

Loading an OS

POST

- When the computer is powered on it only has access to the ROM, which is only big enough to hold a few instructions
- The computer then carries out the power on self test (POST), which checks that everything needed for the computer to come to life is available.
- It also clears the registers of the CPU and loads the address of the first instruction in the boot program into the CPU

Boot Program

- The boot program is stored in the ROM, and contains the skeleton of the BIOS (only the main structure of the BIOS, because it contains user-defined options, which need to be modifiable, and ROM is read only) The user-defined parts of the BIOS would be stored in the CMOS RAM
- The CPU then sends signals to check all the hardware is working. Hardware containing its own BIOS will be incorporated into the systems BIOS.

Ready to load the OS

- Once all the initial checks are done the PC is ready to load the OS. It will first check drive A, and if any bootable medium is present it will boot from it. If not it will check for drive C or return an error message to the user
- If the OS is Windows the boot program will look for IO.SYS and MSDOS.SYS, and load them.
- IO.SYS holds extensions to the ROM BIOS and contains a routine called SYSINIT which controls the rest of the boot procedure.
- SYSINIT now takes control and loads MSDOS.SYS which works with the BIOS to manage files and execute programs

Loading the OS

- The OS searches the root directory for a boot file such as CONFIG.SYS which tells the OS how many files may be opened at the same time., and may also contain instructions to load various device drivers.
- The OS tells MSDOS.SYS to load a file called COMMAND.COM which is in three parts. The first part is a further extension to the I/O functions and it joins the BIOS to become part of the OS. The second part contains resident OS commands, such as DIR and COPY
- The files CONFIG.SYS and AUTOEXEC.BAT are created by the user so that the PC starts up in the same configuration each time it is switched on.