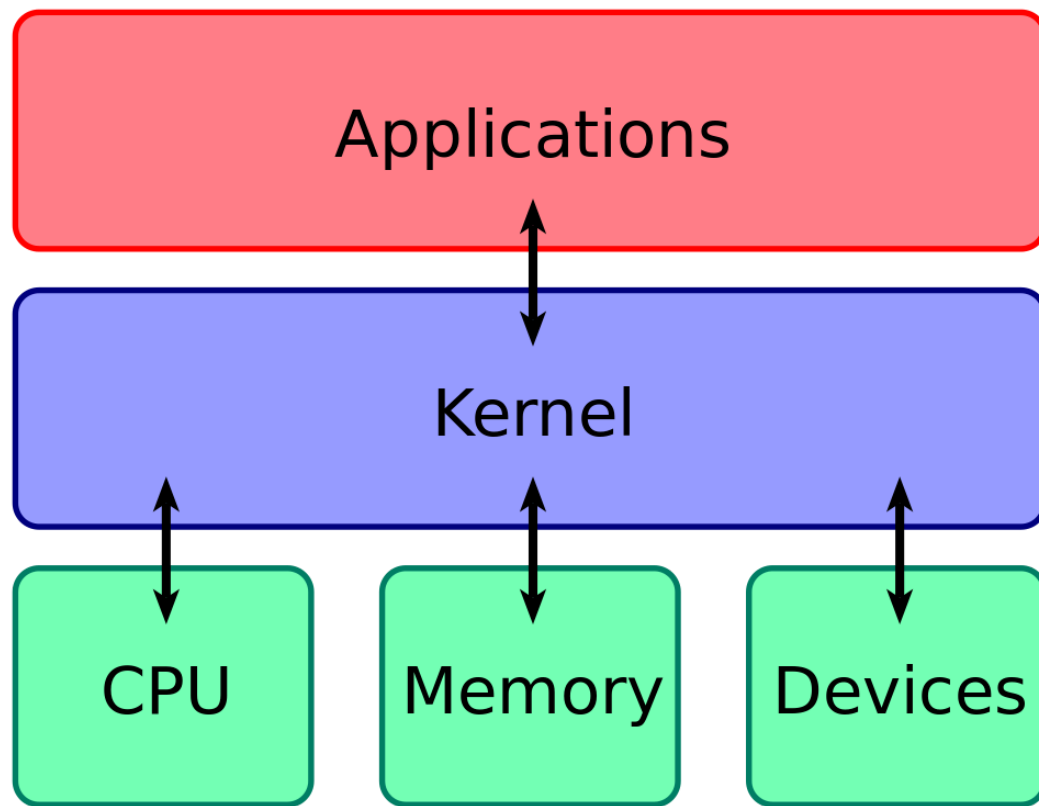


All the Unix-like operating systems. Credits: Guillem, Wereon, Hotmocha, Christoph S. Public domain.

1.3 GNU and the Rise of Linux

GNU is a recursive acronym for *GNU's Not Unix!*.^[20] It was created in 1983 by Richard Stallman to create a Unix-like system in which all software shall be free.^[21] Software that comes under GNU receives a GNU General Public License (GPL).^[22] This software guarantees their users that they can run, share, modify and study the software without any legal issues.^[23] This agreement is known as a copyleft, as it is the opposite of a copyright. At the time of writing this thesis, there are currently 385 software in the GNU library.^[24] All this software helped create a family of operating systems, which came to be known as Linux.^[25]

1.4 Kernel



An oversimplified definition of a kernel. Bobbo, CC BY-SA 3.0

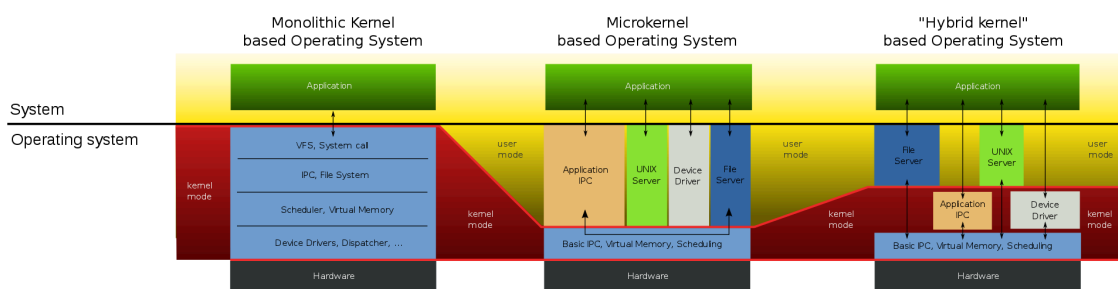
To better understand what Linux is, one needs to have an understanding of what the Linux kernel is and, subsequently, what a kernel is. A kernel is a part of the operating system that links the CPU, memory and devices to the application running on the computer. Consequently, everything that happens in a computer is controlled by the kernel. This is why it is said to be the core of the computer's operating system.^[26]

There are five different types of kernels.

1. **Monolithic kernels** are where the entire operating system is working within the kernel.^[27]
2. **Microkernels**, also spelt **u-kernels** or **μ-kernels** are where a bare minimum amount of space is allocated to the operating system. While this is done to

ensure that the operating system does not consume too much space in the kernel, it also limits the performance of the operating system due to size constraints.^[28]

3. **Nanokernels** are those microkernels in which the length of the code written for the kernel is very small.^[29] The term microkernel was created by Jonathan Shapiro in the paper The KeyKOS NanoKernel architecture.^[30]
4. **Exokernels** are where the operating system can be used with the help of an external memory device. Just like microkernels, exokernels also use low-level specifications for operating systems to ensure they do not consume too much space.^[31] It was developed by the MIT Parallel and Distributed Operating Systems group.^[32]
5. **Hybrid kernels** combine the features from both the monolith kernel and the microkernel.^[33] While it makes use of the aspects of both types of kernel, it is more closely linked to monolith kernels, with the founder of Linux, Linus Torvalds even stating that this name is a mere marketing gimmick.^[34]



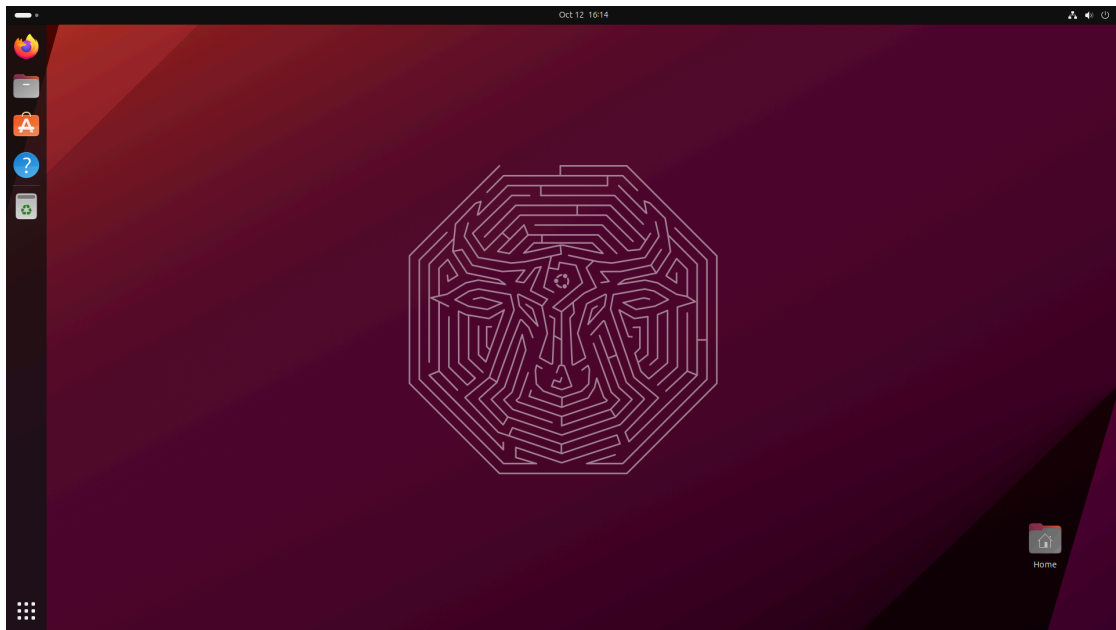
The difference between a monolithic kernel, a microkernel and a hybrid kernel's structure. Golftheman, Public domain.

With the explanation of Unix, GNU GPL and kernels covered, it can be stated that Linux has a monolith Unix-like kernel.^[35] The kernel was written by Linus Torvalds on 17 September 1991, after which it received the GNU GPL and was subsequently available as a free and open-source file.^[36]

1.5 An Introduction to Linux

Linux is a free and open-source (FOSS) family of operating systems.^[37] Open-source software means that one can not only look into the source code but also modify it to create their operating system and release it either as a FOSS itself but also as a

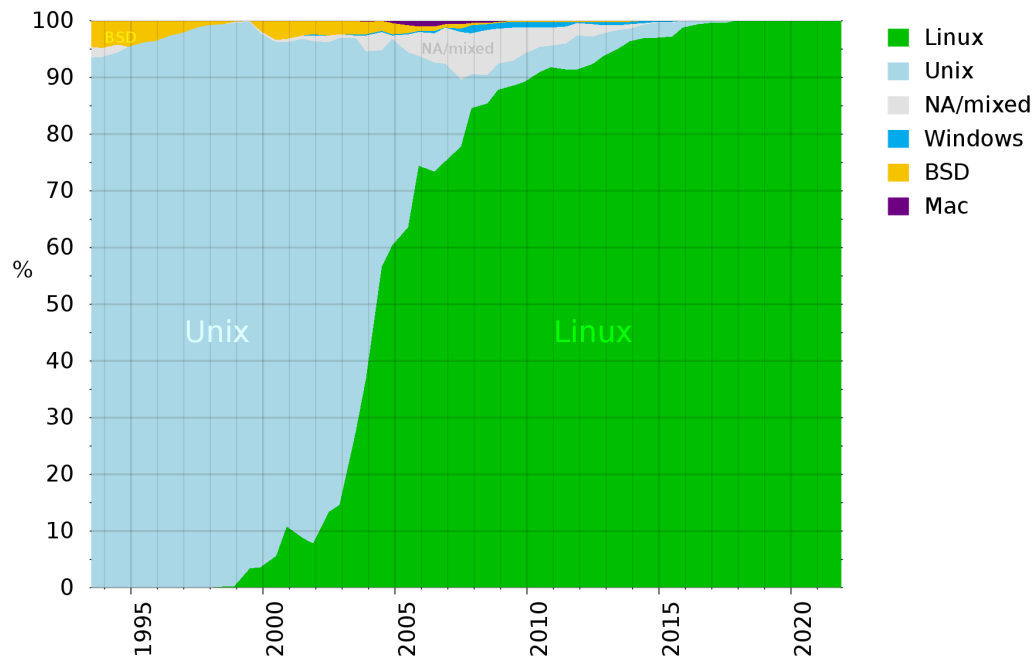
commercial product.^[38] This term must be remembered, as it will be an important part of the results and discussion of the thesis.



The desktop screen of Ubuntu, the most well-known Linux distro. Image from Ubuntu's website.

One thing to note here is that Linux is called a family of operating systems and not an operating system itself. This is because, due to its FOSS nature, many software developers were able to modify the source code and release their version of a Linux-based operating system. These operating systems are known as **Linux distributions** or **distros**.^[39] Furthermore, software developers have modified the source codes of these Linux distros to further make their operating systems called forks.^[40] For example, Debian is a distro that was developed from Linux. Later on in 2004, the source for Debian was used to develop Ubuntu, which is to this day the most commonly used distro.^[41] After this, several Ubuntu-based distros were created including Kubuntu, Linux Mint,^[42] Pop!_OS^[43] and Rock Cluster Distribution (used in supercomputers rather than personal computers).

As mentioned previously, Unix was used in almost all supercomputers in 1995. From the mid-2000s, however, Linux overtook Unix to become the most commonly used operating system and since 2018, all the supercomputers around the world have only been using Linux as their operating system.^[44]



The decline of Unix and the rise of Linux. Benedikt.seidl, public domain

Other than distros, another concept that must be talked about here is **flavour**. Unlike a distro where the entire operating system is modified to create an entirely new version, in a flavour, only certain tweaks are made to target the operating system for a specific purpose.^[45] For example, Edubuntu is a Ubuntu flavour for educational purposes, Ubuntu Server is used for servers, Ubuntu Studio is used by graphic designers, Ubuntu Kylin is a Chinese adaptation of the operating system etc.



All the Ubuntu flavours (L-R, T-B: Ubuntu Budgie, Kubuntu, Lubuntu, Ubuntu Mate, Edubuntu, Ubuntu Kylin, Ubuntu Studio, Xubuntu, Ubuntu Unity and Ubuntu Cinnamon). Photo taken from Ubuntu's official website.

1.6 Android

Now that everything about Linux distributions and flavours has been discussed, the topic at hand can be better understood.

Android Open Source Project (AOSP), better known by its shortened name Android,^[46] is a Linux distribution targeted primarily for handheld touchscreen devices such as mobile phones and tablets and, in some cases, laptops. It was publicly unveiled on 5 November 2007 by a team of developers at Google called Open Handset Alliance (OHA) as their flagship project.^[47] The OHA members are contractually forbidden from creating a fork of Android.^[48]



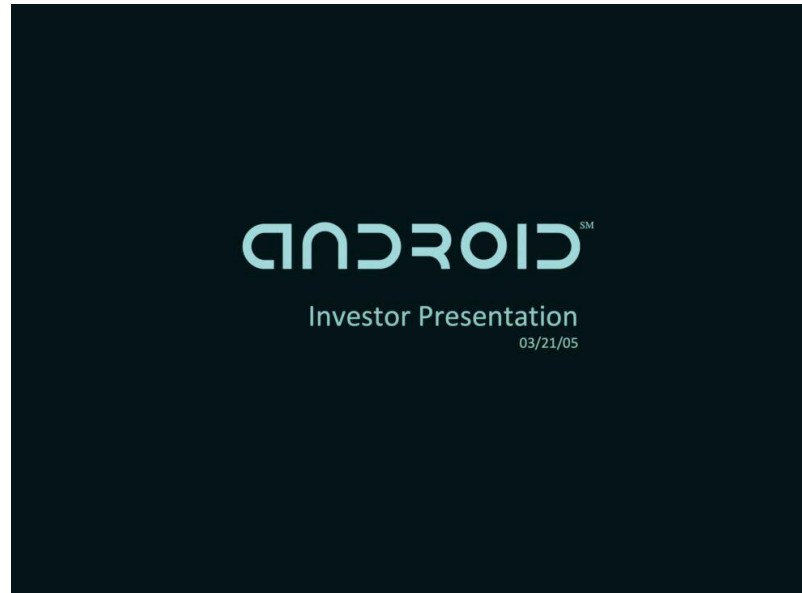
Android's logo since 2023. Logopedia, public domain.

Although AOSP is free for anyone to use under the Apache 2.0 licence, the name and trademark for Android are reserved for Google.^[49] Therefore, any company that uses Android as its operating system is contractually bound to pre-install other Google services along with its operating system.^[50] Some of the most notable services by Google in this proprietary category include the Google App, Google Chrome, Gmail, Play Store, Google Play Games, Google Files, Google Messages, Google Contacts, Google Meet etc.

Android Inc. was founded by Andy Rubin, Rich Miner, Nick Sears and Chris White in October 2003 in Palo Alto, California. The initial idea was to make an operating system for digital cameras that would be connected to personal computers.^[51] In April 2004, however, they concluded that their target audience was not wide enough.^[52] Consequently, the purpose of the operating system was changed to handheld devices that would rival the likes of Symbian and Windows Mobile.^[53]

With an initial struggle in finding an investor for the operating system, Android Inc. was approached by the co-founders of Google, Larry Page and Sergey Brin, who purchased

the company for \$5 crore after a brief conversation with Andy Rubin.^[54] After the company was sold, the team at Android Inc. went on to be the key members of the OHA.^[55]

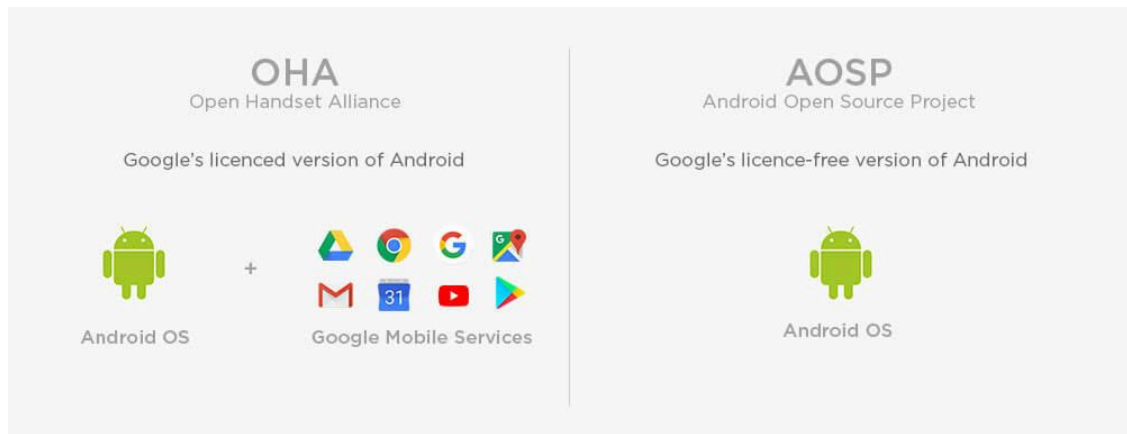


Left: A demo version of what the home screen of Android was supposed to look like at the time of being pitched. Several features did not make it in the final version. The operating system was designed by Andy McFadden and programmed by Brian Swetland and Chris White. Chet Haase, Ars Technica.^[56]

Right: The first slide of the presentation that was shown by the Android Inc. team when they were pitching the operating system to Google. The typeface used to write Android here continued to remain the operating system's logo until 2014. Chet Haase, Ars Technica.

1.7 Android as an Open-Source

As discussed in the previous heading, although AOSP is a FOSS, Android is a proprietary software, that is, it has a copyright and a trademark. It does not allow the OHA to develop forks of Android, and any company that wishes to use the Android operating system for its devices is contractually bound to also include other Google services in their device. This alone makes Android no longer a FOSS, but rather just open source. This, however, comes with several caveats.



A summary of the difference between Android and AOSP. Famoco.

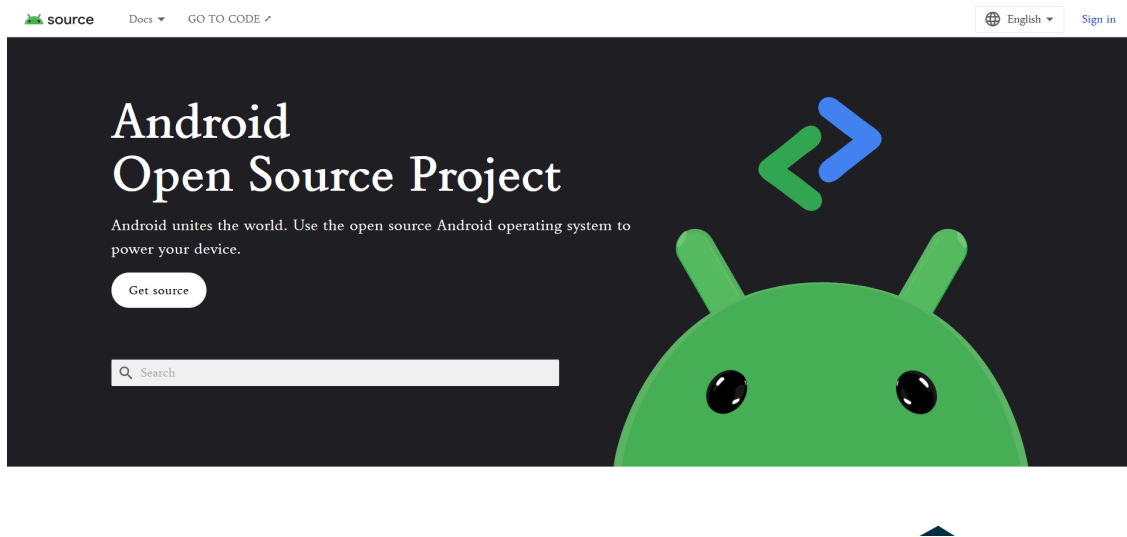
Not only does Google own a lot of apps on its licenced version of Android, it also has the copyright on several features of the operating system. This even includes the push notifications we receive on our Android phones. Furthermore, AOSP also does not have a Play Store, so the end user is expected to be dependent on the developer of the operating system to add an alternative to it. The good news is, however, that the Play Store can be added to AOSP.^[57]

There are several examples of operating systems that have used AOSP to create a unique interface unique to the smartphone. Some of these include MiOS, the OS in Redmi phones,^[58] OxygenOS, the OS in OnePlus phones,^[59] Funtouch OS, the OS in Vivo phones^[60] etc.

CHAPTER 2

LITERATURE REVIEW

2.1 AOSP documentation^[61]



The landing page of the AOSP website, on which the documentation can be found

On the official website of AOSP, the links for two web pages can be seen on the landing page: The documentation and the source code. The thesis here is more focused on the documentation for now, as the final prototype will be covered in a later chapter. The webpage has been divided into several categories, of which the pages that explain the structure of AOSP shall be discussed one by one, while the instructions for creating a fork will be ignored.

Overview^[62]

Unlike all other categories in the documentation, it does not have any subcategories. It briefly touches on all the points that shall be explained in detail in the other categories using images accompanied by one sentence each. It begins by explaining the reason why a developer should choose AOSP over any other software development kit (SDK). The links to each category are linked on the page, giving a tagline and a brief description of each of them.

Overall, there is not much in this category, and thus does not have anything concrete to be discussed about.

AOSP overview^[63]

The page begins by giving the reader a brief description of what Android is and then begins to explain what AOSP is about. However, it does not make it clear that the Android that most people use is not the same Android that is being offered here. Just like in the case of the overview category, it follows with a couple of reasons on why any operating system developer should use AOSP to create their fork rather than any other FOSS distro.

Afterwards, it explains the terms that are to be frequently used throughout the documentation. These terms include Android app developer, Android debug bridge (ADB), Android compatible device, Approver, Compatibility Definition Document (CDD), Compatibility Test Suite (CTS) etc. Most of the terms in this list are relatively self-explanatory, so they were not needed. These only ended up making the list unnecessarily long.

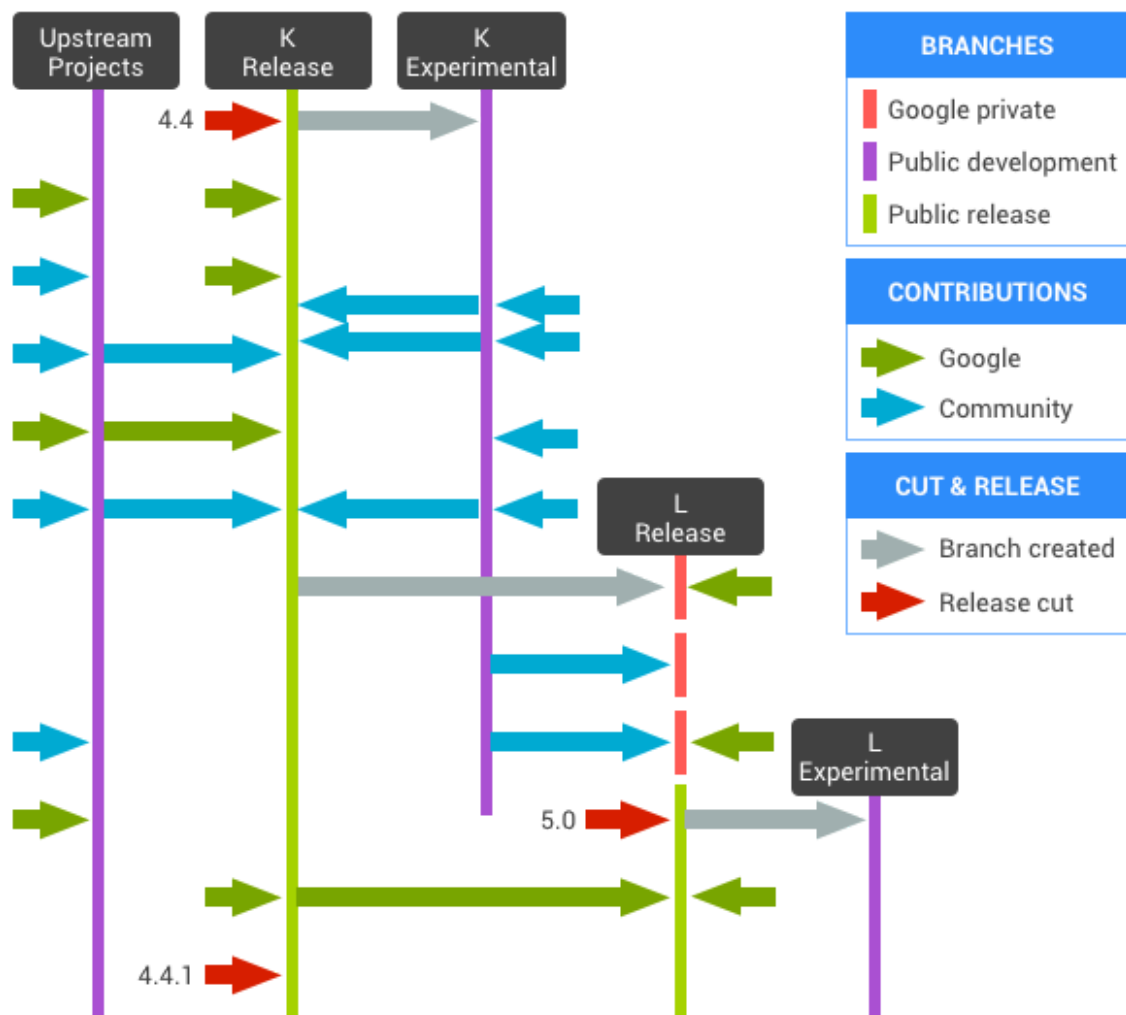
The subcategory ends with explaining the philosophy behind Android OHA and giving a brief history of Google's contribution to the Android project. This entire section was somewhat irrelevant because, as aforementioned, AOSP is separate from OHA. The bias makes it clear that Google created this documentation and is therefore written in its favour.

Android software management^[64]

The code management and structure are explained with a paragraph and a diagram. It gives a brief understanding of the versions of AOSP that were released. Since the versions that are yet to be released are not mentioned, we can safely assume that it is indeed only talking about AOSP and not OHA, as the responsibility of releasing an enhanced version is only partly in the hands of Google, as can be seen from the diagram on the next page.

Whenever there is a current release of Android, it is released in the form of a branch. This way, anyone can either create a fork out of the latest release, or from a previous version that either they have been working on, or they feel has better features than the successor. Google often partners with certain smartphone companies that are to showcase how well the new Android version can run.

It also explains how the fork works, what all the technical terms mean and what caveats one might face whenever a new version gets released. The bullet point system used for writing the terms and caveats coupled with the formal language makes the subcategory difficult to read.



All the previous and current releases of AOSP. Image taken from the document.

The document ends with a topic called *Private code lines*, which mentions that certain code lines are kept private by Google and thus shall not be available for developers to use in their forks. The last paragraph says that although most developers may not agree with this policy, it has been done with the *best intentions* in mind. What those intentions are never told here, therefore a good guess has to be that those are source codes that will fundamentally change the operating system, and thus must not be tampered with.

Brand guidelines^[65]

This page is not important due to two reasons. First, it is a brand manual that one must follow to make their fork. Second, which is the more important reason, it has been mentioned at the beginning of the page that it is outdated. The date of removal has been mentioned as 1 April 2024, but it is still here even as of writing this thesis. In any case, the subcategory is unimportant for the thesis and consequently needs no further discussion.

AOSP frequently asked questions (FAQ)^[66]

The subcategory answers some of the most important questions relevant to the thesis, although there is no need to give a detailed explanation of most of these topics as they have been discussed in the previous chapter. However, there was one question that the thesis had only partial knowledge about, and that was better explored here.

As aforementioned, the private code lines are not displayed to the public, which is a concern that has been answered here. When a new Android version is released, as aforementioned, Google partners with smartphone companies to launch their devices with the new version. Therefore, if another developer were to outperform Google and make a better operating system, the smartphone company's customers would not find any appeal behind the newly launched product. From a business point of view, this sounds like a reasonable explanation.

After this category, the documentation proceeds with instructions on how to create a fork. However, as aforementioned, it is not a part of the required literature review and will thus not be covered.

2.2 Apache License Version 2.0^[67]

Considering AOSP is a FOSS under the Apache 2.0 licence, one must go through the document to better understand what the terms of usage for the operating system fork are. As previously discussed, there are many copyleft licences, each having a different set of conditions. The purpose of this literature review is to go through the original licence document and discover what Google must have had in mind while deciding to choose Apache License Version 2.0 over any other alternative.



The logo of Apache License Version 2.0

The Apache License Version 2.0, as mentioned in the original document, was created in January 2004. At the beginning of the document, the definitions of the words that are to be used frequently are mentioned. These words are as follows.

Term	Definition
Licence	The terms and conditions mentioned in the document for using the software.
Licensor	The person who holds the licence. An interesting detail to note

	here is that the word used for the owning right here is <i>copyright</i> instead of <i>copyleft</i> , even though the latter has been in usage since 1985, whereas this document was written in 2004.
Legal Entity	Those who have the power to act against anyone who breaks the rules of the licence.
You	The person who has been granted permission by the licence to use the software. To distinguish it from the second person pronoun, the word has been used with a capital Y, which shall also be followed in this thesis.
Source	The original software from which the changes have been made.
Object	The software that is developed from the source.
Work	The licensed software, either in the form of source or object. To differentiate from the verb, Work has been mentioned with a capital W throughout the document, something that will also be done here in the thesis.
Derivative Works	As the name suggests, the software that has been derived from a Work. The licence mentions that the derivative work must not be an edition of the work.
Contribution	Any positive and accepted changes that have been made to a Work or a derivative work. The licence says that for the contribution to be considered, it must be submitted to and accepted by the licensor.
Contributor	The person who contributes. If a licensor contributes, then they shall also be considered a contributor for the sake of clarity in the document.

Right in the beginning, it is made clear that the contributors have given You the complete right to do what You want with the software. For any modification that is done on the user's part, there shall be no financial, regional or legal blockade. However, if You try misusing the contract by suing the contributors over copyright infringement, then the licence shall be cancelled. The line is essentially important because, under normal circumstances, it could have been an exploitable loophole.

Afterwards, the conditions of redistribution are mentioned in numbered bullet points, as briefly mentioned:

1. Derivative works from the user's original work also get to use this licence
2. The user must mention which files in the source have been modified
3. Only the modified parts can have any copyright and not the whole source
4. Any additional *notes* must be easy to read and do not modify the licence in any manner

These bullet points help establish the fact that one can use AOSP forks and create modifications on the parts that are not under copyright, without fearing any legal consequences for having *stolen* the work of another individual or company.

The licence further states that the contributor can submit their contributions to the licensor, but cannot add any additional licence agreements to it. The contributor is furthermore prohibited from using the trademarks by the licensor unless it is a part of the agreement. Any theft or infringement that happens to the contributor is their responsibility, nor is the licensor responsible for any damage to the user.

The final condition mentioned in the document states that the contributor is allowed to ask for donations or other forms of monetary support, keeping in mind that these are their actions and in no way is the licensor responsible for any unforeseen consequences for such actions of the contributor. The webpage finally ends with a tutorial on how to apply to an Apache License Version 2.0, which the thesis can ignore due to it not being a part of the problem statement.

As is evident from the licence, it sets a clear border between the fork on which the developer is working from the work of the original developer. Keeping the source code unprotected from any legal obstacles, ensures that no one can lay claim to the original files just because they own them via their contributions. The only way any developer can have a copyright or trademark on any work is if they have themselves made it. By submitting the contributions to the licensor, both parties will be aware of what changes the contributor wishes to make and any undesirable consequences can be eliminated before the final product makes it to the market.

CHAPTER 3

PROBLEM IDENTIFICATION

3.1 Privacy

Privacy has become an important issue in the contemporary digital environment. On 22 November 2021, the Joint Parliamentary Committee of India introduced the Personal Data Protection bill, after the Supreme Court ruled privacy as a fundamental right in 2017. This bill, however, was withdrawn by the government in April 2022.^[68]

Proprietary apps collect user data to tailor to the user's needs and offer better engagement. This, however, comes at the expense of the lack of consent provided by the user, which is used by companies for unethical means. Some examples of this include tracking the device location at all times, giving news specifically biased to suit our narrative, getting ads based on what we searched etc. Not only do these companies collect, but also sell our data to third-party sources, who then sell the data to other sources, thereby putting our data in public.^[69]

Since this is done so that the user uses the app more and the company that owns the app profits more, using FOSS is a better alternative as they do not benefit from collecting data. Furthermore, if a proprietary app can be deleted, it no longer poses a threat to privacy. This is one of the major reasons why Google apps cannot be deleted from the OHA version of Android, as Google will not want the user to remove the apps that help profit the company. On the other hand, smartphones that use AOSP allow users to delete Google apps but often do not allow the users to delete their apps.^[70]

The only way to ensure the user's privacy is under no threat is to specifically provide FOSS alternatives at the time of the first launch. Afterwards, if the user downloads apps and gives away their data, it shall be done with their consent. In 2014, the government of India adopted a policy to adopt FOSS for this very reason.^[71]

The problem statement to the problem identification shall be: *How might we create an operating system that tackles the issue of privacy?*

3.2 Storage

Most smartphones come with a storage of 32GB or more. However, not all this space is available for the user. System files, the files necessary to run the operating system of the smartphone, can take as much as 27GB.^[72] And while it is understandable why these files are required, other apps are not system apps and also cannot be uninstalled. As mentioned under the previous heading, OHA does not allow the user to uninstall Google apps, even if the user does not use them. Some of these apps take up a lot of space, not only from their app size but also from user data and cache memory.

As aforementioned, if a company uses AOSP to create their fork, it adds their apps that cannot be uninstalled.^[73] As a result, there shall always be a sizeable amount of memory allocated to running apps that the users do not want. This discourages the user from trying other apps that also serve the same purpose, as this would only add additional burden on the smartphone. For example, if Google Search is a system app, no one will download DuckDuckGo, a FOSS search engine that does not take up as much space or collect user data.^[74] The user might instead save the space for photos and videos. The photos and videos were previously backed up on the cloud by Google Photos, but on 1 June 2021, it announced a 15GB limit on the storage space, after which the user will have to pay for more storage.^[75] Since Google Photos is an app that cannot be uninstalled, users are discouraged from using other alternatives.

The solution to the problem shall be to allow the users to delete as many apps from the operating system as realistically possible, including the GUI of the operating system itself.^[76] This is possible when the operating system is not for profit, thus not having any bias towards an app that can benefit it in any economic manner.

The problem statement to the problem identification shall be: *How might we create an operating system that solves the problem of low storage on the user's smartphone?*

3.3 Customisation

Customisation allows the phone to be personalised in the way the user finds it appealing by tailoring their visual or technical outlook to the liking of the user. It also helps smartphone companies make their product look more distinguishable in the market, which is the reason why so many companies use AOSP instead of OHA, as it allows them to make their OS look different right off the bat.^[77] Not only does using AOSP instead of OHA allow them to make their layout look different, but also gives them the option to add more features to customise the smartphone. For example, Redmi phones allow the user to change the font.^[78] Similarly, OnePlus phones allow the user to use a more enhanced dark theme compared to OHA smartphones.^[79]

There is, however, still a lot of scope for an OS to allow the user to customise their look. As aforementioned, for example, there is yet to be a smartphone that allows the user to delete the GUI completely. The sizes of the app icons can also not be individually customised, as it shall be hard to program the icons to not overlap in such circumstances, which is not feasible. However, if we negate any requirement to avoid overlapping and leave the choice completely in the hands of the users, there shall be no more need to do any such thing.

The thesis proposes to introduce customisation in the operating system of the smartphone on such a scale that, if fully customised, no two smartphones using the same OS shall look the same. This includes letting the user change the rotation of the app icons and widgets, the colour of the notification bar, changing the app icon sizes individually and, as said previously, even allowing the user to completely delete the GUI.

The problem statement to this problem identification shall be: *How might we create an operating system that allows the user to completely modify the visual and technical aspects of their smartphones?*

Combining all three problem identifications, we get the following problem statement: *How might we create an operating system that solves the problems of privacy, storage and customisation?*

CHAPTER 4

USER RESEARCH AND METHODOLOGY

4.1 Secondary Research

Keeping in mind someone who does not know what FOSS is and how it works, the secondary research contained data about everything from how FOSS came into existence to how it is relevant to the topic of the thesis. The process of understanding the problem statement of the thesis created a snowball effect, resulting in the reader having to learn about several topics sparsely relevant to the final solution. However, the procedure was not in vain, as it was able to explain several topics that would have otherwise not been fully understood.

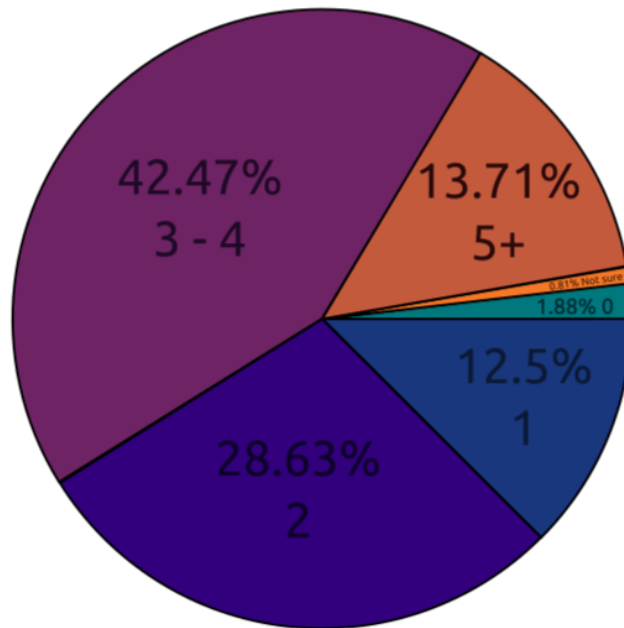
Terminologies such as a distro or Unix-like, which are normally never referenced to in a conversation regarding AOSP were discussed in detail here, which not only made it easier to explain the final solution but also helped the reader understand that these terms are still in use, even if they are not commonly referred to in a conversation.

4.2 Primary Research

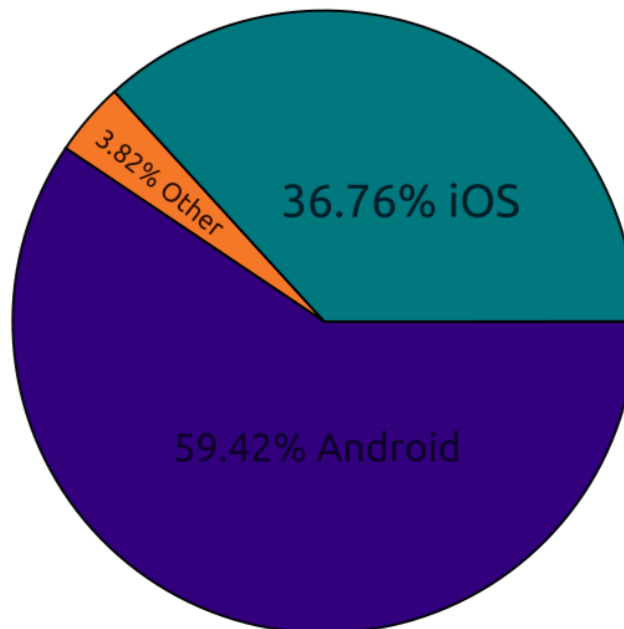
Now that everything regarding the secondary research has been discussed, we can move on to the primary research which, unlike secondary research, has not been mentioned until now. For this step, surveys and interviews were conducted in forums and websites where the user base for Android users, Linux distro users and enthusiasts of GNU software are abundant. As these were the target audience for the problem statement, it was easier to find a considerably large sample size for surveys and interviews.

The sample size of the surveys far exceeded the initial expectation, with over 6,800 participants. The questions in the survey were related to topics regarding Linux and Android, as well as advice on what the final result should look like when the user opens it for the first time.

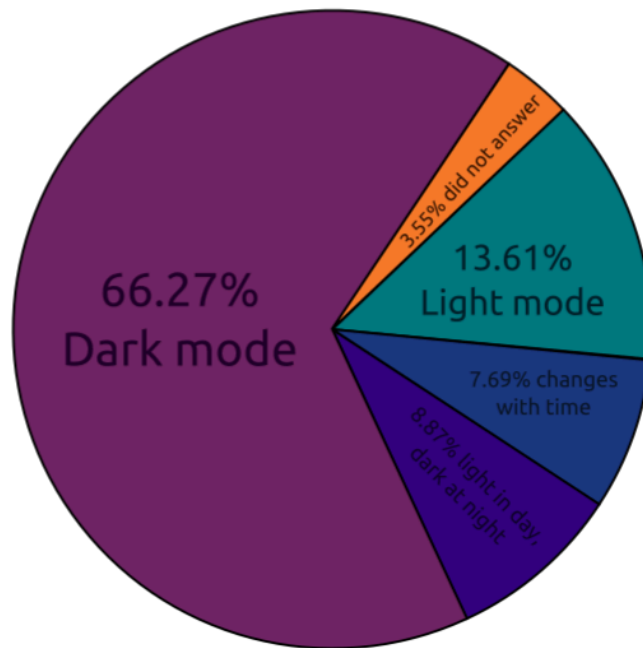
How many phones have you had?



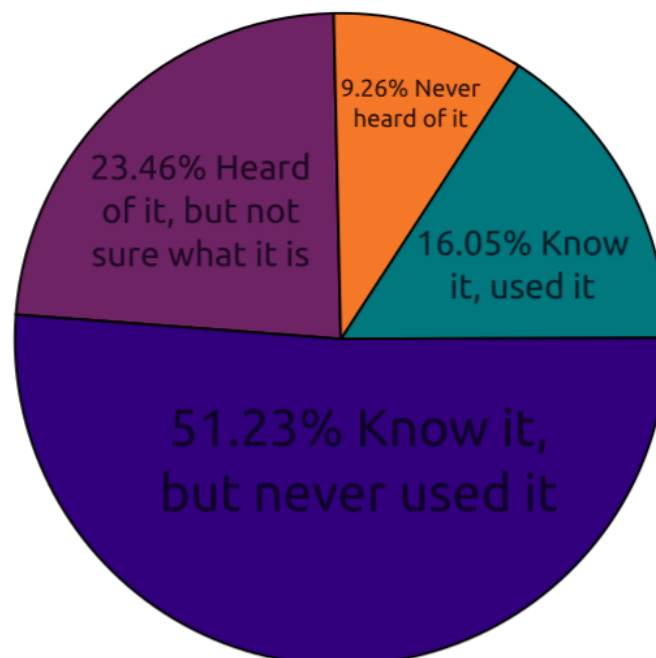
What OS does your phone have?



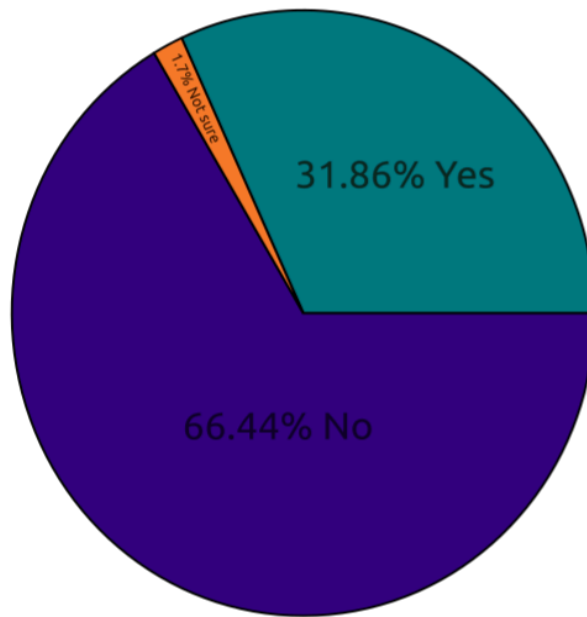
Which theme is your phone set to?



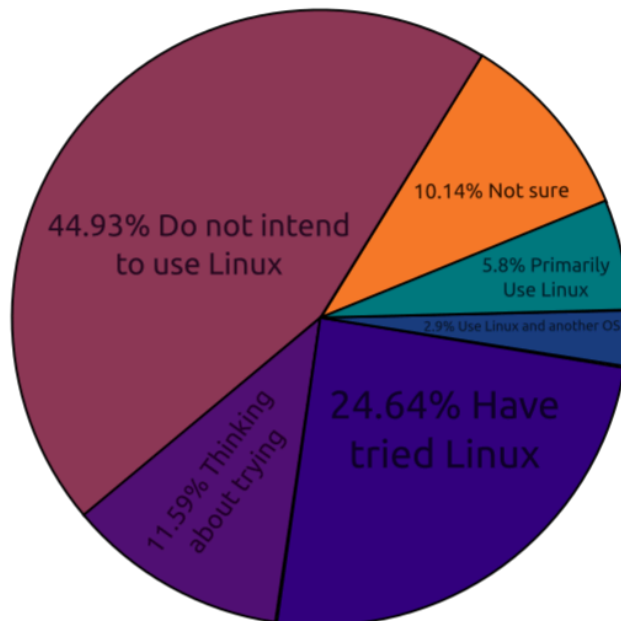
How well do you know about Linux?



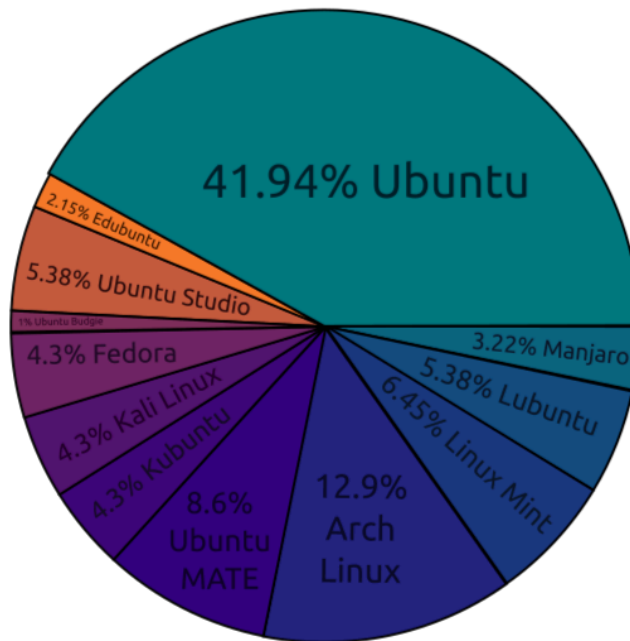
Have you ever used Linux?



Would you use Linux?



Which is the best Linux distro?



Based on the questionnaire, three major conclusions can be drawn.

1. Most people do not know that Android is a distro of Linux. As a result, even though the majority of smartphone users have used Android, they do not feel confident about using Linux.
2. Among the Linux users, Ubuntu is the most commonly used distro. The initial hypothesis was that Linux Mint might be one of, if not the most popular distro, considering how similar it looks to Windows 10. However, as it turns out, it is even less commonly used than some Ubuntu flavours.
3. Even though the majority of smartphone users prefer Android, iOS is not very far behind. Consequently, the final solution must keep the interests of both operating system users in mind when creating the prototype.

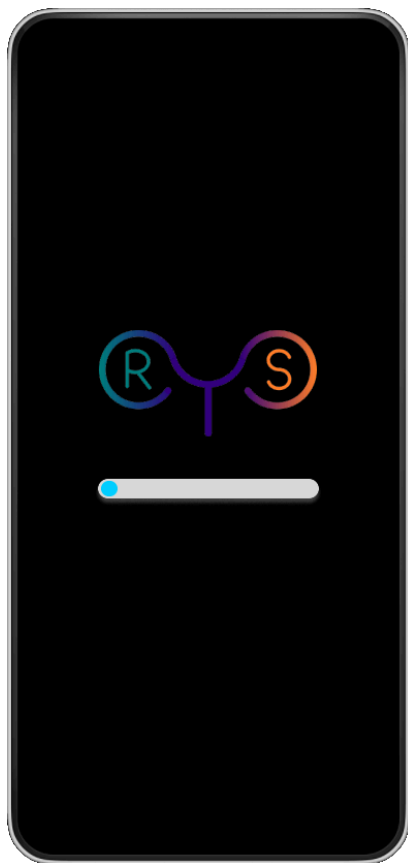
CHAPTER 5

CONCEPT DEVELOPMENT RESULTS

Upon application of the knowledge acquired by the means of research, the concept that was developed turned out to be as follows.

6.1 Introduction to RoyOS

When the operating system first boots up, the logo of RoyOS is shown.



The logo has been made symmetrical with an R and an S on either side, thus spelling R-O-Y-O-S. The R and S are made using bezier curves, and not from an existing font. However, when the operating system first loads, the system font used is Ubuntu. The colours of the logo as well as the original wallpaper are in the gradient from blue to purple to orange. As observed from the survey and secondary research, Ubuntu is the most important Linux distro, therefore aesthetics of RoyOS have been made as close to Ubuntu as possible.

The idea here was to be similar to Ubuntu yet different. The colour combination also suits better for dark mode, as it was evident from the survey that it is the theme in which most users prefer using their devices.

Since the OS is a FOSS, any fork that is created from RoyOS can replace it with its logos. These logos can be of the operating system and/or the smartphone company. Compared to AOSP, the RoyOS boots up much faster due to reasons discussed ahead. After the bootup, the user is introduced to the homepage of the operating system. Upon

first glance, it looks like any other AOSP that the user must have seen before. The familiarity was necessary to make the user adapt to the layout quickly.

6.2 Solving the Problem of Privacy



The first screen of the homepage



The second screen of the homepage

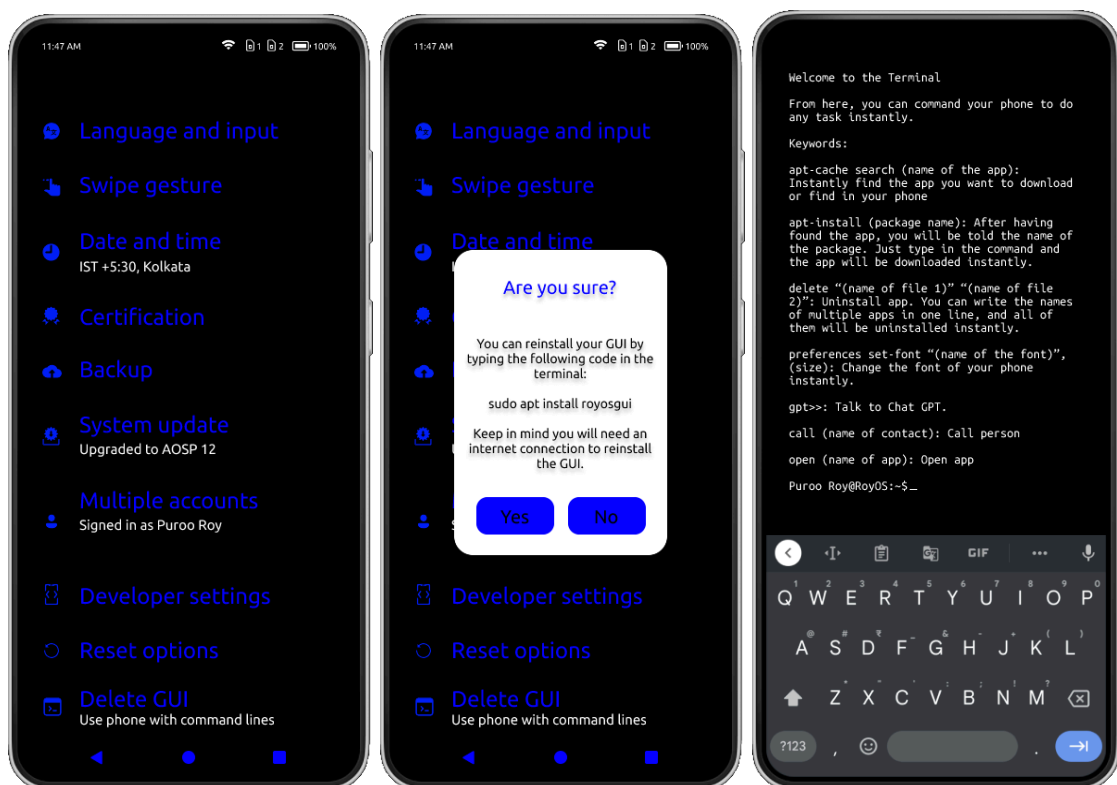
Upon a closer glance, the user might observe that the apps installed on the phone mostly comprise of FOSS, with the only exception being Google Play Store. As aforementioned, the reason behind this is that FOSS does not take data from the user to sell it to third-party sources, because they are not based on any profiting model, thus having nothing to gain from selling the data.

The reason why Google Play Store is still provided is because it is the only place where the user can find all apps that run on Android. However, the user still has the option to uninstall the Play Store and download an alternative, or even download nothing at all. This can be better explained in the next topic.

6.3 Solving the Problem of Storage

Every app on the device can be uninstalled. As all the apps are FOSS at the time the user opens the device for the first time, none of them have any contractual power to remain on the device. This allows the user to delete anything they do not require to save space in their phone or to replace the FOSS apps with more preferred alternatives.

Furthermore, RoyOS also allows the user to delete the GUI of the operating system. Doing so will show a terminal, from where the user can input command lines to make the phone work. The instructions to use commonly used commands are written beforehand.

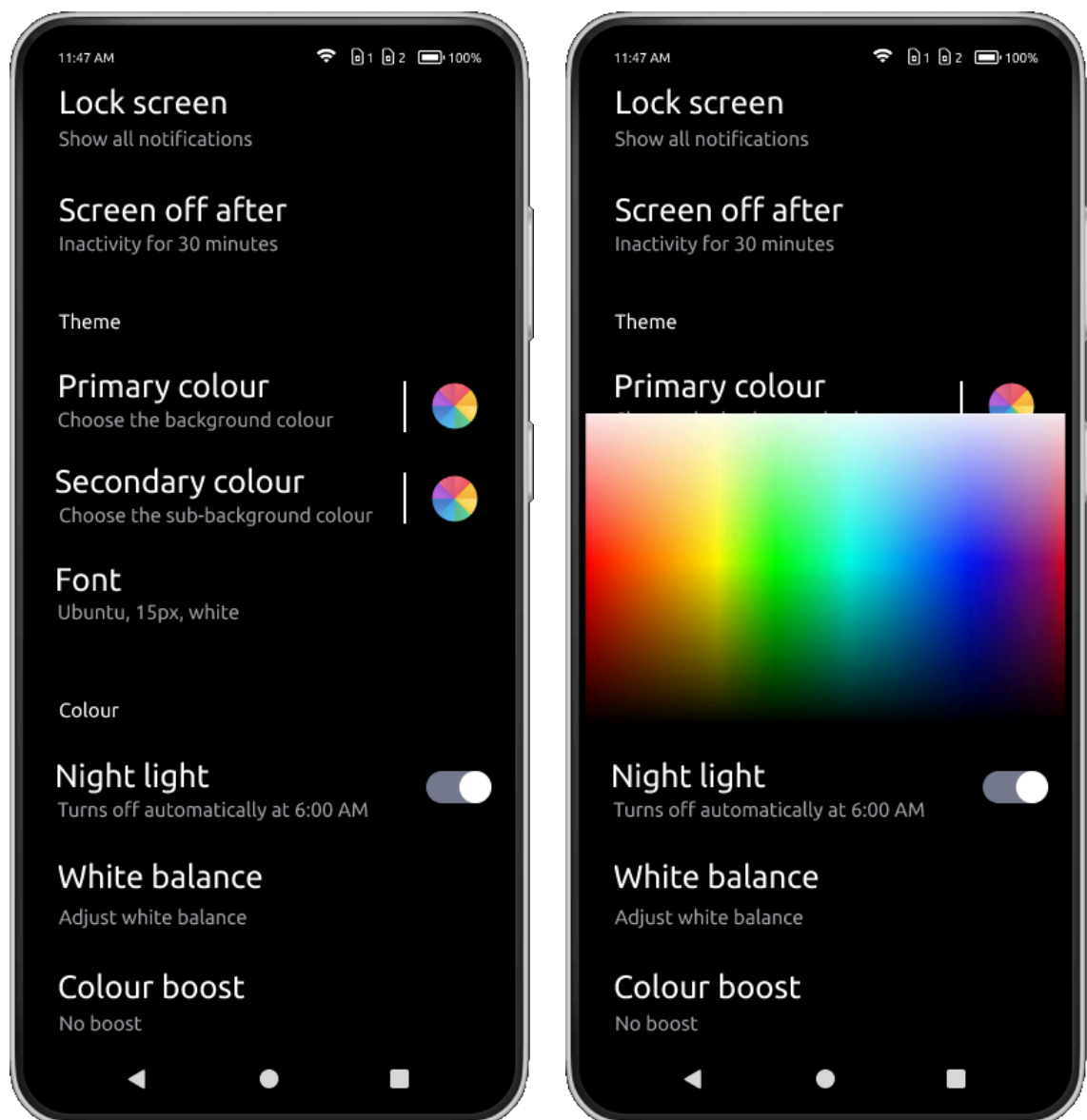


A step-by-step tutorial on how to delete the GUI of the OS. (r) After deleting the GUI.

When the alert to confirm the deletion of the GUI comes up, it also mentions how to bring it back. Being similar to the command line for installing an app, it will help the user easily remember what has to be typed, and also help remember the command line to install an app in the device without using the Play Store.

6.4 Solving the Problem of Customisation

Customisation is the ultimate selling point for the operating system since it not only gives the user a huge range of customisation options but also gives the software developer working on a fork a better headstart. If the developer were to work on an AOSP fork, they would have had to add the options manually, which would have taken up a lot of time and effort on the programming end.



To change the theme of the GUI, the user has to go to Display in the Settings app and under the Theme category, select the primary and secondary colours. The Primary colour is the colour that the background of the GUI shall be, while the secondary colour is the colour used for the buttons and alert windows.

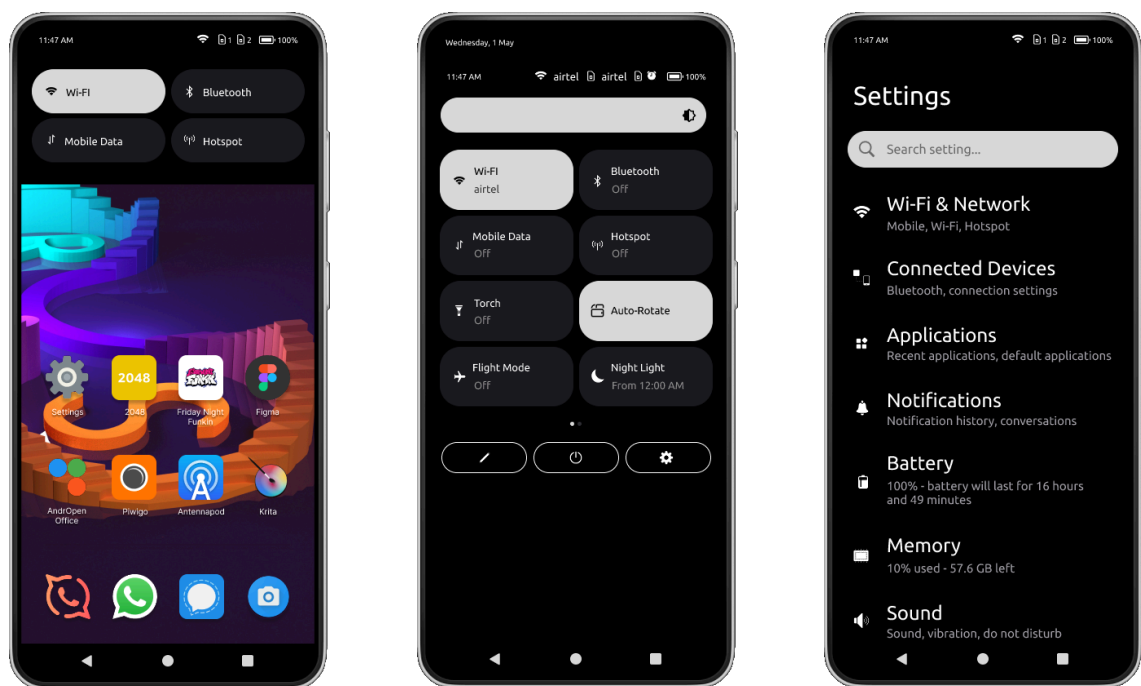
As an example, let us take red as the primary colour and green as the secondary colour. The images below show how the changes look after getting applied. This is a further leap from what the AOSP offers, where the user can only choose between a light and dark theme.



Furthermore, the font of the phone can also be changed. By default, the AOSP gives the option to change the font size between four options under the Accessibilities setting. As

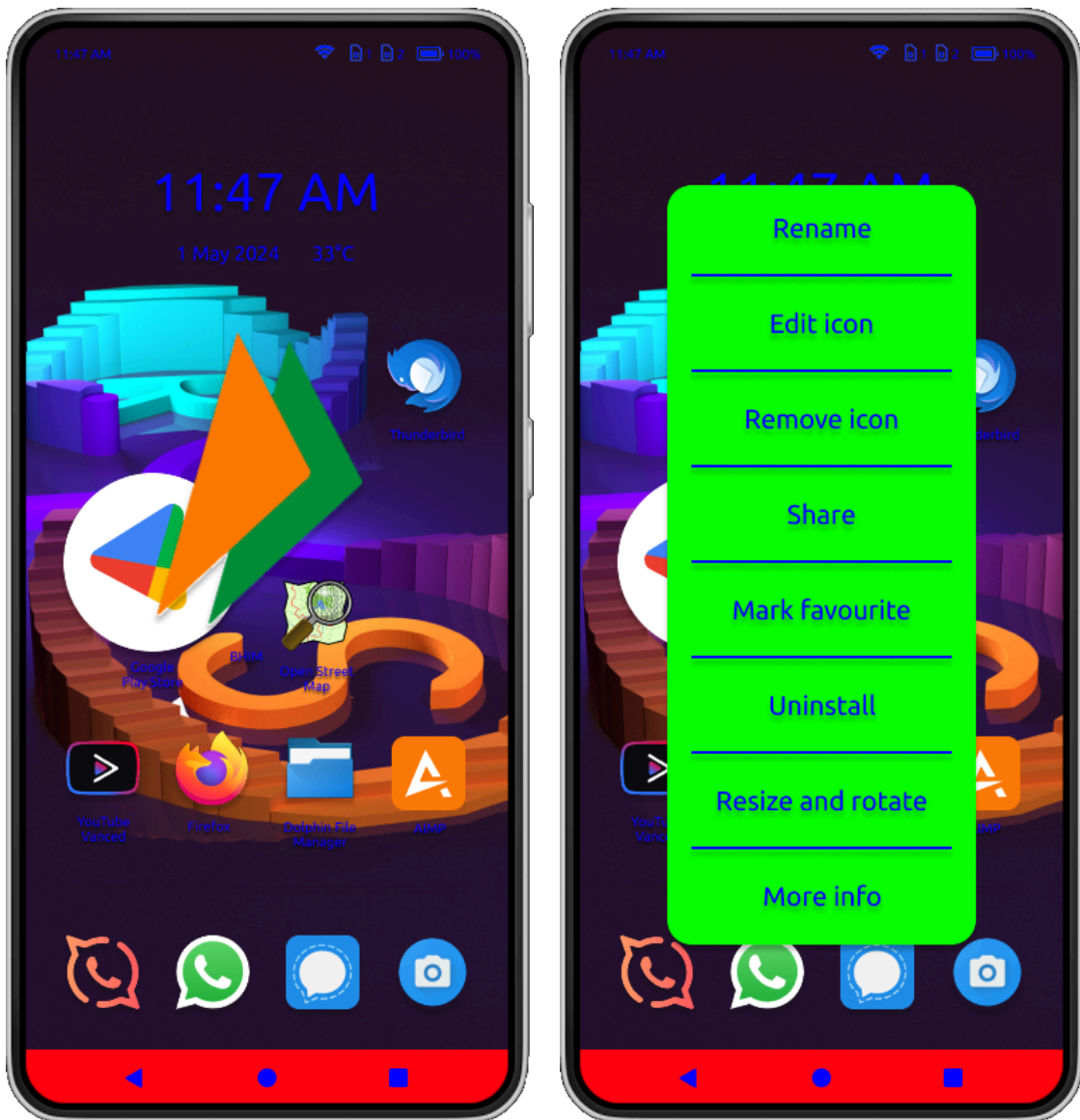
the name suggests, this setting is better suited for helping the user read the text better, not to stylise it. The font colour is the opposite of what the theme has been set to: The text colour is white if the GUI is set to dark theme, and it is black if the GUI is set to light theme. Here, the user can change the font typeface, colour and size to their preference.

Following the previous example, in this case, we changed the primary font colour to blue and the secondary font colour to yellow. The primary font colour shall be used by the main text, while the secondary font colour shall be used for the subtitles, which are written under the main text. Combining previous changes, the differences can be



observed.

Apart from changing the theme and font, RoyOS also allows the user to set the size and rotation of each app individually.



Left: The icons on the home screen resized. Right: The option to resize and rotate an app appears when the user long-presses the app.

CHAPTER 6

CONCLUSION

With the help of the newly developed fork, which has been made available to the public in the form of a FOSS, new developers can directly access an OS that is not only highly customisable but also has no proprietary app preinstalled on it. No longer do they have to create a fork from a parent operating system that has barebone customisation and is filled with unnecessary system software.

This, thereby, can end the monopoly that Google has held over the open-source Linux distribution operating systems for smartphones, similar to how Linux did to Unix about 30 years ago. As AOSP is free for everyone to modify to any extent, it is also safe to assume the developers of Android expect this to inevitably happen. This is not to say RoyOS will completely outperform AOSP in the smartphone market, as Google can patronise its FOSS distro to any extent to eliminate the competition. In a third scenario, Google may replace the current AOSP with something similar to RoyOS, or acquire RoyOS to replace the AOSP. Either way, any software developer aiming to create a fork will have a better base to start from in future.

The issue of privacy was the simplest to solve, the key being to not have any contractual agreement for pre-installing any app on the OS. Keeping it entirely based on FOSS at the time the user first boots up the operating system has to be a fundamental part of the OS for the problem statement to be solved. Therefore, since the base itself does not allow any proprietary app to exist by default, the companies owning such apps will have to establish contracts individually with each smartphone or operating system company that uses RoyOS as its parent operating system.

Similarly, the issue of storage was solved with the same approach. If no app has signed a contract to permanently exist on the operating system, they can all be removed from it. It is, however, up to the user to remove the apps. Like in the case of privacy, the apps that establish contracts with companies owning forks of RoyOS can undo the solution

here. This, however, is the freedom that the fork has. As discussed in the literature review of Apache License Version 2.0, if a derivative work replaces the pre-existing feature of the source, it owns the copyright of the contribution. If the change here is having eliminated the solutions of privacy and storage, it is the responsibility of the contributor rather than the source, thereby making RoyOS free of any blame for the restoration of the problem.

Finally, the problem of customisation was where the majority of the work had to be done. Being a conceptual solution, the complexity that has to go under the programming was not discussed throughout the thesis. The programming behind the solution includes changing the theme to all the colours shown in the colour palette, changing the font style, colour and size, individually changing the rotation and size of the icons etc.

In conclusion, with the current progress in the world of FOSS and Unix-like distros, it is possible to easily create a smartphone OS that tackles the issue of privacy, storage and customisation. Whether it will be done by an individual, a company or an organisation that overlooks profit and solely focuses on providing such a fork is an entirely different matter.

REFERENCES

- [1]. M. Douglas McIlroy, “A Research UNIX Reader: Annotated Excerpts from the Programmer’s Manual”, Dartmouth College, 1971-1986
- [2]. AT&T, “A Brief History: The Bell System”, The Bell Labs, 6 September 2008
- [3]. D.M. Ritchie, “The development of the C language”, ACM Digital Library, 1 March 1993
- [4]. D. Fiedler, “The Unix Tutorial, Part 3: Unix in the Microcomputer Marketplace”, Byte Magazine Volume 8 Number 10 - Unix, October 1983
- [5]. “Development over Time”, TOP500.org, 1993-2023
- [6]. Apple, Inc, “Kernel and Device Drivers Layer”, Mac Technology Overview, 2004
- [7]. Q. Fennessey, C. Karish, “The name Unix is now the property of X/Open”, Google Groups, 11 October 1993
- [8]. M. King, “Novell Completes Sale of UnixWare Business to the Santa Cruz Corporation”, Micro Focus Novell News Press Archive, 6 December 1995
- [9]. C.J.M. Arts, “Software Agreement between Western Electric Company, Incorporated and Katholieke Universiteit”, Katholieke Universiteit, 1 December 1974
- [10]. “Unix Server Platform Hardware Requirements”, Faircom Product Documentation, 3 May 2020
- [11]. D.M. Ritchie, “The Evolution of the Unix Time-Sharing System”, AT&T Bell Laboratories Technical Journal, October 1984, pp 1577-93
- [12]. J. Byrd, “The Unix File System”, Idaho State University, Unix Workshop, 5 February 1997

- [13]. G. Todino, J. Strang, J. Peek, “Learning the Unix Operating System (In a Nutshell) Fourth Edition”, O’Reilly Media, Inc., 11 December 1997
- [14]. Deaniyoer, “Maximum files in a Directory”, The UNIX and Linux Forums, 24 July 2001
- [15]. D. Munoz, “After All These Years, the World Is Still Powered By C”, Toptal, LLC, July 2015
- [16]. “What is Google Chrome?”, Javatpoint, 2011-2021
- [17]. N. Froyd, “Securing Firefox with WebAssembly”, Mozilla Hacks: Mozilla Developer Newsletter, 25 February 2020
- [18]. V. Martin, “There can be only one! A summary of the Unix standardization movement”, Crossroads - The ACM Student Magazine, Volume 1, Issue 3, pp 9-11, 1 February 1995
- [19]. E. Raymond, R. Landley, “OSI Position Paper on the SCO-vs.-IBM Complaint”, Open Source Initiative, 10 October 2003 (Final revision 23 April 2008)
- [20]. R. Stallman, “Overview of the GNU System”, GNU Operating System, 2 November 2021
- [21]. R. Stallman, “RMS Lecture at KTH (Sweden), 1986”, GNU Operating System, 30 October 1986
- [22]. Free Software Foundation, Inc., “GNU General Public License, version 1”, GNU Operating System, 1 February 1989
- [23]. “Top Open Source Licenses”, Black Duck Software, 19 July 2016
- [24]. “GNU Software”, GNU Operating System, 27 December 2021
- [25]. M. Kalin, “A guide to understanding Linux software libraries in C”, Red Hat

Open Source, 4 February 2021

- [26]. “Kernel Definition”, The Linux Information Project, 31 May 2005
- [27]. R. Buyya, “Chapter 1: Operating System Models”, pp 1-6, The Design of PARAS Microkernel, 1 June 2000
- [28]. Blackberry Limited, “Microkernel Architecture”, updated 2024
- [29]. D.L. Mills, P.H. Kamp, “The Nanokernel”, University of Delaware, Free BSD Project, 2 August 2004
- [30]. A.C. Bomberger, A.P. Frantz, W.S. Frantz, A.C. Hardy, N. Hardy, C.R. Landau, J.S. Shapiro, “The KeyKOS Nanokernel Architecture”, USENIX Association, pp 95-112, April 1992
- [31]. D.R. Engler, M.F. Kaashoek, J. O’Toole, “Exokernel: an operating system architecture for application-level resource management”, Association for Computing Machinery Digital Library, pp 251-266, December 1995
- [32]. “MIT Exokernel Operating System”, Massachusetts Institute of Technology, 5 March 1998
- [33]. Messiahandr, “Hybrid Kernel”, OSDev.org, 25 August 2005 (last updated 5 October 2024)
- [34]. L. Torvalds, “Hybrid kernel, not NT”, Real World Technologies, 9 May 2006
- [35]. S. Vaughan-Nichols, “Twenty Years of Linux According to Linus Torvalds”, ZDNET, 13 April 2011
- [36]. The info valley, “What Is Linux: An Overview of the Linux Operating System”, Medium, 11 April 2020
- [37]. J.W. Eckert, “Linux+ Guide to Linux Certification 3rd Edition”, Cengage Learning, 24 July 2012

- [38]. “What is Free Software?”, GNU Operating System, 1 January 2024
- [39]. C. Hoffman, “How Software Installation & Package Managers Work on Linux”, How-To Geek, 28 September 2016
- [40]. A. Dash, “Forking is a Feature”, Dashes: A Blog about Making Culture, 10 September 2010
- [41]. S. Stahie, “Top Five Reasons Why Ubuntu Is The Most Used OS”, Softpedia, 7 September 2015
- [42]. S. Roy, “Linux Mint History and Development”, Unixmen, 1 May 2024
- [43]. “Pop!_OS”, DistroWatch, 2 February 2024
- [44]. A. Prakash, “Linux Runs on All of the Top 500 Supercomputers, Again!”, It’s Foss, 11 January 2023
- [45]. Canonical Ltd., “Ubuntu flavours”, Ubuntu, 2024
- [46]. R. Amadeo, “Google’s iron grip on Android: Controlling open source by any means necessary”, ARS Technica, 21 July 2018
- [47]. E. Fors, M. Lange, P. Dobrow, J. Gordon, K. Golin, P. Alfieri, “Industry Leaders Announce Open Platform for Mobile Devices”, Open Handset Alliance, 5 November 2007
- [48]. E. Moyer, “Alibaba: Google just plain wrong about our OS”, CNET, 15 September 2012
- [49]. “AOSP frequently asked questions”, Android Source Documentation, FAQ, 1 February 2024
- [50]. B. King, “Is Android Really Open-Source? And Does It Even Matter?”, Make Use Of, 2 December 2021
- [51]. S. Earley, “Google’s Android OS: Past, Present, and Future”, Phone Arena, 17

August 2011

- [52]. J. Alabaster, “Android founder: We aimed to make a camera OS”, PC World, News, 16 April 2013
- [53]. C. Welch, “Before it took over smartphones, Android was originally destined for cameras”, The Verge, Google, Mobile, 16 April 2013
- [54]. L. Eadicicco, “THE RISE OF ANDROID: How a failing startup became the world’s biggest computing platform”, Tech Insider, 27 March 2015
- [55]. B. Elgin, “Google Buys Android for Its Mobile Arsenal”, Bloomberg Businessweek, Tech, 17 August 2005
- [56]. C. Haase, “Androids: The team that built the Android operating system”, “Excerpt: How Google bought Android - according to folks in the room”, 13 August 2021
- [57]. W. Sattelberg, “How to install the Google Play Store on any Android Device”, Android Police, Applications, 11 April 2024
- [58]. D. Delima, S. Suvarna, “Xiaomi to Launch MiOS as Replacement for 13-Year-Old MIUI Android Skin, Tipster Claims”, NDTV Gadgets 360, Mobile News, 10 October 2023
- [59]. H. Adamu, “OxygenOS guide: Everything you need to know about the ONePlus experience”, Android Police, Operating Systems, Guides about Operating Systems, 28 November 2023
- [60]. S. Dhapola, A. Bhatia, C. Nayak, S. Pradeep, “Vivo’s Funtouch OS 14 is here: Everything new with the Android 14 update”, The Indian Express, Tech Desk Agency, 8 October 2023
- [61]. Android Open Source Project, “Android OS Documentation”, Android Source, Documentation, 2024

- [62]. Android Open Source Project, “Get Started”, Android Source, Documentation, 2024
- [63]. Android Open Source Project, “AOSP Overview”, Android Source, Documentation, 2024
- [64]. Android Open Source Project, “Android software management”, Android Source, Documentation, 2024
- [65]. Android Open Source Project, “Brand guidelines”, Android Source, Documentation, 2024
- [66]. Android Open Source Project, “AOSP frequently asked questions”, Android Source, Documentation, 2024
- [67]. Apache Software Foundation, “Apache License, Version 2”, January 2004
- [68]. Lok Sabha, “Bulletin - Part 1 (Brief Record of Proceedings)”, pp 15, bullet point 13, 3 August 2022 / Sravana 12, 1944 (Saka)
- [69]. S. Fussell, “The Most Important Things to Know About Apps That Track Your Location”, TIME, Tech, Smartphones, 1 September 2022
- [70]. A. Joy, “How to delete old apps on your Android phone or tablet”, Android Police, Applications, 15 April 2024
- [71]. Ministry of Communication & Information Technology, Department of Electronics and Information Technology, “Policy on Adoption of Open Source Software for Government of India”, F. No. 1(3)/2014 - EG II, 24 June 2014
- [72]. A. Encarnacion, “How much storage does Android’s system files take?”, NoypiGeeks, Android, Explained, 18 June 2020
- [73]. G. Cobucci, “MIUI: Which system and pre-installed apps can be deleted?”, Xiaomi Today, 21 April 2021

- [74]. G. Edelman, “DuckDuckGo’s Quest to Prove Online Privacy is Possible”, WIRED, Security, 16 June 2021
- [75]. D. Bohn, “Google Photos will end its free unlimited storage on June 1st, 2021”, TheVerge, Tech, Google, Apps, 11 November 2020
- [76]. Y. Gopalan, “How To Elevate Device Security With A Custom Operating System”, Forbes Technology Council, Innovation, 25 April 2024
- [77]. ACL Digital, “Unleashing the Potential of Android Open Source Project (AOSP), Alten Group, 4 June 2024
- [78]. G. Cobucci, “How to change font on Xiaomi and Redmi MIUI”, Xiaomi Today, 27 September 2020
- [79]. C.S. Brown, “Oxygen OS guide: Everything you need to know about OnePlus’ Android Skin”, Android Authority, Mobile, Android OS, 22 June 2023

APPENDIX

Android	9, 10, 11, 13, 14, 15, 20, 23, 27, 29, 35
AOSP	9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 21, 22, 23, 28, 31, 32, 35
Apache	9, 16, 18, 36
AT&T	1, 3
Copyleft	4, 16, 17
Copyright	3, 4, 10, 11, 17, 18, 19, 36
Developer	3, 7, 9, 11, 12, 13, 15, 19, 31, 35
Device	5, 6, 9, 10, 13, 15, 20, 28, 30, 31
Distro	7, 8, 13, 23, 27, 28, 35, 36
Fork	7, 9, 10, 12, 13, 14, 15, 16, 18, 19, 21, 35, 36
FOSS	3, 6, 7, 10, 13, 15, 20, 21, 22, 28, 29, 30, 35, 36
GNU	4, 6, 23
Google	2, 9, 10, 11, 13, 14, 15, 16, 20, 21, 29, 35
GPL	4, 6
Kernel	5, 6
Linux	1, 3, 4, 5, 6, 7, 8, 9, 23, 27, 28, 35
OHA	9, 10, 13, 20, 21, 22
Parent	35
Proprietary	3, 9, 10, 20, 35
Smartphone	11, 14, 15, 20, 21, 22, 27, 28, 35, 36
Software	1, 2, 3, 4, 6, 7, 10, 12, 13, 16, 17, 18, 23, 31, 35
Source code	2, 3, 5, 7, 12, 15, 19
Unix	1, 2, 3, 4, 6, 7, 8, 24, 35, 36
Windows	2, 9, 27