Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Fundamental Windows

IS Academy - Infra

People matter, results count.

Contents

- **Introduction**
  - ➢ Introduction to Windows OS
  - ➢ Workstation server
- **Windows Kernel**
  - ➢ Architecture of Windows kernel
  - ➢ Modes of kernel
  - ➢ Components of Kernel
  - ➢ Executive Services
- **Windows Booting Process**
  - ➢ Flow of booting process
  - ➢ Hibernation
  - ➢ Safe boot options
- **Windows File System**
  - ➢ What is file system
  - ➢ Types of file system
  - ➢ Windows file system
    - ✓ FAT
    - ✓ NTFS

Management Mechanisms

- ➢ Registry
- ➢ Services
- ➢ WMI

Windows Device Management

- ➢ Device Drivers
- ➢ Power Manager

Windows Process Management

- ➢ Process & Threads
- ➢ Processing types

Windows Memory Management

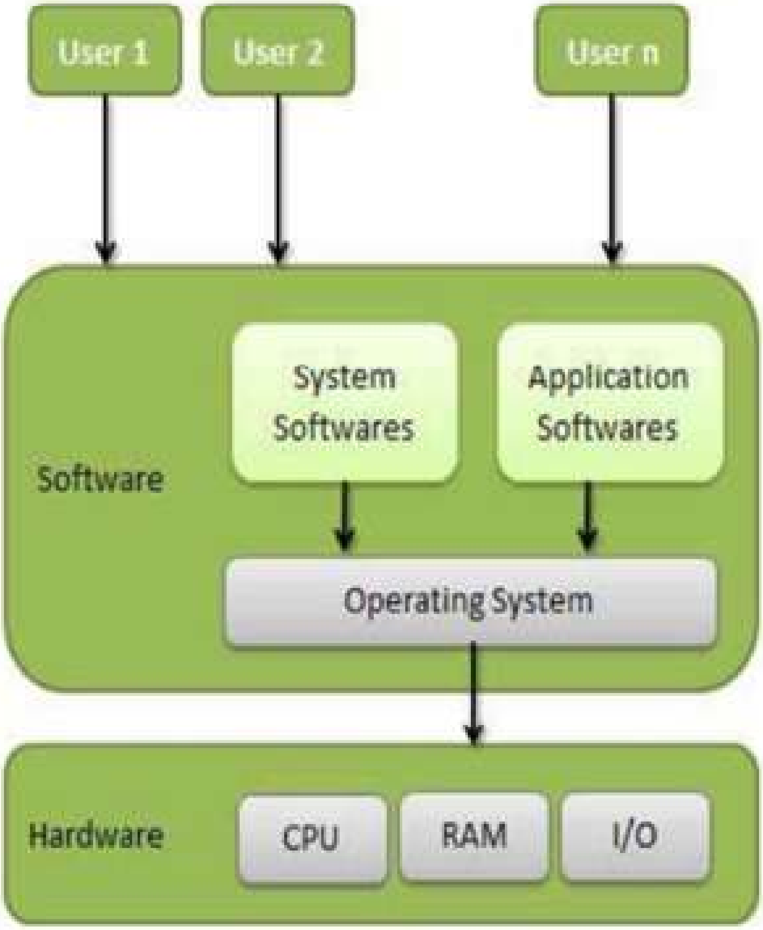
- ➢ Virtual Address Space
- ➢ Paging
- ➢ Memory Pool
- ➢ Virtual memory
- ➢ File Mapping
- ➢ Page Faults

WINDOWS OPERATING SYSTEM

OPERATING SYSTEM

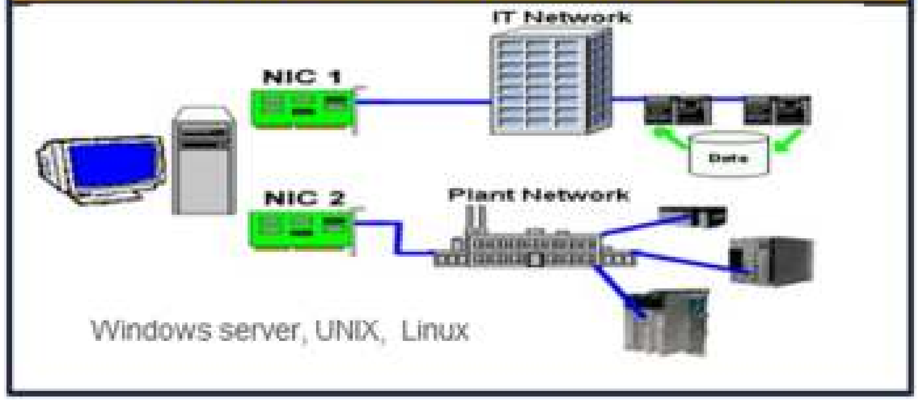
- **Definition:**

  An operating system (OS) is a software, consisting of programs and data, that runs on computers, manages computer hardware resources, and provides common services for execution of various application software.



TYPES OF OPERATING SYSTEM

Network(Server) Operating System



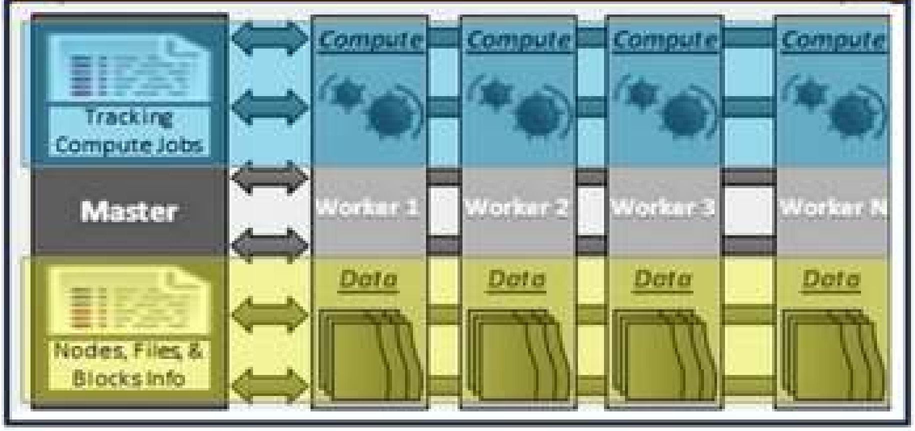
Windows server, UNIX, Linux

Real Time Operating System

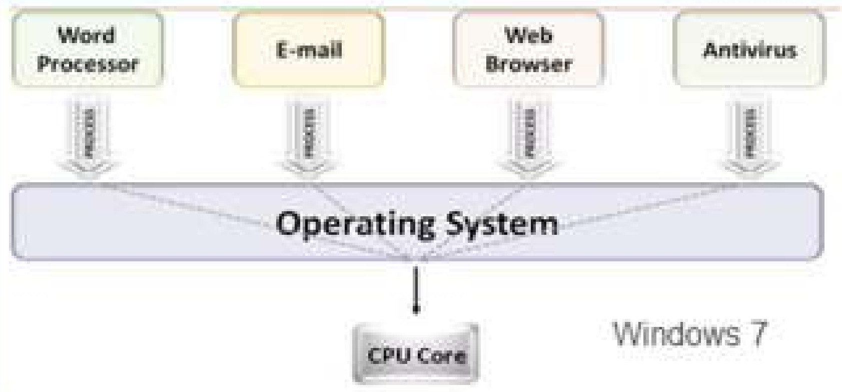
Roadside Equipment Unit
Traffic Signal Controller

Traffic control Systems

Distributed Operating System



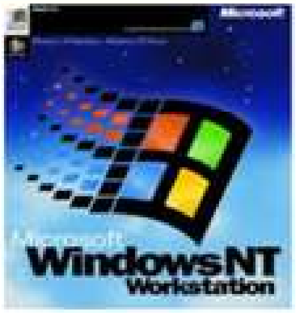
Multi-tasking Operating System



Windows 7

WORKSTATION

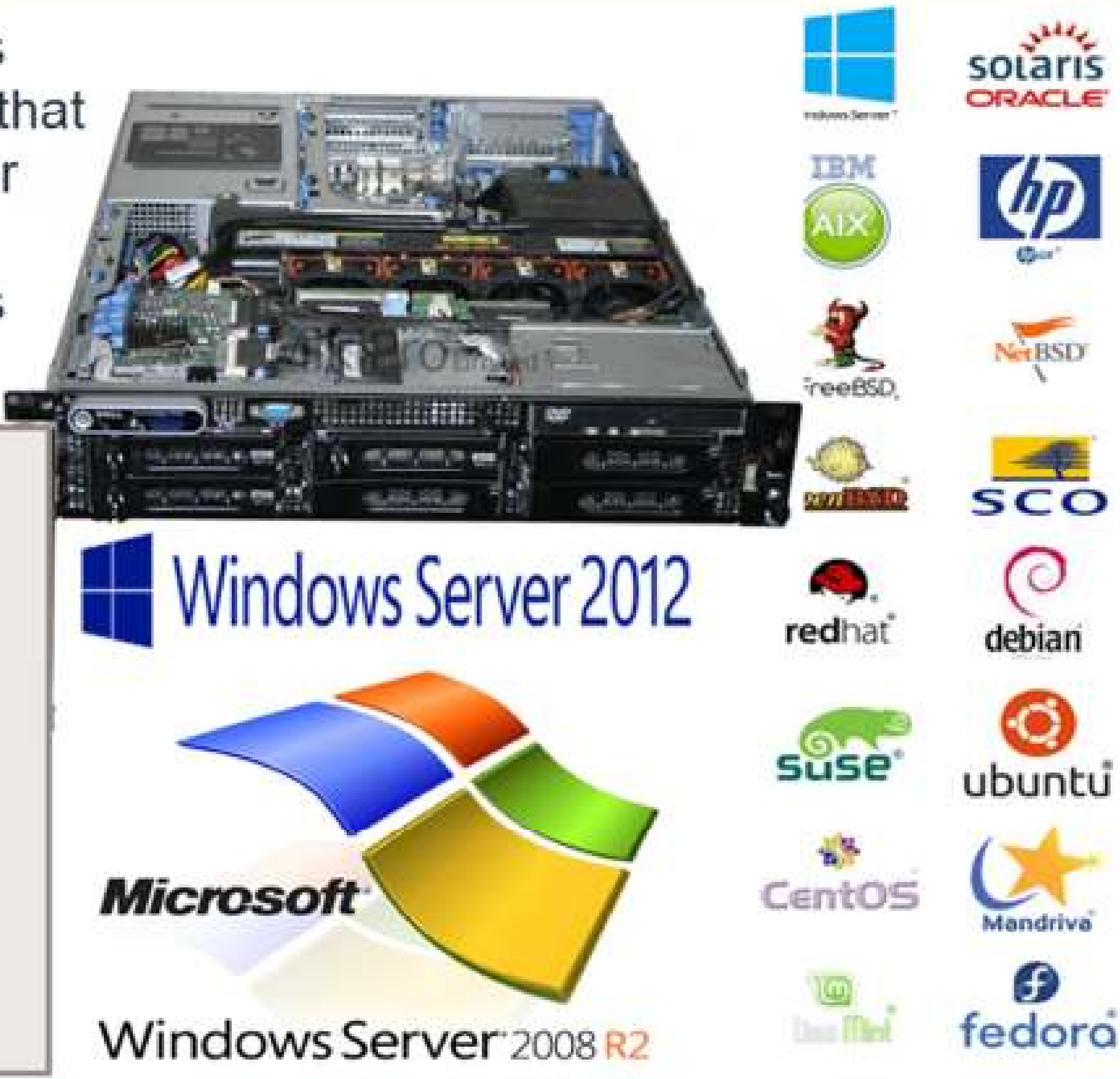
Workstation operating system is primarily designed to run end user applications. Those applications can be text processor, presentation software, games, etc. It runs on lower end hardware.



SERVER

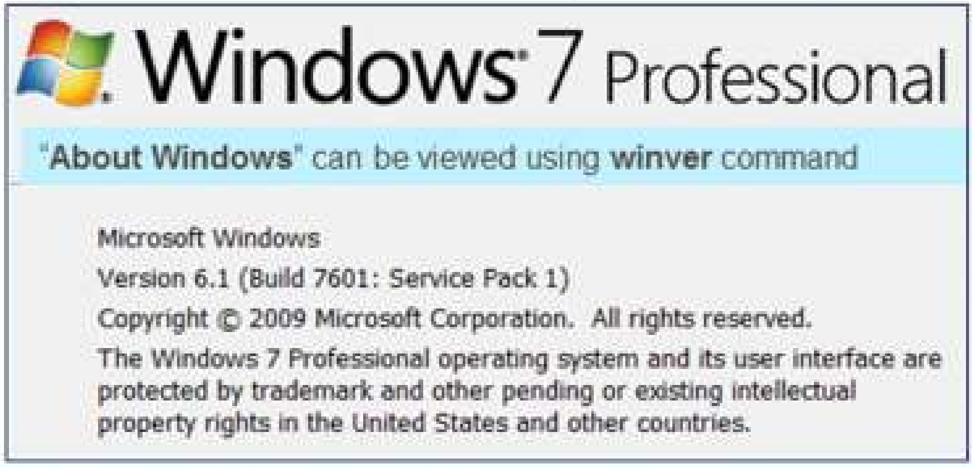
A server operating system is designed to run on servers, that operate within a client/server architecture to serve the requests of client computers on the network.

Authentication
Mail Service
Web Server
Chat server
File server
Webcast



CLIENT & SERVER VERSION, EDITIONS

Operating system (Client)	Version	Operating system (Server)	Version
Windows 10	10.0	Windows Server 2016 Technical Preview*	10.0
Windows 8.1	6.3	Windows Server 2012 R2	6.3
Windows 8	6.2	Windows Server 2012	6.2
Windows 7	6.1	Windows Server 2008 R2	6.1
Windows Vista	6.0	Windows Server 2008	6.0
Windows XP 64-Bit Edition**	5.2	Windows Server 2003/ 2003 R2 **	5.2
Windows XP**	5.1	Windows 2000 Server**	5.0



"**About Windows**" can be viewed using **winver** command

Microsoft Windows
Version 6.1 (Build 7601: Service Pack 1)
Copyright © 2009 Microsoft Corporation. All rights reserved.
The Windows 7 Professional operating system and its user interface are protected by trademark and other pending or existing intellectual property rights in the United States and other countries.

Editions	Version
Windows 7 Ultimate	6.1
Windows 7 Enterprise	6.1
Windows 7 Professional	6.1
Windows 7 Home Premium	6.1
Windows 7 Home Basic	6.1
Windows 7 Starter	6.1

Note: * Yet to be released. Currently available for only technical Preview
** Microsoft ends support – No updates will be released.

32 bit Vs 64 bit

- **Addresses:** The terms 32 - bit and 64 - bit are referring to the CPU, or processor. The number represents how the data is processed. It is processed either as 2^32 or 2^64 . The larger the number is, the larger the amount of data that can be processed at any one time.
- **RAM per OS:** A 32 - bit operating system can handle up to 4 GB of RAM, and a 64 - bit processor can handle up to 16 Exabytes of RAM. The problem is that Windows and most motherboards can't handle this much RAM.
- **RAM per process:** RAM limit of 4GB on x86 for processes (always). If you think this is not important, try running a huge MSSQL database intensive application. It will use > 4GB itself if you have it available and run much better.
- **Wider programs available:** From an x64 you can run both x86 and x64 programs.
- **Faster:** Some calculations are faster on a 64-bit CPU.
- **Exclusive programs available:** Several new programs only support x64.

It's your turn now..

- Identify different editions of windows 8.1 and windows 10
- What is the OS version defined for Windows 8.1?
- Explore the hardware used for running server OS
- Identify the advantages and disadvantages of Windows OS

WINDOWS KERNEL ARCHITECTURE

KERNEL ARCHITECTURE

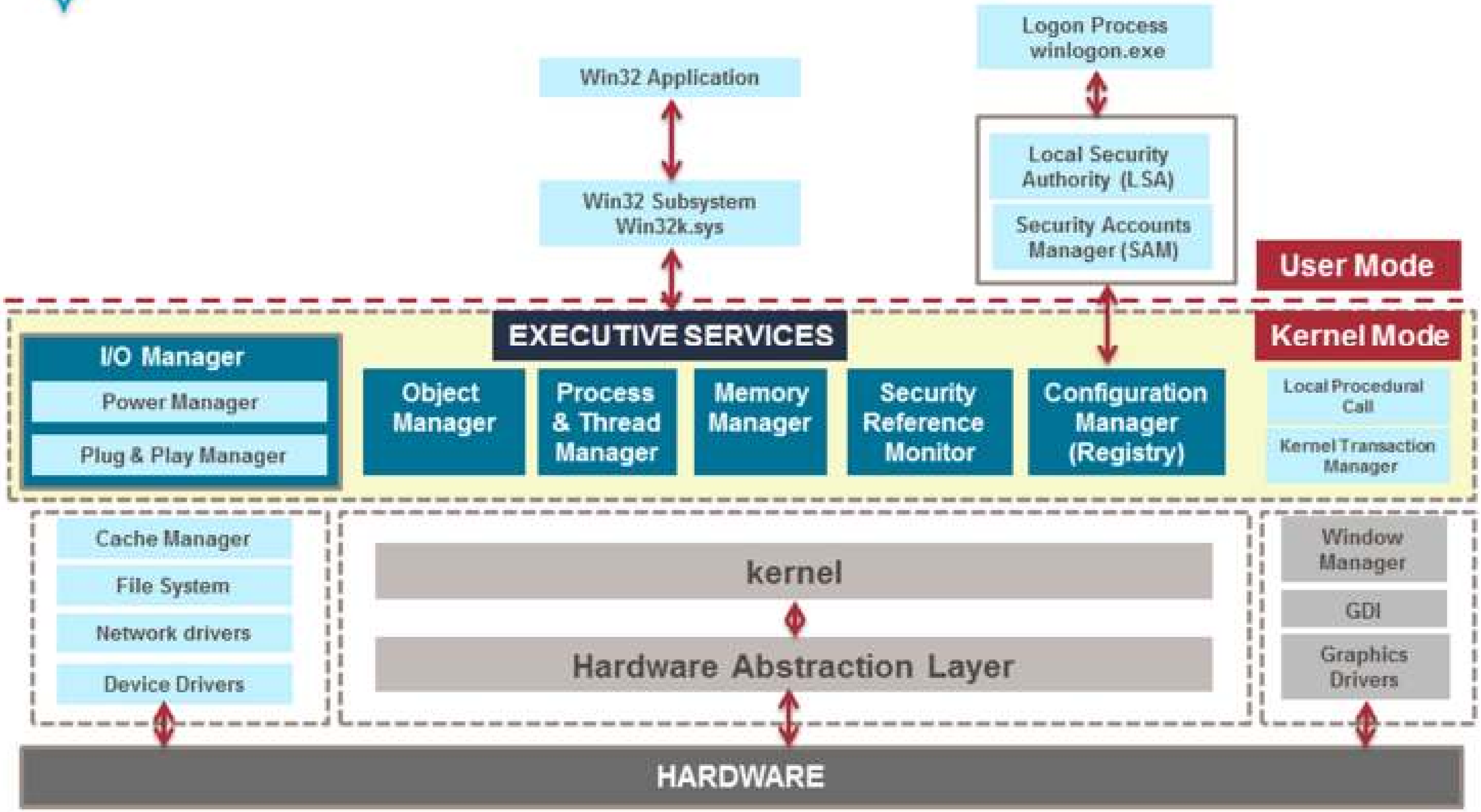
KERNEL

- The **kernel** is the main component (heart)of a computer operating systems.
- It is a bridge between applications and the actual data processing done at the hardware level.
- It provides basic low-level operations such as scheduling threads or routing hardware interrupts.

KERNEL MODE AND USER MODE

- A processor in a computer running Windows has two different modes: user mode and kernel mode.
- The processor switches b/w two modes depending on what type of code is running on the processor.
- Applications run in user mode, and core operating system components run in kernel mode.
- While many drivers run in kernel mode, some drivers may run in user mode.

WINDOWS KERNEL ARCHITECTURE



Logon Process winlogon.exe

Win32 Application

Local Security Authority (LSA)

Security Accounts Manager (SAM)

Win32 Subsystem Win32k.sys

User Mode

EXECUTIVE SERVICES

Kernel Mode

I/O Manager

Power Manager

Plug & Play Manager

Object Manager

Process & Thread Manager

Memory Manager

Security Reference Monitor

Configuration Manager (Registry)

Local Procedural Call

Kernel Transaction Manager

Cache Manager

File System

Network drivers

Device Drivers

kernel

Hardware Abstraction Layer

Window Manager

GDI

Graphics Drivers

HARDWARE

SAM-Security Accounts Manager; LSA- Local Security Authority
KTM-Kenel Transaction Manager; LPC- Local Procedural Call

KERNEL ARCHITECTURE

HARDWARE ABSTRACTION LAYER

- Windows runs on many different configurations of the personal computer. Each configuration requires a layer of software that interacts between the hardware and the rest of the operating system.
- This layer abstracts (hides) the low-level hardware details from drivers and the operating system, it is called the hardware abstraction layer (HAL).
- The HAL includes hardware-specific code that controls I/O interfaces, interrupt controllers and multiple processors.

EXECUTIVE SERVICES

The Windows operating system uses the term *executive layer* to refer to kernel-mode components that provide a variety of services to device drivers, including:

1. Object management
2. Memory management
3. Process and thread management
4. Security Reference Monitor
5. Input/output management
6. Configuration management

EXECUTIVE SERVICES

MEMORY MANAGER

- Manages physical memory (RAM) for the operating system.
- Managing the allocation and de-allocation of memory virtually and dynamically.
- Supporting the concepts of memory-mapped files and shared memory

PROCESS & THREAD MANAGER

- A *process* is a software program that is currently running in Windows.
- A *thread* is an object that identifies which part of the program is running.
- It handles the execution of all threads in a process.
- Scheduling and synchronization

OBJECT MANAGER

- An object is a collection of data that the OS manages. Ex: Files, Devices, Registry keys
- Windows has more than 25 types of objects.
- Managing the creation and destruction of objects.
- Keeping track of objects assigned to each process.
- Managing the lifetime of an object

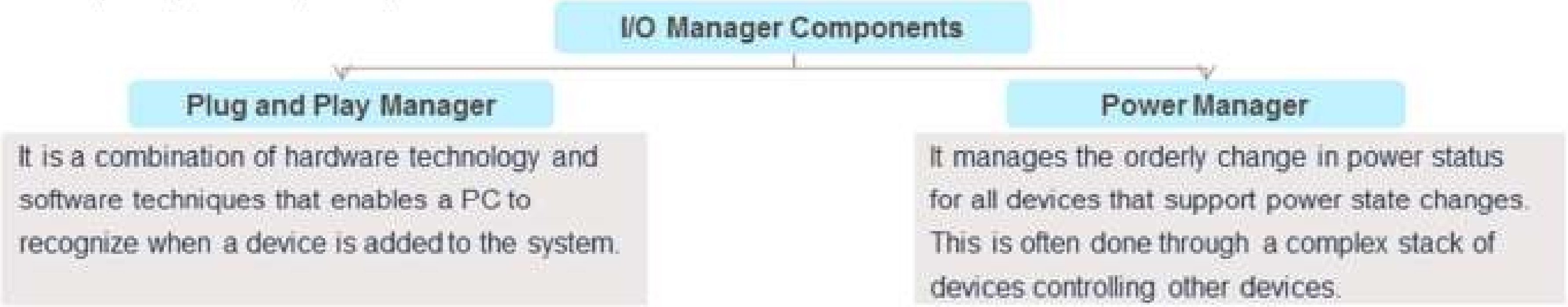
SECURITY REFERENCE MONITOR

- Windows uses an access control list (ACL) to determine which objects have what security.
- An access control entry (ACE) describes access rights associated with a particular SID.
- **Discretionary ACL**: ACEs that describe the access rights for a protected object.
- **System ACL**: ACEs that describe the auditing and alarm policy for a protected object.

EXECUTIVE SERVICES (Contd..)

I/O MANAGER:

- It manages the communication between applications and the interfaces provided by device drivers.
- Device drivers provide software connection between the devices like Keyboard and the operating system.
- I/O request packets (IRPs): Communicates between OS and device drivers



I/O Manager Components

Plug and Play Manager

It is a combination of hardware technology and software techniques that enables a PC to recognize when a device is added to the system.

Power Manager

It manages the orderly change in power status for all devices that support power state changes. This is often done through a complex stack of devices controlling other devices.

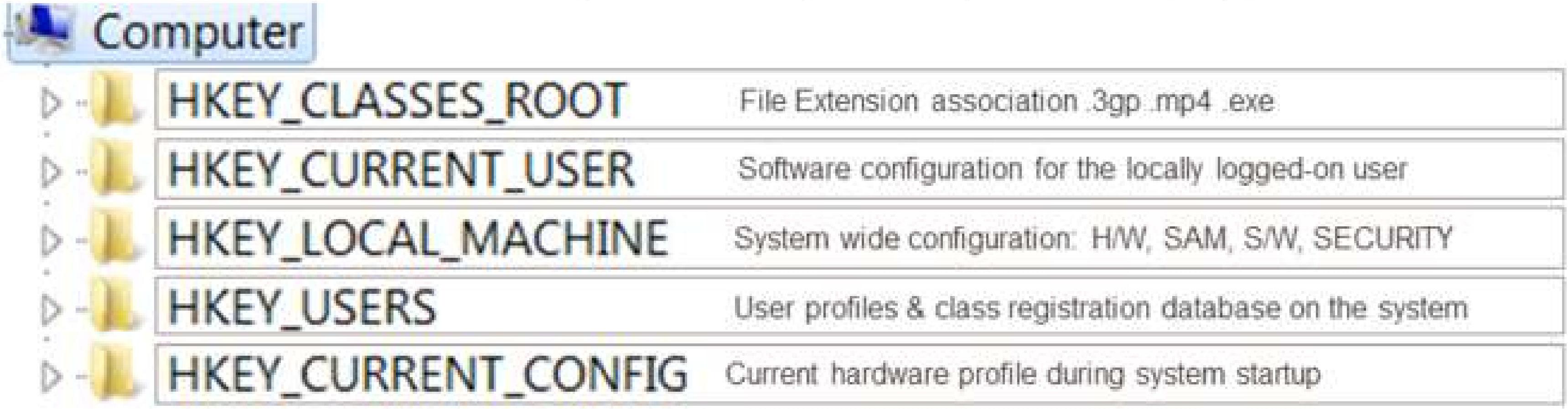
CACHE MANAGER

- Co-ordinates with Memory, I/O Manager and drivers to provide a common cache for regular file I/O.
- Windows Cache Manager operates on file blocks.

FILE SYSTEM DRIVERS

EXECUTIVE SERVICES (Contd..)

CONFIGURATION MANAGER: Responsible for implementing Windows Registry



Computer

HKEY_CLASSES_ROOT	File Extension association .3gp .mp4 .exe
HKEY_CURRENT_USER	Software configuration for the locally logged-on user
HKEY_LOCAL_MACHINE	System wide configuration: H/W, SAM, S/W, SECURITY
HKEY_USERS	User profiles & class registration database on the system
HKEY_CURRENT_CONFIG	Current hardware profile during system startup

LOCAL PROCEDURAL CALLS:

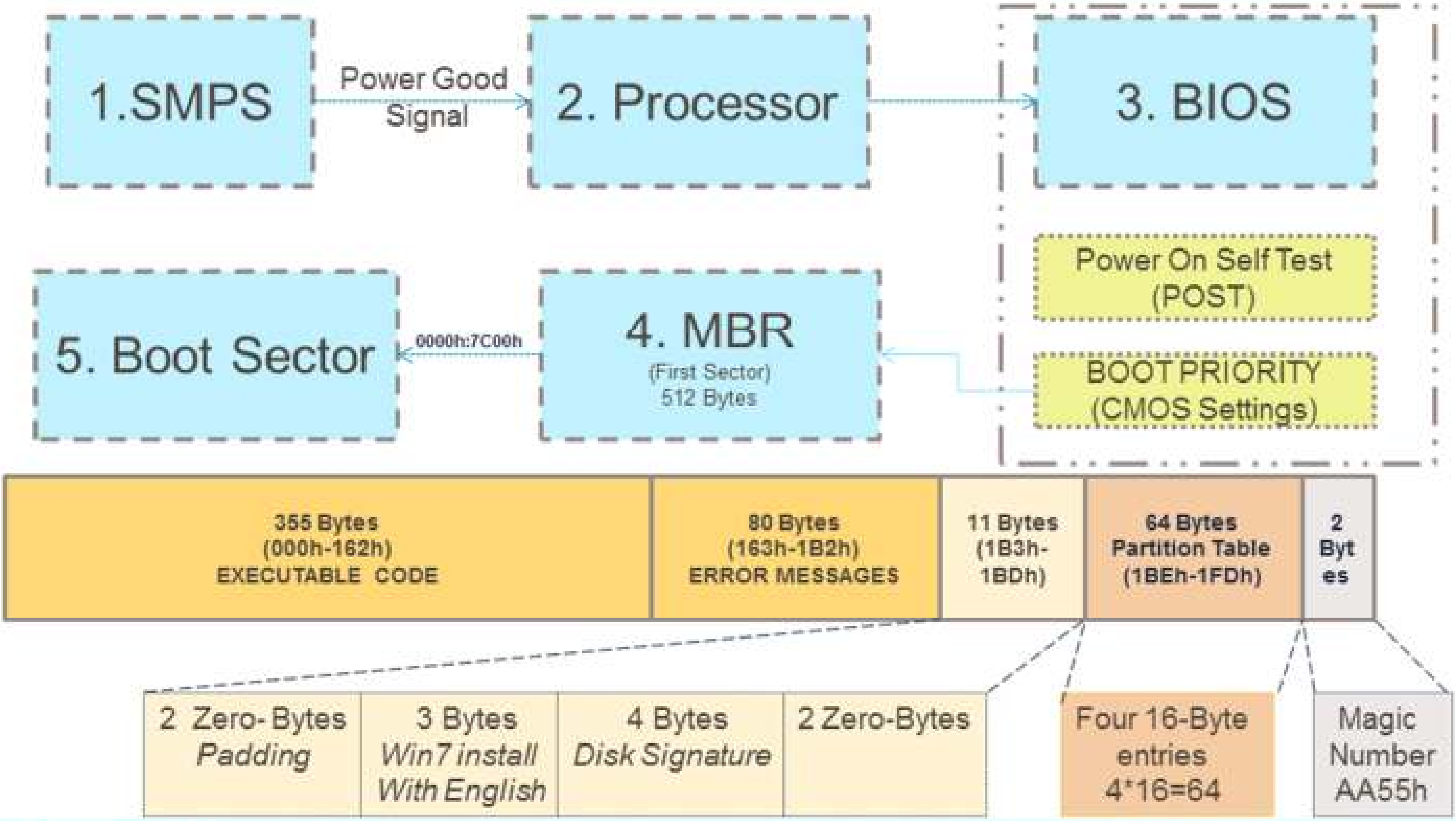
- LPC is a high speed message based communication mechanism implemented in the NT kernel.
- LPC can be used for communication between two user mode processes, between a user mode process and a kernel mode driver or between two kernel mode drivers.

KERNEL TRANSACTION MANAGER (KTM):

- It implements transaction processing in kernel mode.
- KTM allows kernel mode components, such as drivers, to perform transactions.

WINDOWS BOOTING PROCESS

BIOS Initialization | OS Loader | Kernel Initialization | Session Initialization | Winlogon Initialization | Explorer Initialization



1.SMPS

Power Good Signal

2. Processor

3. BIOS

Power On Self Test (POST)

BOOT PRIORITY (CMOS Settings)

4. MBR
(First Sector)
512 Bytes

0000h:7C00h

5. Boot Sector

355 Bytes (000h-162h) EXECUTABLE CODE

80 Bytes (163h-1B2h) ERROR MESSAGES

11 Bytes (1B3h-1BDh)

64 Bytes Partition Table (1BEh-1FDh)

2 Bytes

2 Zero-Bytes *Padding*

3 Bytes *Win7 install With English*

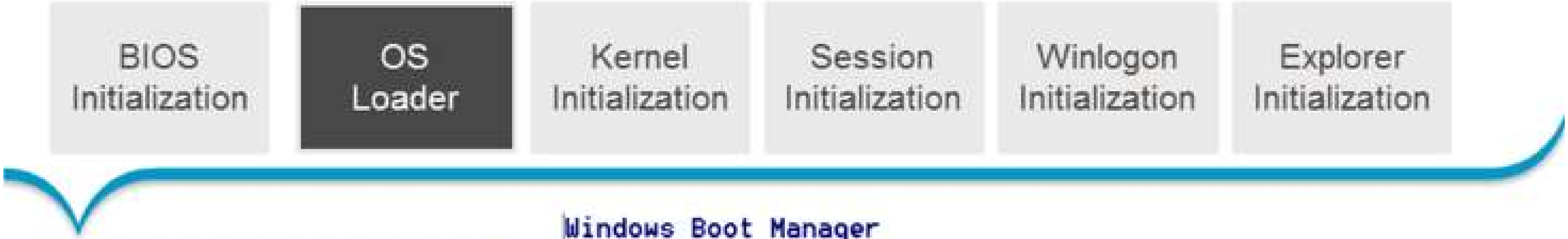
4 Bytes *Disk Signature*

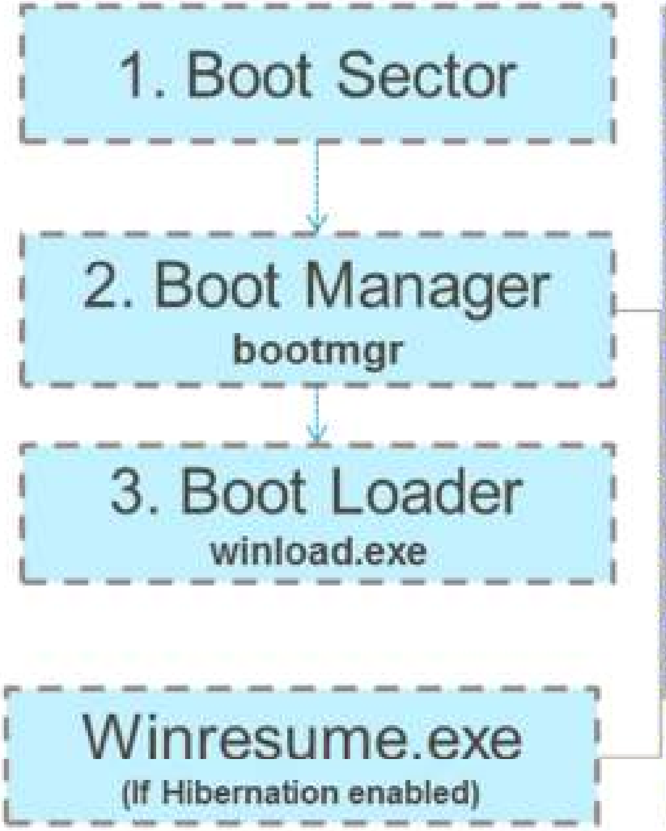
2 Zero-Bytes

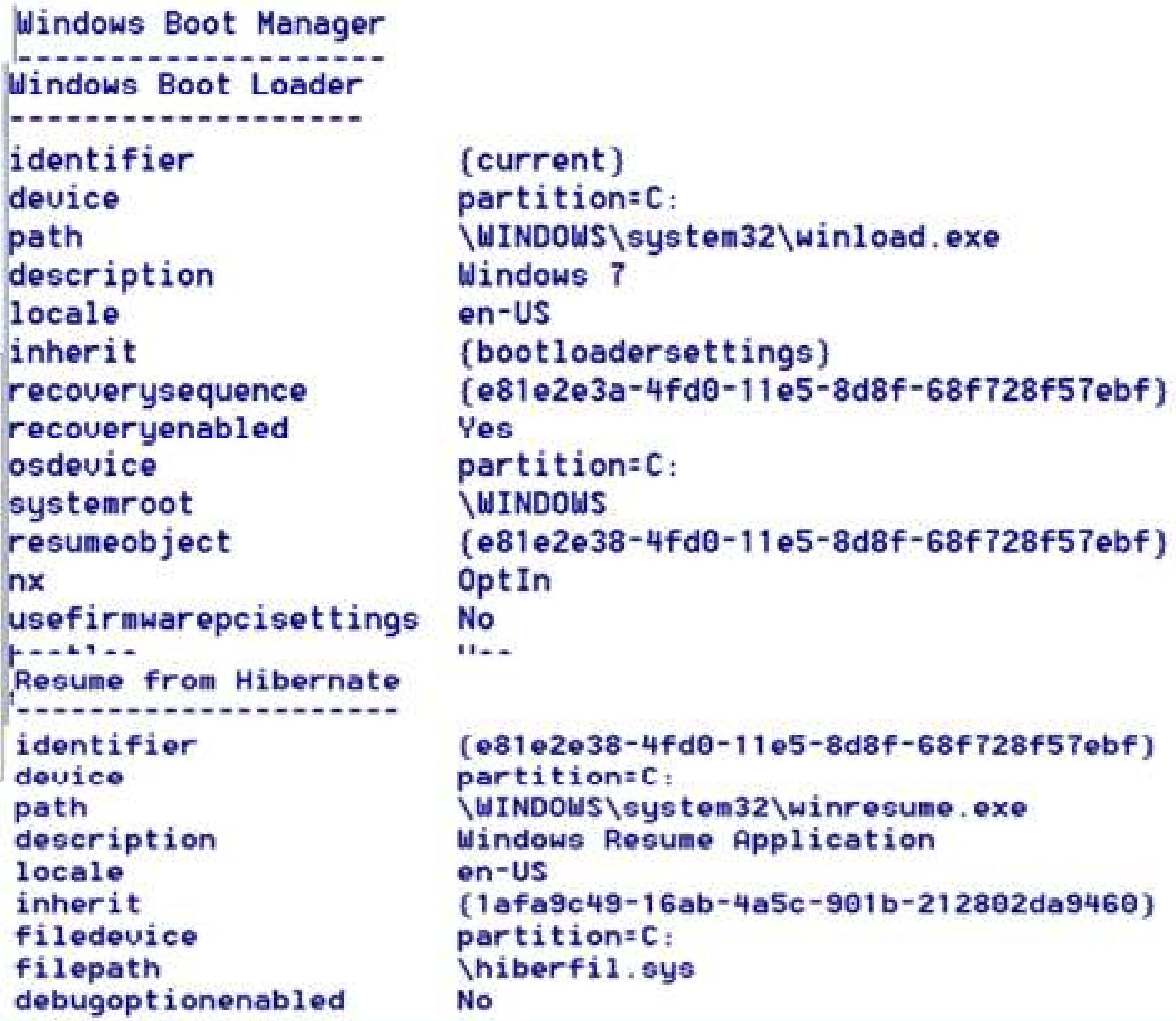
Four 16-Byte entries 4*16=64

Magic Number AA55h

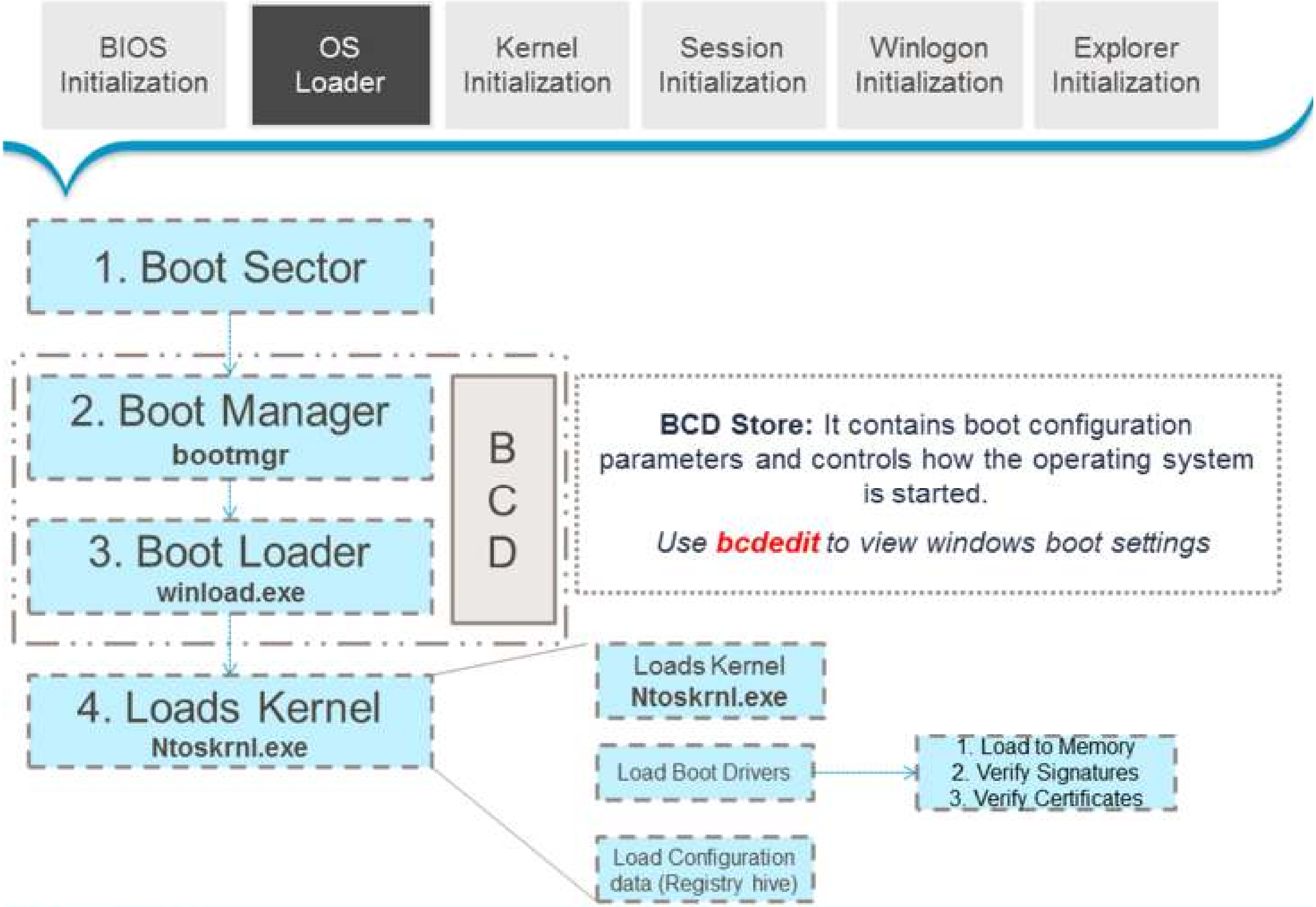
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING







```
Windows Boot Manager
--------------------
Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \WINDOWS\system32\winload.exe
description             Windows 7
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence        {e81e2e3a-4fd0-11e5-8d8f-68f728f57ebf}
recoveryenabled         Yes
osdevice                partition=C:
systemroot              \WINDOWS
resumeobject            {e81e2e38-4fd0-11e5-8d8f-68f728f57ebf}
nx                      OptIn
usefirmwarepcisettings  No
Resume from Hibernate
---------------------
identifier              {e81e2e38-4fd0-11e5-8d8f-68f728f57ebf}
device                  partition=C:
path                    \WINDOWS\system32\winresume.exe
description             Windows Resume Application
locale                  en-US
inherit                 {1afa9c49-16ab-4a5c-901b-212802da9460}
filedevice              partition=C:
filepath                \hiberfil.sys
debugoptionenabled      No
```



BIOS Initialization

OS Loader

Kernel Initialization

Session Initialization

Winlogon Initialization

Explorer Initialization

1. Boot Sector

2. Boot Manager
bootmgr

3. Boot Loader
winload.exe

4. Loads Kernel
Ntoskrnl.exe

B
C
D

BCD Store: It contains boot configuration parameters and controls how the operating system is started.

*Use **bcdedit** to view windows boot settings*

Loads Kernel
Ntoskrnl.exe

Load Boot Drivers

1. Load to Memory
2. Verify Signatures
3. Verify Certificates

Load Configuration data (Registry hive)

BIOS Initialization | OS Loader | **Kernel Initialization** | Session Initialization | Winlogon Initialization | Explorer Initialization

Boot Loader

winload.exe

Phase 0

Ntoskrnl.exe

1. Initialize kernel **(ntoskrnl.exe)**
2. Initialize **hal.dll, bootvid.dll**
4. Start the **debugger**
5. Initialize **routines for executive services**

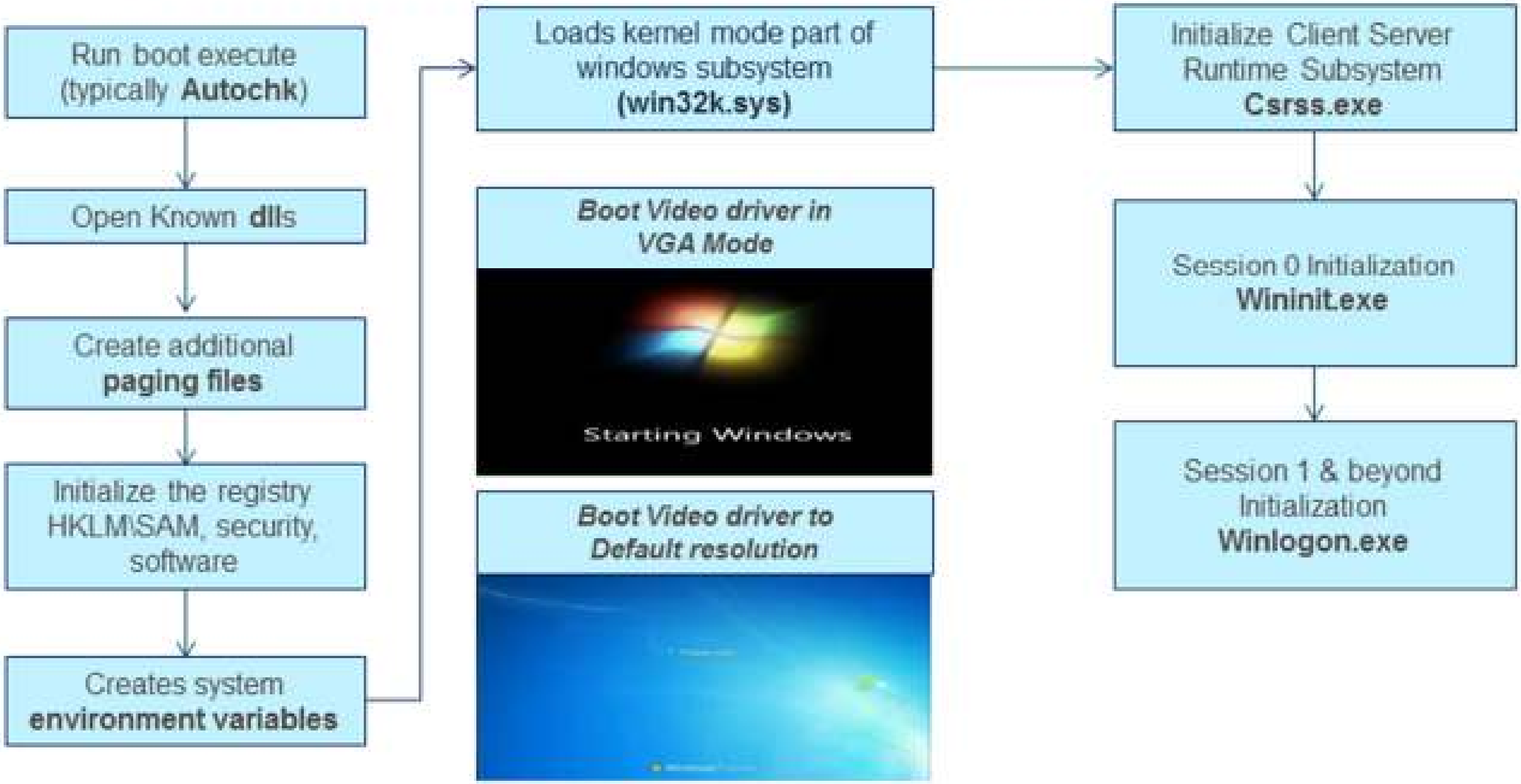
Use **msinfo32** to view boot drivers
System Information → Software Environment → System Drivers

Phase 1

1. Initialize kernel **(ntoskrnl.exe)**
2. Initialize **hal.dll, bootvid.dll**
4. Start the **debugger**
5. Initialize **routines for executive services**

BIOS Initialization | OS Loader | Kernel Initialization | **Session Initialization** | Winlogon Initialization | Explorer Initialization

Session Manager Sub System Smss.exe



Run boot execute (typically **Autochk**)

Open Known **dlls**

Create additional **paging files**

Initialize the registry HKLM\SAM, security, software

Creates system **environment variables**

Loads kernel mode part of windows subsystem **(win32k.sys)**

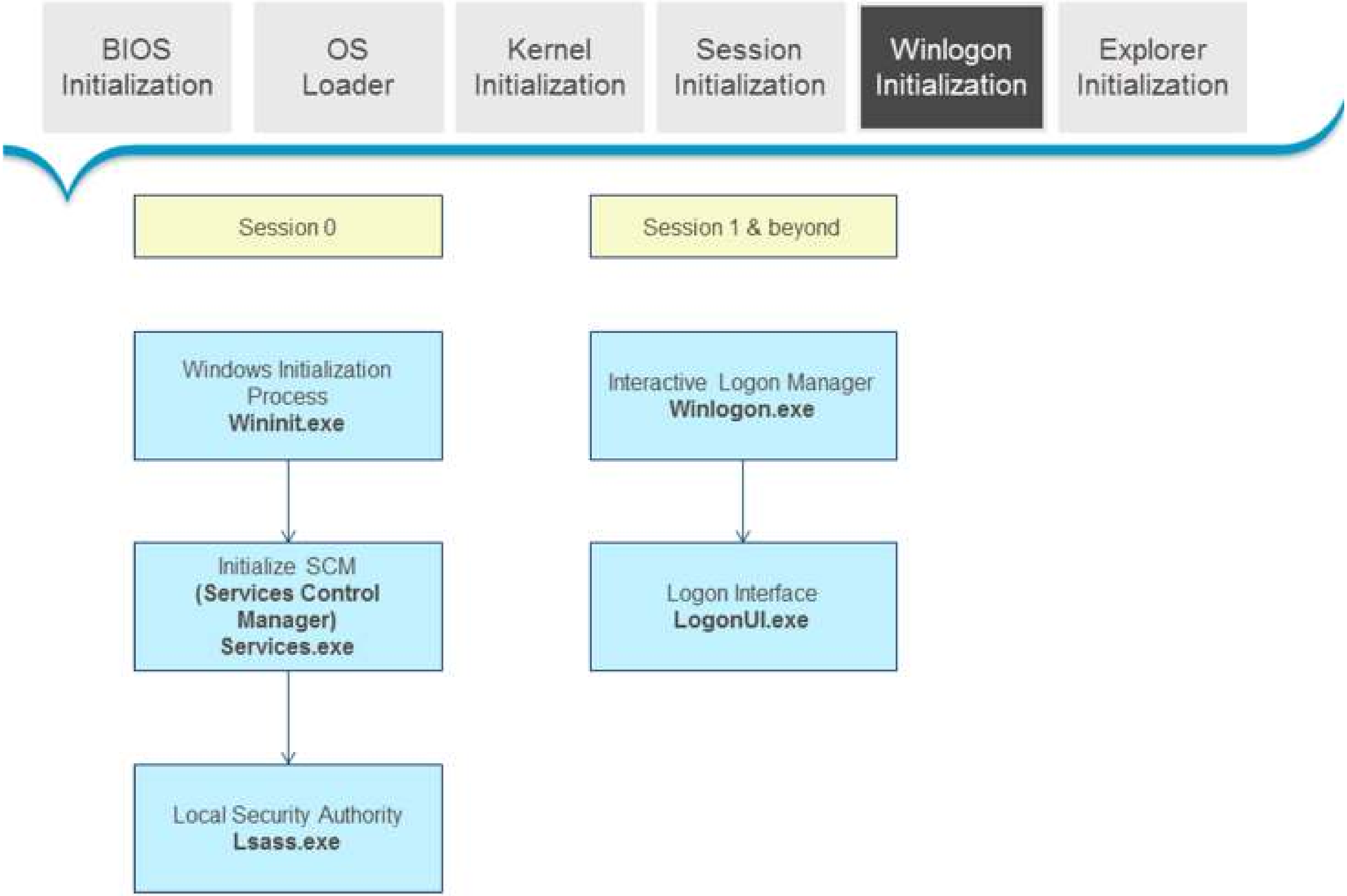
Boot Video driver in VGA Mode

Boot Video driver to Default resolution

Initialize Client Server Runtime Subsystem **Csrss.exe**

Session 0 Initialization **Wininit.exe**

Session 1 & beyond Initialization **Winlogon.exe**



BIOS Initialization

OS Loader

Kernel Initialization

Session Initialization

Winlogon Initialization

Explorer Initialization

Session 0

Windows Initialization Process
Wininit.exe

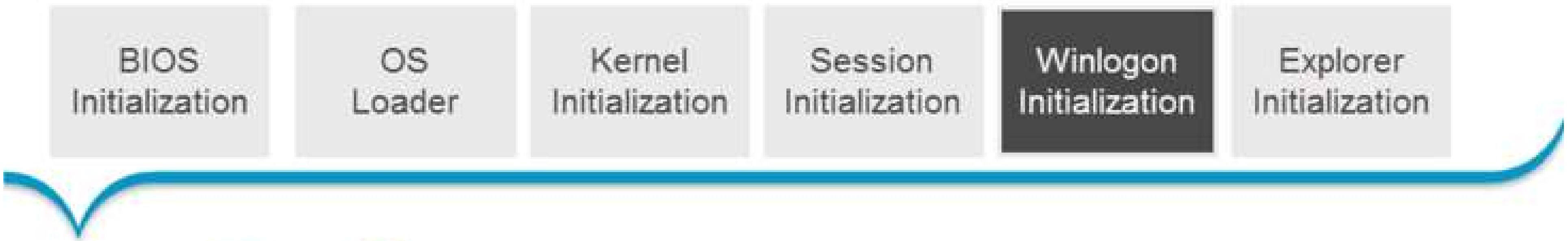
Initialize SCM
(Services Control Manager)
Services.exe

Local Security Authority
Lsass.exe

Session 1 & beyond

Interactive Logon Manager
Winlogon.exe

Logon Interface
LogonUI.exe



BIOS Initialization

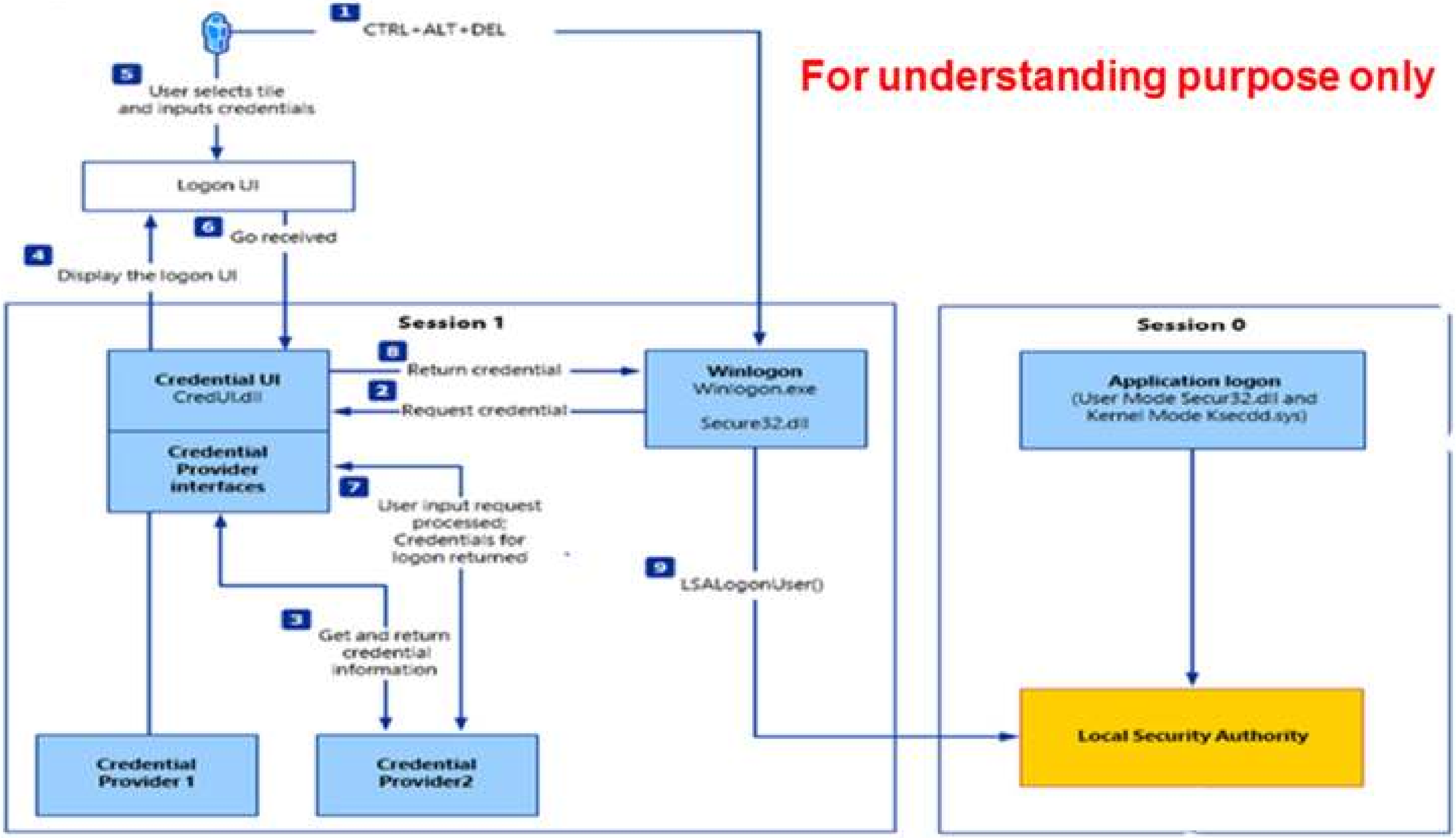
OS Loader

Kernel Initialization

Session Initialization

Winlogon Initialization

Explorer Initialization



For understanding purpose only

1 CTRL+ALT+DEL

5 User selects tile and inputs credentials

Logon UI

6 Go received

4 Display the logon UI

Session 1

Credential UI CredUI.dll

Credential Provider interfaces

8 Return credential

2 Request credential

Winlogon Winlogon.exe

Secure32.dll

7 User input request processed; Credentials for logon returned

3 Get and return credential information

9 LSALogonUser()

Credential Provider 1

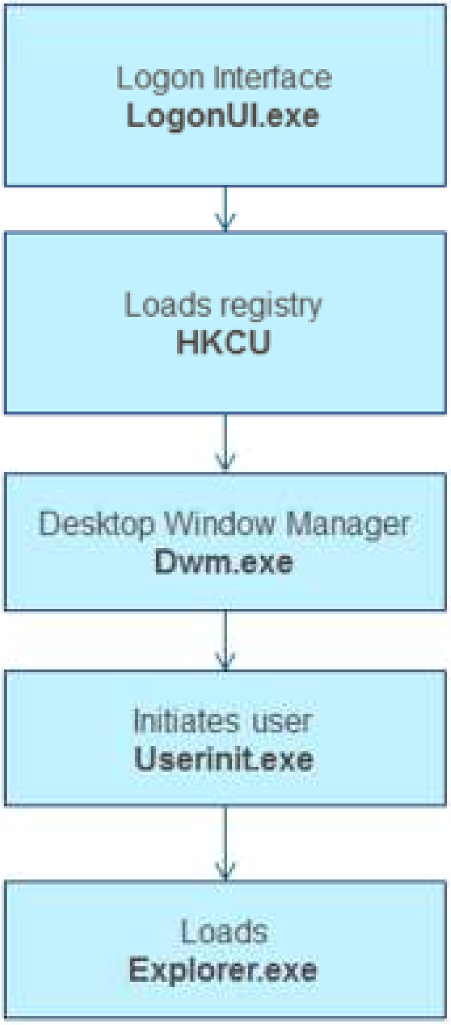
Credential Provider2

Session 0

Application logon (User Mode Secur32.dll and Kernel Mode Ksecdd.sys)

Local Security Authority

BIOS Initialization | OS Loader | Kernel Initialization | Session Initialization | Winlogon Initialization | **Explorer Initialization**



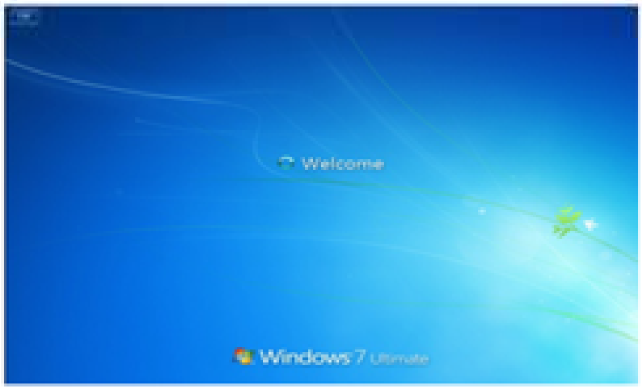
Logon Interface
LogonUI.exe

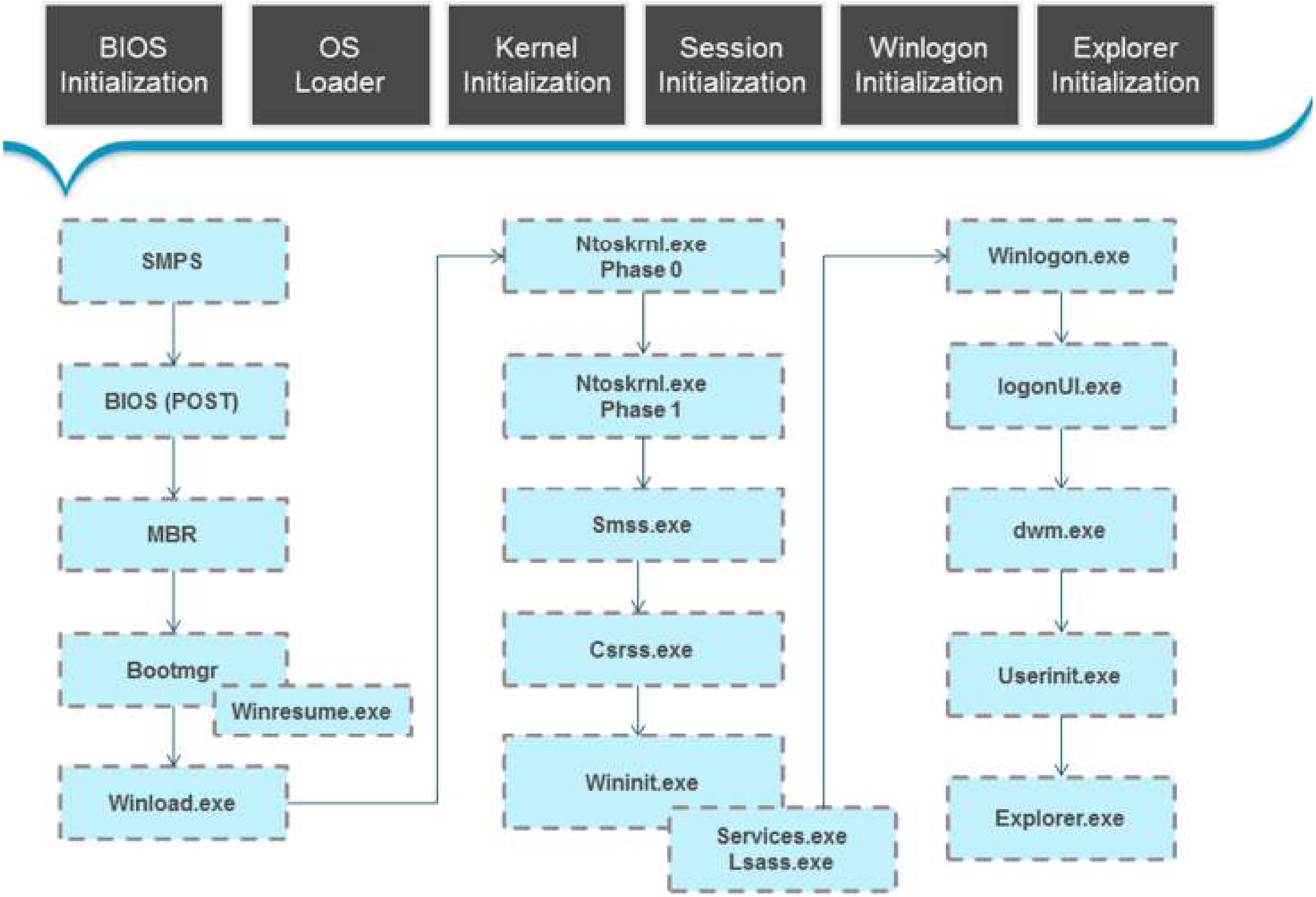
Loads registry
HKCU

Desktop Window Manager
Dwm.exe

Initiates user
Userinit.exe

Loads
Explorer.exe





BIOS Initialization

OS Loader

Kernel Initialization

Session Initialization

Winlogon Initialization

Explorer Initialization

SMPS

BIOS (POST)

MBR

Bootmgr

Winresume.exe

Winload.exe

Ntoskrnl.exe Phase 0

Ntoskrnl.exe Phase 1

Smss.exe

Csrss.exe

Wininit.exe

Services.exe
Lsass.exe

Winlogon.exe

logonUI.exe

dwm.exe

Userinit.exe

Explorer.exe

SAFE BOOT OPTIONS

- In safe mode, we can have access to only basic files and drivers (mouse, monitor, keyboard, mass storage base video, default system services)
- Safe mode helps us to diagnose problems.
- If a newly added device or a changed driver is causing problems, we can use safe mode to remove the device or reverse the change
- Windows Safe Mode bypasses startup programs and drivers that are not required for Windows to load and will allow you to fix Windows problems.

Safe Mode (SAFEBOOT_OPTION=Minimal):

This option uses a minimal set of device drivers and services to start Windows.

Safe Mode with Networking (SAFEBOOT_OPTION=Network):

This option uses a minimal set of device drivers and services to start Windows together with the drivers that you must have to load networking.

Safe Mode with Command Prompt (SAFEBOOT OPTION = Minimal(Alternate Shell)):

This option is the same as Safe mode, except that Cmd.exe starts instead of Windows Explorer.

Safe mode and Safe mode with Networking load the Vga.sys driver instead.

SAFE BOOT OPTIONS

Last Known Good Configuration:

This option starts Windows by using the previous good configuration.

Directory Service Restore Mode:

This mode is valid only for Windows-based domain controllers. This mode performs a directory service repair.

Enable Boot Logging:

This option turns on logging when the computer is started with any of the Safe Boot options except Last Known Good Configuration. The Boot Logging text is recorded in the Ntbtlog.txt file in the %SystemRoot% folder.

Starts Windows Normally:

This option starts Windows in its normal mode.

Reboot

This option restarts the computer.

Return to OS Choices Menu:

On a computer that is configured to starting to more than one operating system, this option returns to the Boot menu.

WINDOWS FILE SYSTEM

FILE SYSTEM IN WINDOWS

- **File System:**

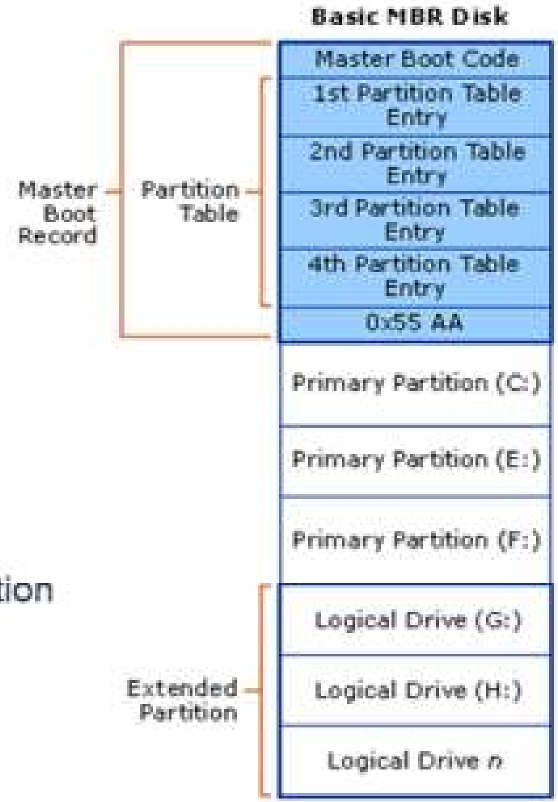
A file system is a method of storing and organizing the computer files and the data they contain to make it easy to find and access them.

- **Types of File System:**

1. FAT-File Allocation Table
2. NTFS- New Technology File System

FAT versions:
FAT12
FAT16
FAT32
exFAT

NTFS versions:
NTFS v1.0 from NT
NTFS v3.0 from Windows 2000
NTFS v3.1 from Windows XP



Basic MBR Disk

Master Boot Record

Partition Table

Master Boot Code

1st Partition Table Entry

2nd Partition Table Entry

3rd Partition Table Entry

4th Partition Table Entry

0x55 AA

Primary Partition (C:)

Primary Partition (E:)

Primary Partition (F:)

Extended Partition

Logical Drive (G:)

Logical Drive (H:)

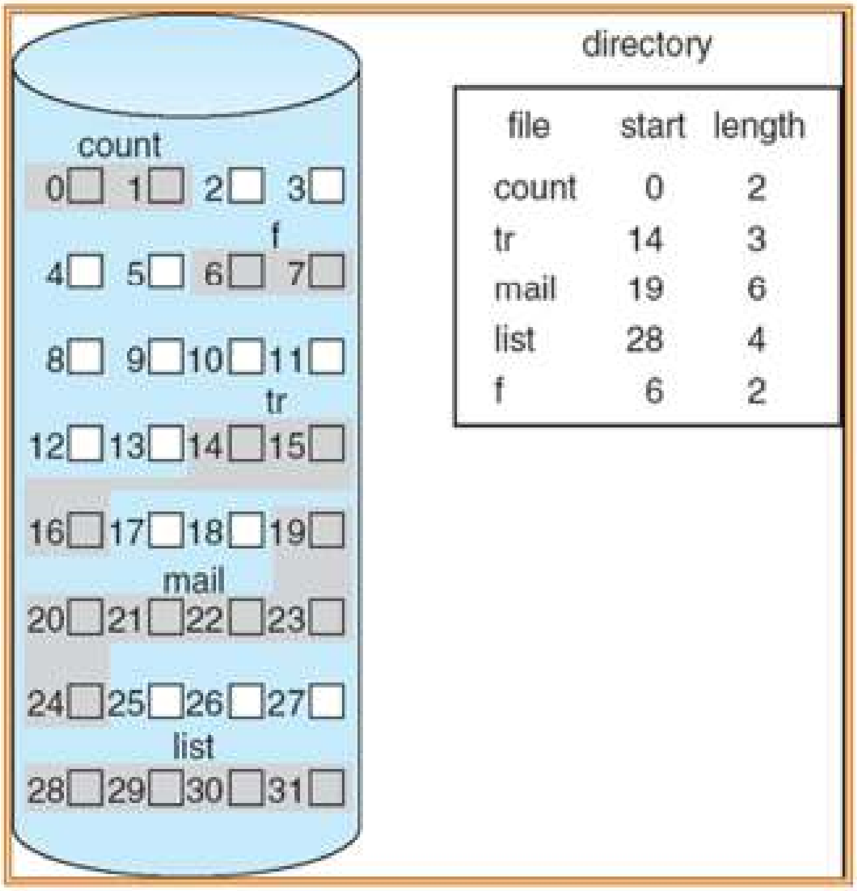
Logical Drive *n*

- **Terminology:**

1. Disk 2. Partition 3. Volume 4. Attribute 5. File operation

ALLOCATION METHODS (OF DATA BLOCKS)

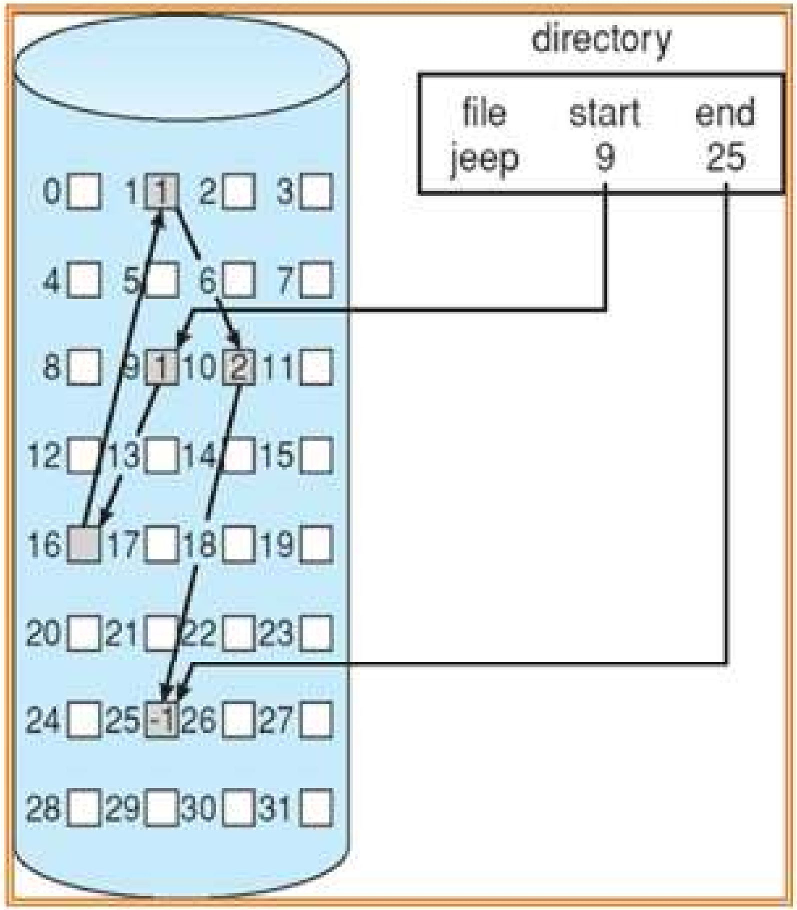
- Contiguous Allocation



directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

- Linked Allocation

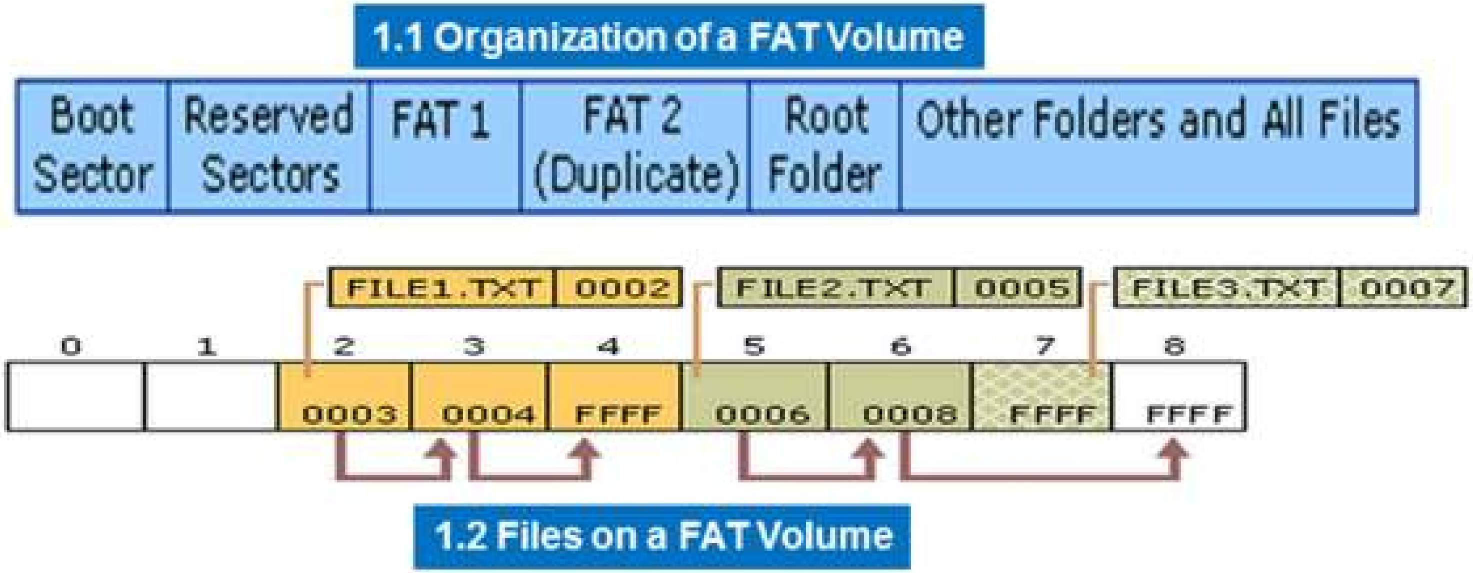


directory

file	start	end
jeep	9	25

FILE ALLOCATION TABLE (FAT32)

- FAT32 is an updated version of File Allocation Table (FAT).
- A disk formatted with FAT is allocated in clusters, whose size are determined by the size of the volume.
- When a file is created, an entry is created in the directory and the first cluster number containing data is established.
- This entry in the FAT table either indicates that this is the last cluster of the file, or points to the next cluster.
- FAT32 supports drive sizes from 512 MB up to 2 TB, although if you create and format a FAT32 partition through Windows 7, the FAT32 partition can only be up to 32 GB.



1.1 Organization of a FAT Volume

Boot Sector	Reserved Sectors	FAT 1	FAT 2 (Duplicate)	Root Folder	Other Folders and All Files

FILE1.TXT	0002

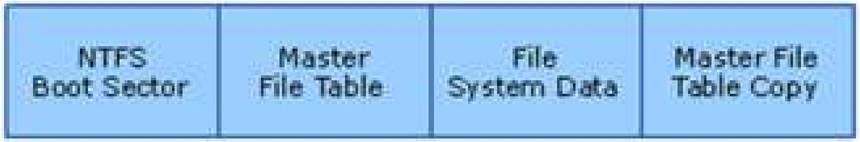
FILE2.TXT	0005

FILE3.TXT	0007

0	1	2	3	4	5	6	7	8
		0003	0004	FFFF	0006	0008	FFFF	FFFF

1.2 Files on a FAT Volume

NTFS



NTFS Boot Sector | Master File Table | File System Data | Master File Table Copy

NTFS Partition Boot Sector

- When you format an NTFS volume, the format program allocates the first 16 sectors for the $Boot metadata file.
- First sector, in fact, is a boot sector with a "bootstrap" code and the following 15 sectors are the boot sector's IPL (initial program loader).
- To increase file system reliability the very last sector an NTFS partition contains a spare copy of the boot sector.

Master File Table :

- Each file on an NTFS volume is represented by a record in a special file called the master file table (MFT). NTFS reserves the first 16 records of the table for special information.
- The MFT consists of a series of 1KB records, one for each file in the partition.
- The first record of this table describes the master file table itself, followed by a MFT mirror record.
- If the first MFT record is corrupted, NTFS reads the second record to find the MFT mirror file, whose first record is identical to the first record of the MFT.
- The locations of the data segments for both the MFT and MFT mirror file are recorded in the boot sector.
- The next ten include a changes log file for system recovery, information about the volume, the index of the root folder and a bitmap showing cluster allocation information.

NTFS ATTRIBUTES

Resident Attributes :

Contains 4 Attributes

Attribute 1	Having the file attributes such as the archive bit, which shows whether the file has been backed up, and timestamps showing when the file was created, last modified and last accessed
Attribute 2	Contains the filenames. Each file can have multiple names NTFS supports names up to 255 Unicode characters
Attribute 3	Security Descriptor, contains the Access Control List (ACL) data for the file.
Attribute 4	The VCN is a sequential number relating to each extent of consecutive clusters on the disk which contain the file, the LCN refers to the location of the first cluster of each extent

Non-resident attributes are ones too large to fit in the MFT record.

FEATURES OF NTFS

Disk Quotas

- Disk quotas are a new feature in NTFS that provide more precise control of network-based storage.
- Disk quotas are implemented on a per-volume basis and enable both hard and soft storage limits to be implemented on a per-user basis.

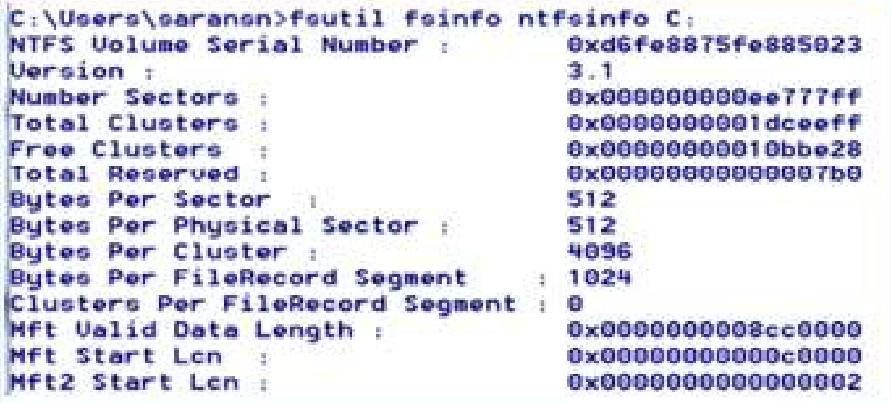
Sparse File Support

- Sparse files allow programs to create very large files, but to consume disk space only as needed.
- A sparse file is a file with an attribute that causes the I/O subsystem to allocate the file's meaningful (nonzero) data.
- NTFS includes full sparse file support for both compressed and uncompressed files.

Reparse point

- A reparse point is a special NTFS feature that Windows uses to identify and manage mount points for drives and junction links for directories.
- Think of a mount point as the place where Windows connects physical volumes to logical entries such as drives and folders.
- This attribute is used when a symbolic link or mount point is created.

NTFS v3.1 settings



```
C:\Users\saranon>fsutil fsinfo ntfsinfo C:
NTFS Volume Serial Number :         0xd6fe8875fe885023
Version :                           3.1
Number Sectors :                    0x00000000000ee777ff
Total Clusters :                    0x000000000001dceeff
Free Clusters  :                    0x0000000000010bbe28
Total Reserved :                    0x00000000000000007b0
Bytes Per Sector  :                 512
Bytes Per Physical Sector :         512
Bytes Per Cluster :                 4096
Bytes Per FileRecord Segment      : 1024
Clusters Per FileRecord Segment : 0
Mft Valid Data Length :             0x0000000008cc0000
Mft Start Lcn  :                    0x00000000000c0000
Mft2 Start Lcn :                    0x0000000000000002
```

FILE ATTRIBUTES

Archive Bit

- The archive bit is a file attribute that is set whenever a file is modified.
- For backups that use archive bits, this bit is turned off after the backup completes, indicating to the system that the file has been backed up.
- If the file is changed again before the next backup, the bit will be turned on.

Hidden bit

- The purpose of the Hidden attribute bit is to make the file invisible in certain applications' file list display.
- "Show hidden files" options should be enabled to view the hidden files.

Compression

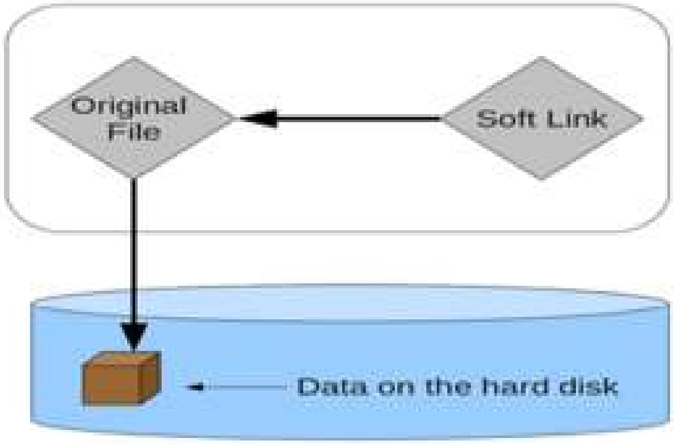
- **Windows** supports compression on individual files, folders, and entire NTFS volumes. Only **NTFS** can read the compressed form of the data.
- The compression algorithms in NTFS are designed to support cluster sizes of up to 4 KB.
- When writing a compressed file, the system reserves disk space for the uncompressed size.

Encrypting File System

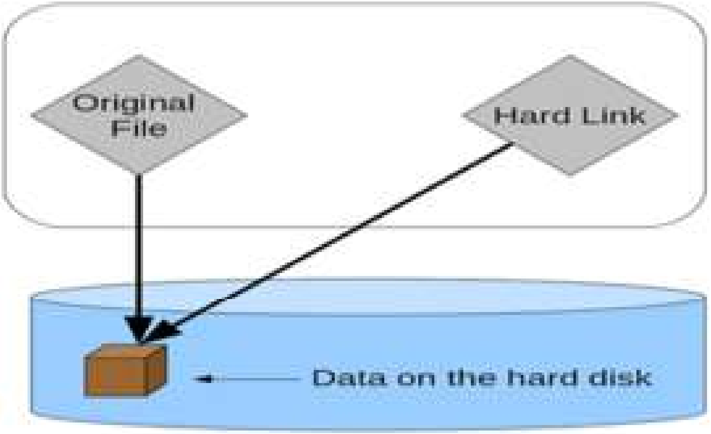
- File and directory-level encryption is implemented in the version of NTFS included with Windows for enhanced security in NTFS volumes.
- Windows uses Encrypting File System (EFS) to store data in encrypted form, which provides security when the storage media are removed from a system.

LINKS IN NTFS

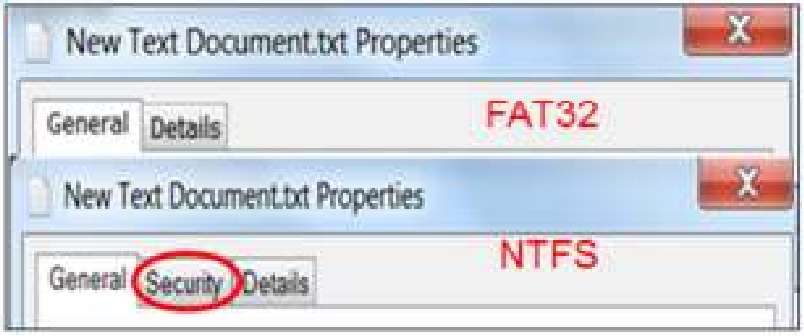
- A symbolic link is any link that redirects the file system from one location to another.
- The junction point is a symbolic link typically used for folders.
- A soft or symbolic link behaves similar to a Windows shortcut.



- A hard link is a direct pointer to the data on the hard disk. A hard disk is identical to the original file, and any modifications made to the hard linked version are made to the original as well, since you are modifying the same physical space on the hard disk



FAT32 Vs NTFS



New Text Document.txt Properties

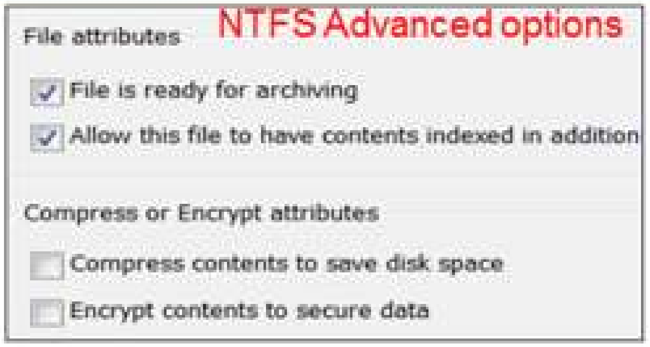
General | Details

FAT32

New Text Document.txt Properties

General | Security | Details

NTFS



NTFS Advanced options

File attributes

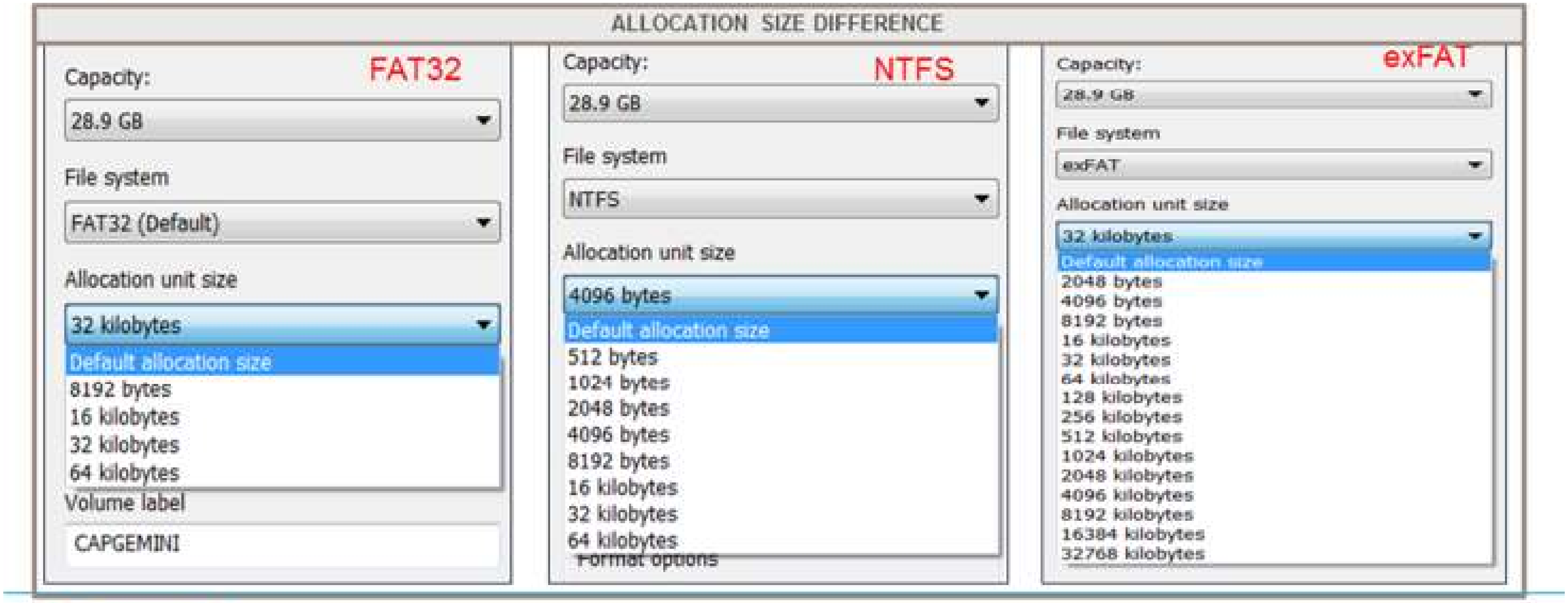
☑ File is ready for archiving

☑ Allow this file to have contents indexed in addition

Compress or Encrypt attributes

□ Compress contents to save disk space

□ Encrypt contents to secure data



ALLOCATION SIZE DIFFERENCE

FAT32

Capacity: 28.9 GB

File system: FAT32 (Default)

Allocation unit size: 32 kilobytes

- Default allocation size
- 8192 bytes
- 16 kilobytes
- 32 kilobytes
- 64 kilobytes

Volume label: CAPGEMINI

NTFS

Capacity: 28.9 GB

File system: NTFS

Allocation unit size: 4096 bytes

- Default allocation size
- 512 bytes
- 1024 bytes
- 2048 bytes
- 4096 bytes
- 8192 bytes
- 16 kilobytes
- 32 kilobytes
- 64 kilobytes

Format options

exFAT

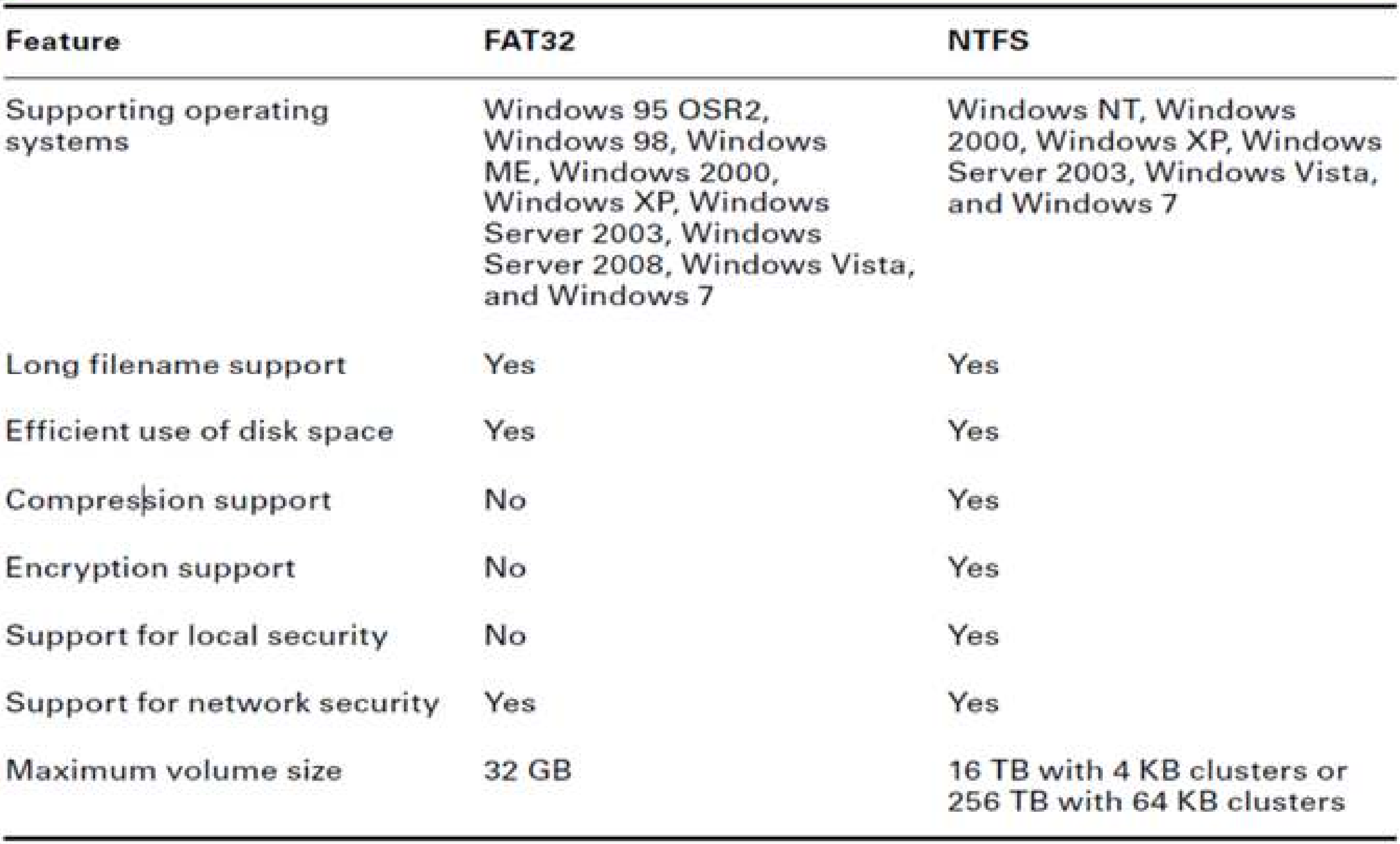
Capacity: 28.9 GB

File system: exFAT

Allocation unit size: 32 kilobytes

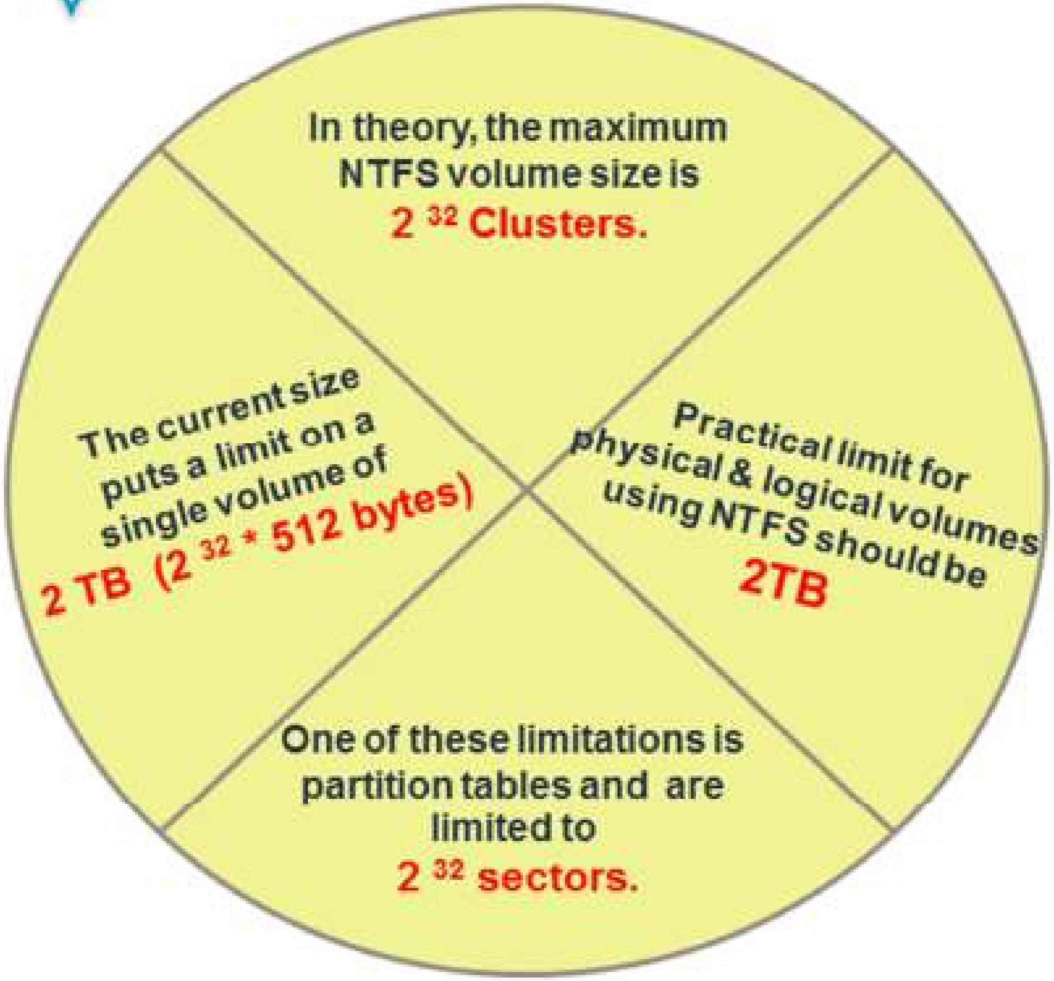
- Default allocation size
- 2048 bytes
- 4096 bytes
- 8192 bytes
- 16 kilobytes
- 32 kilobytes
- 64 kilobytes
- 128 kilobytes
- 256 kilobytes
- 512 kilobytes
- 1024 kilobytes
- 2048 kilobytes
- 4096 kilobytes
- 8192 kilobytes
- 16384 kilobytes
- 32768 kilobytes

FAT32 Vs NTFS



Feature	FAT32	NTFS
Supporting operating systems	Windows 95 OSR2, Windows 98, Windows ME, Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, and Windows 7	Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows 7
Long filename support	Yes	Yes
Efficient use of disk space	Yes	Yes
Compression support	No	Yes
Encryption support	No	Yes
Support for local security	No	Yes
Support for network security	Yes	Yes
Maximum volume size	32 GB	16 TB with 4 KB clusters or 256 TB with 64 KB clusters

PRACTICAL LIMITS OF NTFS



In theory, the maximum NTFS volume size is 2^{32} Clusters.

The current size puts a limit on a single volume of 2 TB (2^{32} * 512 bytes)

Practical limit for physical & logical volumes using NTFS should be 2TB

One of these limitations is partition tables and are limited to 2^{32} sectors.

Cluster size	NTFS Max Size
512 bytes	2,199,023,255,040 (2TB)
1024 bytes	4,398,046,510,080 (4TB)
2048 bytes	8,796,093,020,160 (8TB)
4096 bytes	17,592,186,040,320 (16TB) Default cluster size
8192 bytes	35,184,372,080,640 (32TB)
16384 bytes	70,368,744,161,280 (64TB)
32768 bytes	140,737,488,322,560 (128TB)
65536 bytes	281,474,976,654,120 (256TB)

To view sector and cluster size:

fsutil fsinfo ntfsinfo C:

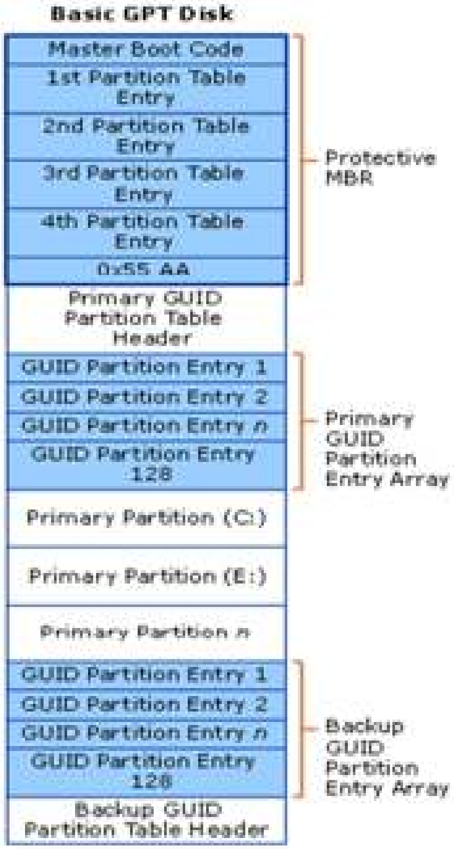
To perform check disk:

chkdsk

GUID PARTITION TABLE (GPT)

- GPT was first introduced as part of the Extensible Firmware Interface (EFI) initiative from Intel.
- GPT header and partition table is written to both the front and the back end of the disk, which in turn provides for better redundancy.
- GPT uses a newer addressing scheme called Logical Block Addressing (LBA).
  - Allows a volume size larger than 2 TB
  - Allow up to 128 primary partitions
  - Used for both 32 - bit or 64 - bit Windows 7 editions
  - Includes Cyclical Redundancy Check (CRC) for greater reliability

Block:	Contents:			
LBA 0	Protective MBR			
LBA 1	Primary GPT Header			
LBA 2	Entry 1	Entry 2	Entry 3	Entry 4
LBA 3	Entries 5 – 128			
LBA 34 to LBA -34	Partition 1			
	Partition 2			
	Remaining Partitions			
LBA – 33	Entry 1	Entry 2	Entry 3	Entry 4
LBA – 2	Entries 5 – 128			
LBA – 1	Secondary GPT Header			



Basic GPT Disk

Master Boot Code
1st Partition Table Entry
2nd Partition Table Entry
3rd Partition Table Entry
4th Partition Table Entry
0x55 AA
Primary GUID Partition Table Header
GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry *n*
GUID Partition Entry 128
Primary Partition (C:)
Primary Partition (E:)
Primary Partition *n*
GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry *n*
GUID Partition Entry 128
Backup GUID Partition Table Header

Protective MBR
Primary GUID Partition Entry Array
Backup GUID Partition Entry Array

🟩 Primary

🟦 Secondary

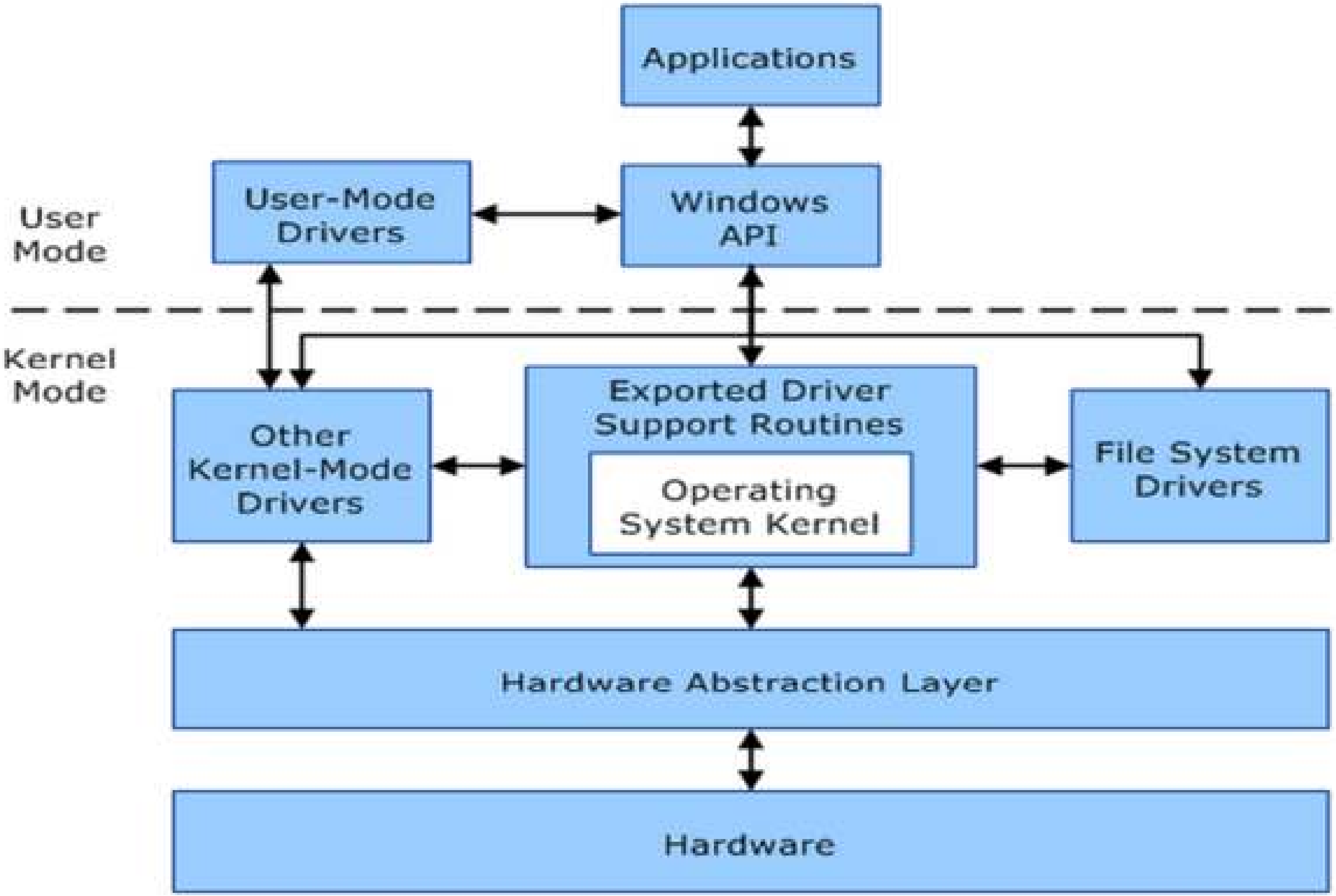
FILE SYSTEMS COMPARISON

For understanding purpose only

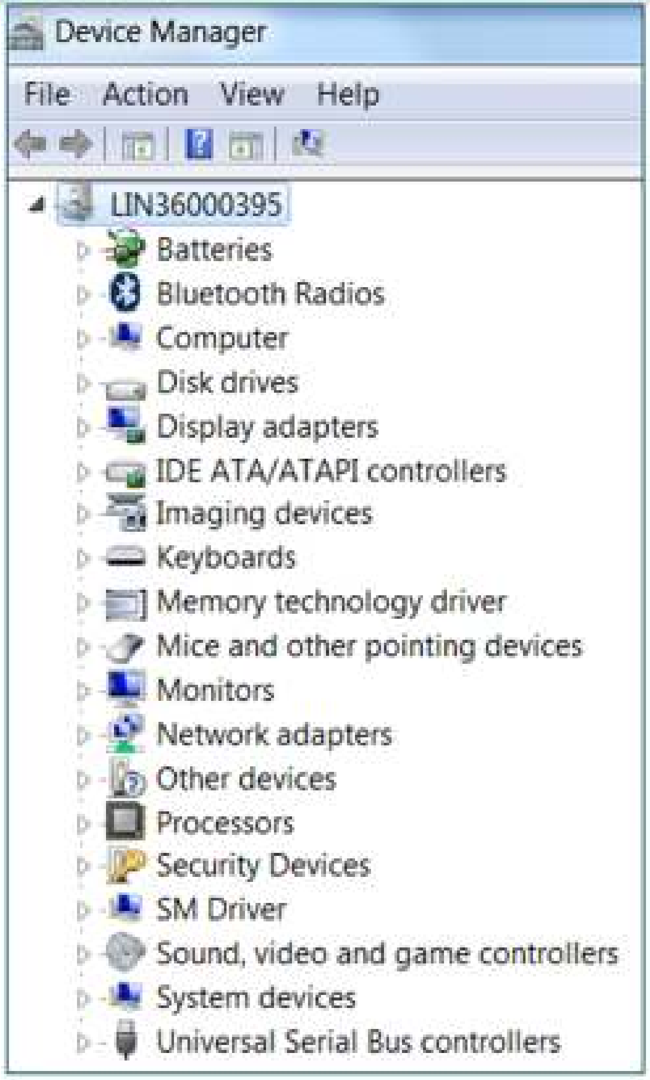
Criteria	NTFS5	NTFS	exFAT	FAT32	FAT16	FAT12
Operating System	Windows 7	Windows 7	Windows 7	Windows 7		
Limitations						
Max Volume Size	2^{64} -1 clusters	2^{32} -1 clusters	128PB	32GB for all OS. 2TB for some OS	4GB	16MB
Max Files on Volume	4,294,967,295 (2^{32} -1)	4,294,967,295 (2^{32} -1)	Nearly Unlimited	4194304	65536 (2^{16})	
Max File Size	2^{64} -1 (16 ExaBytes) minus 1KB	2^{44} -1 (16 TeraBytes) minus 64KB	16EB	4GB minus 2 Bytes	2^{32} -1 (Limit Only by Volume Size)	16MB (Limit Only by Volume Size)
Max Clusters Number	2^{64} -1	2^{64} -1	4294967295	4177918	65520	4080
Max File Name Length	Up to 255	Up to 255	Up to 255	Up to 255	Stnd - 8.3 Extn- to 255	Up to 254
File System Features						
Unicode File Names	Unicode Character Set	Unicode Character Set	Unicode Character Set	System Character Set	System Character Set	System Character Set
System Records Mirror	MFT Mirror File	MFT Mirror File	No	Second Copy of FAT	Second Copy of FAT	Second Copy of FAT
Boot Sector Location	First and Last Sectors	First and Last Sectors	Sectors 0 to 11 Copy in 12 to 23	First Sector and Copy in Sector #6	First Sector	First Sector
File Attributes	Standard and Custom	Standard and Custom	Standard Set	Standard Set	Standard Set	Standard Set
Alternate Streams	Yes	Yes	No	No	No	No
Compression	Yes	Yes	No	No	No	No
Encryption	Yes	No	No	No	No	No
Object Permissions	Yes	Yes	Yes	No	No	No
Disk Quotas	Yes	No	No	No	No	No
Sparse Files	Yes	No	No	No	No	No
Reparse Points	Yes	No	No	No	No	No
Volume Mount Points	Yes	No	No	No	No	No
Overall Performance						
Built-In Security	Yes	Yes	Yes minimal ACL only	No	No	No
Recoverability	Yes	Yes	Yes if TFAT activated	No	No	No
Performance	Low on small volumes High on Large	Low on small volumes High on Large	High	High on small volumes Low on large	Highest on small volumes Low on large	High
Disk Space Economy	Max	Max	Max	Average	Minimal on large volumes	Max
Fault Tolerance	Max	Max	Yes if TFAT activated	Minimal	Average	Average

WINDOWS MANAGEMENT

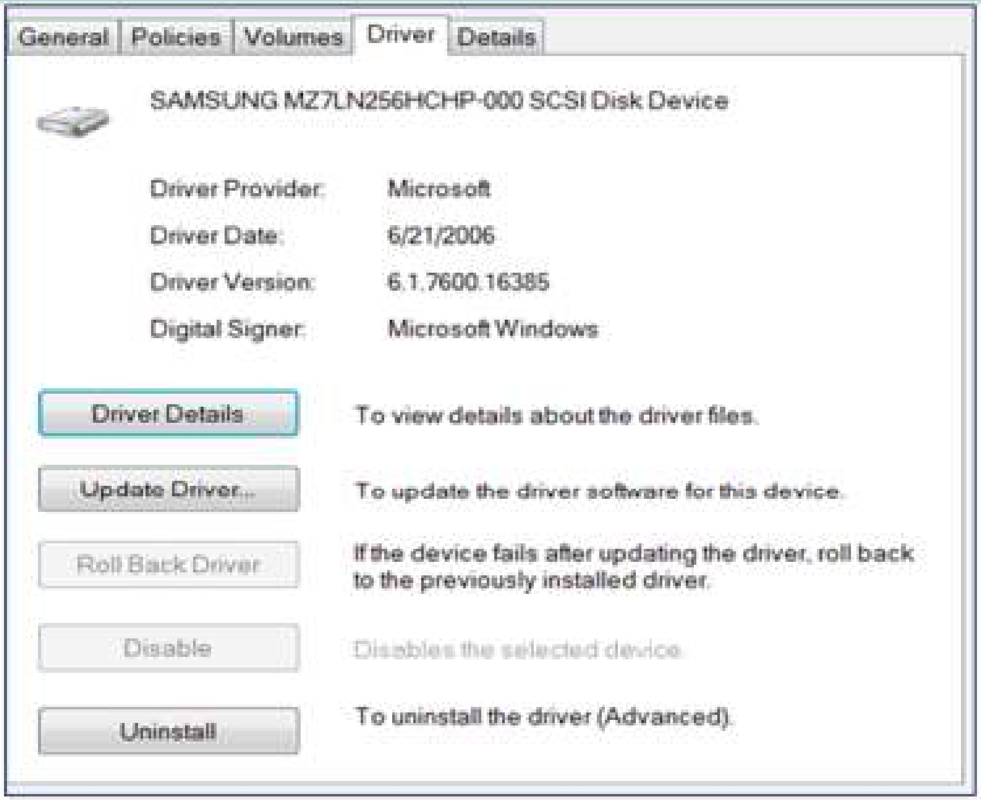
DEVICE MANAGEMENT



DEVICE MANAGEMENT



- LIN36000395
  - Batteries
  - Bluetooth Radios
  - Computer
  - Disk drives
  - Display adapters
  - IDE ATA/ATAPI controllers
  - Imaging devices
  - Keyboards
  - Memory technology driver
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
  - Processors
  - Security Devices
  - SM Driver
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers



SAMSUNG MZ7LN256HCHP-000 SCSI Disk Device

Driver Provider: Microsoft

Driver Date: 6/21/2006

Driver Version: 6.1.7600.16385

Digital Signer: Microsoft Windows

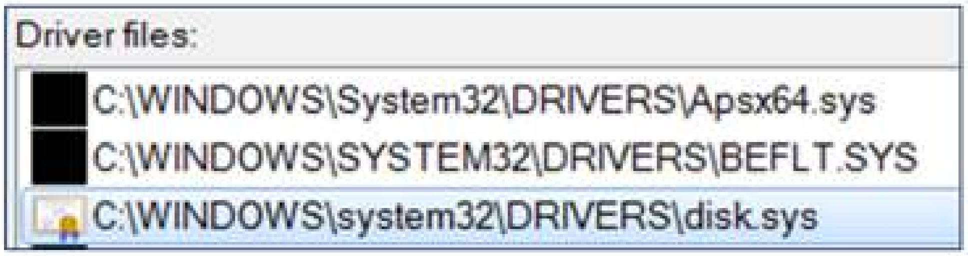
Driver Details — To view details about the driver files.

Update Driver... — To update the driver software for this device.

Roll Back Driver — If the device fails after updating the driver, roll back to the previously installed driver.

Disable — Disables the selected device.

Uninstall — To uninstall the driver (Advanced).


Driver files:

C:\WINDOWS\System32\DRIVERS\Apsx64.sys
C:\WINDOWS\SYSTEM32\DRIVERS\BEFLT.SYS
C:\WINDOWS\system32\DRIVERS\disk.sys

SERVICE CONTROL MANAGER

- The Service Control Manager (SCM) maintains a database of the installed services and driver services that allow the operating system to start successfully, and provides a unified and secure means of controlling them.
- The database, which is stored in the Windows system registry, includes configuration and security information about each service or driver service.
- System administrators should use the **Services** snap-in or the **sc.exe** command-line tool to query or configure services

POWER MANAGER

State	Power Consumption	Software Resumption	Hardware Latency
S0 (fully on)	Maximum	Not applicable	None
S1 (light sleep)	Less than S0, more than S2	System resumes where it left off (returns to S0)	Less than 2 seconds
S2 (deep sleep)	Less than S1, more than S3	System resumes where it left off (returns to S0)	2 or more seconds
S3 (deepest sleep)	Less than S2; processor is off	System resumes where it left off (returns to S0)	Same as S2
S4 (hibernating)	Trickle current to power button and wake circuitry	System restarts from saved hibernate file and resumes where it left off prior to hibernation (returns to S0)	Long and undefined
S5 (fully off)	Trickle current to power button	System boot	Long and undefined

- Windows Kernel-Mode WMI Library
- Windows provides a general mechanism for managing components. This system is called Windows Management Instrumentation (WMI). To satisify Windows Driver Model (WDM) requirements, you should implement WMI for your driver so that your driver can be managed by the system

WINDOWS PROCESS MANAGEMENT

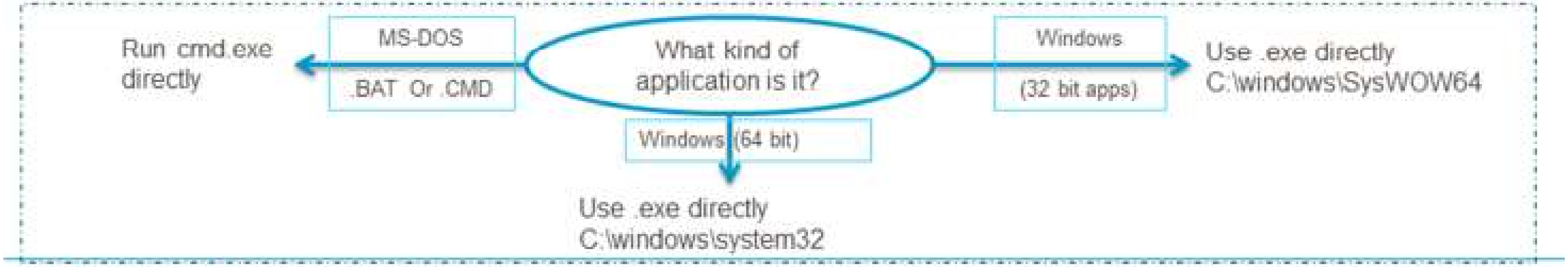
PROCESS AND THREAD

What is a process?

- Represents an instance of a running program in a sequential manner.
- The resources are allocated when a process is created or while in execution to run a program
- Process is defined by 1. Address spaces 2. Resources (Handles) 3. Security profile (token)

What is a thread?

- An execution context within a process
- All threads in a process share the same per-process address space
- Unit of scheduling (threads run, processes don"t run)
- A process can have multiple **threads** simultaneously executing the same function.



What kind of application is it?

MS-DOS

.BAT Or .CMD

Run cmd.exe directly

Windows

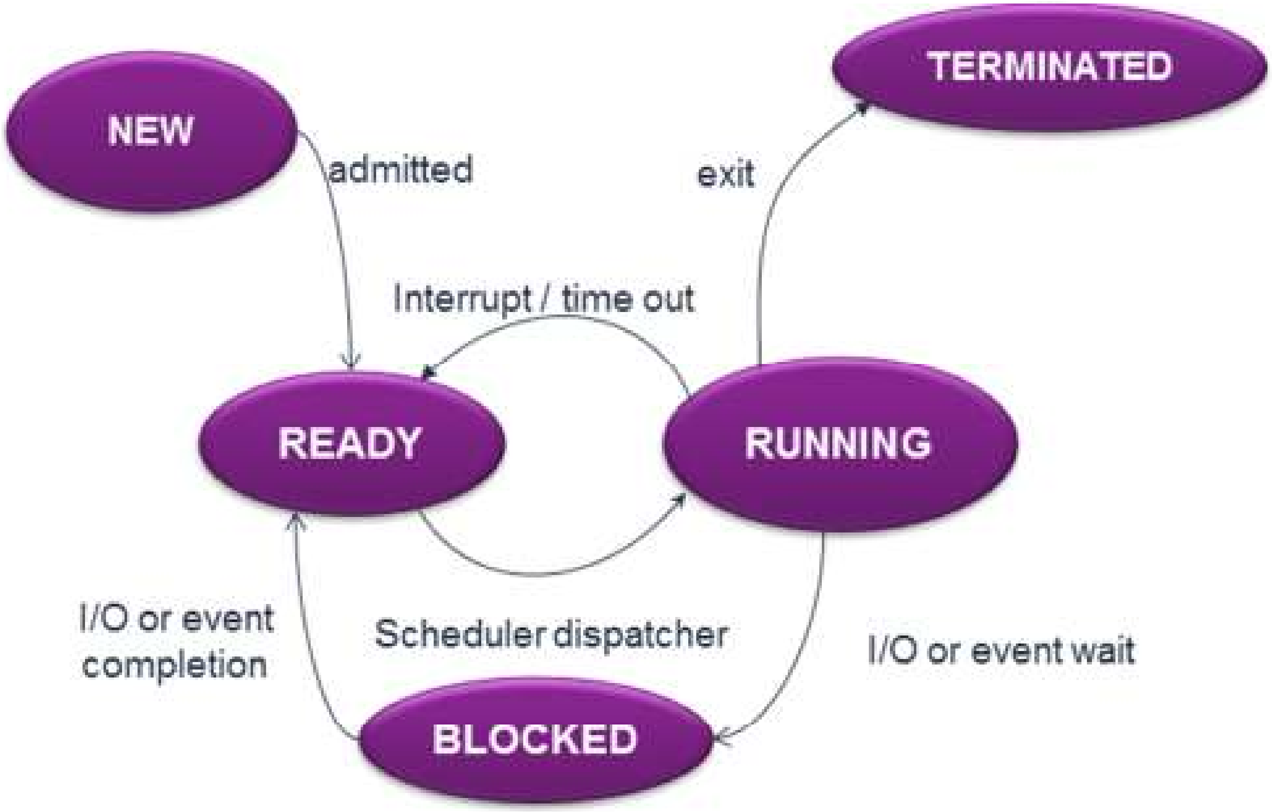
(32 bit apps)

Use .exe directly C:\windows\SysWOW64

Windows (64 bit)

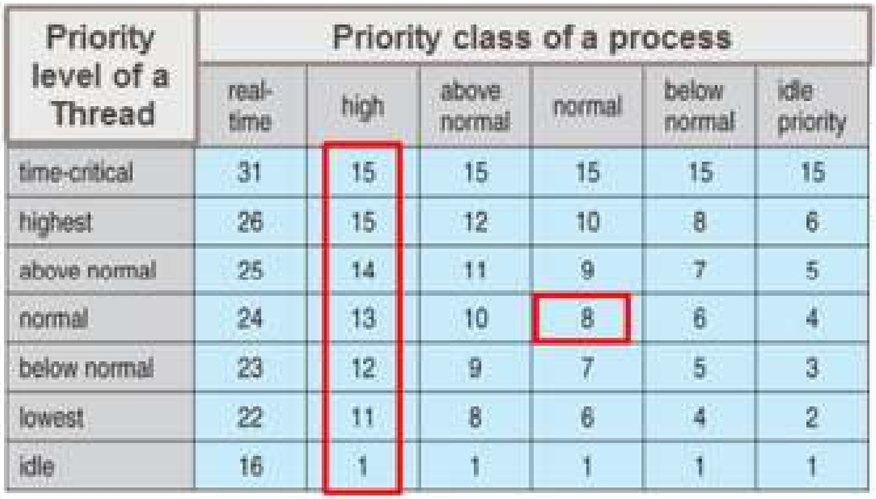
Use .exe directly C:\windows\system32

PROCESS STATES



THREAD SCHEDULING PRIORITY

- Threads are scheduled to run based on their **scheduling priority**.
- Only the **zero-page thread** can have a priority of zero.
- The process priority class and thread priority level are combined to form the **base priority** of each thread.
- The scheduler maintains a queue of executable threads for each priority level and known as ***ready threads***.
- When a processor becomes available, the system performs a ***context switch***. .
  - Find the highest priority queue that contains ready threads.
  - Remove the thread at the head of the queue, load its context, and execute it.



Priority level of a Thread	Priority class of a process					
	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

	Lowest	Highest
Priority level	0	31
Variable class	1	15
Real time class	16	31

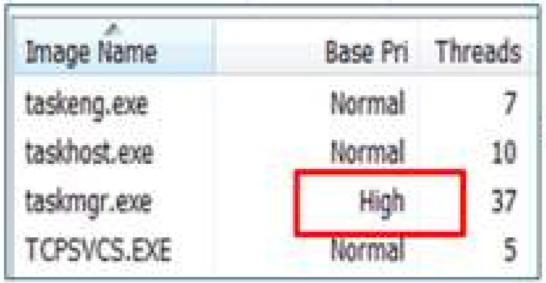


Image Name	Base Pri	Threads
taskeng.exe	Normal	7
taskhost.exe	Normal	10
taskmgr.exe	High	37
TCPSVCS.EXE	Normal	5

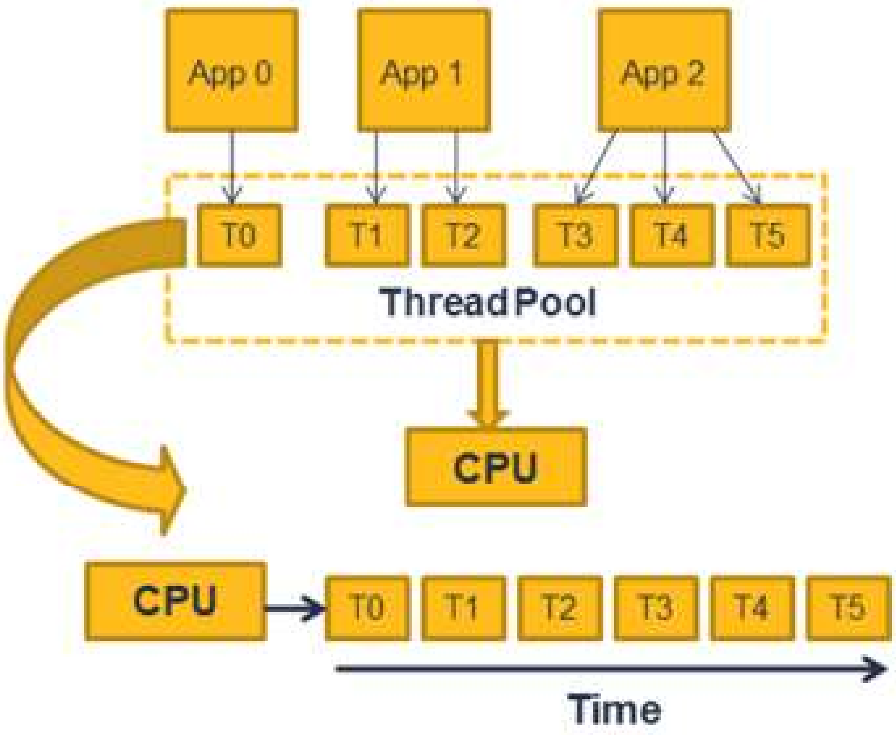
For process with **High priority**,
Dispatcher assign base priority for threads b/w **11 and 15.**
If no ready thread is found, dispatcher will execute **idle thread.**

Most process and thread will occupy **Normal priority.**

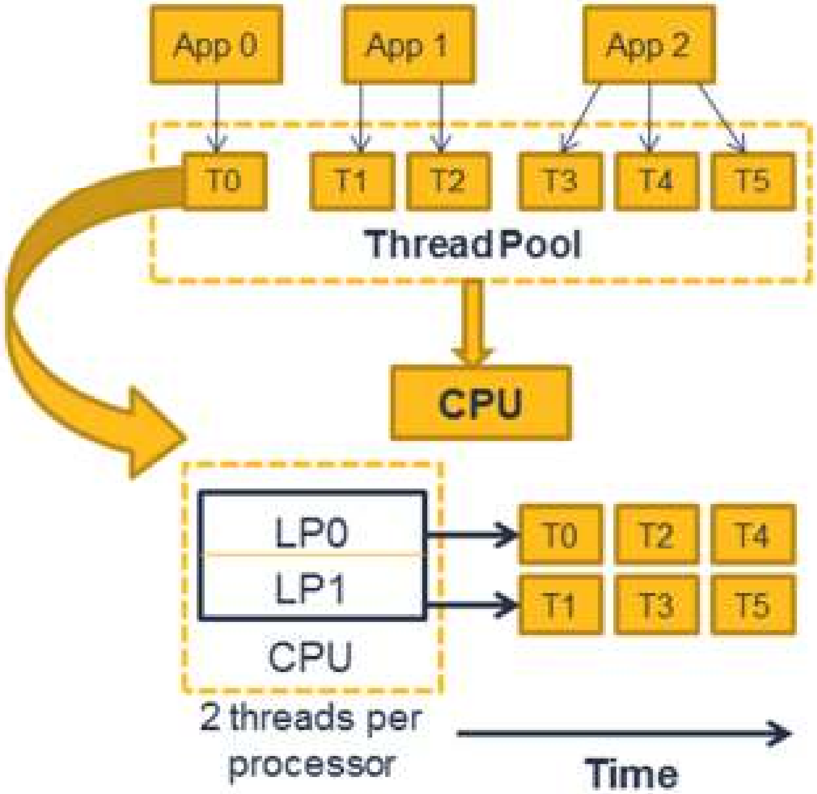
HYPERTHREADING

Hyper-Threading : It enables different parts of the CPU to work on different tasks concurrently. In this way, a CPU with Hyper-Threading appears to be more than one CPU.

Multithreading



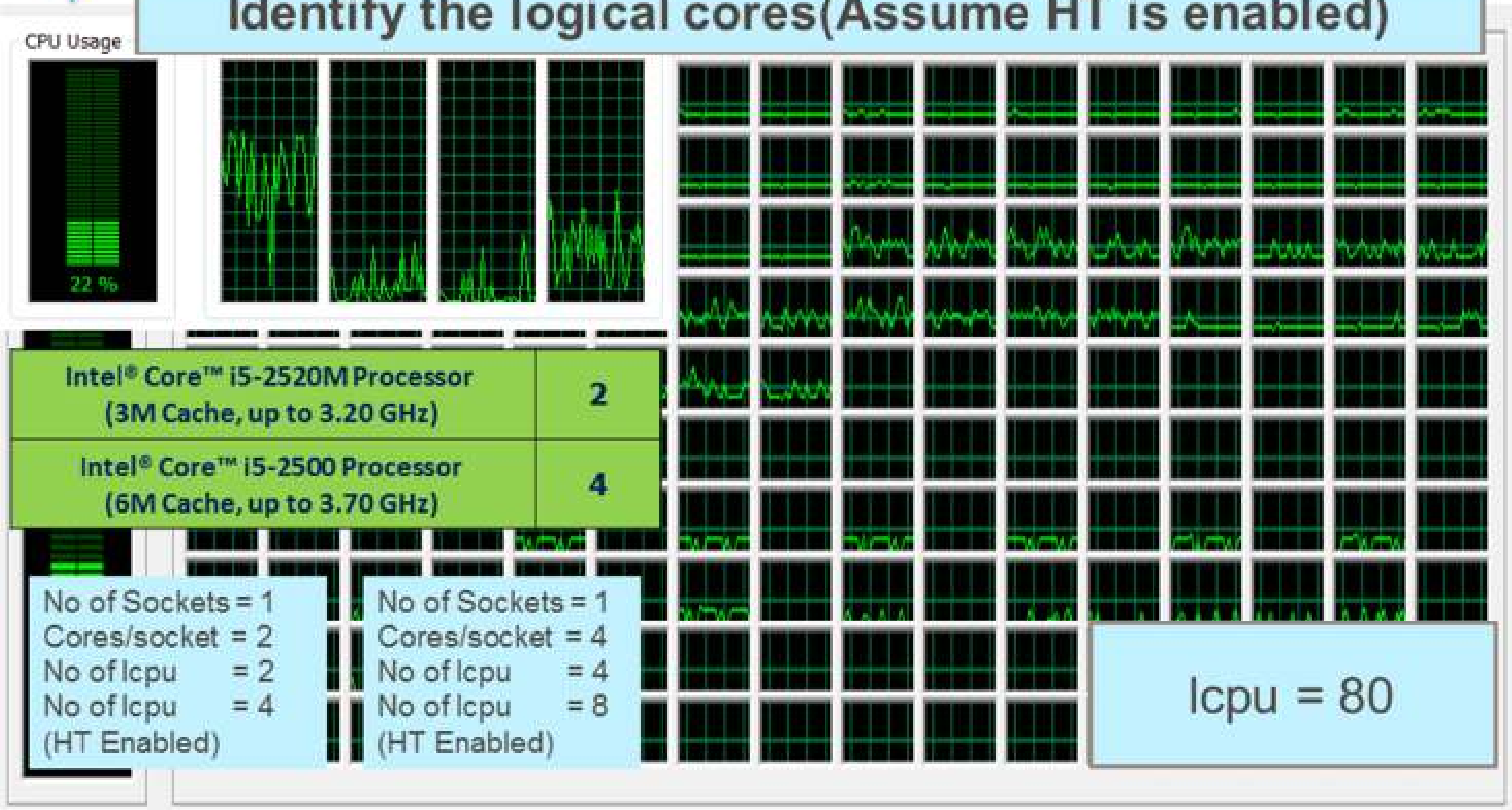
Hyper-threading Technology



2 threads per processor

PROCESSOR CORES

Identify the logical cores(Assume HT is enabled)



Intel® Core™ i5-2520M Processor (3M Cache, up to 3.20 GHz)	2
Intel® Core™ i5-2500 Processor (6M Cache, up to 3.70 GHz)	4

No of Sockets = 1
Cores/socket = 2
No of lcpu = 2
No of lcpu = 4
(HT Enabled)

No of Sockets = 1
Cores/socket = 4
No of lcpu = 4
No of lcpu = 8
(HT Enabled)

lcpu = 80

It's your turn now…

- Why process has to be divided into multiple threads?
- How do processes terminate and Why?
- The system 2 sockets each with dual core processor. Identify the no of logical cores if hyper-threading is enabled.
- What is the base priority for System Idle process?

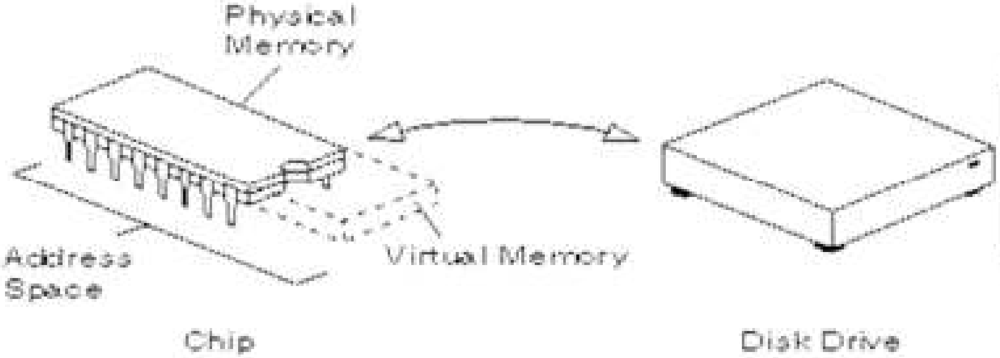
WINDOWS MEMORY MANAGEMENT

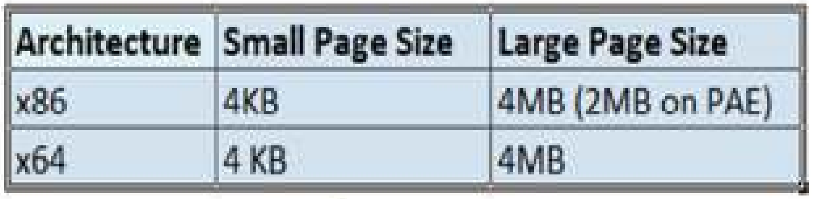
MEMORY MANAGEMENT

- **Kernel memory** is owned by Windows and is used to provide system services to applications.
- **Virtual memory** consists of physical memory plus the amount of space in the page file, which is stored on the hard disk.
- **Cached memory** holds data or program code that has been fetched into memory during the current session but is no longer in use now.
- **Free memory** represents RAM that does not contain any data or program code and is free for use immediately
- **Virtual Address Space:** Set of virtual memory addresses that a process can use;

PAGING

- Paging is one of the **memory-management** schemes by which a computer can store and retrieve data from secondary storage for use in main memory.
- In the paging memory-management scheme, the operating system retrieves data from secondary storage in same-size blocks called **pages.**
- **The Page Frame Number database** contains lists that represent the physical memory pages of the system. The kernel uses the lists to track which pages are "in use" (allocated to working sets), free, available, and so on.
- **Page-in:** Pages moved from Hard disk to physical memory(RAM)
- **Page-out :** Pages moved out from RAM to Hard disk



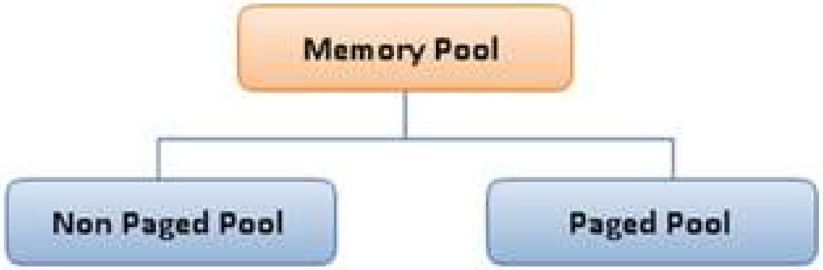


Architecture	Small Page Size	Large Page Size
x86	4KB	4MB (2MB on PAE)
x64	4 KB	4MB

MEMORY POOL

- **Page Table**: An internal data structure used to translate virtual addresses into their corresponding physical addresses;

- The memory manager creates the following memory pools that the system uses to allocate memory:



Memory Pool

Non Paged Pool

Paged Pool

- **Paged Pool** is a noncritical kernel memory used by the operating system kernel. Noncritical portions of kernel memory can be paged to disk and don't have to reside in physical memory (RAM).

- **Non-paged Pool** is a critical kernel memory used by the operating system kernel. Critical portions of kernel memory must operate in physical memory (RAM) and cannot be paged to disk.

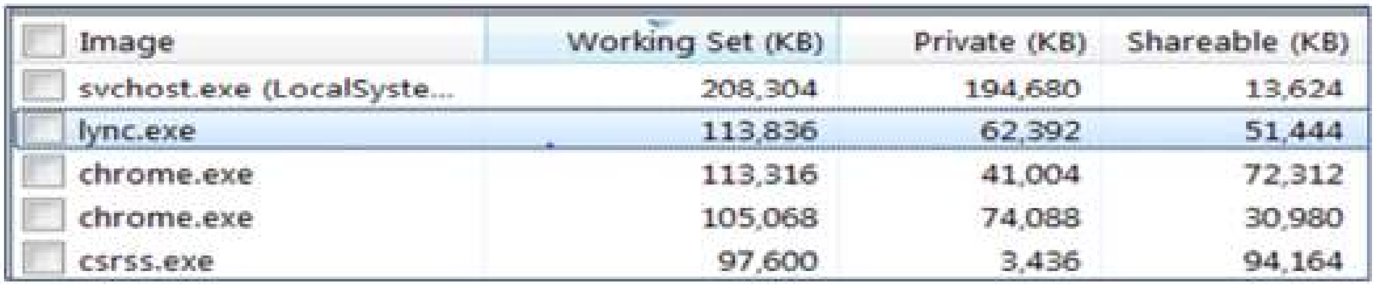
WORKING SET

- **Working Set** is the Amount of physical memory currently in use by the process; Set of pages in the virtual address space of a process that are currently resident in physical memory.

- **Private Working Set** is the amount of memory that is dedicated to that process and that cannot be shared to other process.

- **Shareable Working Set** can be surrendered if physical RAM begins to run scarce.

Working set (WS) = Private WS + Shareable WS



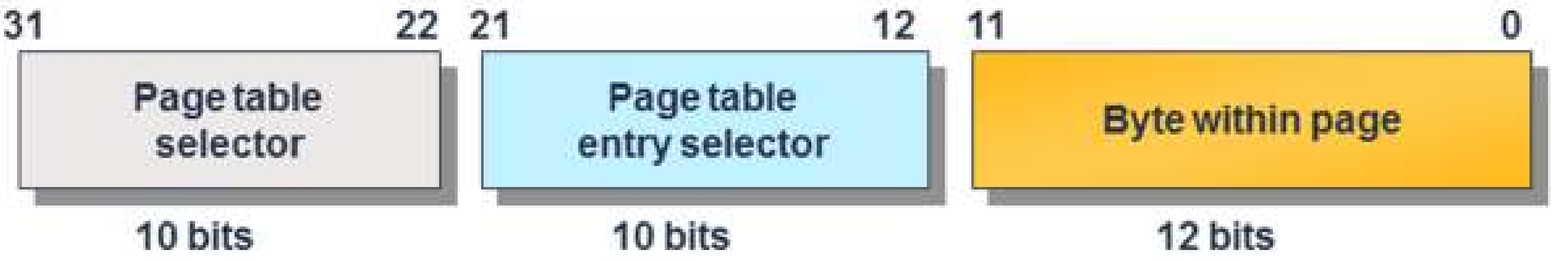
□ Image	Working Set (KB)	Private (KB)	Shareable (KB)
□ svchost.exe (LocalSyste...	208,304	194,680	13,624
□ lync.exe	113,836	62,392	51,444
□ chrome.exe	113,316	41,004	72,312
□ chrome.exe	105,068	74,088	30,980
□ csrss.exe	97,600	3,436	94,164

PAGE FAULT

- A ***page fault*** occurs when a process accesses a page of memory that's not currently in its working set.
- Some page faults require page contents to be retrieved from disk; others can be resolved without accessing the disk.
- A ***hard page fault*** must be resolved by reading page contents from the page's *backing store*, which is in the system paging file.
- A ***soft page fault*** can be resolved without accessing the page file.
- ***Demand-zero fault*** : A process references an allocated virtual page for the first time.

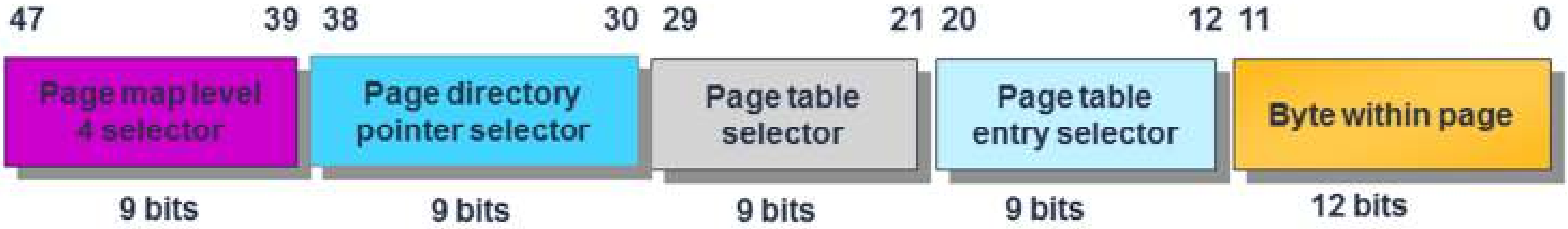
VIRTUAL ADDRESS INTERPRETATION

x86 32-bit



31–22	21–12	11–0
Page table selector	Page table entry selector	Byte within page
10 bits	10 bits	12 bits

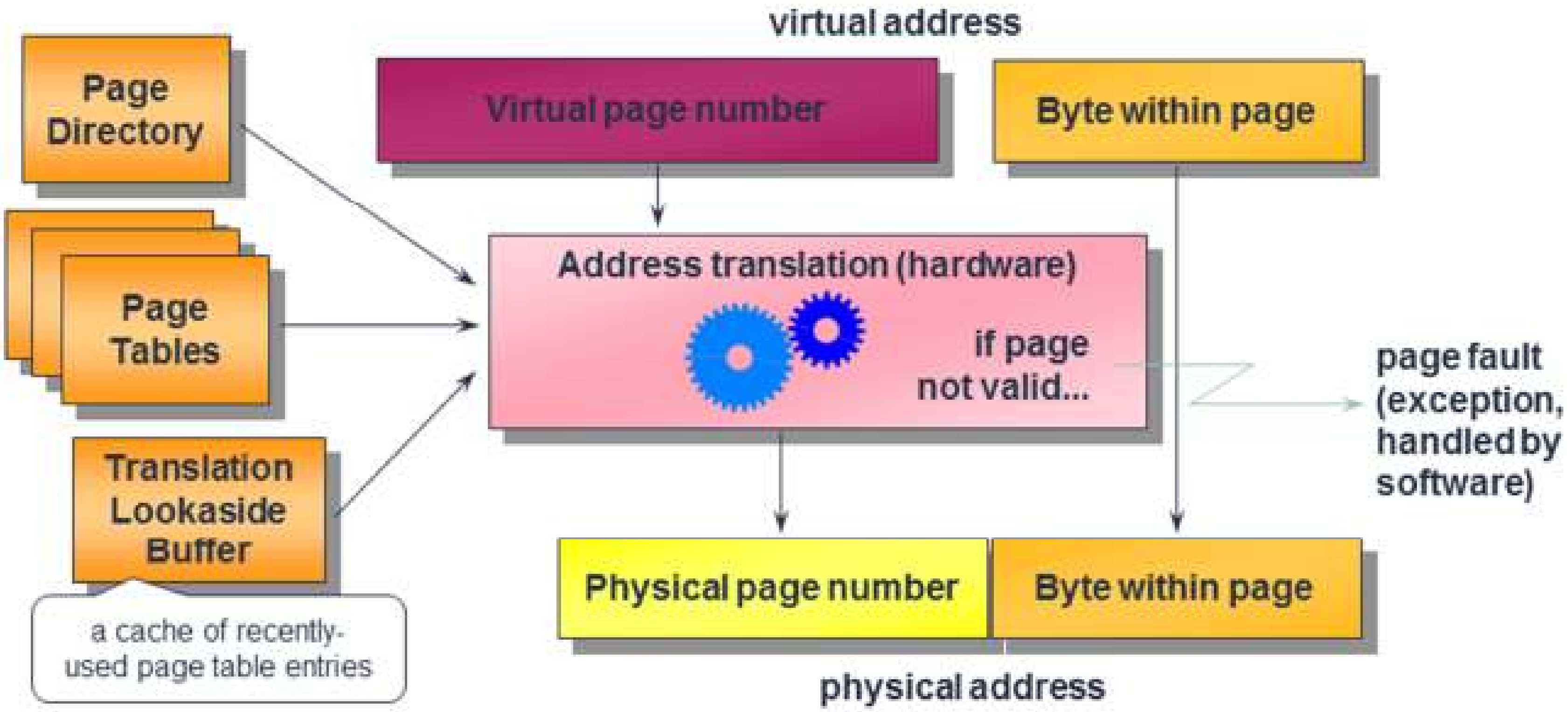
x64 64-bit (48-bit in today's processors)



47–39	38–30	29–21	20–12	11–0
Page map level 4 selector	Page directory pointer selector	Page table selector	Page table entry selector	Byte within page
9 bits	9 bits	9 bits	9 bits	12 bits

VIRTUAL ADDRESS TRANSLATION

- The hardware converts each valid virtual address to a physical address



virtual address

Virtual page number

Byte within page

Page Directory

Page Tables

Translation Lookaside Buffer

a cache of recently-used page table entries

Address translation (hardware)

if page not valid...

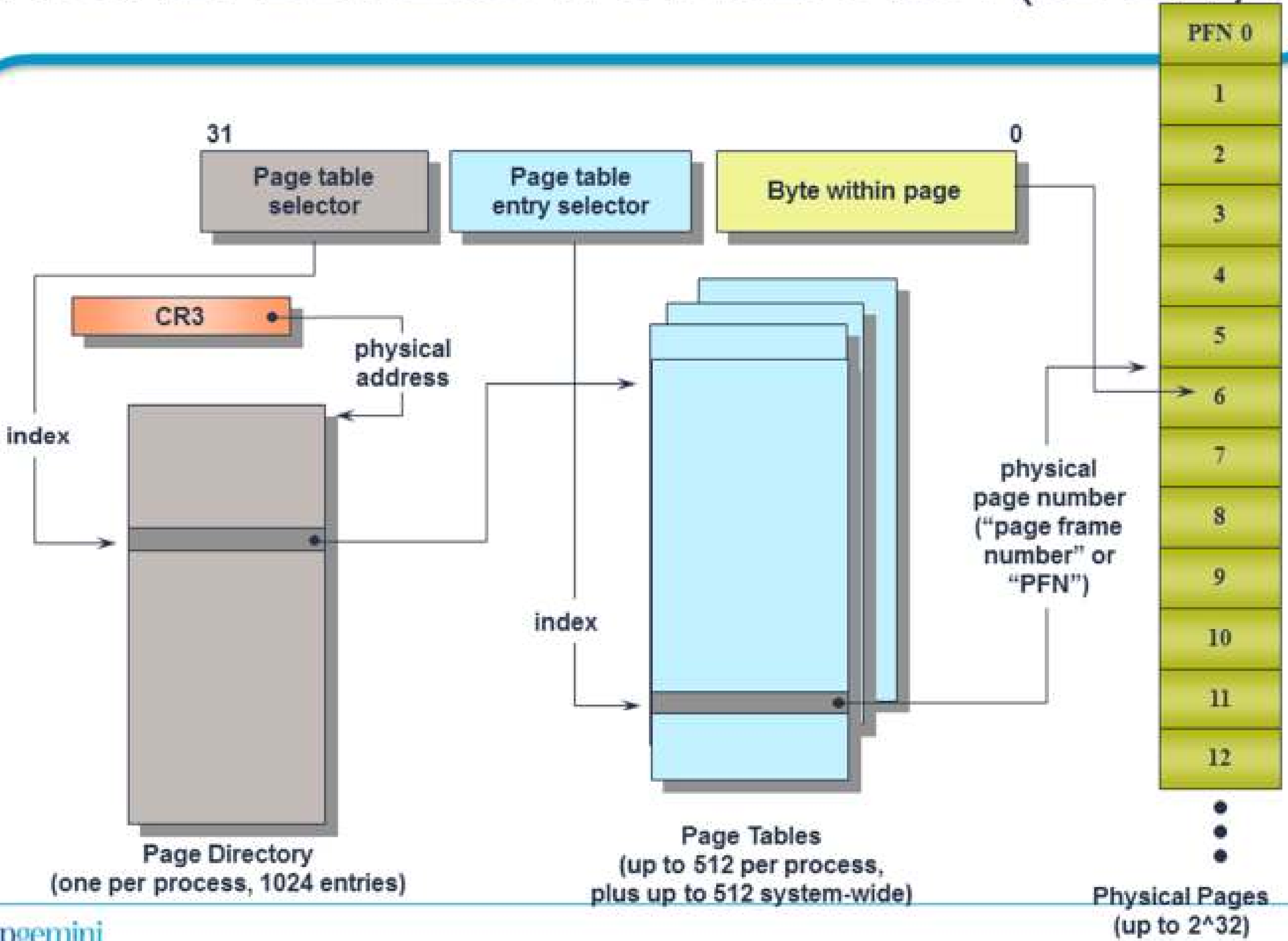
page fault (exception, handled by software)

Physical page number

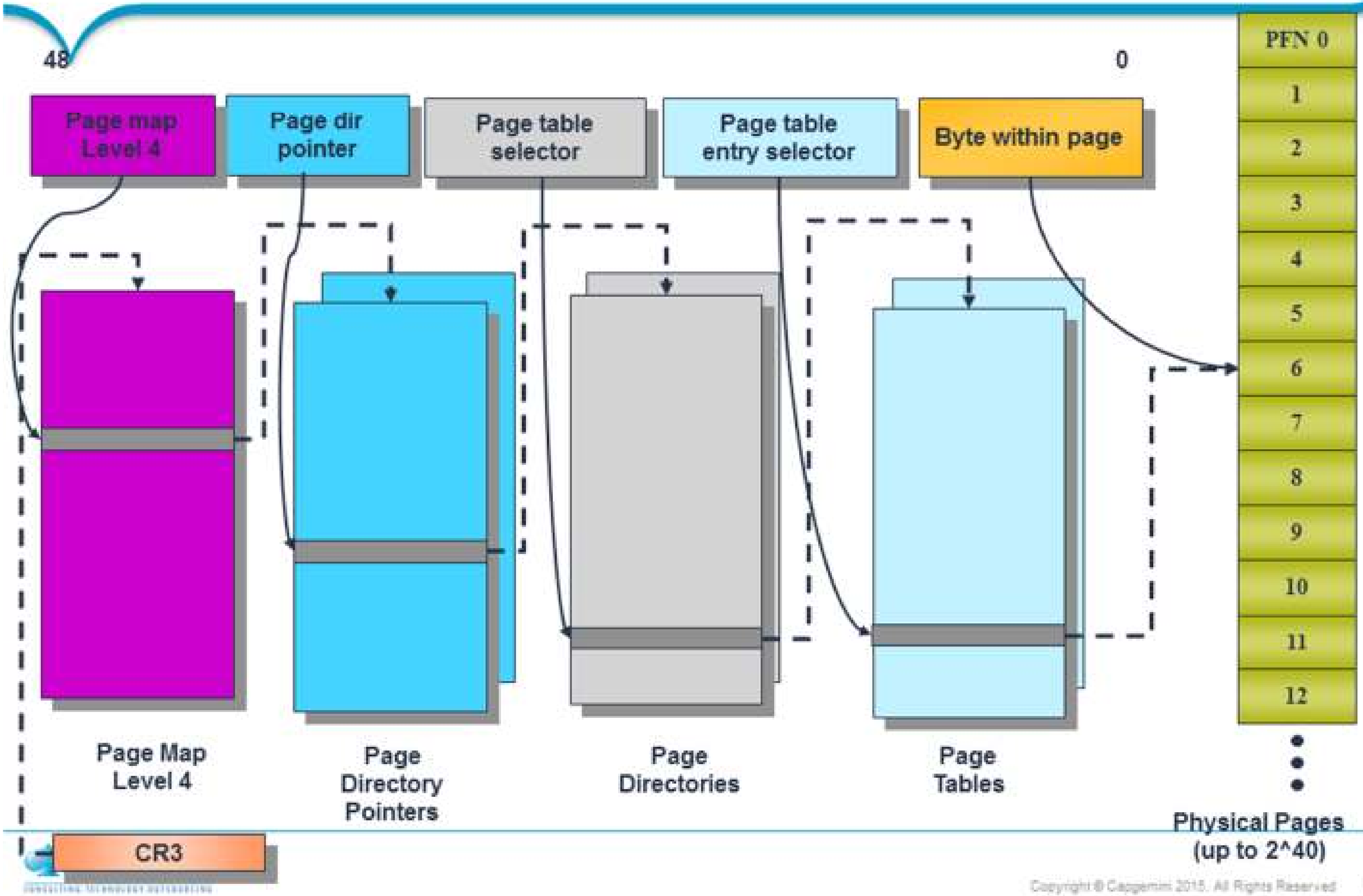
Byte within page

physical address

VIRTUAL ADDRESS TRANSLATION (32-BIT)



VIRTUAL ADDRESS TRANSLATION (64-BIT)



48

0

Page map Level 4

Page dir pointer

Page table selector

Page table entry selector

Byte within page

PFN 0

1

2

3

4

5

6

7

8

9

10

11

12

Page Map Level 4

Page Directory Pointers

Page Directories

Page Tables

Physical Pages (up to 2^40)

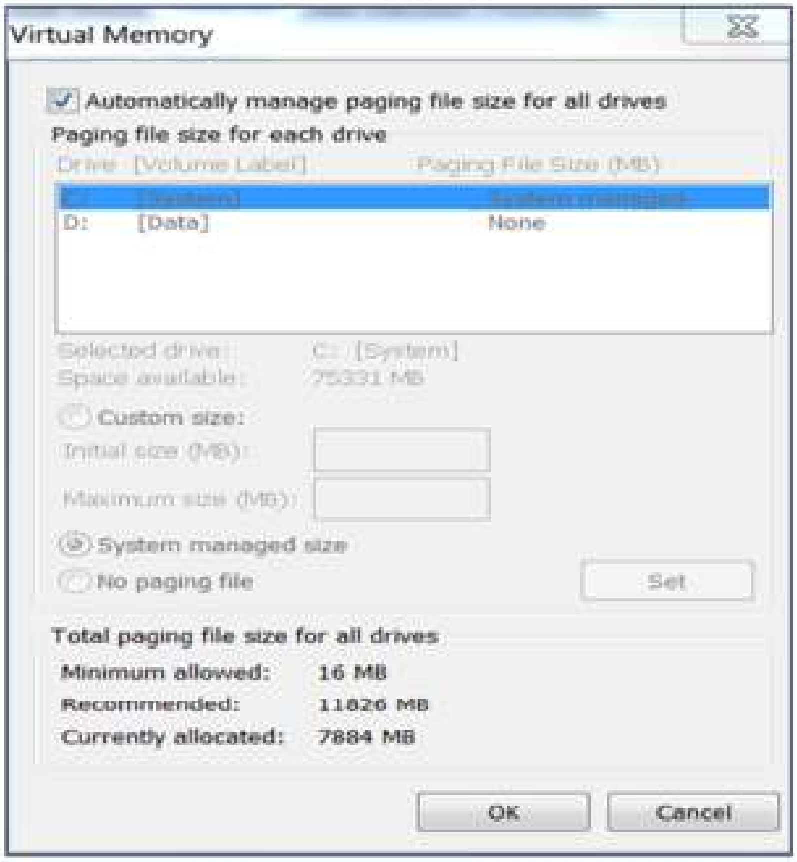
CR3

PAGE FILE SETTINGS

- *Go to "System Properties" → Click "Advanced" → Under Performance, "Settings"- →Choose "Advanced" → Click "Change"*

- Maximum Page file size = **16TB.**
- Max No of page files = **16 paging files**
- Recommended Page File to be Allocated:

  ➔ **1.5 * Size of your RAM**



Virtual Memory

☑ Automatically manage paging file size for all drives

Paging file size for each drive

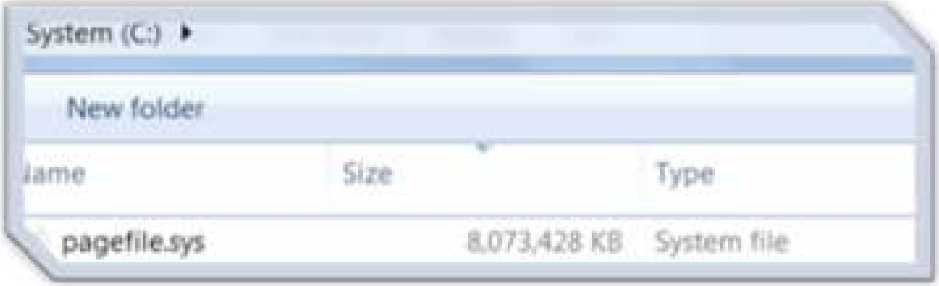
Drive [Volume Label]		Paging File Size (MB)
C:	[System]	System managed
D:	[Data]	None

Selected drive: C: [System]
Space available: 75331 MB

○ Custom size:
Initial size (MB):
Maximum size (MB):
● System managed size
○ No paging file

Total paging file size for all drives

Minimum allowed: 16 MB
Recommended: 11826 MB
Currently allocated: 7884 MB



System (C:) ▸

New folder

Name	Size	Type
pagefile.sys	8,073,428 KB	System file

Settings In Registry:

HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management\PagingFiles

PHYSICAL MEMORY LIMITS

Version	Limit on X86	Limit on X64
Windows 7 Ultimate	4 GB	192 GB
Windows 7 Enterprise	4 GB	192 GB
Windows 7 Professional	4 GB	192 GB
Windows 7 Home Premium	4 GB	16 GB
Windows 7 Home Basic	4 GB	8 GB
Windows 7 Starter	2 GB	N/A

Version	Limit on X86	Limit on X64
Windows 10 Enterprise		
Windows 10 Education		
Windows 10 Pro		
Windows 10 Home		

It's your turn!!!!

Note: For all 32-bit (x86) editions, the maximum limit is 4GB

Version	Limit on X64
Windows Server 2008 R2 Datacenter	2 TB
Windows Server 2008 R2 Enterprise	2 TB
Windows Server 2008 R2 Foundation	8 GB
Windows Server 2008 R2 Standard	32 GB
Windows HPC Server 2008 R2	128 GB
Windows Web Server 2008 R2	32 GB

Version	Limit on X64
Windows Server 2012 Datacenter	
Windows Server 2012 Standard	
Windows Server 2012 Essentials	
Windows Server 2012 Foundation	
Windows Storage Server 2012 Workgroup	
Windows Storage Server 2012 Standard	
Hyper-V Server 2012	

Note: From server 2008 R2 versions are available only on x64

It's your turn now…

- Do graphics cards and other devices affect memory limits? If yes, explain How?
- What will be the impact if I disable page file?
- In which way performance will be good?

  Placing page file on different partitions/drives on same disk;

  Placing page file on different partitions/drives on different disks;
- Size of RAM is 16GB. Calculate the recommended page file.

Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

THANK YOU

People matter, results count.