Assignment 1

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?

Single-user systems should maximize use of the system for the user. A GUI might "waste" CPU cycles, but 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. If the system does not complete a task in a certain time frame, it may cause a breakdown of the entire system it is running. Therefore when writing an operating system for a real-time system, the writer must be sure that his scheduling schemes don't allow response time to exceed the time constraint.

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

Certain instructions could be executed only when the CPU is in kernel mode. Similarly, hardware devices could be accessed only when the program is executing in kernel mode. Control over when interrupts could be enabled or disabled is also possible only when the CPU is in kernel mode. Consequently, the CPU has very limited capability when executing in user mode, thereby enforcing protection of critical resources.

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

1. Such operating systems cannot be modified or updated. So if we notice some critical bug in OS we have no way to correct it.
2. All passwords and other important credentials of users must be kept in an unprotected memory, since we cannot store them in a protected memory.

5. Give two reasons why caches are useful. What problems do they solve? What problems 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?

1. Caches allow faster access to data by storing data in a physically faster device (e.g.: memory instead of disk).
2. Caches reduce cost of finding data by requiring only a small number of items to be searched.
3. Caches reduce the load on slow physical devices, by handling most accesses to data on a faster, more capable device.

Caches keep the data in a cheap-to-access place so that it solves the slow access of data.

Caches cause consistency problems, because the real data can be different from the data that is stored in the cache. For example when a data is updated in the disk, if it also not updates the cache it will cause the above problem.

To eliminate a device with an equal sized cache,

The replaced cache and the device should have equivalent state-saving capacity (that is, if the device retains its data when electricity is removed, the cache must retain data as well) and although faster storage tends to be more expensive, in this case it should be affordable. But it is more difficult to implement these things.

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? b) Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

(A) 1. One user can read the private data of another – privacy
2. One user can change or damage the data of another – integrity
3. One user can prevent another from getting anything done – denial of service

(B) We can have a same degree of security if we can ensure that the operating system prevents any sharing of data between users, either for reading or writing, and fairly shares the computer. But we can never be sure that the software will not have any bugs. So we can never be sure that we prevent all sharing of data and fairly allocate computer resources.

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

Basis for Comparison	Symmetric multiprocessing	Asymmetric multiprocessing
Basic	Each processor run the tasks in OS	Only master processor run the tasks of OS
Processes	Processor takes processes from a common ready queue, or there may be a private ready queue for each processor.	Master processor assigns processes to the slave processors, or they have some predefined processes.
Architecture	All processors have the same architecture	All processors may have the same or different architecture
Communication	All processors communicate with other processors through a shared memory	Processors are controlled by master processor so that other processors do not communicate with each other
Failure	If processor fails the processing capacity of system reduces	If master processor fails, a slave become the master and continues. And if a slave fails that task is passed on to another processor
Ease	Somewhat complex as all the processors need to be synchronized to maintain load balance	Simple as the master processor access the data structure

Advantages

- Multiprocessing systems can save money by sharing power supplies, housings and peripherals.
- Can execute programs quickly
- Increased reliability

Disadvantages

- Multiprocessors are very complex in both hardware and software. Additional CPU cycles are required to manage the cooperation, so per-CPU efficiency goes down.

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 is also called a thin client. These are terminals which can implement web based computing. In order to fulfill its computational needs it is heavily dependent on other server. Example: Remote Desktop Services

Tradition computers on the other hand are standalone systems which have its own CPU and all the computational needs can be fulfilled by the system alone.
Example: Personal computer used at homes.

Network computers are used in following places.

1. Most of the financial firms outsource their IT operations to other companies. They establish a special area called ODC (Offshore Development Centers). In ODC thin clients are used to connect to the onsite servers. In this way data security can be maintained as all the data is maintained on the server and user has no way to copy the data on a local computer in an unauthorized manner. Even if the thin client is stolen the data is safe on the onsite server.

2. It also provides hardware resource optimization as the cost of cable, buses and I/O can be minimized by this approach and also the processing power can be utilized by the user session that needs it the most.

3. Software maintenance can be reduced by using network computer as all the software patches and updates and OS migration can be rolled out for all users in one go.

9. What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?
Interrupt signal tells the processor that there is some event that needs immediate attention. Basically its purpose is to interrupt the processor.

Trap (or exception) is a software generated interrupt caused either by an error (division by zero or invalid memory access) or by a specific request from the user program that an OS service be performed. Interrupt triggered by signal from either hardware or software but not tied to any specific instruction.

Yes it can be generated intentionally by a user program through exception or explicit call. It can be used for debugging of a program. Trap can be called to catch any arithmetic errors or array out of bound.

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?
b) How does the CPU know when the memory operations are complete?
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.
a) The CPU can initiate a DMA operation by writing values in to special registers that can be independently accessed by the device. When the device receives a command from the CPU, the device also initiates the corresponding operation.

b) Once the operation is completed DMA interrupts the CPU

c) Both the CPU and the DMA controller are bus masters. A problem would be created if both the CPU and the DMA controller want to access the memory at the same time. Accordingly, the CPU should be momentarily prevented from accessing main memory when the DMA controller seizes the memory bus. However, if the CPU is still allowed to access data in its primary and secondary caches, a coherency issue may be created if both the CPU and the DMA controller update the same memory locations.

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.

An operating system for a machine of this type would need to remain o in control (or monitor mode) at all times. This could be accomplished by two methods:

a. Software interpretation of all user programs (like some BASIC, Java, and LISP systems, for example). The software interpreter would provide, in software, what the hardware does not provide.

b. Require meant that all programs be written in high-level languages so that all object code is compiler-produced. The compiler would generate (either in-line or by function calls) the protection checks that the hardware is missing.

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.

Advantages

- Many people can work on them
- Many people can debug them
- Ease of access and distribution
- Rapid update cycles
- View and modify the source code – For students and programmers
- Typically open source operating systems are free for some forms of use, usually just requiring payment for support services

Disadvantages

- Some open source operating systems do not offer paid support programs. Some companies avoid open source projects because they need paid support, so that they have some entity to hold accountable if there is a problem or they need help fixing an issue
- Some open source OS have lack of discipline in the coding, means that backward compatibility is lacking making upgrades difficult. But because of frequent new releases users have upgrade frequently.