iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** IX **Month of publication:** September 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Classification of Operating System

Ajay Kumar*[1], Bharat Kumar*[2], Chirag Dawar*[3], Dinesh Bajaj* [4]

[1, 2, 3, 4] *(B.Tech 3rd sem) Dept. of Computer Science and Engineering, Dronacharya College of Engineering, Gurgaon 122001, India*

Abstract- This paper is all about the operating system and its classification. Operating system (OS) is a bridge between system and user. It is software which handles computer software and hardware resources. The operating system performs basic tasks, such as recognizing input from the keyboard, sending output to the display screen, controlling and allocating memories, keeping track of files and directories on the disk and controlling input and output devices. Application programs generally require an operating system to function. It is difficult to present a complete as well as deep amount of operating systems developed till today. So, this paper tries to overview only a subset of the available operating systems and its different categories. OS systems are being developed by a large number of academic and commercial organizations for the last several decades. This paper, therefore, concentrates on the different categories of operating systems with special emphasis to those that had deep impact on the evolution process. The aim of this paper is to provide a brief timely commentary on the different categories important operating systems available today. Examples of popular OS include Android, Linux, Osx, Microsoft windows, Windows phone.

Keywords-Time Sharing OS; Real Time OS; Distributed OS; Memory allocation; Quantum; SMP (symmetric Multiprocessor systems); Global aggregate OS; GUI (Graphical User Interface).

I. INTRODUCTION

An *operating system* (OS) is a software that manage computer hardware and software resources and provides common services for computer programs. Itis an interface between hardware and user which is responsible for the management and coordination of activities and the sharing of the resources of a computer, that acts as a host for computing applications run on the machine. The operating system is an essential component of the system software in a computer system. Application programs usually require an operating system to function. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and will frequently make a system call to an OS function or be interrupted by it. Operating systems offer a number of services to application programs and users. Applications access these services through application programming interfaces (APIs) or system calls. An operating system is required to perform a variety of functions. Some of the important functions are: processor management, memory management, file management, device management, error handling, interrupt handling, resource control, protection, multi-access. Operating systems can be found on almost any device that contains a computer—from cellular phones and video game consoles to supercomputers and web servers. Examples of popular modern operating systems include Android, BSD, iOS, Linux, OSX, QNX, Microsoft Windows,[3] Windows Phone, and IBM z/OS.

II. CLASSIFICATION

Operating system classification depends on:
Purpose, Task processing mode, Way of user-system interaction, Means of multi-processing, Architecture and on this basis operating system can be broadly classified as:

1. on the basis of purpose-
   - Time sharing operating system
   - Real time operating system
   - Distributed operating system
2. On the basis of processing-
   - single processor system

ISSN: 2321-9653

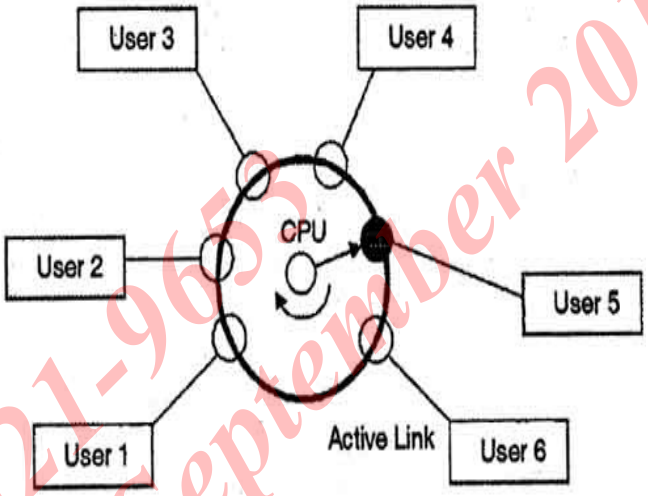
- multi-processor system

3. On the basis of tasking-
   - single tasking
   - multi-tasking
4. On the basis of number of users-
   - single user
   - multi user

1. Time Sharing Operating System

A time sharing system allows many users to share the computer resources simultaneously. In other words, time sharing refers to the allocation of computer resources in time slots to several programs simultaneously. In an interactive system, many users directly interact with the computer from terminals connected to the computer system. They submit small execution requests to the computer and expect results back immediately, after a short enough delay to satisfy their temperament. We need a computer system that supports both multi programs and multi processes. The processes appear to be executing simultaneously, each at its own speed. Apparent simultaneous execution of processes is achieved by frequently switching the processor from one process to another in a short span of time. These systems are often called time-sharing systems. It is essentially a rapid time division multiplexing of the processor time among several processes. The time sharing systems were developed to provide an interactive use of the computer system. A time shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. It allows many users to share the computer resources simultaneously. As the system switches rapidly from one user to the other, a short time slot is given to each user for their executions.

The time sharing system provides the direct access to a large number of users where CPU time is divided among all the users on scheduled basis. The OS allocates a set of time to each user. When this time is expired, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that they each have their own CPU and they are the sole owner of the CPU. This short period of time during that a user gets attention of the CPU; is known as a *time slice or a quantum.* The concept of time sharing system is shown in figure.



In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status.

As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

The time-shared systems are more complex than the multi-programming systems. In time-shared systems multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped out within a short time.

Examples - Menu driven programs, Dialogue driven programs etc.

2. Real Time Operating System

It is a multitasking operating system that aims at executing real-time applications. Real-time operating systems often use specialized scheduling algorithms so that they can achieve a deterministic nature of behavior. The main object of real-time operating systems is their quick and predictable response to events. They either have an event-driven design or a time-

sharing one. An event-driven system switches between tasks based on their priorities while time-sharing operating systems switch tasks based on clock interrupts. A real-time systems reacts to outside events reaching in a given, application specific time, if this deadline is not met, the answer can be considered erroneous. Real Time System is usedat those Places in which we Requires higherand Timely response. These types ofsystems are used in reservation so whenwe specify the request, the CPU will performat that Time.

There are two Types of realtime System:

a. Hard Real Time System:

In the Hard Real Time System, time is fixed and we change any moments of the time of processing means CPU will process the data as we enters the data.

b. Soft Real Time System:

In the Soft Real Time System, some moments can be change means after giving the command to the CPU, CPU performs the operation after a micro second.

Examples- OSE, QNX, RT Linux, VX Works, Windows CE

3. Distributed Operating System

Distributed means data is stored and processed on multiple locations. Distributed Operating System is a model where distributed applications are running on multiple computers linked by communications. A distributed operating system is an extension of the network operating system that supports higher levels of communication and integration of the machines on the network .It is a software over a collection of independent, networked, communicating, and physically separate computational nodes.[1] Each individual node holds a specific software subset of the global aggregate operating system. Each subset is a composite of two distinct service provisioners.[2] The first is a ubiquitous minimal kernel, or microkernel, that directly controls that node's hardware. Second is a higher-level collection of *system management components* that coordinate the node's individual and collaborative activities. These components abstract microkernel functions and support user applications.

This system looks to its users like an ordinary centralized operating system but runs on multiple, independent central processing units (CPUs).When a data is stored on to the multiple computers those are placed in different locations. Distributed means in the network, network collections of computers are connected with each other than if we want to take some data from other computer, then we uses the distributed processing system and we can also insert and remove the data from out location to another location. In this data is shared between many users and we can also access all the input and output devices are also accessed by multiple users. Distributed operating system allows the following important services: Data migration, Computation migration and process migration. Advantages and disadvantages of this system are: security, resiliency, scalability, reliability but it has one disadvantage that is very hard to develop such systems or complexity.

Examples-IRIX, DYNIX, AIX, Mach/OS, OSF/01 etc.

4. Single Processor versus Multi-Processor system

Generally a Computer has a single processor means a computer have a just one CPU for processing the instructions but if we are running multiple jobs, then this will decrease the speed of CPU. On a single-processor system, there is one main CPU capable of executing a general-purpose instruction set, includinginstructions from user processes. A single processor system can execute only one process at any point of real time, though there might be many processes ready to be executed.

In contrast the multiprocessing OS have two or more CPU in a single operating system if one CPU will fail, then other CPU is used for providing backup to the first CPU. With the help of multi-processing, we can execute many jobs at a time. All the operations are divided into the number of CPU if first CPU completed its work before the second CPU, then the work of second CPU will be divided into the first and second. A multiprocessor system is one that has more than one processor on-board in the computer. They execute independent streams of instructions simultaneously. They share system buses, the system clock, and the main memory, and may share peripheral devices too.

In multiprocessor systems, many processors can execute operating system programs simultaneously. Consequently, kernel path synchronization is a major challenge in designing multiprocessor operating systems. We need a highly concurrent kernel to achieve real gains in system performance. Synchronization has a much stronger impact on performance in multiprocessor systems than on single processor systems. Many known single processor synchronization techniques are ineffective in multiprocessor systems. Multiprocessor systems need very sophisticated, specialized synchronization schemes. Multiprocessor operating systems are expected to be fault tolerant, that is, failures of a few processors should not halt the entire system, a concept called graceful degradation of the system.

A multiprocessor system can execute many different processes simultaneously at the same real time. However, the number of processors in the system restricts the degree of simultaneous process executions. In single processor systems, concurrency is only achieved in the form of execution inter leavings; only one process can make progress in the kernel mode, while others are blocked in the kernel waiting for processor allocation or some events to occur.

There are two primary models of multiprocessor operating systems:

a. Symmetric Multiprocessor System:

In this system, each processor executes the same copy of the resident operating system, takes its own decisions, and cooperates with other processors for smooth functioning of the entire system. Challenge in symmetric multiprocessor systems is to balance the workload among processors rationally.

b. Asymmetric multiprocessor System:

In this system, each processor is assigned a specific task, and there is a designated master processor that controls activities of the other subordinate processors. The master processor assigns works to subordinate processors.

5. Single Programming versus Multi Programming Operating System

When a single program is allowed to run at a time, the system is grouped under a single-tasking system. Example-MS-Dosis an operating system which performs the single task.

Whereas multitasking is the process of letting the operating system perform multiple tasks at what seems to the user simultaneously. In SMP (symmetric Multiprocessor systems) this is the case, since there are several CPU's to execute programs on - in systems with only a single CPU this is done by switching execution very rapidly between each program, thus given the impression of simultaneous execution. This process is also known as task switching or timesharing. Practically all modern OS has this ability. Multitasking is, on single-processor machines, implemented by letting the running process own the CPU for a while and when required gets replaced with another process, which then owns the CPU.

There are two types of multi-tasking operating system which are:

a. Preemptive Multi-Tasking Operating System:

Preemption means forcefully taking away of the processor from a process and allocating it to another process. Preemption can take place any time, whether the process is in the user space or in the kernel space. Most modern systems allow arbitrary preemption in the user space. (Without such preemption, time-sharing of the processor is not possible.) They may or may not have arbitrary preemptions of kernel executions. In the rest of this section, by a preemptive system we mean a preemptive kernel. An interrupt breaks the current program execution and starts a new program execution in the same process context. Preemption also breaks the current program execution, but starts a new program execution in a different process context. Designers of an operating system must decide whether the system may or may not arbitrarily interleave executions of different processes while the processes execute operating system programs.

b. Non Preemptive Multi-Tasking Operating System:

In a non-preemptive system, the processor executes the operating system on behalf of a process, the processor cannot be arbitrarily preempted (that is, taken away) from the process and allocated to another process. The processor scheduler is called upon only when the current process, running in the kernel space,

voluntarily relinquishes the processor or is about to return to the user mode. In a fully preemptive system,the operating system, in addition, may preempt the processor from a process (in the kernel space) at any point in time and allocate the processor to another process. Many general-purpose operating systems are non-preemptive or partly preemptive, but they implement a few selective preemption points where a process can be forced to relinquish the processor while it is in the kernel space. There are operating systems that claim to be fully preemptive. They are actually partly preemptive because there are some regions in the kernel where kernel executions are non- preemptive. It is substantially easier to design and develop non-preemptive systems compared to preemptive ones.

6. Single User versus Multi User Operating System

Single user type of operating system has to deal with one person at a time. An example of this type of operating system can be found on mobile phones. There can be only one user using that mobile and that person runs one application at a time.

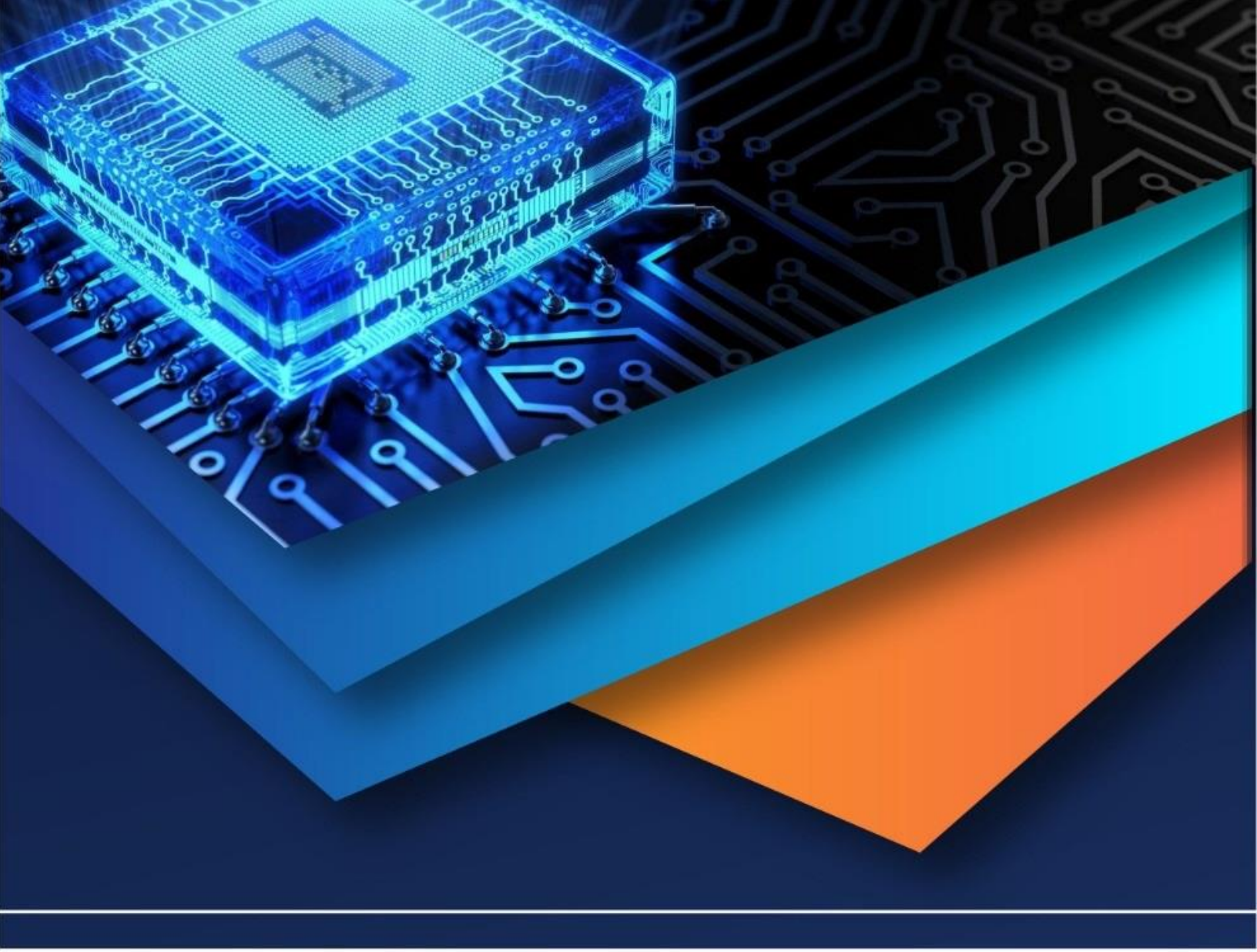
A multi user system is one that can be used by more than one user. The system provides an environment in which many users can use the system at the same time or exclusively at different times. Each user can execute his applications without any concern about what other users are doing in the system .When many users run their applications at the same time. Security is a major design issue in multiuser operating systems. Each user has a private space in the system where he maintains his programs and data, and the operating system must ensure that this space is visible only to him and authorized ones, and is protected from unauthorized and malicious users. The system needs to arbitrate resource sharing among active users so that nobody is starved of system resources. Multiuser systems may need an accounting mechanism to keep track of statistics of resource usage by individual users.

III. CONCLUSION

In this paper we have presented the Heterogeneous Operating Systems with examples. We described the information regarding operating system and issues or benefits of operating system. It is an informative paper about operating system. Given the current state of the operating system market and the research field, GUI may be used to provide a bridge between both fields and promote the development of more flexible and cooperative operating systems. This would provide system administrators and programmers with the flexibility needed to develop user-friendly operating environments and applications that are notlimited by the choice of a single operating system.

REFERENCES

[1]Dixit.J.B, Fundamentals of computers and programming in c, Laxmi Publication pvt.Ltd., second edition 2010.
[2]Sharma.A.K., Fundamentals of computers and programming in c, DhanpatRai Publications, 2013, pp-7.50-7.67
[3]http://en.wikipedia.org/wiki/Operating_system
[4]http://whatis.techtarget.com/definition/operating-system-OS
[5]http://www.webopedia.com/TERM/O/operating_system.html
[7]http://ecomputernotes.com/fundamental/disk-operating-system/time-sharing-operating-system
[8]http://en.wikipedia.org/wiki/Distributed_operating_system
[9]http://ecomputernotes.com/fundamental/disk-operating-system/distributed-operating-system



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY