

## HomeWork 1

**Due Date:** Wednesday 2/17/2021

**HW Delivery:** submit on **Canvas** by the due date, before midnight

**Total Points:** 60

**General rules:** Create homework, compose specifications your work using a common *document-creation* tool, such as Microsoft® Word. Each question is worth 3 points.

**Hints:** Refer to the wwweb or lecture notes for this class to answer the questions below. Be concise, complete, and precise.

1. List the 4 key critical components of a working computer system; take a view from lowest level HW to the highest level.

Computer systems compromise of four components:

- Hardware Resources
  - CPU, memory, IO devices
- Operating System
  - Controls and coordinates use of HW among various applications and competing users
- Application Programs (Apps)
  - Use system resources to solve computing problems
  - Word processors, compilers, web browsers, database systems, video games
- Users (You)
  - Human users, other computers, and networks

2. Why is it meaningful, even necessary to have an Operating System? What does an OS provide?

It is necessary to have an OS because it helps manage all computer hardware and software resources (*resource allocator*). It also decides between conflicting request for efficient and fair resource use (*control program*).

The OS keeps Hardware running, use resources efficiently, maintain a file system, support multitasking, and manage virtual memory.

3. Explain the essence of a true **interrupt**. What causes interrupts? How can a user application know, that is has been interrupted?

The true essence of an interrupt handles asynchronous events. It uses a signal that is triggered by hardware or IO devices and sent to the CPU by way of system bus. It stops whatever it is doing and transfer execution to a fixed location.

A trap or exception is a software generated interrupt caused by an error, a user, or software request.

4. An interrupt disrupts the regular flow of program execution and consumes compute resources. Why is it necessary for an OS to support and manage interrupts?

It is necessary to support interrupts because it can occur during a routine it must be able to save current state and then restore the state before returning. Management of interrupt is important because it allows for high-priority interrupt to preempt the execution of a low-priority interrupt. Interrupts are used so heavily for time-sensitive processing, efficient interrupt handling is required for good system performance.

5. Which are the 5 key “management” functions of an OS?

The 5 key management functions of an OS are user interface, resource management, tasks management, files management, and utilities.

6. Characterize the type of information residing in main memory, versus information on secondary storage.

Main memory (also called rewritable memory) is volatile meaning it loses content when power is turned off. It is commonly implemented as dynamic random-access memory (DRAM). Secondary storage is an extension of main memory and can hold large quantities of data permanently. Most common secondary storage devices include HDDs or SSDs. Many programs use it as a source and destination for their processing.

7. What is the meaning of the acronym EULA? Why use EULAs? Discuss the expenses of EULAs?

EULA stands for *end-user license agreement* and it generally protects software companies from errors in programs. The licensing means you as the user are authorized to use the software, not the entire program.

8. Characterize an Operating System’s function of: Resource allocation, resource sharing, accounting, protection, and security

Resource allocation: make available time, memory, peripherals

Resource sharing: to more than one user

Accounting: track actual usage

Protection: avoid other users from interfering

Security: manage who can see and interpret generated information

9. Describe **System Call**. How can system calls be made to fit a user’s specific needs? List a sample system call.

A *System Call* is a request done by the user to the OS for help, regarding computing needs. It takes in a value of 0 or more, an example can be copying a file from one destination to another.

10. Argue in which kind of language (level of language) an OS should be implemented.

Programmers use high-level languages like C and C++ to create the kernel, utilities and some applications. While also incorporating assembly for performance-critical situations, but not a lot is written in assembly because it is not portable.

11. Outline the **hierarchy** of at least 5 different **storage** resources and technologies for holding information; mention relative speeds and capacities.

1) Registers →

- Technology: custom memory with multiple ports CMOS
- Capacity: < 1 KB
- Speed (MB/Sec): 20,000 – 100,000

- 2) Cache →
  - Technology: on-chip or off-chip CMOS SRAM
  - Capacity: < 16 MB
  - Speed (MB/Sec): 5,000 – 10,000
- 3) Main Memory →
  - Technology: CMOS SRAM
  - Capacity: < 64 GB
  - Speed (MB/Sec): 1,000 – 5,000
- 4) SSD →
  - Technology: flash memory
  - Capacity: < 1 TB
  - Speed (MB/Sec): 500
- 5) Magnetic Disk →
  - Technology: magnetic disk
  - Capacity: < 10 TB
  - Speed (MB/Sec): 20 – 150

12. Describe in detail what is essential in a **Real Time** compute environment.

In a **Real Time**, compute environment has to be well-defined and follow fixed time constraints. The processing must be done with the defined constraints or the system will fail. A real-time functions correctly only if it returns the correct result within its time constraints.

13. Specify in your own words what constitutes an **Operating System**.

An OS is a resource manager, that overlooks how hardware and software resources are being used. It also is interrupt driven, deciding between conflicting requests for efficiency and fair resource use, preventing errors and improper use.

14. What are essential steps to get an Operating System to work at the moment a computer is turned on (AKA **booted up**)?

Computer startup begins with loading up the bootstrap program which then loads the operating system. This in return initializes all aspects of the system and starts execution. To prevent the bootstrap program from being overwritten it is stored in EPROM (aka firmware).

15. Characterize in detail what is a “process”. Describe typical process needs; types, cause of start, and end of process.

A process is a program in execution, and it needs resources such as CPU, memory, files, etc., to accomplish its task. There exists single-threaded process which contains only program counter (pc) specifying location of next instruction to execute. This means it executes each process sequentially, one at a time. The other type is multi-threaded process where it can contain one pc per each thread. This allows for many processes to be executed at a time, by multiplexing CPUs among processes and threads. The cause of start for a process could be user or OS. End of process could be user or OS like terminating a process.

16. What is an “idle Loop”? Why is it part of an OS? Should be an Idle Loop?

An “idle loop” is the part of an OS where the code recognizes that there are no user requests that need to be handled. It is part of the OS because without it the processor being occupied with something it will potentially freeze.

17. What is a “utility program?” Why have one? Give an example. Should the utility program be an integral part of the OS?

A utility program is a type of system software that can handle low-level details and other tasks behind the scenes, such as maintenance and repairs not handled by the OS. It makes it easier for users create, copy, delete, and store files on storage devices. It should be an integral part of the OS because it makes it convenient for the user.

18. What is “disk fragmentation?” What causes it? What is the impact on user programs?

Disk fragmentation is when user files are scattered over a diverse cluster. It is caused when the file system allows gaps to develop between different pieces of a file. These gaps in the file system make retrieval a longer process than it should be, because it will need to be looking for all the pieces in different places in order to bring the file together.

19. What is DMA? Why does an OS support DMA? What is its main advantage?

Direct Memory Access (DMA) is used for high-speed IO devices that are able to transmit information at close to memory speeds and is managed by the OS. The device controller transfers blocks of data from buffer to storage directly to or from main memory without CPU intervention.

20. List some similarities and differences between Unix and Linux operating systems?

UNIX is a command-line based, textual input and output-based OS, and Linux was later developed and is UNIX-like OS that is basically an open-source version of UNIX.

- Open sources include GNU/Linux and BSD UNIX
- Both can be used with a VMware Player which is used to run guest OS for exploration