

UNIT 1: Overview of Operating System

★ Definition of Operating System (ऑपरेटिंग सिस्टम की परिभाषा)

Operating System (OS) एक ऐसा **system software** है जो **user** और **computer hardware** के बीच interface का काम करता है।

यह computer के hardware resources को manage करता है और programs को services provide करता है।

👉 “Operating system ek aisa software hai jo computer ko चालू (start), चलाने (run) और नियंत्रित (control) करने का काम करता है और user aur hardware ke बीच ek interface provide karta hai.”

📌 Functions of Operating System (मुख्य कार्य)

1. Process Management (प्रक्रिया प्रबंधन):

- CPU par चलने वाली processes ko manage करता है।
- Process scheduling aur termination handle करता है।

2. Memory Management (स्मृति प्रबंधन):

- Main memory (RAM) ko allocate aur deallocate करता है।
- यह सुनिश्चित करता है कि हर process ko proper memory मिले।

3. File Management (फ़ाइल प्रबंधन):

- Files aur folders ko store, access aur secure करने का काम करता है।
- File permissions aur directories को maintain करता है।

4. Device Management (डिवाइस प्रबंधन):

- Input-output devices (जैसे keyboard, printer) ke साथ communication control करता है।
- Device drivers ke through hardware ko manage करता है।

5. Security and Protection (सुरक्षा और संरक्षण):

- Unauthorized access ko रोकता है।
- System resources को misuse से बचाता है।

6. User Interface Provide करना:

- Command line interface (CLI) ya Graphical User Interface (GUI) provide करता है जिससे user आसानी से interact कर सके।

◆ Types of Operating System (प्रकार)

1. Batch Operating System
2. Time-Sharing Operating System
3. Multiprogramming Operating System
4. Real-Time Operating System
5. Distributed Operating System
6. Network Operating System

(अगर आप चाहें तो हर type को detail में समझा सकता हूँ।)

📌 Examples of Operating System (उदाहरण):

- Windows
 - Linux
 - Unix
 - macOS
 - Android
-

✨ Summary (सारांश):

- Operating System एक **interface** है user और hardware के बीच।
- यह **resources manage** करता है (CPU, memory, files, devices)।
- यह **security, process management, file management** जैसी services देता है।

☀ Introduction of Operating System (ऑपरेटिंग सिस्टम का परिचय)

- ① **Operating System** एक **system software** है जो computer के hardware और user के बीच interface का काम करता है।
 - ② यह **hardware resources** को **manage** करता है जैसे CPU, memory, input-output devices, files इत्यादि।
 - ③ यह **programs** को **services provide** करता है ताकि वे efficiently (दक्षता से) run हो सकें।
 - ④ यह **user** को आसान तरीका देता है **computer use** करने का CLI (Command Line Interface) या GUI (Graphical User Interface) के माध्यम से।
 - ⑤ यह **system** की **security** और **protection** को संभालता है Unauthorized access और errors को control करता है।
-

✨ Need of Operating System (OS की आवश्यकता क्यों?):

1. User सीधे hardware से बात नहीं कर सकता, इस लिए OS एक interface प्रदान करता है।
 2. यह computer resources को organize और control करता है।
 3. Programs को services देता है ताकि वे आसानी से execute हों।
 4. Errors और security issues को handle करता है।
-

📌 Main Objectives of Operating System (मुख्य उद्देश्य):

- **Convenience** (सुविधा): User को system use करना आसान बनाना।
 - **Efficiency** (दक्षता): Hardware का पूरा उपयोग करना।
 - **Resource Sharing** (संसाधनों का साझा उपयोग): कई users/processes के बीच CPU, memory, files share करना।
 - **Security** (सुरक्षा): Unauthorized access से system की रक्षा करना।
-

📋 Features of Operating System (विशेषताएँ):

- Multi-tasking support
- Memory management
- File management
- Device control
- Security features
- User-friendly interface (CLI/GUI)

Examples (उदाहरण):

- Windows
- Linux
- Unix
- macOS
- Android

★ Different Types of Operating System (ऑपरेटिंग सिस्टम के प्रकार)

1 Batch Operating System (बैच ऑपरेटिंग सिस्टम)

- इस सिस्टम में **jobs** को **groups (batches)** में divide किया जाता है।
- Similar jobs एक साथ run होते हैं, बिना user के सीधे interaction के।
- Example: पुराने mainframe systems में।

Main point:

- ✓ User सीधे interact नहीं करता।
 - ✓ Jobs को बैच में submit करके execute कराया जाता है।
-

2 Time-Sharing Operating System (टाइम शेयरिंग ऑपरेटिंग सिस्टम)

- एक ही समय पर कई users को system share करने की सुविधा देता है।
- CPU time को छोटे-छोटे time slices में divide किया जाता है।
- Example: UNIX, Multics.

Main point:

- ✓ एक ही समय में multiple users.
 - ✓ Interactive system.
-

3 Multiprogramming Operating System (मल्टीप्रोग्रामिंग ऑपरेटिंग सिस्टम)

- Memory में एक साथ कई programs load रहते हैं।
- CPU एक program खत्म होते ही दूसरा program run करता है, जिससे CPU idle नहीं रहता।
- Example: Modern mainframe systems.

- **Main point:**
 - ✓ Maximum CPU utilization.
 - ✓ Programs simultaneously run होते हैं।
-

4 Real-Time Operating System (रीयल टाइम ऑपरेटिंग सिस्टम)

- यह system ऐसे कामों के लिए होता है जहाँ **time constraints** strict होते हैं।
- Example: Air traffic control, industrial robots।

Main point:

- ✓ Immediate response देता है।
 - ✓ Hard real time और soft real time types होते हैं।
-

5 Distributed Operating System (डिस्ट्रिब्यूटेड ऑपरेटिंग सिस्टम)

- कई computers आपस में connected रहते हैं और एक system की तरह काम करते हैं।
- Resources share होते हैं और users को एक single system जैसा experience मिलता है।

Main point:

- ✓ Resource sharing.
 - ✓ Reliability high होती है।
-

6 Network Operating System (नेटवर्क ऑपरेटिंग सिस्टम)

- यह system servers और network devices को manage करता है।
- Example: Windows Server, Novell NetWare.

Main point:

- ✓ Network पर file sharing और printer sharing।
- ✓ Multi-user environment support करता है।

★ Functions of Operating System (ऑपरेटिंग सिस्टम के मुख्य कार्य)

1 Process Management (प्रक्रिया प्रबंधन)

- CPU पर चलने वाले programs (processes) को manage करता है।
- Process ko create करना, schedule करना और terminate करना।
- Example: Multitasking, multiprogramming में process management ज़रूरी है।

② Memory Management (स्मृति प्रबंधन)

- Main memory (RAM) को अलग-अलग processes में allocate और deallocate करता है।
 - Memory protection और efficiency maintain करता है।
-

③ File Management (फाइल प्रबंधन)

- Data को files के रूप में store करना, access करना और organize करना।
 - Directories maintain करना, file permissions handle करना।
-

④ Device Management (डिवाइस प्रबंधन)

- Input/Output devices (जैसे printer, keyboard) को control और coordinate करना।
 - Device drivers के द्वारा hardware से communication manage करना।
-

⑤ Security and Protection (सुरक्षा और संरक्षण)

- Unauthorized access से system को बचाना।
 - Data और resources को protect करना।
-

⑥ Error Detection and Handling (त्रुटि जाँच और प्रबंधन)

- Hardware और software errors को detect और report करना।
 - Smooth operation के लिए error-handling mechanisms provide करना।
-

⑦ Networking (नेटवर्किंग प्रबंधन)

- कई systems के बीच communication enable करना।
 - Network resources को manage करना।
-

⑧ User Interface Provide करना (उपयोगकर्ता इंटरफ़ेस देना)

- User को system से interact करने का तरीका देता है।
 - CLI (Command Line Interface) या GUI (Graphical User Interface) के द्वारा।
-

★ Services of Operating System (ऑपरेटिंग सिस्टम की सेवाएँ)

- ✓ **Program Execution:** Program को load और execute करना।
- ✓ **I/O Operations:** Input और Output devices से communication।
- ✓ **File System Manipulation:** File create, delete, read और write करना।
- ✓ **Communication Services:** Inter-process communication support करना।
- ✓ **Resource Allocation:** CPU, memory और devices को efficiently distribute करना।
- ✓ **Accounting:** System resource usage का record रखना।
- ✓ **Protection:** Data और resources को सुरक्षित रखना।

★ UNIX / LINUX Architecture (यूनिक्स / लिनक्स की संरचना)

UNIX और LINUX का architecture layered structure में होता है। इसमें मुख्यतः चार हिस्से होते हैं:

♦ 1. Hardware Layer (हार्डवेयर लेयर)

- यह सबसे नीचे की layer है।
- इसमें **CPU, memory, input-output devices** आते हैं।
- यही actual physical resources हैं जिन पर पूरा system चलता है।

♦ 2. Kernel (कर्नेल)

- यह OS का **core part** है।
- Hardware और बाकी system के बीच interface का काम करता है।
- Kernel के मुख्य कार्य:
 - **Process Management** (प्रक्रियाओं को नियंत्रित करना)
 - **Memory Management** (स्मृति प्रबंधन)
 - **Device Management** (डिवाइस नियंत्रित करना)
 - **File System Management** (फाइल सिस्टम प्रबंधन)
- यह हमेशा **running state** में रहता है।

♦ 3. Shell (शेल)

- यह **user** और **kernel** के बीच का interface है।
- User के commands को लेता है और उन्हें kernel को pass करता है।
- यह एक **command line interpreter** है।
- Example: bash shell, sh shell, ksh shell.

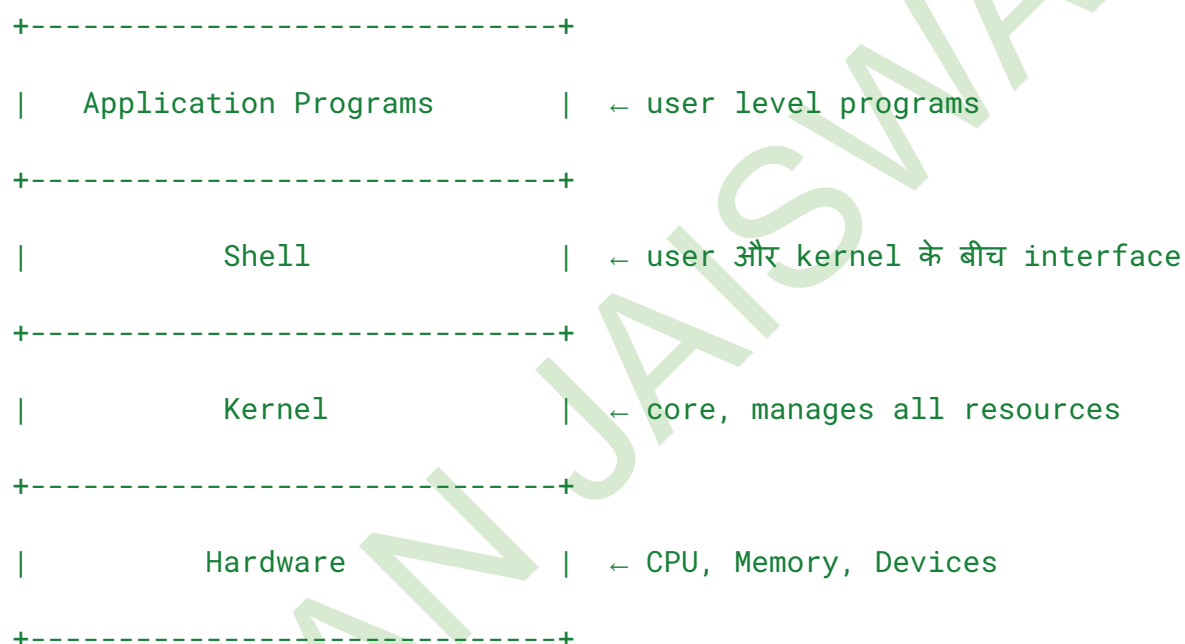
♦ 4. Application Programs (एप्लिकेशन प्रोग्राम्स)

- User द्वारा use किए जाने वाले programs।
- ये programs shell के ऊपर run होते हैं और user के काम को आसान बनाते हैं।
- Example: text editors, compilers, web browsers, utilities.

✨ Diagram of UNIX / LINUX Architecture (सरल चित्र):

pgsql

CopyEdit



✓ Summary (सारांश):

- **Hardware** – physical resources
- **Kernel** – system का heart, resources manage करता है
- **Shell** – user से commands लेता है
- **Applications** – user द्वारा इस्तेमाल किए जाने वाले tools

★ Kernel (कर्नेल)

📌 Definition (परिभाषा):

Kernel Operating System का **core part** (मुख्य भाग) होता है जो सीधे computer के **hardware** के साथ interact करता है और सभी resources (CPU, memory, devices) को manage करता है।

👉 “**Kernel** वह हिस्सा है जो **user** के **programs** और **hardware** के बीच **interface** का काम करता है और **system resources** को **control** करता है।”

🌟 Functions of Kernel (मुख्य कार्य):

① Process Management (प्रक्रिया प्रबंधन)

- Kernel CPU पर चलने वाली processes को manage करता है।
- Process create करना, terminate करना और schedule करना।

② Memory Management (स्मृति प्रबंधन)

- RAM को allocate और deallocate करता है।
- हर process को उसकी जरूरत के हिसाब से memory देता है।

③ Device Management (डिवाइस प्रबंधन)

- Input/Output devices जैसे keyboard, mouse, printer को control करता है।
- Device drivers के द्वारा hardware से communication करता है।

④ File System Management (फाइल प्रबंधन)

- Data को files और directories में manage करता है।
- File access control और storage handle करता है।

⑤ Security और Protection

- Unauthorized access को रोकता है।
- Process isolation और data protection देता है।

📌 Types of Kernel (कर्नेल के प्रकार):

1. Monolithic Kernel

- पूरा kernel एक बड़े single program के रूप में चलता है।
- Example: UNIX Kernel, Linux Kernel.

2. Microkernel

- Kernel के अंदर सिर्फ basic services होती हैं, बाकी services user space में चलती हैं।
- Example: Minix, QNX.

3. Hybrid Kernel

- Monolithic और microkernel का मिश्रण।
- Example: Windows NT kernel, macOS kernel.

✓ Summary (सारांश):

- Kernel OS का सबसे महत्वपूर्ण हिस्सा है।
- यह सभी hardware resources को manage करता है।
- Kernel के बिना Operating System काम नहीं कर सकता।

★ 1. Services of Operating System (ऑपरेटिंग सिस्टम की सेवाएँ)

Operating System user और programs को कई **services** provide करता है ताकि system को use करना आसान और efficient हो सके।

✨ मुख्य Services:

① Program Execution (प्रोग्राम निष्पादन)

- Program को load करना, execute करना और terminate करना।

② I/O Operations (इनपुट/आउटपुट संचालन)

- Input devices (keyboard) और output devices (printer) के साथ communication।

③ File System Manipulation (फाइल सिस्टम प्रबंधन)

- File create, delete, read और write करने की सुविधा देना।

④ Communication Services (संचार सेवाएँ)

- Processes के बीच data exchange के लिए support देना (Inter-Process Communication)।

⑤ Error Detection and Handling (त्रुटि जाँच और प्रबंधन)

- Errors detect करना और उनको handle करना।

⑥ Resource Allocation (संसाधन आवंटन)

- CPU, memory और devices को सही तरीके से allocate करना।

⑦ Accounting (लेखा-जोखा रखना)

- कौन सा process कितने resources use कर रहा है उसका record रखना।

⑧ Protection and Security (सुरक्षा और संरक्षण)

- Unauthorized access से data और resources की सुरक्षा करना।
-

★ 2. System Calls (सिस्टम कॉल्स)

📌 Definition (परिभाषा):

System Call एक ऐसा तरीका है जिसके द्वारा **user programs** Operating System से services माँगते हैं।

👉 “System call वह interface है जो user program को OS की services तक पहुँचने देता है।”

✨ मुख्य Functions of System Calls:

① Process Control (प्रक्रिया नियंत्रण)

- Process को create, terminate, suspend या resume करने के लिए system calls।
- Example: `fork()`, `exit()`, `wait()`

② File Management (फाइल प्रबंधन)

- File create, delete, open, read, write करने के लिए system calls।
- Example: `open()`, `read()`, `write()`, `close()`

③ Device Management (डिवाइस प्रबंधन)

- Devices को allocate, deallocate और control करने के लिए।
- Example: `ioctl()`, `read()`, `write()`

④ Information Maintenance (सूचना प्रबंधन)

- System time, date, process information लेने के लिए।
- Example: `getpid()`, `alarm()`, `sleep()`

⑤ Communication (संचार)

- Processes के बीच message भेजने और प्राप्त करने के लिए।
 - Example: `pipe()`, `shmget()`, `msgsnd()`
-

★ System Programs (सिस्टम प्रोग्राम्स)

📌 Definition (परिभाषा):

System Programs वे programs होते हैं जो **Operating System** के ऊपर **run** होते हैं और user को **basic environment** और **utilities** provide करते हैं ताकि user अपने tasks आसानी से perform कर सके।

👉 “**System programs OS की services का उपयोग करके user को सुविधाएँ और support प्रदान करते हैं।**”

✨ Main Features (मुख्य विशेषताएँ):

- ✓ यह OS का हिस्सा नहीं होते लेकिन OS के ऊपर चलते हैं।
- ✓ यह user और application programs के बीच एक सहायक layer की तरह काम करते हैं।
- ✓ यह utilities और services provide करते हैं जो computer use करना आसान बनाती हैं।

📋 Types of System Programs (प्रकार):

① File Management Programs (फ़ाइल प्रबंधन प्रोग्राम्स)

- Files को create, delete, copy, move करने के लिए।
- Example: `cp`, `mv`, `rm` (Linux commands)

② Status Information Programs (स्थिति जानकारी देने वाले प्रोग्राम्स)

- System की status या performance बताने वाले tools।
- Example: `top`, `ps`, `uptime` (Linux)

③ File Modification Programs (फ़ाइल संशोधन प्रोग्राम्स)

- Files को modify करने के लिए editors।
- Example: `vi editor`, `nano`, `notepad`

④ Programming Language Support (भाषा समर्थन प्रोग्राम्स)

- Compilers, assemblers और interpreters।
- Example: `gcc compiler`, `python interpreter`

⑤ Program Loading and Execution (प्रोग्राम लोड और निष्पादन प्रोग्राम्स)

- Programs को memory में load करना और run कराना।
- Example: `loader`, `linker`

⑥ Communication Programs (संचार प्रोग्राम्स)

- Network communication के लिए programs।
- Example: `ssh`, `ftp`, `telnet`

⑦ Background Services (बैकग्राउंड सेवाएँ)

- Daemons या background में चलने वाले processes जो system को smooth चलाते हैं।
- Example: `cron daemon`, `httpd` (web server)

✓ Summary (सारांश):

- ✓ System Programs = Utilities + Tools जो OS के ऊपर चलकर user को support देते हैं।
- ✓ इनकी मदद से files, programs और devices manage करना आसान हो जाता है।