Colegiul National Bilingv
"George Cosbuc"

The Impact of C, UNIX, and Pioneers Dennis Ritchie & Ken Thompson

Elev: Sorin-Andrei Tudose
Clasa: $12R_2$

Profesor coordonator:
Kerestély Melinda Laura

Table of Contents

Synopsis

What has always fascinated me is computer programming and software engineering. After a few years of using a computer as a means of playing games I started wondering how those application run and their build process. There were a few questions for which I really needed to find the answer, such as: How are games and other applications built?, How are different pieces of information displayed on the screen?, How one can master the necessary skills to write a computer program?, etc. So I started doing research. Even thought, at that time, the documentation I was reading overwhelmed me, I managed to understand that if one had the necessary knowledge, the possibilities in software engineering are endless. Since I wrote my first `"Hello world"` in C++, when I was 15 years old, I knew that hardware and software was the area I wanted to excel in. I find it truly magical how people can manipulate every pixel of a computer screen and display any desired content.

As a result, creating code and building applications, that help in solving different problems and enable us to be more effective is what I am keen on pursuing in my near future. It is undoubtedly a complex field where mathematical and physics skills are required, but the excitement and passion for this field helps me overcome all challenges.

Having said that, I decided to write my Grade 12 Graduation Paper about **Dennis Ritchie** and **Ken Thompson** who revolutionized, in **1969** at **Bell Telephone Laboratories**, the way people used computers. Not only did they create the **UNIX** operating system which is the root of many modern operating systems, but they also developed the **C** programming language. I, for one, am amazed of the things they created considering the fact that resources were not as broadly available as they are nowadays. There were almost no applications designed for creating software and no internet that can assist with the process. In addition, they also had to write most of the code is Machine Language, a low level programming language which takes a long time to master.

Throughout this thesis we will go on a journey tracing the origins of computers.

Introduction

These days most people own at least one modern computer for personal usage. As a result many do not know that the original design looked nothing like it does today.

First and foremost, there were no operating systems, programming languages, mice or keyboards. Moreover, there was no **physical storage** (*solid-state drives* or *hard disk drives*), **RAM** memory (*random access memory*) or **ROM** memory (*read only memory*). **ENIAC** (*Electronic Numerical Integrator and Computer*) (*Figure 1*), the very first computer designed at the University of Pennsylvania in 1945 to serve as a military tool to properly calculate the angle of elevation of a gun. This computer consumed around 150,000 watts which is significantly higher than modern computers (200-500 watts).

So how did ENIAC work?

Not only did it use cables and switches, which were used to start, shut down and control the flow of data, but also plugboards, electrical switchboards into which electrical plugs are inserted to create different temporary circuits, for transmitting instructions to the machine. As a result, for each task the computer needed manual calibration which meant rewiring the plugboards and adjusting the switches. This "programming" method was not efficient in terms of flexibility and speed.

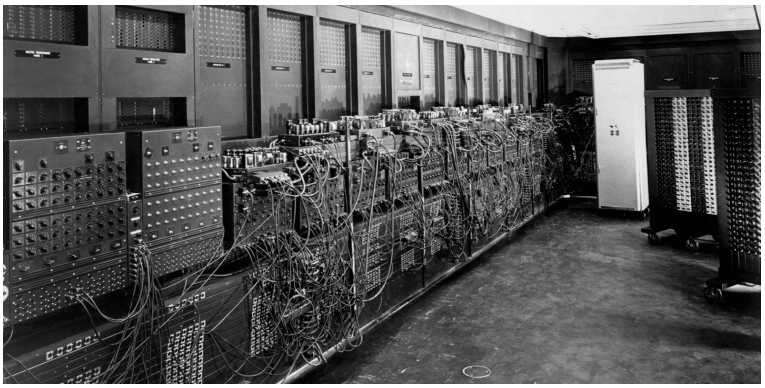

Figure 1: ENIAC

"Preparing ENIAC for a series of runs was an incredibly involved process. First, detailed instructions had to be written defining the problem and a procedure for solving it. These instructions were programmed by adjusting switches manually and inserting thousands of cables into as many as forty large plug boards. A team of five operators might work several days on the external wiring and many more days searching for errors and correcting them." — **Breakthrough to the Computer Age, Harry Wulforst, Charles Scribner's & Sons Pub., 1982**

After experimenting with ENIAC, people found different ways to make significant improvements:

- (1949) **EDSAC** and **EDVAC** - Computers get memory as tubes with mercury
- (1949) **BINAC** - Programming languages were introduced (**William Schmitt** – Short Order Code)

- (1952) **IBM 701** - The first computer to use the assembler-style language (*Figure 2*) and a repository to store frequently used blocks of data.

```
1         *ASM     XOPTS(LEASM)
2         DFHEISTG DSECT
3         OUTAREA  DS    CL200                    DATA OUTPUT AREA
4         *
5         TESTLE   DFHEIENT CODEREG=R3
6                  MVC OUTAREA(40),MSG1
7                  MVC OUTAREA(4),EIBTRMID
8                  EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(43) FREEKB ERASE
9                  EXEC CICS RECEIVE
10                 MVC OUTAREA(13),MSG2
11                 EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(13) FREEKB ERASE
12                 EXEC CICS RETURN
13        *
14        MSG1     DC     C'xxxx: ASM program invoked. ENTER TO END.'
15        MSG2     DC     C'PROGRAM ENDED'
16                 DFHREGS
17        CODEREG  EQU    R3
18                 END
```

Figure 2: IBM Assembler Language (ASM)

- (1969) The **UNIX Time-Sharing** Operating System

Chapter 1: Dennis Ritchie & Ken Thompson: C & UNIX Pioneers

Chapter 2: UNIX, the operating system revolution

Chapter 3: The C programming language

C architecture: Unveiling the language's design

Impact on programming languages & software development

The purpose behind C++

Chapter 4 - The everlasting Legacy

Relevance of C and UNIX systems in today's technology

The future of software development. 50 Years of C: Is it still worth learning?

Conclusion