1) **Power On Self Test phase (POST)**

The pre-boot sequence begins when the power is turned on. The computer runs Power-On Self Test(POST) routines to determine the amount of physical memory and the other hardware components present. If the computer has a plug and play BIOS, the hardware is recognized and configured. The computer’s BIOS locates the boot device, and then loads and runs the Master Boot Record(MBR).An MBR is generally 512 Bytes in size and consists of bootstrap code and partiion table that is 64 bytes. Every partition entry is 16 bytes in size making the no of partitions in MBR based hard disk limited to only 4.

## Kernel Load

After configuration selection, the Windows XP Professional kernel (NTOSKRNL.EXE) loads and initializes. NTOSKRNL.EXE also loads and initializes device drivers and loads services. If you press *Enter* when the Hardware Profile/Configuration Recoverymenu appears, or if NTLDR makes the selection automatically, the computer enters the kernel load phase. The screen clears and a series of white rectangles appears across the bottom of the screen.

During the kernel load phase, NTLDR does the following:

* Loads NTOSKRNL.EXE but does not initialize it.
* Loads the hardware abstraction layer file (HAL.DLL).
* Loads the HKEY\_LOCAL\_MACHINE\SYSTEM registry key from %*systemroot%*\System32\Config\System.
* Selects the control set it will use to initialize the computer. A *control set* contains configuration data used to control the system, such as a list of the device drivers and services to load and start.
* Loads device drivers with a value of 0x0for the Start entry. These typically are low-level hardware device drivers, such as those for a hard disk. The value for the List entry, which is specified in the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\ServiceGroupOrder subkey of the registry, defines the order in which NTLDR loads these device drivers.

Kernel Initialization

When the kernel load phase is complete, the kernel initializes, and then NTLDR passes control to the kernel. At this point, the system displays a graphical screen with a status bar indicating load status. Four tasks are accomplished during the kernel initialization stage:

1. **The Hardware key is created.??**On successful initialization, the kernel uses the data collected during hardware detection to create the registry key HKEY\_LOCAL\_MACHINE\HARDWARE. This key contains information about hardware components on the system board and the interrupts used by specific hardware devices.
2. **The Clone control set is created.??**The kernel creates the Clone control set by copying the control set referenced by the value of the Current entry in the HKEY\_LOCAL\_MACHINE\SYSTEM\Selectsubkey of the registry. The Clone control set is never modified, as it is intended to be an identical copy of the data used to configure the computer and should not reflect changes made during the startup process.
3. **Device drivers are loaded and initialized.??**After creating the Clone control set, the kernel initializes the low-level device drivers that were loaded during the kernel load phase. The kernel then scans the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services subkey of the registry for device drivers with a value of 0x1 for the Start entry. As in the kernel load phase, a device driver's value for the Group entry specifies the order in which it loads. Device drivers initialize as soon as they load.

If an error occurs while loading and initializing a device driver, the boot process proceeds based on the value specified in the ErrorControl entry for the driver.

Table 18.4 describes the possible ErrorControl values and the resulting boot sequence actions.

**Table 18.4??***ErrorControl Values and Resulting Action*

| **ErrorControl value** | **Action** |
| --- | --- |
| 0x0 (Ignore) | The boot sequence ignores the error and proceeds without displaying an error message. |
| 0x1 (Normal) | The boot sequence displays an error message but ignores the error and proceeds. |
| 0x2 (Severe) | The boot sequence fails and then restarts using the LastKnownGood control set. If the boot sequence is currently using the LastKnownGood control set, the boot sequence ignores the error and proceeds. |
| 0x3 (Critical) | The boot sequence fails and then restarts using the LastKnownGood control set. However, if the LastKnownGood control set is causing the critical error, the boot sequence stops and displays an error message. |

ErrorControl values appear in the registry under the subkey HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services*\name\_of\_service\_or\_driver\*ErrorControl.

1. **Services are started.??**After the kernel loads and initializes devices drivers, the Session Manager (SMSS.EXE) starts the higher order subsystems and services for Windows XP Professional. Session Manager executes the instructions in the BootExecute data item, and in the Memory Management, DOS Devices, and SubSystems keys.

Table 18.5 describes the function of each instruction set and the resulting Session Manager action.

**Table 18.5??***Session Manager Reads and Executes These Instruction Sets*

## Logon

The logon process begins at the conclusion of the kernel initialization phase. The Win32 subsystem automatically starts WINLOGON.EXE, which starts the Local Security Authority (LSASS.EXE) and displays the Logon dialog box. You can log on at this time, even though Windows XP Professional might still be initializing network device drivers.

Next, the Service Controller executes and makes a final scan of the HKEY\_ LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services subkey, looking for services with a value of 0x2 for the Start entry. These services, including the Workstation service and the Server service, are marked to load automatically.

The services that load during this phase do so based on their values for the DependOnGroup or DependOnService entries in the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\ Services registry subkey.

A Windows XP Professional startup is not considered good until a user successfully logs on to the system. After a successful logon, the system copies the Clone control set to the LastKnownGood control set.

For more information on LastKnownGood configuration, see Lesson 3, "Using Startup and Recovery Tools," later in this chapter

**Hibernation** (or **suspend to disk**) in computing is powering down a computer while retaining its state. Upon hibernation, the computer saves the contents of its [random access memory](https://en.wikipedia.org/wiki/Random_access_memory) (RAM) to a [hard disk](https://en.wikipedia.org/wiki/Hard_disk) or other [non-volatile storage](https://en.wikipedia.org/wiki/Non-volatile_storage). Upon resumption, the computer is exactly as it was before entering hibernation.

On Windows computers, hibernation is available only if **all** hardware and [device drivers](https://en.wikipedia.org/wiki/Device_driver) are ACPI and [plug-and-play](https://en.wikipedia.org/wiki/Plug-and-play)–compliant. Hibernation can be invoked from the [Start menu](https://en.wikipedia.org/wiki/Start_menu) or the command line.[[4]](https://en.wikipedia.org/wiki/Hibernation_(computing)#cite_note-4)

[Windows 95](https://en.wikipedia.org/wiki/Windows_95) supports hibernation through hardware manufacturer-supplied drivers and only if compatible hardware and BIOS are present. Since Windows 95 supports only [Advanced Power Management](https://en.wikipedia.org/wiki/Advanced_Power_Management) (APM), hibernation is called Suspend-to-Disk. [Windows 98](https://en.wikipedia.org/wiki/Windows_98) and later support ACPI. However, hibernation often caused problems since most hardware was not fully ACPI 1.0 compliant or did not have [WDM](https://en.wikipedia.org/wiki/Windows_Driver_Model) drivers. There were also issues with the FAT32 file system.[[5]](https://en.wikipedia.org/wiki/Hibernation_(computing)#cite_note-5)

[Windows 2000](https://en.wikipedia.org/wiki/Windows_2000) is the first Windows to support hibernation at the operating system level (OS-controlled ACPI S4 sleep state) without special drivers from the hardware manufacturer. A hidden system file named "hiberfil.sys" in the root of the [boot partition](https://en.wikipedia.org/wiki/Boot_partition) is used to store the contents of RAM when the computer hibernates. In Windows 2000, this file is as big as the total RAM installed.

Remote Installation Services (RIS) is a Windows component that you can install with Windows Server 2003 or add at any time after the operating system is installed. RIS is an automated installation technology that you can use to create installation images of operating systems or of complete computer configurations, including desktop settings and applications. These installation images can then be made available to users at client computers. RIS is typically used during large-scale deployments when it would be too slow and costly to have administrators or end users interactively install the operating system on individual computers.

RIS is a Windows component that you can install with Windows Server 2003 or add at any time after the operating system is installed. Services that install with RIS include the Remote Installation service, TFTPD, and the SIS service.