# Homework 1

1. A more common definition, and the one that we usually follow, is that the operating system is the one program running at all times on the computer—usually called the kernel. Along with it, there are two other types of programs: system program, which are associated with the operating system but are not part of it, and applications programs, which include all programs not associated with the operation of the system.
2. To run an HPL (High-Level Programming Language, such as C, C++, Java) program, the source code of that program should be generally translated into ①Executable code by mainly two types of system software – ②Compiler or ③Interpreter. Then the OS should prepare necessary resources for it. Once all those resources are ready, that program turns into ④Process and there is a special data structure in OS, namely ⑤task to store the necessary information for running that program, including the identification information, location information etc.

①②③④⑤：IDE (Integrated Development Environment, such as VC, Eclipse), Assembler, Interpreter, Translator, Process Control Block, Executable code, Thread, Task, Job, Process, Partition Control Block, Thread Control Block, File Control Block, Machine Code, Compiler, Terminator.

1. As software, OS itself should be executed in the computer system. When you turn on the computer, the first executed program is LOADER for checking all instruments are OK, which is stored usually in a ROM.

A. BIOS B. POST C. LOADER D. TESTER

1. 1. When you turn on the power to a computer, the first program that runs is usually a set of instructions ( kept in the computer's read-only memory – ROM), abbreviated as ① POST which examines the system hardware to make sure everything is functioning properly. Once it has successfully completed, the software loaded in ROM (sometimes called the ② BIOS or firmware [**固件**]) will begin to activate the computer's disk drives. If everything is fine, it is then the turn of ③ Bootstrap Loader to load the operating system into memory and allows it to begin operation.

①②③： OS, BIOS, MBR, POST, bootstrap loader, machine code, instructions

1. One of the most important aspects of operating systems is the ability to Multitasking. A single program cannot, in general, keep either the CPU or the I/O devices busy at all times. Single users frequently have multiple programs running. Even in these kinds of systems an environment, in which the various system resources (for example, CPU, memory, and peripheral devices) are utilized effectively, is provided, but they do not provide for user interaction with the computer system. Time sharing (or Multiplexing ) is a logical extension. In time-sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
2. Time sharing and multiprogramming require that several jobs be kept simultaneously in memory. If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is Medium Therm Scheduling. When the operating system selects a job from the job pool, it loads that job into memory for execution. In addition, if several jobs are ready to run at the same time, the system must choose among them. Making this decision is 2nd Short Term Scheduling.
3. [Short answer (1 pt)] What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

Both of Trap and Interrupt are a halt in the CPU work, hardware interrupt are called Interrupt and software interrupt are called Trap (or Exception) called by the programmer. The trap is useful for the programmer to call a subroutine (Interrupt Service Routine) associated with the trap, and can handle the conditions that are raised by the interrupt.