DaxOS

A PROJECT REPORT

submitted by

By

Nihal Narayan (MBT17CS081)

Antony S. Chirayil (MBT17CS023)

Mathew Koshy (MBT17CS068)

R Midhun Suresh (MBT17CS095)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

January, 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEEERING

MAR BASELIOS COLLEGE OF ENGINEERING & TECHNOLOGY

Bethany Hills, Nalanchira

Thiruvananthapuram 15

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAR BASELIOS COLLEGE OF ENGINEERING & TECHNOLOGY

Nalanchira, Thiruvananthapuram.

CERTIFICATE

This is to certify that the report entitled ***DaxOS*** *submitted by* ***Nihal Narayan (MBT17CS081)***, ***Antony S. Chirayil (MBT17CS023)***, ***Mathew Koshy (MBT17CS068)***, ***R Midhun Suresh (MBT17CS095)*** *to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and Technology is a bonafide record of the project work carried out by him/her under my/our guidance and supervision.This report in any form has not been submitted to any other University or Institute for any purpose*

Ms. Gayathri K. S.	**Mr. V. S. Shibu**	**Dr. Tessy Mathew**
Project Coordinator	Guide	Head of the Department

Place: Thiruvananthapuram

Date: 12/01/2021

Acknowledgement

We would like to take this opportunity to extend our sincere gratitude to **Mr. Shibu VS** for guiding us through the process of this project. We would also like to thank the seminar coordinator, **Mrs. Gayathri KS**, for providing us with this opportunity which has allowed us to explore a domain which would otherwise be beyond the confines of our academic course.

Nihal Narayan
Mathew Koshy
Antony S. Chirayil
R Midhun Suresh

ABSTRACT

Every computer science enthusiast is proficient in their operating system of choice. They may even know its underlying working from an old operating system course they took in college. However, it is usually the case that their knowledge and understanding is limited to theory and writing low-level system code is often considered an insurmountable challenge. This project hopes to change this attitude by developing a minimal yet functional 32-bit operating system that can be used in conjunction with theoretical teaching to promote and introduce systems programming. A minimal kernel guarantees easier to read source code (as opposed to the 27 million SLOC Linux kernel) and provides a gentler introduction to kernel development. The kernel will include a full keyboard and mouse driver and will have support for VGA text-mode and graphics. It will also contain a limited libc implementation with a streamlined build process. Additionally, this project will serve as an illustration for good development practices (code reuse, clean architecture, unit testing). The 32-bit kernel will be written in C with a little of assembly for the truly low-level aspects.

Contents

List of Figures

List of Tables

Nomenclature

Chapter 1

Introduction

arabic

The Kernel is the fundamental interconnect between hardware and software of a computer system. Writing a kernel (kernel programming) is considered to be a difficult endeavor because development has to start from a bare metal state. DAX OS is a minimal 32-bit hobbyist operating system that can be used to provide a gentle introduction to students who wish to explore the domain of systems programming. The project is open source and licensed under GNU General Public License v3.0 to ensure unrestricted access and complete transparency. DAX OS comes with a terminal driver, keyboard and mouse driver and basic memory management. The project also uses appropriate development practises such as unit-testing and version control.

1.1 Existing Technologies

1.1.1 Linux

Linux Kernel, created by Linus Torvals, is extremely large and contains about 27.8 million lines of code. It is free and open- source and compiling takes a long time. Kernel code is written using the GNU/GCC to the standard C programming language.

1.1.2 Windows Research Kernel

Windows Research Kernel is not free and open-source.It is difficult to get access as it need to get registered through University. It is based on older windows XP / Server 2003 code. It is outdated and not in development.

1.1.3 MINIX

Minix microkernel, the best introductory kernel and developed by Andrew S. Tanenbaum. However, the kernel has a micro-kernel design instead of the familiar monolithic kernel. Traditional unix based kernels (eg: linux) follow monolithic design. It was created for educational purposes and its is now open-source.

1.2 Proposed System

We propose to build a 32-bit kernel that has the following functionality:

1. **Keyboard Driver**

   Dax-OS will include a fully functional PS/2 keyboard driver. The PS/2 keyboard driver will convert scan-codes generated when the user presses a key on the keyboard to an integer character code. It must be noted that all keyboards practically used in modern day utilize the USB standard. We stick with the PS/2 protocol because the USB protocol is massive and difficult to implement. However there are practically no disadvantages from such a decision because most motherboards will emulate USB keyboards as PS/2 keyboards.

2. **Mouse Driver**

   Similar to the keyboard driver, the operating system will also contain a functional mouse driver. The mouse can be utilized to move the terminal window up and down.

3. **Terminal Display Driver**

   DaxOS is a terminal based operating system i.e it does not support windowing. The display support will be implemented using VGA text-mode and real-mode.

4. **Basic Memory Management**

   DaxOS will use a flat memory model. Since it is an 32-bit operating system, it will support at most 4GB of addressable memory. A custom memory allocator will also be implemented.

5. **Graphics Support**

   DaxOS will provide basic graphics capability using the above mentioned VGA real-mode to draw primitive shapes with the available 16 colors.

1.3 Technology Stack

The bulk of the operating system is written in the C programming language. Certain functionality such as writing data to ports and loading tables (IDT, GDT) are implemented using either GCC inline assembly or using normal x86 assembly.

The project uses the GCC cross-compiler & binutils to target the generic i686 platform; which is a generic 32-bit Intel P6 architecture. The GNU Assembler and Linker are also used. The assembly syntax style used is AT&T.

The entire compilation process is driven using GNU Make which uses Makefiles to build the project. The compilation is initiated using BASH shell scripts. The kernel is tested using the qemu-i386 emulator and is developed on a stable Xubuntu distribution.

DAX OS uses git as its version control system of choice. The git repository is uploaded on Github and every new feature is developed on its own separate branch. Team communication and coordination are actualized using discord with Github integration enabled.

Chapter 2

Design Diagrams

Write the literature review here.

Citations should be included as [**?**], or [**?**, **?**] if there are more than one references to cite.

Chapter 3

Chapter heading

Write the body of the thesis. Include as many chapters as needed.

3.1 Sections in chapter

Logically divide the content of the chapter into sections and subsections.

3.1.1 An example subsection

Within a subsection, content division may again be included as shown below.

3.1.1.1 Examples of equations

A few examples of various types of equations are shown in Eq. 3.1, to Eq. 3.7.

$$A(z) = \frac{1 - B(z)}{1 - B(\frac{z}{\gamma})} \tag{3.1}$$

$$A(\frac{z}{\gamma}) = \sum_{i=1}^{M} \gamma^i x_i z^{-i}, 0 < \gamma < 1 \tag{3.2}$$

$$a_n^{'} = \begin{cases} a_n b_n, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

$$a_n = \alpha - \beta \cos \frac{2\pi n}{N-1} \tag{3.4}$$

where $\alpha = 0.54$ and $\beta = 1 - \alpha = 0.46$.

$$\frac{\delta C}{\delta a_k} = 2E[(s[n] + \sum_{k=1}^{M}(a_k s_{n-k}))b_{n-i}] = 0 \tag{3.5}$$

$$A = \|\mathbf{x}(n) - \eta_k b_i \mathbf{C} y_j\| \tag{3.6}$$

$$a_1 = b[T^{(1)}] \tag{3.7}$$

3.1.1.2 Examples of figures

Figures may be included as in Fig. **??**. To increase or decrease the size of the figures, adjust the 'scale' parameter in the code.

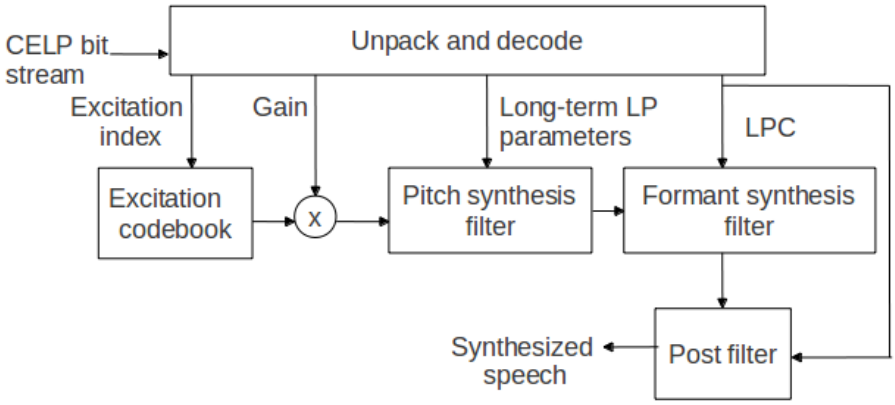


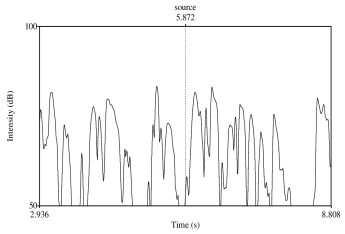
Figure 3.1: An Example Figure

If the size of the figures needs to be made uniform, use the parameters 'width' and 'height' as shown in Fig. 3.2.

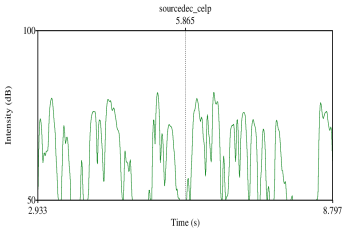
3.1.1.3 Examples of tables

Tables may be formatted as in Table 3.1.

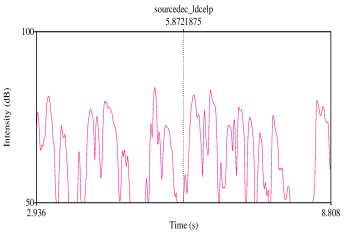
If vertical lines are not required, tables may be formatted as in Table 3.2.



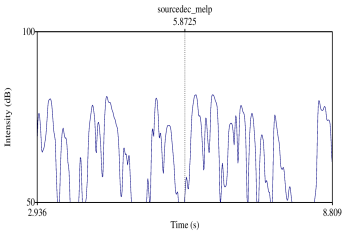
(a) Original Sound File



(b) Decoder 1 output file



(c) Decoder 2 output File



(d) Decoder 3 output file

Figure 3.2: An Example showing Sub-figures

Table 3.1: An Example Table

Parameter	Algorithm 1	Algorithm 2	Algorithm 3
file1.wav	3.27	3.93	2.73
file2.wav	3.53	4.07	2
file3.wav	3.53	4.47	2.8
Average Value	**3.44**	**4.16**	**2.51**

Table 3.2: An Example Table

Parameter	Algorithm 1	Algorithm 2	Algorithm 3
file1.wav	3.27	3.93	2.73
file2.wav	3.53	4.07	2
file3.wav	3.53	4.47	2.8
Average Value	**3.44**	**4.16**	**2.51**

Chapter 4

Conclusion

Write the Conclusion here.

Chapter 5

Appendix

Include appendices here (optional)

Chapter 6

List of Publications

Write the list of publications here.

1. Publication 1
2. Publication 2
3. Publication 3

REFERENCES

[1] write the references

[2] write the references