Comp 314: Graphic Design
ID: 3441988

Andrew Mackay

# Operating Systems Assignment 1

1.

Von Neumann architecture accounts for the fundamental structure of most computers, and has to do with how process instructions and data are stored in the same main memory. Data is fetched from primary memory and brought into the CPU/instruction register within the Von Neumann architecture modal. From there the instruction is decoded and may cause operands to be fetched from memory and stored in some internal register. Reading and writing of the data to and from the main memory is done on one data bus and fetching instructions for execution is on secondary memory. The processor has the ALU that can implement the basic functions and operations and it has the temporary storage that can store only a few words of data. The memory unit sees only a stream of memory addresses. It does not know how they are generated or what they are for . This cycle is iterated repeatedly until the task is completed.

2.

Drivers are a form of software that aids hardware's functioning and performance. For every driver there is an associated controller. When initiating an I/O operation, the device driver loads the appropriate registers in the device controller. The device controller then examines the contents of these registers to then determine what action to take next . The controller then begins to transfer the data from the device to its local buffer. Once the data transfer is completed, the device controller then informs the device driver that it has completed the operation. The device driver will then give control to other parts of the operating system. (boron.physics,2022). Trap is an interrupt that is caused by software. This is what software like drivers use to interrupt a process and drivers use this to move memory temporarily.

3.

A device controller is the I/O managing processor within a device whereas the driver is An operating system component that provides uniform access to various devices and manages I/O to those devices, all external devices and internal devices used with a computer have drivers to help them function more optimally.

The movement of the data between the peripheral devices and its local buffer storage is done by the device controller. The device driver is a software that runs the device. The device driver acts an interface between the device controller and the operating system. The device is connected to the computer when the device driver is installed on the computer. An example of the device driver is the USB driver that is installed in the computers through which the user and the system can communicate with the device. Any type of controller(gaming) or audio interface, or any external feature will have a device driver that will be downloadable, and will also have updates over time.

4.

The Kernel is the core of the system and without it the computer would not function. The kernel stores and organizes a lot of information. So it has data about which processes are running in the system, their memory requirements, files in use etc. To handle all this, three important structures are used. These are process table, file table and v node/ i node information.

The clustered system provides the high availability service because in this system a group of hosts provide efficient system by acting as a single system. These clusters are used for load balancing, back up, high availability service because it helps to achieve an enhancement in processing power. In clustered system, the chance of having a single point of failure is eliminated.Depending on what operation needs to be done an OS can switch between both user mode and Kernel mode for different tasks.

5.

The layered approach breaks up the operating system into different layers. This allows engineers to change the inner workings. Within the layered approach, the bottom layer is the hardware, and at the highest is the UI. The main advantage is within the complexity of the system it simplifies our understanding and can help pinpoint issues and simplify construction as well. “All the layers can be defined separately and interact with each other as required. Also, it is easier to create, maintain and update the system if it is done in the form of layers. Change in one layer specification does not affect the rest of the layers.” (javatpoint,2022) This ties into the Microkernal where all non essential elements are stripped/structured and implemented as system and user level programs.Within the monolithic structure functionality of the OS is invoked with simple function calls within the kernel, which is one large program .Device drivers are loaded into the running kernel and become part of the kernel. What makes the microkernal appealing in system design is that while extending the operating system becomes easier, it is more secure and compartmentalizes things in a more understandable way .

6.

System calls are performed by the kernal and are treated differently than functional calls. The parameters can be passed in registers directly. Sometimes there can be more parameters than registers and this can prove to be an insufficient method. Secondly parameters can be stored in a block or table within memory, then passed to the address block as a parameter within a register. The last method would be to push the parameters onto a stack which is popped off by the operating system. (umass, 2022)

7.

The linker and loader are parts of a program execution process where the source program is originally stored as binary code, is taken to be compiled into an object file which will then be further manipulated by the linker and loader. Linking and loading is also associated with Relocation, which is an activity that matches final addresses to program parts and adjusts code and data in the program.

A linker is an important utility program that takes the object files, produced by the assembler and compiler, and other code to join them into a single executable file. The linker takes the object file and turns it to an executable file which then the loader will convert to a program in memory.

A loader is a vital component of an operating system that is accountable for loading programs and libraries. The loader takes the binary executable file and loads it into the program memory, where it is only then possible to be run on a CPU core.

8.

The difference between mechanism and policy can be thought of in terms of the mechanism being how something will be done, while policy determines what will be done. Separation of Mechanism and policy is in essence saying that user access or lack of access should not inhibit the process of decisions on which operations should be made. Two level implementation is used by operating systems to separate the mechanism from policy. The separation from mechanism and policy is also what distinguishes a microkernal from a monolithic one, which implies this technique is common in most modern computers and interfaces. This is important for flexibility as policies and mechanisms change and are reliant on each other, so if one changes so must the other unless there is the separation of policies.

9.

Within the operating system there are built in techniques of monitoring performance such as counters, per-process reports, and system-wide reports. Besides for counter based tools there are also tracing tools which are event specific inquiries and include Per-process tracing, and System-wide tracing.

Besides from these there are additionally more user accessible downloadable programs that monitor the operating systems performance and data logs, one of these a built in program called “Task Manager”. Real time monitoring tools are concerned with measuring the current system state and provide up to date information about the system performance. Log-based monitoring tools record system performance information for post-processing and analysis and to find trends in the system performance. (cse, 2022).

10.

Virtualization is a technology that creates an abstract layer over the hardware of a computer into several different execution environments, which creates a virtual separate environment that makes it seem like it is its own is running on its own private computer, each of which have their own sort of Operating System. These computers are typically referred to as “Virtual Machines”(IBM, 2019). They each are thought of to also have their own operating system. . Virtualization allows operating systems to run as applications within other operating systems. At first blush, there seems to be little reason for such functionality. To run something like Java programming you need the Java virtual machine installed on your computer to compile and run the code.Within the scope of virtualization is Emulation, which involves simulating or imitating computer hardware within software, and is typically used when the source CPU type is different from the target CPU type.

11.

The two modes of the operating system are Kernal mode and User mode. The mode in which a user application requests a service from the OS through a system call, is called a kernel mode whereas the mode in which the system executes on behalf of a user application is called a user mode.

The kernel mode is represented by 0 while user mode is represented by 1 . The mode bit, is added to the hardware of the computer to indicate the current mode: kernel “0”or user “1”. With the mode bit, we can distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user. When the computer system is executing on behalf of a user application, the system is in user mode. However, when a user application requests a service from the operating system (via a system call), the system must transition from user to kernel mode to fulfill the request.

12.

“API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses”(AWS,2022). A typical API is code that is called upon through front end services, like in the header of an html file, which allows cross communications between applications. Operating system API’s can be referred to sometimes as “Toolkits” and are built into the system. System call provides the services of the operating system to the user programs via Application Program Interface(API). It provides an interface between a process and operating system to allow user-level processes to request services of the operating system. System calls are the only entry points into the kernel system.

Essay questions:

1.

As we know the Operating system acts as an interpreter between the computer hardware and program applications.

If we consider a simple program such as a instant messenger system like AOL, MSN in pseudo code it might be written something like 1.Create chat directory 2.Establish GUI framework, integrated with sidebar list of contacts. 3. Establish connection with server program/internet to connect to contacts 4. Enable/ Get input from User to send to other contact or contacts within list/create and or connect back end. (getstream, 2022).

The app that I am proposing is not competitive with todays apps, but more-so a throw back to a simpler time therefor doesn't require quite as much planning on all ends but still is not an overly easy task to create either.

The GUI interacts with the operating system directly but keeps a more aesthetically user friendly interface than a program written to run in the command prompt or an IDE text editor. Contacts have stored information associated with them such as display name, web address, and possibly additional information like phone number or home address but that data is only accessible if they are prompted to accept your friend request and upon them doing so it becomes available through clicking their display name. The search bar needs to return values within the list specifically, but also have an alternate search bar with the add a contact feature that can search more broadly outside the contact list for the options to add a new contact.

Obviously this is a vastly scaled down version of what it would take to make a instant messenger but is a project I am interested in so nonetheless am trying to get my bearings on how to go about doing it without saying anything too specific and just keeping it general.

Von neuman architecture defines a computer consisting of a central processing unit, control unit, internal memory or Ram to store data and instructions and external memory. It contains several registers that would ease out the computation process.The processing unit is the core of the system and it does all kinds of logical and arithmetic computations. The program counter is used to store the address of next instruction to be executed by the system. This is how the constructions for this code would be carried out and processed in this order. The main components being the Control Unit, which It takes care of the correct implementation of the tasks and their completion time. The Processing Unit , which deals with the Computation of info. Memory, Information Storage. Input, which is to input information into the computer. Output, what is displayed as a result of the input.
Memory and the processing unit use the Memory Data Register(MDR) and Memory Address Register(MAR) to communicate. The memory address register or "MAR" holds the address of the current instruction which will be fetched from memory, or the address in memory in which data is to be transferred. The memory data register or "MDR" holds the contents found at the address held in the MAR, or data which is to be transferred to primary memory.

So any input as a message is sent to another client which triggers the input output component of Von neuman architecture. After they click enter it initiates a sequence of passing instructions to the central processing unit which is then interpreted and passed along and loaded into the internal memory then the instructions are stored in RAM and passed to registers which message is relayed to the internet server in which the application has to be connected to for contacts to receive messages. If there is no connection to the internet even the input of the info itself triggers this process and makes the message still output by it appearing in text as the user is writing, which isn't the ultimate goal of the program, but it is still output according to computer science as the message is displaying on the screen within the textbox.

2.

The system call is a software triggered interrupt which then enables a process to to request a kernal service. A system call always refers to comminication from a procedure to the kernal, these procedures could be executing procedures or equipment/hardware related. Applications needs RAM and IO to function, these are provided by system calls. System calls are used when a program wants access to some hardware or software which is controlled by the operating system. It isn't often for an application program to directly access system hardware. What happens is that it calls interface routines which are carried out by the operating system. There are various reasons for this. First, even though the hardware varies, the OS can mask the differences, so that application programs don't have to support a huge range of different hardware for the same purpose. Secondly, when you have multiple applications all trying to use the same resources, the OS can control access to the resources, and resolve access conflicts. When the user application requests for a service from the operating system or an interrupt occurs or system call, then there will be a transition from user to kernel mode to fulfill the requests.

The C standard library is simply a library of prewritten C language functions that when downloaded can be referenced as shortcuts. The functions are optomized for performance, it saves considerable development time, and the functions are portable.(programiz, 2022) The C programming language does not require a virtual machine like Java, and C#, so when compiled it is going to be directly handled by the host operating system.

Unix write systems have API, which is an actualized portion of C library such as glibc. We can obtain the complete list of processes by using the "ps" command. A library or API is given by the systems that cooperate between an operating system and ordinary projects. This furnish system calls with wrapper capacities which uncover a normal capacity calling tradition for utilizing the system call and make it measured. The wrapper's primary capacity is to put every one of the ions in the fitting processor registers which are to be passed to the system. Therefor the library, which is between the OS and the application, builds portability. Control is exchanged to the kernel by the real system call as in Unix-like systems, fork and execve system calls are utilized by execve C and fork library capacities. A change to kernel mode isn't caused by the call to the library work and is usually a subroutine call. So making the system call within in the application's code isn't doable.

A shared memory object is created for the "Hello world!" program code, the program memory-maps a shared-memory object of the specified size which then allows writing to the object. The "MAP_SHARED" flag specifies changes to the shared-memory object will now be visible . We write to the shared-memory object by calling the sprintf() function and writing to the ptr. (pointer). After each write the pointer is incremented by the number of bytes written. The consumer process, reads and outputs the contents of the shared memory. Similarly, exceptions generate an interrupt too. If the operating system has a plan for interrupt handling, then the operating system can handle system calls and exceptions too.

3.

A single operating system is used on majority of computers, but there exists a technology called virtualization which can also be known as virtual machines, that enable someone to switch between multiple operating systems.A virtual computer has its own RAM, hard disc, and processor, among other things. This is essentially a separate computer within a real computer that works on shared physical hardware resources (RAM, processor, and hard drive). The software used is called a hypervisor — a small layer that enables multiple operating systems to run alongside each other, sharing the same physical computing resources. When a hypervisor is used on a physical computer or server (also known as bare metal server) in a data center, it allows the physical computer to separate its operating system and applications from its hardware. Then, it can divide itself into several independent "virtual machines."(ibm.com,2022)

There is a difference between cloud computing and virtualization in the sense that while virtualization may use cloud infrastructure or container software, it is different in the way that cloud computing has a pool of resources known as an environment, that aren't physically stored anywhere but exist in the cloud. In virtualization, software called a hypervisor sits atop/connects to the physical hardware and creates a virtual environment called a virtual machine. This doesn't use cloud technology, but if the virtual resources of the virtual machine are pooled in a cloud then it is considered a cloud operating system(redhat, 2022).

Containers provide an alternative to virtual machines although are similar in nature. Rather than spinning up an entire virtual machine, containerization packages together everything needed to run a single application or microservice (along with runtime libraries they need to run). The container includes all the code, its dependencies and even the operating system itself.(ibm.com,2022)

The main benefit is running applications native to other operating systems that will not run on your original OS, using virtualization to switch to a OS that does run that application and running it that way. This is mainly an issue with compatibility between applications and operating systems, where it is erased with this technology. Cloud computing uses virtualization as its base but there many types of cloud computing and when that term is referred to it is usually referring to something different than a cloud operating system, although virtual machines can be called that as well.

The performance of the virtual machine is relative to the host operating system, although there are certain tasks which are preferable to execute through a virtual machine, furthermore a virtual machine which operates on cloud computing has a different set of attributes comparatively.

With cloud virtualization you can deliver the virtual machine resources to a group or team of people, but with a virtual machine it is associated with a single user, although virtual machines will have a longer life span if not functioning on a cloud as cloud lifespan of storage is more short term. Non cloud based virtual machines are stateful, which means they require backup storage whereas cloud based virtual machines are stateless meaning they don't require backup storage(redhat,2022).

So it may come down to what you are trying to achieve, from a business at large or from a single users personal computer. Its hard to imagine a use case for a cloud operating system for  a teenager who uses their computer for  writing high school level essays, talking to their friends online, and playing video games. Whereas if a entire business had many employees who were working on complex software related issues within the company system they may benefit from a cloud operating system, also it is worth keeping in mind exactly what we mean by “cloud computing” as it can be somewhat ambiguous.

## References

-(2022). Retrieved 2 June 2022, from https://online.vitalsource.com/reader/books/9781119320913/epubcfi/6/16[%3Bvnd.vst.idref%3Dc02]!/6[c02-body-0157]

-C Standard Library Functions
C Standard Library Functions. (2022). Retrieved 8 July 2022, from https://www.programiz.com/c-programming/library-function

-Containers vs. Virtual Machines (VMs): What’s the Difference?
Containers vs. Virtual Machines (VMs): What’s the Difference?. (2021). Retrieved 9 July 2022, from https://www.ibm.com/cloud/blog/containers-vs-vms

-What is an API? - API Beginner's Guide - AWS
What is an API? - API Beginner's Guide - AWS. (2022). Retrieved 12 July 2022, from https://aws.amazon.com/what-is/api/#:~:text=API%20stands%20for%20Application

-How to Build a Chat App: The Ultimate Guide

How to Build a Chat App: The Ultimate Guide. (2022). Retrieved 12 July 2022, from https://getstream.io/blog/build-chat-messaging-app/

-WHAT'S THE DIFFERENCE BETWEEN CLOUD AND VIRTUALIZATION?
In-text: (What's the difference between cloud and virtualization?, 2022)
What's the difference between cloud and virtualization?. [online] Available at: <https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization> [Accessed 13 July 2022].

Operating System and Process Monitoring Tools
Operating System and Process Monitoring Tools. (2022). Retrieved 27 July 2022, from https://www.cse.wustl.edu/~jain/cse567-06/ftp/os_monitors/

I/O Structure
I/O Structure. (2022). Retrieved 10 August 2022, from http://boron.physics.metu.edu.tr/ozdogan/Operati

West-Eberhard, M.
West-Eberhard, M. (2008). Phenotypic Plasticity. Encyclopedia Of Ecology, 2701-2707. doi: 10.1016/b978-008045405-4.00837-5

Education, I.
Education, I. (2019). virtualization-a-complete-guide. Retrieved 10 August 2022, from https://www.ibm.com/cloud/learn/virtualization-a-complete-guide

Layered Structure of Operating System - javatpoint
Layered Structure of Operating System - javatpoint. (2022). Retrieved 13 August 2022, from https://www.javatpoint.com/layered-structure-of-operating-system

Anon
(2022). Retrieved 13 August 2022, from https://lass.cs.umass.edu/~shenoy/course