International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 561-564 www.ijaem.net ISSN: 2395-5252

Evolution of Operating System

Hitesh Kumaar S[1], Midhun Chakkaravarthy K[2,] Mohan Prasanth S[3]
Dr.M. Sujithra M.C.A, M.Phil., PhD, Dr.A.D. Chitra M.C.A, M.Phil., PhD,

1,2,3,4 M.Sc. Software Systems,Coimbatore Institute of TechnologyCoimbatore
Assistant Professor,Department of Data Science, Coimbatore Institute of Technology, Coimbatore
Assistant Professor,Department of Software Systems,Coimbatore Institute of Technology, Coimbatore

---

Date of Submission: 15-11-2020 Date of Acceptance: 30-11-2020

ABSTRACT - The early computers of the late 1940s had no operating system. Human operators scheduled jobs for execution and supervised the use of the computer's resources. Because these early computers were very expensive, the main purpose of an operating system in these early days was to make the hardware as efficient as possible. Now, computer hardware is relatively cheap by comparison with the cost of the personnel required to operate it, so the purpose of the operating system has evolved to encompass the task of making the user as efficient as possible.
Keywords--Jobs,Batch,Spool.

I. INTRODUCTION

An operating system functions in much the same way as other software. It is a collection of programs that are loaded into memory and executed by the processor. When the computer is powered down it exists only as a collection of files on a disk drive. The main difference is that, once it is running, it has a large measure of control over both the processor itself and other system resources. In fact, the operating system only relinquishes control to allow other programs to execute. An application program is frequently given control of the processors for short periods of time in order to carry out its allotted task, but control always reverts to the operating system, which can then either use the processor itself or allocate it to another program.

The operating system, then, controls the operation of the computer. This includes determining which programs may use the processor at any given time, managing system resources such as working memory and secondary storage, and controlling access to input and output devices. In addition to Controlling the system itself, the operating must provide an interface between the system and the user which allows the user to interact with the system in an optimal manner.

Increasingly these days, the operating system provides sophisticated networking functionality, and is expected to be compatible with a growing range of communication devices and other peripherals. In recent years, the implementation of an application programming interface (API) has been a feature of most operating systems, making the process of writing application programs for those operating systems much easier, and creating a standardized application environment.

II. EVOLUTION OF OS

Need for evolution:Computer operating systems provide a set of functions needed and used by most application programs on a computer, and the links needed to control and synchronize computer hardware. On the first computers, with no operating system, every program needed the full hardware specification to run correctly and perform standard tasks, and its own drivers for peripheral devices like printers and punched paper card readers. The growing complexity of hardware and application programs eventually made operating systems a necessity for everyday use.

A. Open shop

Each user was allocated a block of time to load and run his/her program, which was input from punch cards. Debugging consisted of inspecting the internal machine states and patching them directly. Device drivers (device-specific routines), functions, compilers, and assemblers had to be explicitly loaded.

B. Operator-driven shop

The computer operator loaded the jobs and collected output. Users debugged programs by inspecting a core dump, which was a hexadecimal listing of the exact contents of memory. The

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 561-564 www.ijaem.net ISSN: 2395-5252

operator could batch jobs or rearrange them according to priority, run time, etc.

C. Serial Programming

Serial Processing: The Serial Processing Operating Systems are those which Performs all the instructions into a Sequence Manner or the Instructions those are given by the user will be executed by using the FIFO Manner means First in First Out. All the Instructions those are Entered First in the System will be Executed First and the Instructions those are Entered Later Will be Executed Later. For Running the Instructions, the Program Counter is used which is used for Executing all the Instructions.

In this the Program Counter will determine which instruction is going to Execute and the which instruction will be Execute after this. Mainly the Punch Cards are used for this. In this all the Jobs are firstly Prepared and Stored on the Card and after that card will be entered in the System and after that all the Instructions will be executed one by One.

Limitations:Serial processing strictly refers to 'sequential'. There aren't any overlaps of successive processing times on objects or distinct subsystems. A standard type of serial system, each object takes the same average amount of time to process and the next object begins processing only after the previous one is done. Hence, the CPU may only work a single task at the time which severely limits the fluidity and responsiveness of a program during heavy workloads.

D. Simple Batch System

A separate computer was used for I/O. Several programs were first loaded onto tape, and then the full tape was read into the main computer. Program output and dumps were written to tape, and then printed from the tape by the auxiliary computer.In this type of system, there is no direct interaction between user and the computer.The user has to submit a job (written on cards or tape) to a computer operator. Then computer operator places a batch of several jobs on an input device. Jobs are batched together by type of languages and requirement. Then a special program, the monitor, manages the execution of each program in the batch. The monitor is always in the main memory and available for execution.

A small resident monitor program reset the main computer after each job, interpreted some simple command language, performed some simple accounting, and did device-independent input and output.

D. Multiprogrammed batch systems

Spool:simultaneous peripheral operations on line: program and data cards were first read to disk, input was thus available quickly when needed since reading from the disk was much faster than reading from cards; output was first copied to disk, then printed from disk by a peripheral processor. Disks were used for intermediate storage: faster than tapes and allowed jobs to be processed in any order. A nucleus (or kernel) contained routines to manage processes (called jobs) and device interrupts. Used interrupts to perform I/O (device tells computer when it is finished a task). Device drivers included in the nucleus. A process (running program) requested assistance from the kernel by making a service call = system call. A scheduler sorted incoming jobs according to priority and processor time needed. Still used a human operator to mount data tapes needed by jobs, make some policy decisions about which jobs to run, and to restart the resident monitor if it failed or was overwritten by a program.Could do multi programming or multitasking: have more than one process somewhere between starting and finishing.
-Example: IBM OS/360

Advantages of multiprogramming systems:

- CPU is used most of time and never become idle
- The system looks fast as all the tasks runs in parallel
- Short time jobs are completed faster than long time jobs
- Multiprogramming systems support multiply users
- Resources are used nicely
- Total read time taken to execute program/job decreases
- Response time is shorter
- In some applications multiple tasks are running and multiprogramming systems better handle these type of applications

Disadvantages of multiprogramming systems:

- It is difficult to program a system because of complicated schedule handling
- Tracking all tasks/processes is sometimes difficult to handle
- Due to high load of tasks, long time jobs have to wait long

E.Interactivemultiprogramming (Timeshared sys tem)

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 561-564 www.ijaem.net ISSN: 2395-5252

Users interact with the computer directly through a command language at a terminal. A command interpreter defines interface. A session lasts from logon to logoff. Text editors allow users to create programs, text files, and data files online instead of with cards or tape.

Time-Sharing Operating Systems is one of the important type of operating system.

Time-sharing enables many people, located at various terminals, to use a particular computer system at the same time. Multitasking or Time-Sharing Systems is a logical extension of multiprogramming. Processor's time is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Time-Sharing Systems and Multiprogrammed Batch Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are implemented by the CPU by switching between them, but the switches occur so frequently. So, the user can receive an immediate response. For an example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation, i.e.; if n users are present, then each user can get a time quantum. Whenever the user submits the command, the response time is in few seconds at most.

An operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems which were designed primarily as batch systems have been modified to time-sharing systems.
- Examples: CTSS, MULTICS, UNIX

Types of Multitasking Operating System:

Preemptive Multitasking - Preemptive multitasking is special task that is assigned **to** computer operating system, in which it takes decision that how much time spent by one task before assigning other task for using the operating system. Operating system has control for completing this entire process, so it is known as "Preemptive".

Cooperative Multitasking - Cooperative multitasking is known as "Non-Preemptive Multitasking". Main goal of Cooperative multitasking is to run currently task, and to release the CPU to allow another task run. This task is performed by calling taskYIELD().Context-switch is executed when this function is called.

Advantages of Timesharing operating systems are

- It provides the advantage of quick response.
- This type of operating system avoids duplication of software.
- It reduces CPU idle time.

Disadvantages of Time-sharing operating systems are

- Time sharing has problem of reliability.
- Question of security and integrity of user programs and data can be raised.
- Problem of data communication occurs.

F.Distributed computing

It is one of the important type of operating system.Multiple central processors are used by Distributed systems to serve multiple real-time applications and multiple users. Accordingly, Data processing jobs are distributed among the processors.

Processors communicate with each other through various communication lines (like high-speed buses or telephone lines). These are known as **loosely coupled systems** or distributed systems. Processors in this system may vary in size and function. They are referred as sites, nodes, computers, and so on.

Following are the two types of distributed operating systems used:

1. Client-Server Systems
2. Peer-to-Peer Systems

Client-Server Systems

Centralized systems today act as server systems to satisfy requests generated by client systems.

Server Systems can be broadly categorized as: Compute Servers and File Servers.

- Compute Server systems, provide an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the client.
- File Server systems, provide a file-system interface where clients can create, update, read, and delete files.

Peer-to-Peer Systems

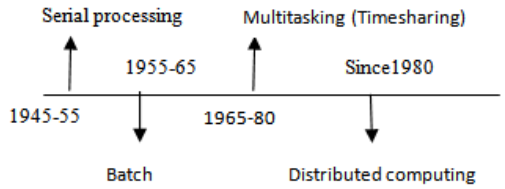
The growth of computer networks - especially the Internet and World Wide Web (WWW) – has had a profound influence on the recent development of operating systems. When PCs were introduced in the 1970s, they were designed for personal use and were generally considered standalone computers. With the beginning of widespread public use of the Internet in the 1990s for electronic mail and FTP, many PCs became connected to computer networks.

In contrast to the Tightly Coupled systems, the computer networks used in these applications consist of a collection of processors that do not share memory or a clock. Instead, each processor has its own local memory.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 561-564 www.ijaem.net ISSN: 2395-5252

The processors communicate with one another through various communication lines, such as high-speed buses or telephone lines.

III. TIMELINE



IV. CONCLUSION

Operating systems have evolved from slow and expensive systems to present-day technology where computing power has reached exponential speeds and relatively inexpensive costs. In the beginning, computers were manually loaded with program code to control computer functions and process code related to business logic. This type of computing introduced problems with program scheduling and setup time. As more users demanded increased computer time and resources, computer scientists determined they needed a system to improve convenience, efficiency, and growth (Stallings, 2009, p. 51). As a result, they created an operating system (OS) to process jobs in batches. Later they created Multitasking and Time-Sharing to run multiple jobs and allow user interaction to improve efficiency. Multitasking brought challenges to manage I/O operations required by multiple jobs in which computer vendors resolved with interrupts.

REFERENCES

[1]. http://lloogg.com/history-and-evolution-of-operating-systems
[2]. http://www2.cs.uregina.ca/~hamilton/courses/330/notes/history/history.html
[3]. https://www.poweradmin.com/blog/the-evolution-of-operating-systems/
[4]. https://www.youtube.com/watch?v=1-7BlGuo4v8
[5]. Concurrent Systems or Operating Systems Bacon J [and Harris T], Addison Wesley 1997 [2003]
[6]. Operating Systems Steven Hand, Michaelmas Term[2010]