CO327 – Operating Systems

Assignment 1

E/14/017

1. We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to "waste" resources? Why is such a system not really wasteful?

   It may be appropriate for the operating systems to forsake this principle in case of single user systems. In such systems, the use of computer resources must be maximized for the user.

   Such a system is not really wasteful because although a graphical user interface may waste CPU cycles, but by maximizing the use of the system, it optimizes the user's interaction with the system.

2. What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

   The main difficulty is keeping the operating system within the fixed time constraints of a real time system.

3. How does the distinction between kernel mode and user mode function as rudimentary form of protection (security) system?

   It's mandatory that certain instructions must be executed only when the operating system is in kernel mode.
   access to hardware devices can be done only when the program is executing in kernel mode.

Interrupts can be controlled only when the CPU is executing in kernel mode.
The privileged instructions can be executed only when the CPU is in kernel mode.
Its capacity is limited, when the CPU is executing in user mode.

Executing privileged instructions in user mode will cause a trap to the operating system. Thus, dual mode of operation provides a rudimentary form of protection.

4. Some early computers protected the operating system by placing it in a memory partition that could not be modified either user job or operating system itself. Describe two difficulties that you think could arise with such a scheme.

   The critical data such as passwords and access control information that are required by or generated by the operating system would have to be passed through or stored in unprotected memory slots and would be accessible to unauthorized users.

   The operating system could never be updated or patched, since it is not modifiable or accessible by the user or the operating system itself.

5. Give two reasons why caches are useful. What problems do they solve? What problem do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

   If the active portions of program and data are placed in a fast-small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. Such a fast-small memory referred as cache.

   Caches are useful when two or more components need to exchange data, and the components perform transfer at different speeds. Caches solve the

transfer problem by providing a buffer of intermediate speed between two components. If the fast device finds the data it needs in the cache, it need not wait for the slower device. The datum in the cache must be kept consistent with the datum in the components. If a component has a data value change, and the datum is also in the cache, the cache must also be updated.

This is especially a problem on multiprocessor system where more than one processes may be accessing a datum. A component may be eliminated by an equal-sized cache, but only if a) the cache and the component have equivalent state-saving capacity (that is, the component retains its data when electricity is removed, the cache must retain data as well), and b) the cache is affordable, because faster storage tends to be more expensive.

6. In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.
   a) What are two such problems?
      Memory management – the system must allocate the memory to several jobs.
      CPU scheduling – the system must choose among several jobs ready to run.

   b) Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.
      We can't ensure the same security in time-shared machine as we have in a dedicated machine. Because in a time-shared machine there are several programs simultaneously in memory. Since any protection scheme devised by a human can inevitably break by a human, more complex scheme.

7. Describe the differences between symmetric and asymmetric multiprocessing. What are three advantages and one disadvantage of multiprocessor system?

Symmetric multiprocessing
All processors are peers.
All processors run a copy of operating system.
Processors communicate with one another as needed.
No special hardware / software needed to differentiate as neither master nor slave exist.

Asymmetric multiprocessing
Processor will have master-slave relationship scheme.
Each processor is assigned by a specific task. (by master processor)
A master process controls the system.
Special hardware or software can only differentiate one master and multiple slaves.

Advantages
Increase throughput – obviously more work can be done in less time. It is because of increase number of processors.
Economy of scale – the cost of multiprocessor system is less when compared to multiple single-processor systems. It is because multiple processors in multiprocessor systems share peripherals, mass storage, and power supplies.

Disadvantage
The solution of activating a back-up copy and restarting the most recent check point in time of system failure is much expensive, since it involves considerably hardware duplication.

8. How are network computers different from traditional personal computers? Describe some usage scenarios in which it is advantageous to use network computers?

   A network computer relies on a centralized computer for most of its services. It can therefore have a minimal operating system to manage its

resources. While a personal computer functions in a standalone manner providing all the required functionality. In network computer people can copy files from one machine to another, hence it is not secure like the personal computer. They are preferred at places at where sharing of resources and data is required.

9. What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps can be generated intentionally by a user program? If so for what purpose?

   An interrupt is a hardware-generated change of flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt, control is then return to the interrupted context and instruction. A trap is a software generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

10. Direct memory access is used for high speed I/O devices in order to avoid increasing the CPU's execution load.
    a) How does the CPU interface with the device to coordinate the transfer?

       All devices have special hardware controllers. Normally, the OS has device drivers (which are kernel programs) that communicate with the controllers. The device drivers have registers, counters and buffers to store arguments and results. Normally these drivers sit in a tight loop to see the I/O through. But these would tie up the CPU during the I/O. with direct memory access, the CPU first load them, and then the device controller takes over.

    b) How does the CPU know when the memory operations are complete?

       The device controller sends an interrupt to the CPU.

c) The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

   You might say there is no interference with the user program, provided you discount interrupts. The DMA controller sends an interrupt when it is done. an this can cause a user process to be suspended.

11. Some computer systems do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is and that it is not possible.

    The main mechanism in which the operating system is able to assert and limit control to hardware resources while servicing user-level programs comes from the ability to classify certain instructions as being privileged and others as being non-privileged. The privileged instructions (often kernel code / system calls) are unbounded by what they can do and thus they can only be executed in privileged mode.
    Non-privileged instructions on the other hand are necessarily constrained by what they can do and it is this distinction in mode of execution at the hardware level that enables the operating system at the software level to very strictly control and limit what the user-level code is able to do and thus by doing so it is able to secure / restrict access to the resources that it is managing.

12. Many SMP systems have different levels of caches; one level is local to each processing core, and another level is shared among all processing cores. Why are caching systems designed this way?

    The different levels are based on access speed as well as size. In general, the closer the cache is to the CPU, the faster the access. However, faster caches are typically more costly. Therefore smaller and faster caches are

placed local to each CPU, and shared caches that are larger, yet slower, are shared among several different processors.

13. Describe a mechanism for enforcing memory protection in order to prevent a program from modifying the memory associated with other programs.

    The processor could keep track of what locations are associated with each process and limit access to locations that are outside of a program's extent. information regarding the extent of a program's memory could be maintained by using base and limits registers and by performing a check for every memory access.

14. Identify several advantages and several disadvantages of open-source operating systems. Include the types of people who would find each aspect to be an advantage or a disadvantage.

    Open-source operating systems have the advantage of having many number of people working upon, ease of distribution, many people debugging codes, and faster updates. and students and programmers can view and modify the source code. They are free to use compared to the commercial operating systems. Some companies prefer commercial operating systems which provide paid support, so that they have someone to hold accountable when they face problem. Backward compatibility is lacking making upgrades difficult.