Assignment 1

Saad Saiyed saadsaiyed7@gmail.com
200415258

PART A:

1. Define the concepts interrupt and trap, and explain the purpose of an interrupt vector.

Ans: For a software to run there are some routines that happens without any exceptions. For some exceptions traps are used to pause the program before running and this normally occurs when someone tries to exchange control over different subroutine in the process. Same goes for an Interrupt, I/O hardware devices invoke it. Every time there is some event with I/O device, it shows the presence of itself to the process via Interrupt. Interrupt vector is a variable which holds the position or memory address of Interrupt handler. Traps are software base exception handler (e.g. division by 0) and Interrupts are hardware base invokers.

2. How does a computer system with von Neumann architecture execute an instruction? Ans: Von Neumann architecture is a design of a computer which consists of Memory storage, ALU (Arithmetic/Logic Unit), CU (Control Unit) and Input/Output. In this architecture instructions are executed sequentially. Von Neumann Bottleneck is type of connection or transfer of data which is performed throughout the process. It is a process which has one way data bus which takes the information and returns with some results. Even if the CPU is more powerful, von Neumann architecture works with one way data bus. ALU performs basic calculation like addition and subtractions and Boolean logics. The data here is volatile and stored temporarily. The size is also small compare to main storage.

3. What role do device controllers and device drivers play in a computer system? Ans: Device controllers handles all the data transfer between the main data storage and data from the device. Device driver is a layer between the Operating system and Device Controller. So if we press any key on the input device, keyboard the data from the particular key goes to device driver through device controller and then from device driver to Operating system and then to response center (all other hardware stuff in CPU) to receive a response. So we can say that only when device drivers are installed in a computer then we can say that the device is connected to the system. Best example is speakers when connected to the ports we need to install something like Dolby Atmos Drives to our system and only then we can use the speaker.

4. Why do clustered systems provide what is considered high-availability service? Ans: In electric circuits the reason why we have parallel connections is the same reason why we have high-availability service in clustered systems. The reason is, if for some reason one of the point fails we have backup and that way we don't have to reboot the system. Each points in clustered system acts as an individual system which are called hosts. These clusters are available for backup and also used for balancing load if one cluster is getting

overloaded. These increases the efficiency of the process and hence clustered systems provide high-availability service.

5. Describe an operating system's two modes of operation. Ans: Two modes in Operating System are User Mode and Kernel Mode. Using these modes rather than one helps us to create extra layer of protection to hardware system. a. Kernel Mode: Operating System runs on Kernel mode. In Kernel Mode, software has full access to all the computer's hardware. All the request of configuration and usage of hardware from any program is reported to Kernel and then processed. b. User Mode: Everything processed by an application is done in User Mode. It has allocated memory which is displayed to no one but itself and is enough to work along all the processes. Any actions to covert the modes are reported to kernel and that way any malicious code in an application can be restricted to do changes to computer's hardware.

6. Define cache, and explain cache coherency. Ans: Cache is a small capacity memory storage which is closer to PC (Processor Core) to increase the efficiency of the program. If a program needs to access some memory address again and again from the main memory, cache is used to store that memory temporarily. Almost all CPUs have independent Cache memory to store duplicate memory in the hierarchy manner. Cache Coherency is mostly useful for multiprocessing systems which have multiple independent cache memory which sometime store same memory multiple times. If one of the CPU updates the value of the cache in its cache memory, the information is passed on to another CPU's cache memory to update the memory. This is done using Cache Coherency.

7. Describe why direct memory access (DMA) is considered an efficient mechanism for performing I/O. Ans: DMA is used to transfer data from main memory to device and so forth to decrease the load to CPU and reduce time to operate any I/O device. If a CPU is running and is busy, I/O device does not need to wait for it to access the memory; rather it can use DMA. That way, any request made by I/O devices are very efficient to get response.

8. Describe why multicore processing is more efficient than placing each processor on its own chip. Ans: There are multiple reasons for multicore processing to be more efficient then placing each processor on its own chip, which are: • Pricing: adding another slot for new chip and producing new chip is expensive in compare to adding a core to one chip making it multicore processor. • Heat control: There is no difference in heat control between multicore and single core processor but advantage of multicore is that you extract more power from multicore processor using less exchange of heat. • Clock Speed: We can increase clock speed of one chip drastically using multiple cores which is not the case for single core chips.

9. Describe the relationship between an API, the system-call interface, and the operating system. Ans: API is the set of programs or functions which are present as an external sources (e.g. google API). Which then communicate with Operating System using System Call. System call is mode of communication between any request a program (API) made to use computer hardware. But there is one layer between the hardware and program which is OS. So System call talks to Kernel. After the response from hardware OS sends the response to program (API) via System Call.

10. Describe some requirements and goals to consider when designing an Operating System. Ans: One of the main reason we have Operating System is to have a layer of protection between User (program) and Computer Hardware. So first thing any operating

system must have a way to communicate with memory and hardware devices and transfer data to and from system calls. Also it should be able to grant the specific access to CPU if user or I/O device requests for. Before designing OS you need to make sure all the process works fast and effectively. The access to main memory should be efficient too as that would be the first part of any software before running anything. Sorting which operand to run first, exception handling, malicious activity and many other things should be prevented and tested constantly.

11. Explain why a modular kernel may be the best of the current operating system design techniques. Ans: Imagine using limousine to get coffee from a drive through of Tim Hortons. You can do it but very slow. And what if you have mini cooper just beside limousine. You can easily decide which one to use. Same goes for modular kernel. Kernel is divided in different modules. Each has its own usage. When one needs to open certain thing other modules are not awakened. Just use the one which needed and use less memory and less runtime. The best example is booting a system. Bootable modules are executed at the time booting OS. As a result the time consumption is much.

12. Distinguish between virtualization and simulation Ans: Virtualization is not real but acts like one. It keeps the system in dark of what is happening. Virtual machine are the best examples of Virtualization. It creates an environment in which operating system has no idea if there is another system using hardware. Simulation in another hand is not real but user can use it to gain access and do certain things. It is like you've access to use fire arms if you've proper license. But you are not allowed to use it carelessly. Pilots use this to learn how to ride a plane.

PART B:

1. Draw a typical computer organization figure that includes the main components of von Neumann architecture. Identify each component, and explain its function and interaction relative to other components. Ans: *From: http://www2.cs.siu.edu/ cs401/Textbook/ch2.pdf

Main components of von Neumann architecture are: 1. CPU (Central Processing Unit): CPU includes three major components; CU (Control Unit), ALU (Arithmetic/Logic Unit) and Registers. CU handles and adjust the manner of execution and process the retrieval to particular request. It also act as a complier which converts code to machine language. Control signals are generated to help sequencing all the requests by CU. So CU orders particular control signals to take the information to particular component. It acts like Uber app which tells drivers where to pick up the traveler and where to drop them. ALU performs all the mathematical operations and Boolean conditions. Registers can be found very close to CU. It is a temporary memory storage for Cache data. Register memory is used to transfer data quickly and repeatedly. That way it cannot be too far from the CU and cannot store chunks of data as it ruins the purpose of not going through memory to get the data. Registers are used to hold operational information and its result. 2. Memory: There are two types of memory storage; RAM (Random Access Memory) and ROM (Read Only Memory). RAM is the part of memory which is used by machine code and programs to execute certain tasks. It is volatile and so all the data inside RAM is erased when system turns of or rebooted. More the RAM, faster the response as programs has more space to store info and work with it and not have to wait to extract memory from the ROM. ROM is a permanent memory which stored even after the system is rebooted. Any data in ROM can be deleted by overriding another data. 3. I/O (Input / Output) Interface: This interface allows user to interact with the system. All the information is received and passed through the memory using I/O Interface. It allows all the different storage like ports, CD, etc.

2. Define system call, and list the main types of system calls. Elaborate on how a system call interacts with a standard C library and hardware under a dual-mode operating system environment. Ans: The call from a program to do certain things with or on system is called system call. It says itself, if we need some emergency we call 911 which is emergency call. In order to communicate with operating system program uses system call as a mod of communication. System call is the only way software talks to kernel. After receiving response to kernel, it is passed to software using API (Application Program Interface). Main types of the system calls are: a) Process control: All the basics of interaction like allocating memory to some program or to delete some memory or to terminate some process. b) File management: To use the memory as a directory, read, write, copy, move, delete, etc. c) Device management: Using inputs of keyboard, showing output to monitor, playing sound, etc. d) Information maintenance: Protecting using layers of abstraction. e) Communication: Privacy, networking, latency, etc.

3. Describe the overall structure of virtual machines, and compare VMware and JVM. Ans:

References

http://www2.cs.siu.edu/ cs401/Textbook/ch2.pdf