Operating Systems: Basic Concepts and Challenges against Virtualization

Abhishek Bhattacharya
Student
Amity School of Engineering and Technology
Amity University, Noida, India

Shagun Paul
Student
Amity School of Engineering and Technology
Amity University, Noida, India

Arunima Jaiswal
Asst. Professor
Amity University, Noida, India

ABSTRACT
Operating systems act as fundamentals to the basic computer systems in today's world. An OS is the communication between the software you use and the hardware that lies underneath. Every electronic signal talks with your operating system to understand the work it has to do or the way it has to behave to get the user's work done.

In this paper, we discuss about basic operating systems, get an overview of the internal processes associated with an operating system and understand its connection with the kernel and the hardware. We then move on to Virtualization by introducing Vagrant, an operating system virtualization software that allows you to run virtual machines on your system and configure them the way you want to suit your needs, and discuss about virtualization on cloud.

Keywords
Operating System, Virtualization, Cloud, Vagrant

1. INTRODUCTION
Microsoft Windows, right? We are talking about the software that talks to your computer's hardware, and is most efficient in establishing a connection between you and your computer.

The operating system makes your computer understand you. And, it does it the most brilliant way. The electronic circuits that you see on a motherboard cannot communicate with you to understand the signal you want it to send. You need an operating system for that job.

So, again, Microsoft Windows? A lot of operating systems have emerged till today. We have more names - Linux, MacOS, Jolicloud et al.

Basically, a definition can be stated by saying about an operating system to be a system that lets other programs run.

Another definition would be it being a system that provides a very controlled access to the resources of a system.

What do we mean by resources?

Resources refers CPU (the scheduling of processes), memory (for memory management), display, mouse, keyboard, storage, networks et al.

You wouldn't want a single process to consume all of the resources available (for eg., all the CPUs or the entire available RAM).

So, by this, we mean - Yes! We're talking about the Microsoft Windows running on your computer. But, we do have eggs in different plates!

2. LITERATURE SURVEY
There has been tremendous research in the field of operating systems, many of which have formed the basis of my work.

3. SUMMARY OF LITERATURE SURVEY
Table 1. Author wise issues addressed in their papers.

Author's Name	Issue Addressed
AS Woodhull et. al. [1]	Explains basic concepts of an operating system.
MA Harrison et. al. [2]	Discusses 'security' in operating systems based on access rights.
J Ousterhout et. al. [3]	Comparing performance of operating systems to the hardware.
AS Tanenbaum et. al. [4]	Securing operating systems and increasing reliability.
S Loannidis et. al. [5]	Sub-operating systems. How files become 'active' as 'code' under certain circumstances.
H Chen et. al. [6]	Live updating of operating systems using virtualization
P Barham et. al. [7]	Talks about Xen and the art of virtualization by linking applications explicitly against n Ilwaco guest OS.
DA Menascé et. al. [8]	Objectifies performance modeling and discusses concepts and applications of virtualization.
JY Hwang et. al. [9]	Explains implementation system virtualization using Xen hypervisor on ARM-based secure mobile phones.
M Hashimoto [10]	Provides a detailed description of a virtualization software.

4 UNDERSTANDING THE COMPUTER SYSTEM ORGANIZATION

Before delving into operating systems, we should understand the computer system organization, i.e., how the computer system is organized in its truest sense and how things work.

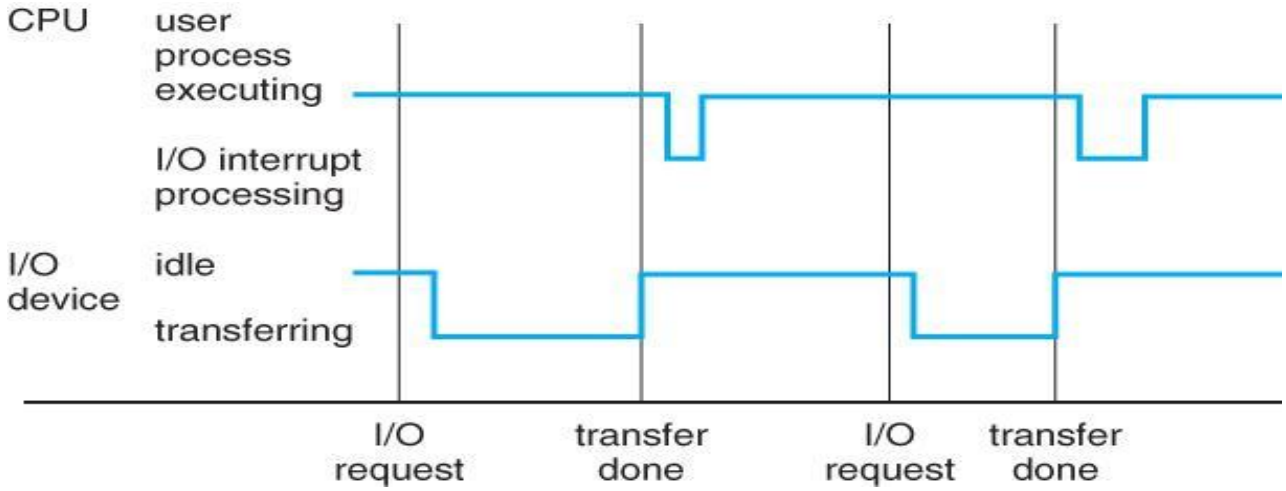

Fig. 1. AS Woodhull [1]

Have you heard of Intel i5 as a quad-core processor? That means the processor, Intel i5 has four different CPUs on a single integrated circuit (IC), and we call it a *multi-core processor*. Your *Micromax Knight* is a multi-core processor with eight cores!

What do we mean by multiprocessing? This means, whenever a process is to be handled by your processor, if its a quad-core processor, four CPUs are going to share the work required, which obviously means that the computation time would be less. - *So, do you now get a reason why your Intel Pentium 4 processor hanged more than your Intel i5?*

The Intel Pentium 4 was a single-core processor and the entire burden was always put on the only core present. This gives you an idea where we're advancing to.

Going deeper, a computer system consists of CPUs and controllers connected through a common bus which provide access to shared memory.

What the device controllers do is take care of the working of specific devices - monitor, graphic cards, audio devices et al.

Now we move on to understanding the needs of a computer as it starts, or is rebooted.

On startup, a computer needs instructions to follow so that it can properly start and detect and load required resources and programs. [1]We call this a ***bootstrap program***. This program is stored in the ROM or the read-only memory that's electrically erasable and programmable (EEPROM). The generic term we give to such type of memory is ***firmware***.

[1,3]A very important phenomena for a computer system is an ***interruption***. This can be from a hardware or a software, whereby a hardware might signal the CPU to trigger an interrupt, using the system bus, and the software can trigger an interrupt using a *system call, or a monitor call.*

What happens when an *interruption is triggered?* The CPU immediately stops whatever it has been doing and shifts execution to a location where the instruction is primarily located.

4.1 Why multiprocessor systems?

[1]Multiprocessor systems, or parallel systems, or tightly coupled systems, have two or more processors which tend to be in close communication, and share the system bus and memory, devices et al.
Multiprocessor systems and their advantaging areas:

Throughput: [1-3]This is getting more work done in less time. As we discussed earlier, presence of more processors distributes a single process and makes the computation faster. However, there's a simple drawback; an increase in processors doesn't deliver an equivalent increase in throughput, just like N programmers wouldn't do the job equivalent to the total work of a single programmer times N.

Scale Economy: Multiprocessor systems definitely cost less than equivalent multiple single core systems because there's a sharing of the resources which reduce the cost considerably.

Increased Reliability: When processes are distributed properly, failure of a particular processor wouldn't halt the working of the system.The remaining processors can easily catch up by being minimally slow.

As *Steve Jobs* has said regarding the hardware of your computer and the operating system:

"*There's no other company that could make a MacBook Air and the reason is that not only do we control the hardware, but we control the operating system. And it is the intimate interaction between the operating system and the hardware that allows us to do that. There is no intimate interaction between Windows and a Dell notebook.*"

4.2 Dual Mode Operation of an Operating System

By dual-mode operation, we refer to a computer's ability to differentiate between user and OS code; and this is generally done by providing hardware support.

There are two types of modes: *user mode* and *kernel mode.* The current mode is indicated by adding a **mode bit** to the hardware, representing kernel by 0 and user by 1.

Following diagram shows the *transition from user mode to kernel mode:*

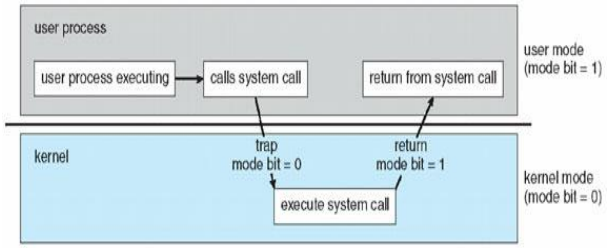

Fig. 2. AS Woodhull [1]

Initially, the operating system has the control and instructions are executed in kernel mode. When control is passed to a user application, the user mode comes in. Whenever a trap or a system call is faced with, the control is again shifted to the operating system, and it is done by setting the mode bit to 0 again.

4.3 Memory Management

[1,5-7]Main memory is an array of thousands of billions of words or bytes, corresponding to individual addresses, and allows a quick access to data required by the CPU and the I/O devices.

The CPU reads the required information from memory during a cycle we call instruction-fetch cycle and does both reading and writing during a data-fetch cycle, generally on a Von Neumann architecture.

Memory management is required of the operating system due to the need of keeping several programs in memory so as to increase the utilization and the speed of the CPU.

Regarding memory management, an OS is responsible for the following:

- ➔ Tracking the memory parts being used.
- ➔ Deciding about data's moving in and out.
- ➔ Memory allocation and de-allocation.

4.4 System Calls

By means of system calls, an interface is provided to the services made available by an operating system. An operating system executes thousands of system calls per second. Almost every action by an operating system is a system call. For instance, each and every read and write actions are separate system calls.

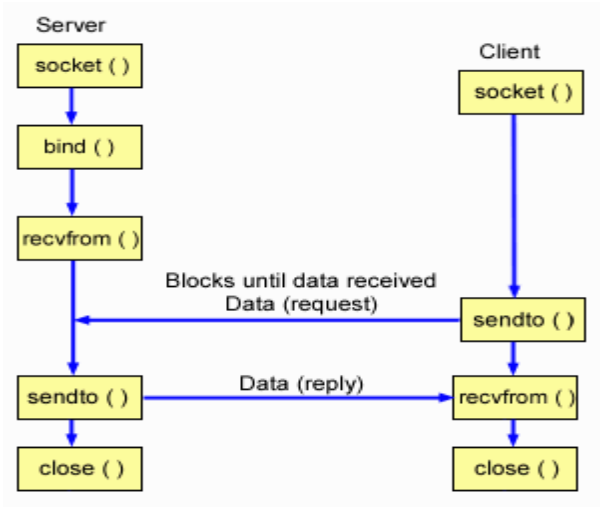

Fig. 3. S. Loannidis [5]

4.5 Operating System Services

While providing the programs with an environment to work, it provides programs and it's users with certain services, most of which are common to all the operating systems.

We'll look at some of the most common services:

[1,7-8]User Interface: An UI is provided by all the operating systems, being either a character user interface, or a graphical user interface.

Program Execution: A system got to be able to load programs to its main memory and execute them for the users.

I/O Operations: A lot of the programs require some sort of input from the user, as an example. Programs generally require input-output operations, hence, an operating system must be able to provide the basic I/O operations.

File-system Manipulation: Programs need the ability to create, modify, delete or store in files in the system. This is again a very basic service every operating system provides to the users and the programs.

Communication: Interaction and message-transfer between programs is required because two different programs running on a computer might require to communicate between them to exchange critical information.

Error Detection: Any sort of disruptive circumstance might occur while a program is executed. For example, while you create a task of printing, the paper might get finished in the printer which would cause the task to be aborted. The user should get to know about such a problem so that he can rectify it. This is done by error detection by an operating system.

Similar errors, like memory problems et al are intimidated to the user by error detection.

5 DIRECT MEMORY ACCESS

[1]Disk drives do large transfers. Such areas of work waste the resources of an expensive general-purpose processor. This is because a processor needs to handle a lot of tasks and processes all through, and burdening the processor with the task of transferring data which just requires keeping track of

the source and destination addresses and the bytes, implies wasting the valuable resources.

Hence, for handling programmed I/O (PIO) operations, computers have a direct-memory-access (DMA) controller which acts as a special-purpose processor.

This process is started by the processor writing a DMA instruction block into the main memory. This command block contains the three things required; pointers to the source and the destination addresses and the number of bytes to be transferred.

What happens is, the CPU only has to write the address of this block to the DMA controller, and then go onto other works.

DMA-request and DMA-acknowledge are a set of wires that help perform the handshake between the DMA-controller and the device controller.

The DMA controller interrupts when the transfer is finished. This can slow down the CPU computation, but it has been observed that use of a DMA-controller considerably increases the throughput.

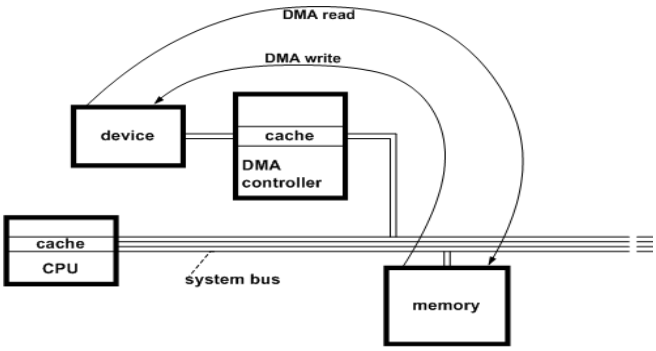

Fig. 4. AS Woodhull [1]

6 PROTECTION AND SECURITY

[4,6]Everything in a computer system is based on and completely vulnerable toward various levels of access.

By access, we refer to the file and memory permissions provided to programs. This is most exploited when in user mode, that is, mode bit being set to 1.

Security system of an operating system protects data from unauthorized access, destruction of memory and inconsistency.

6.1 Access Control

[1,3-4]Each file and directory in a file system, is assigned a group or a particular owner who have access to the file based on their level.

[3,4]Solaris 10 is a great example that advances the protection provided by Sun Microsystems by adding the principle of least privileges via *RBAC, i.e, role-based access control.*

This facility revolves around the concept of privileges. A privilege is a right to execute a system call.

Privileges are assigned to processes based on their requirements so as to restrict improper usage, or what we call, *escalation of privileges*.

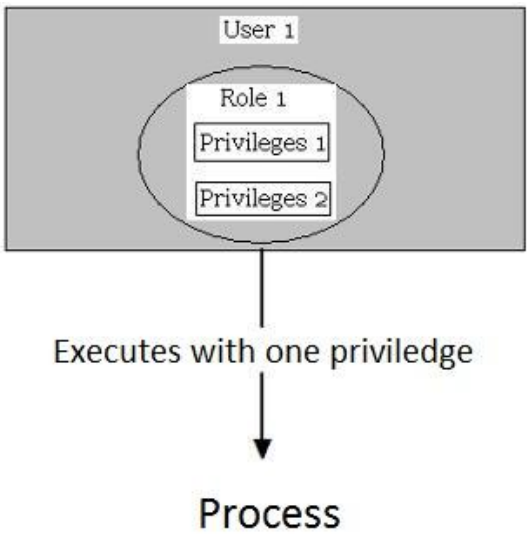

Fig. 5.

7 VIRTUALIZATION

[8-9]The basic concept of virtualization is a level of abstraction that defines the *Virtual Machine Monitor* is added below the Operating System abstraction. This has the capability to host a lot of OSs along with controlling access to the hardware resources.

Virtualization in an x86 architecture:

Privileges are provided in four levels: rings 0 up to 3.

In a general, non-virtualized environment, the applications run in ring 3 while the OS runs in ring 0.. This gets changed in a virtualized environment where the Virtual Machine Monitor runs at ring 0, the guest-OS at ring 1 and again the applications at ring 3.

7.1 A Vagrant Instance

[10]Vagrant is a tool with which we build development environments that are totally complete. With a very acute focus on automation, the work becomes way too easy.

We're going to work with Vagrant on Virtual Box because it's free, available on every major platform and built-in into Vagrant. We'll install Virtual Box and Vagrant, and get it up and running.

7.1.1 Vagrant – Up and Running

```
$ vagrant init hashicorp/precise32

$ vagrant up
```

[10]Just these two commands would get you a fully running virtual machine in Virtual Box running Ubuntu 12.04 LTS. You can ssh into this machine with **vagrant ssh,** and when we're done playing around, we can just delete it by running **vagrant destroy**.

You need a directory to be set up to start working with Vagrant, for which Vagrant has a built-in command:

```
$ mkdir vagrant_getting_started
$ cd vagrant_getting_started
$ vagrant init
```

Fig. 6. VagrantUp [10]

That's it. That's all you need to get yourself up with Vagrant. For an elaborated guide, a reference to their website would be useful.

8 CHALLENGES TO VIRTUALIZATION

[7,9]I've managed to find and discover quite a few *challenges* to virtualization which I am now going to discuss.

- Availability and performance issues due to depletion of resources.

  When you move from physical to virtual hardware, issues regarding performance and availability of your data become prevalent. Virtualization translation layer slows down I/O intensive operations.

- Lack of virtualization in terms of application
  During migration from physical to virtual, app availability is certainly lost as virtualization isn't provided for applications.

- Virtualization features are left unused
  Services such as software switching and support for VLAN segmentation. They are not linked with the Application Delivery Network because there is no information sharing between the network and the VM.

- Storage network is overrun
  Data files and the operating system resides on shared storage in virtual environments, while they are stored on internal storage in physical environments.

- Rapid increase in files
  Virtual environments happen to exceed the number of files and the size while they're stored on the shared environment.
  Furthermore, the operating system images happen to be cloned leading to rarely used images.

- Data pipes are not able to handle the volume
  Storage network gets congested due to a dramatic increase in storage and passing of large amounts of data from different guests all of which causes flooding, bottlenecks and congestion.

- Complexity in management
  Management tools in don't work together in the virtual environment. Data center management are different components in hypervisor and the host system which don't go together.

9 CONCLUSION AND FUTURE SCOPE

All this was a basic introduction to Operating Systems and virtualization in OS. We use an operating system entirely when we use "any" sort of system that has some interface or not, has user interaction or not, has a properly built UI or not. An OS interacts on our behalf with the hardware which is of utmost importance when we're working with various devices. We talk about virtualization because it's an evolving field where work is yet to be done. The future lies in a more integrated approach in terms of processing, devices controlled by high end operating systems and a complete virtualization of the platforms we would be using regularly.

10 REFERENCES

[1] AS Woodhull 1992. Operating System Concepts

[2] MA Harrison 1976. Protection in Operating Systems

[3] J Ousterhout 1990. Why aren't operating systems getting faster as fast as hardware.

[4] AS Tanenbaum 1987. Operating Systems: Design and Implementation

[5] S Loannidis 2002. Sub-operating systems: A new approach to application security.

[6] H Chen 2006. Live updating operating systems using virtualization.

[7] P Barham 2003. Xen and the art of virtualization

[8] DA Menascé 2005. Virtualization: Concepts, applications and performance modeling.

[9] JY Hwang 2008. Xen on ARM: System virtualization using Xen hypervisor for ARM-based secure mobile phones

[10] M Hashimoto 2013. Vagrant: Up and Running