

# Operating Systems 1 assignment 1

RAJ PATIL  
CS18BTECH11039

A brief history of operating systems

## PLAGIARISM STATEMENT

I certify that this assignment is my own work, based on my personal study and research and that I have acknowledged all materials and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

**Name: Raj Patil**



**Date: 13<sup>th</sup> November 2019**

**Signature**

Acknowledgement:

- Operating System Concepts: Galvin et al
- Modern Operating Systems: Andrew Tanenbaum

**This is a synopsis regarding the history of operating systems and the fundamental notions tending to one.**

An operating system(referred to as OS from here onwards) is an intermediary between the conventional user and the hardware; its purpose is to make the process of using computers efficient and convenient.

The purpose of an OS can be distributed generically into the following

- allow for more accessible communication between the user and the hardware(provide a high-level abstraction for the programmer to work with-however,note that these abstractions are generally used by application programs which provide another level of simplification for the physical user)
- implementing security measures to protect the operations, which is achieved by using a modal operating system(discussed later)
- acting as a resource manager: efficient allocation of valuable resources such as compute time and/or power and priorities for using the hardware, for instance
  - this notion can be extended to the idea of sharing resources between multiple users which is a rapidly growing sector currently(cloud computing (time multiplexing & space multiplexing)) as it allows us to take advantage of the economies of scale.

Now, with the basics out of the way, focusing on the evolution of operating systems over time.

In a crude sense, OSs have been somewhat evolving in correspondence to the computer architecture they have been paired with over the years.

Moving chronologically, the first computer was made by Charles Babbage(quoted as an "analytical engine" by him), and it did not have an os and was purely mechanical. Little progress was made in the construction of digital computers until World War II, and the first digital computer was made of vacuum tubes. This was the first generation of digital computers, and all the programming was done in machine language or by wiring electrical circuits by connecting cables to plugboards to control a machine's essential functions. The procedure was made slightly more convenient with usage of punched cards instead of plugboards in the early 1950s, but it was rudimentarily the same.

In the second generation of computers(1955-65) transistors and batch systems were employed, and this resulted in a radical revamp of the current computational scenario. Clear boundaries could be established between the kinds of jobs in the field such as designers, builders, operators, programmers, and maintenance personnel. In these mainframe computers (as they were called), programmer had to punch out code in FORTRAN on cards and submit it to operators for it to be processed. The output was collected later on in a different room - they needed to be housed in large air-conditioned rooms. This resulted in a significant amount of time being wasted walking around the room carrying cards and called for the need to employ batch processing wherein a collection of jobs was copied on a magnetic tape using a relatively cheaper computer and then processed as batches - this saved computer time to an extent. An example of the copying computer is the IBM 1401 and the one on which the actual computation was done: IBM 7094. The same kind of copying computer was later used to print out the output from an output tape. A typical job started out with an SJOB card, indicating max run time, and some other metadata about the programmer. This was followed by an SFORTRAN card which commanded the OS to load the FORTRAN compiler from the system tape - followed immediately by the program to be compiled which preceded a \$LOAD card directing the OS to load the object program just compiled. Next, a \$RUN and SEND card told the OS to run the program and mark the program's end respectively. Typical OSs during this phase were the Fortran Monitor System and IBSYS( for the IBM 7094)

A lot of novel implementations were included in the third generation(1965-1980). Firstly, there was a divide into two basic categories: scientific computers such as the 7094 and commercial computers like the 1401. However, with the introduction of Integrated Circuits(ICs) IBM combined the two machine types into one compatible for both jobs into the IBM 360. The 360 was the first to employ multiprogramming- allowing multiple programs to be loaded in the memory at once, each with a separate partition which allowed for efficient usage of the computer such as when one job was tending to I/O, another job could be prioritised by CPU rather than waiting. Another valuable addition was the technique of spooling(simultaneous peripheral operation on line) which enabled the ability to read jobs form cards onto the disk as soon as they were brought to the computer room. Hence, whenever a job was finished, the OS

could load a new job from the disk and run it without the need of operators and those 2 IBM 1401 for input and output. This result in a major decrement in the carrying of tapes.

Moreover, timesharing was also introduced ( a variant of multiprogramming) which allowed for greater user satisfaction as a whole even though the time taken to complete one job increased. Another significant growth in this generation was the proliferation of mini-computers which were also employing timesharing on a single-user level: in terms of the different tasks performed by one user. This later developed as the UNIX OS. Due to it being open-source, lots of variations were developed leading to inconsistencies. For restoring compatibility over all the UNIX flavours, IEEE developed a standard for UNIX - POSIX which is supported by most versions of UNIX. In the same timeline(around 1987), a UNIX clone- MINIX was released for educational purposes which later evolved into MINIX 3 which supports high modularity(increased security) resulting in greater reliability.

In the fourth generation(1980-present), a lot of emphases was laid on the growth of the personal computation space which was permitted by the development of large scale integration circuits containing relatively larger amounts of transistors in a smaller area of silicon. What the minicomputer meant for a university/company, a microprocessor chip-enabled a personal computer to hold that place for a single individual. In the early 1980s, the IBM PC was developed, and Bill Gates made a Basic interpreter for it, and after a couple of failed business deals, IBM asked Gates for an OS for the IBM PC. This led to the production of MS-DOS(Microsoft disk operating system) which quickly engulfed the IBM PC market. This was perhaps the first influential operating system tending to personal computers, which is reflected in the current market share statistics where Microsoft holds a significant chunk with its Windows line of operating systems. However, MS-DOS was reasonably primitive and employed a CLI(command-line interface). This esoteric nature of personal computers led Steve Jobs to further dive into the idea of a GUI(graphical user interface) which, after a failure(LISA) took the form of Macintosh which became widely popular due to its user-friendly nature.

Fast-forwarding to the present day, the basic principles governing the development of an operating system remain the same, albeit considering the significant advancements in the computer architectures we are experiencing.