Assignment Questions
Subject – OPERATING SYTEM

1. **Define operating system and list the basic services provided by operating system**
   ANS:- ऑपरेटिंग सिस्टम (Operating System) :- ऑपरेटिंग सिस्टम (OS) एक सॉफ़्टवेयर प्रोग्राम है जो कंप्यूटर या मोबाइल डिवाइस के हार्डवेयर और सॉफ़्टवेयर के बीच मध्यस्थ के रूप में कार्य करता है। यह उपयोगकर्ता को कंप्यूटर हार्डवेयर और सॉफ़्टवेयर संसाधनों का प्रबंधन करने और उनका उपयोग करने की अनुमति देता है। ऑपरेटिंग सिस्टम कंप्यूटर पर चलने वाले अन्य प्रोग्रामों को नियंत्रित और समन्वित करता है, जिससे उपयोगकर्ता के लिए कार्यों को सरल और सुचारू रूप से पूरा किया जा सके।

ऑपरेटिंग सिस्टम द्वारा प्रदान की जाने वाली बुनियादी सेवाएँ (Basic Services Provided by Operating System):

1. **प्रोसेस प्रबंधन (Process Management)**:
   - ऑपरेटिंग सिस्टम कंप्यूटर पर चल रहे विभिन्न प्रोग्रामों या प्रक्रियाओं (Processes) का प्रबंधन करता है, जैसे कि प्रोसेस को बनाना, समाप्त करना, और उनके बीच संसाधनों का आवंटन करना।
2. **मेमोरी प्रबंधन (Memory Management)**:
   - यह सिस्टम की मेमोरी (RAM) का प्रबंधन करता है। ऑपरेटिंग सिस्टम यह तय करता है कि कौन सा प्रोसेस कितनी मेमोरी उपयोग करेगा और संसाधनों का उचित वितरण करता है।
3. **फ़ाइल प्रबंधन (File Management)**:
   - ऑपरेटिंग सिस्टम फ़ाइलों और डायरेक्टरीज़ का प्रबंधन करता है, जैसे फ़ाइलों को बनाने, पढ़ने, लिखने, संपादित करने और हटाने की प्रक्रिया को नियंत्रित करना।
4. **इनपुट/आउटपुट (I/O) प्रबंधन (Input/Output Management)**:
   - यह ऑपरेटिंग सिस्टम कंप्यूटर के विभिन्न इनपुट और आउटपुट डिवाइसों (जैसे कीबोर्ड, माउस, प्रिंटर) के साथ संचार करता है और उन्हें नियंत्रित करता है।
5. **सुरक्षा और पहुंच नियंत्रण (Security and Access Control)**:
   - ऑपरेटिंग सिस्टम सुरक्षा सुनिश्चित करता है और उपयोगकर्ताओं को संसाधनों तक पहुंच को नियंत्रित करता है। यह पासवर्ड, फायरवॉल, और अन्य सुरक्षा उपायों का उपयोग करता है।
6. **नेटवर्क प्रबंधन (Network Management)**:
   - यह कंप्यूटर के नेटवर्क से जुड़े संसाधनों का प्रबंधन करता है, जैसे नेटवर्क पर डेटा भेजना और प्राप्त करना, और अन्य कंप्यूटरों से कनेक्टिविटी स्थापित करना।
7. **यूज़र इंटरफ़ेस (User Interface)**:
   - ऑपरेटिंग सिस्टम उपयोगकर्ताओं के लिए एक इंटरफ़ेस प्रदान करता है, जिससे वे कंप्यूटर से इंटरैक्ट कर सकते हैं, जैसे कि ग्राफिकल यूज़र इंटरफ़ेस (GUI) या कमांड-लाइन इंटरफ़ेस (CLI)।
8. **प्राथमिक कार्यों का समन्वय (Task Scheduling)**:
   - ऑपरेटिंग सिस्टम विभिन्न कार्यों को प्रबंधित करता है और उन्हें प्राथमिकता के अनुसार शेड्यूल करता है, जिससे प्रोसेसों के बीच समय का उचित वितरण होता है।

2. Define process and Explain process states in details

ANS:- प्रक्रिया (Process) :- प्रोसेस (Process) एक कार्यक्रम (Program) है जो कंप्यूटर में निष्पादन (Execution) के दौरान चल रहा होता है। इसे एक सक्रिय (Active) कार्यक्रम माना जाता है, जिसे CPU द्वारा

संसाधित किया जा रहा होता है। जब एक प्रोग्राम को चलाया जाता है, तो वह प्रोसेस के रूप में बदल जाता है। प्रोसेस में कोड, डेटा, और प्रोग्राम काउंटर (Program Counter) जैसी कई जानकारी शामिल होती है, जो कंप्यूटर द्वारा कार्यान्वित की जाती है।

प्रक्रिया की स्थिति (Process States) विस्तार से:

प्रोसेस के जीवनकाल (Life Cycle) में विभिन्न अवस्थाएँ होती हैं। इन अवस्थाओं को प्रोसेस की स्थिति कहा जाता है। प्रत्येक प्रोसेस के जीवनकाल में यह विभिन्न अवस्थाओं से गुजरता है, और हर अवस्था में प्रोसेस एक विशेष कार्य करता है। ऑपरेटिंग सिस्टम प्रोसेस के जीवन चक्र को नियंत्रित करता है और उसे उचित संसाधन प्रदान करता है। मुख्य रूप से निम्नलिखित प्रोसेस की अवस्थाएँ होती हैं:

1. नया (New):

- **इस अवस्था में एक नया प्रोसेस बनाया जाता है, लेकिन इसे अभी तक निष्पादित नहीं किया गया होता। प्रोसेस को तैयार करने के लिए आवश्यक सभी जानकारी को ऑपरेटिंग सिस्टम द्वारा इकट्ठा किया जाता है, जैसे प्रोसेस का कोड, डेटा, और अन्य संसाधन। यह प्रोसेस शुरू होने से पहले की स्थिति होती है।**

2. तैयार (Ready):

- **जब प्रोसेस को CPU द्वारा निष्पादित होने के लिए तैयार किया जाता है, तो वह "तैयार" अवस्था में होता है। इसका मतलब है कि प्रोसेस के पास सभी आवश्यक संसाधन (जैसे मेमोरी) होते हैं, लेकिन उसे अभी CPU द्वारा संसाधित नहीं किया जा रहा होता। प्रोसेस को एक निश्चित समय में CPU का समय मिलने का इंतजार होता है।**
- **तैयार अवस्था में प्रोसेस हमेशा CPU का उपयोग करने के लिए तैयार रहता है।**

3. निष्पादित हो रहा (Running):

- **जब प्रोसेस को CPU द्वारा समय मिलता है और वह कार्य करने लगता है, तो उसे "निष्पादित हो रहा" (Running) कहा जाता है। इस अवस्था में प्रोसेस अपनी कार्यक्षमता को पूरा कर रहा होता है, जैसे गणनाएँ करना, इनपुट प्राप्त करना या आउटपुट देना।**
- **केवल एक प्रोसेस को एक समय में CPU मिलता है, इसलिए यदि CPU पर कोई अन्य प्रोसेस चल रहा है, तो प्रोसेस तैयार स्थिति में रहता है।**

4. अवरोधित (Blocked / Waiting):

- **जब प्रोसेस किसी बाहरी संसाधन (जैसे I/O उपकरण) का इंतजार करता है, तो वह "अवरोधित" अवस्था में चला जाता है। उदाहरण के लिए, यदि एक प्रोसेस डेटा प्राप्त करने के लिए डिस्क से संपर्क करता है, तो उसे इस संसाधन के उपलब्ध होने का इंतजार करना पड़ता है।**
- **अवरोधित अवस्था में प्रोसेस निष्पादन नहीं कर सकता, और केवल तब ही तैयार अवस्था में वापस लौटता है जब वह आवश्यक संसाधन प्राप्त कर लेता है।**

5. समाप्त (Terminated / Exit):

- **जब प्रोसेस अपना कार्य पूरा कर लेता है, या किसी कारणवश उसे समाप्त किया जाता है, तो वह "समाप्त" अवस्था में चला जाता है। प्रोसेस के निष्पादन के बाद, ऑपरेटिंग सिस्टम उसे समाप्त कर देता है और उसके द्वारा उपयोग किए गए संसाधन मुक्त कर दिए जाते हैं।**

6. सस्पेंड (Suspended):

- **कभी-कभी प्रोसेस को अस्थायी रूप से निष्क्रिय किया जा सकता है, जो "सस्पेंड" अवस्था में जाता है। इसे तैयार या अवरोधित स्थिति में भी रखा जा सकता है। प्रोसेस को पुनः सक्रिय करने के लिए इसे तैयार अवस्था में वापस लाया जाता है। इसे आमतौर पर मेमोरी को बचाने के लिए किया जाता है।**

3. Explain different types of OS.

ANS:- ऑपरेटिंग सिस्टम (Operating System) :- ऑपरेटिंग सिस्टम (OS) एक सॉफ़्टवेयर है जो कंप्यूटर हार्डवेयर और सॉफ़्टवेयर के बीच मध्यस्थ का काम करता है। विभिन्न कार्यों और आवश्यकता के आधार पर ऑपरेटिंग सिस्टम के विभिन्न प्रकार होते हैं। आइए जानते हैं ऑपरेटिंग सिस्टम के प्रमुख प्रकार:

1. बैच प्रोसेसिंग ऑपरेटिंग सिस्टम (Batch Processing OS):

- **परिभाषा**: बैच प्रोसेसिंग ऑपरेटिंग सिस्टम एक प्रकार का ऑपरेटिंग सिस्टम होता है, जो बिना किसी यूज़र इंटरएक्शन के कार्यों (जैसे प्रोग्राम, जॉब्स) को एक बैच में समूहित करके निष्पादित करता है।
- **विशेषताएँ**:
  - इसमें उपयोगकर्ता को परिणाम के लिए इंतजार नहीं करना पड़ता है।
  - जॉब्स को बैच में रखा जाता है और वे एक बाद एक निष्पादित होते हैं।
  - ये सिस्टम पुराने कंप्यूटरों में पाए जाते हैं।
- **उदाहरण**: IBM की 7090 बैच प्रोसेसिंग सिस्टम।

2. टाइम-शेयरिंग ऑपरेटिंग सिस्टम (Time Sharing OS):

- **परिभाषा**: टाइम-शेयरिंग ऑपरेटिंग सिस्टम एक ऐसा OS है जो कई उपयोगकर्ताओं के बीच CPU के समय को विभाजित करता है, ताकि हर उपयोगकर्ता को निष्पादन के लिए कुछ समय मिल सके।
- **विशेषताएँ**:
  - यह मल्टीपल यूज़र्स के लिए रिसोर्सेज़ को शेयर करता है।
  - CPU का समय छोटे छोटे टुकड़ों में बांटा जाता है।
  - उपयोगकर्ता को त्वरित प्रतिक्रिया मिलती है।
- **उदाहरण**: UNIX, Multics।

3. मल्टीटास्किंग ऑपरेटिंग सिस्टम (Multitasking OS):

- **परिभाषा**: मल्टीटास्किंग ऑपरेटिंग सिस्टम वह ऑपरेटिंग सिस्टम है जो एक साथ कई कार्यों (tasks) को निष्पादित कर सकता है। यह एक प्रोसेस से दूसरे प्रोसेस में स्विच करता है, ताकि उपयोगकर्ता को एक साथ कई कार्य करने की सुविधा मिल सके।
- **विशेषताएँ**:
  - इसमें एक ही समय में एक से अधिक प्रक्रियाओं का निष्पादन होता है।
  - यह सिस्टम कार्यों के बीच स्विच करता रहता है।
- **उदाहरण**: Windows, Linux।

4. मल्टीप्रोसेसिंग ऑपरेटिंग सिस्टम (Multiprocessing OS):

- **परिभाषा**: मल्टीप्रोसेसिंग ऑपरेटिंग सिस्टम वह सिस्टम है, जो एक साथ कई प्रोसेसों को निष्पादित करने के लिए एक से अधिक CPU (प्रोसेसर) का उपयोग करता है।
- **विशेषताएँ**:

    - इसमें दो या दो से अधिक प्रोसेसर होते हैं जो समानांतर में कार्य करते हैं।
    - इससे कार्यों की गति बढ़ जाती है और दक्षता भी बढ़ती है।
- **उदाहरण**: UNIX, Linux (कुछ संस्करणों में), Windows Server।

5. मल्टीथ्रेडेड ऑपरेटिंग सिस्टम (Multithreaded OS):

- **परिभाषा**: मल्टीथ्रेडेड ऑपरेटिंग सिस्टम में एक ही प्रोसेस के भीतर विभिन्न थ्रेड्स होते हैं, जिन्हें एक साथ निष्पादित किया जाता है। हर थ्रेड को प्रोसेसर का समय प्राप्त होता है।
- **विशेषताएँ**:
    - एक ही कार्य को छोटे-छोटे हिस्सों में विभाजित कर उसे कई थ्रेड्स द्वारा संसाधित किया जाता है।
    - यह प्रोसेस को अधिक कुशल और तेज बनाता है।
- **उदाहरण**: Windows NT, Linux।

6. नेटवर्क ऑपरेटिंग सिस्टम (Network Operating System):

- **परिभाषा**: नेटवर्क ऑपरेटिंग सिस्टम एक ऐसा ऑपरेटिंग सिस्टम है जो नेटवर्क पर विभिन्न कंप्यूटरों को जोड़ने और उनके बीच संसाधन साझा करने की अनुमति देता है।
- **विशेषताएँ**:
    - यह एक नेटवर्क पर स्थित विभिन्न उपकरणों (जैसे सर्वर, कंप्यूटर) को नियंत्रित करता है।
    - यह उपयोगकर्ताओं को फाइल शेयरिंग, प्रिंटर, और अन्य नेटवर्क सेवाओं की अनुमति देता है।
- **उदाहरण**: Novell NetWare, Microsoft Windows Server।

7. डिस्ट्रीब्यूटेड ऑपरेटिंग सिस्टम (Distributed Operating System):

- **परिभाषा**: डिस्ट्रीब्यूटेड ऑपरेटिंग सिस्टम एक ऐसा सिस्टम है, जो कई कंप्यूटरों को एक साथ जोड़कर उन्हें एक यूनिट के रूप में कार्य करने के लिए प्रबंधित करता है।
- **विशेषताएँ**:
    - इसमें विभिन्न कंप्यूटर एक साथ मिलकर काम करते हैं, जिससे प्रदर्शन और विश्वसनीयता में वृद्धि होती है।
    - यह सिस्टम एक केंद्रीय स्थान से सभी कार्यों को नियंत्रित करता है।
- **उदाहरण**: Google's Android, Apache Hadoop।

8. सिंगल-यूज़र ऑपरेटिंग सिस्टम (Single-user Operating System):

- **परिभाषा**: सिंगल-यूज़र ऑपरेटिंग सिस्टम वह सिस्टम है, जो केवल एक ही उपयोगकर्ता को कंप्यूटर का उपयोग करने की अनुमति देता है।
- **विशेषताएँ**:
    - इसमें एक समय में केवल एक उपयोगकर्ता ही सिस्टम के साथ इंटरैक्ट कर सकता है।
    - यह सिस्टम सरल और छोटे उपकरणों के लिए उपयुक्त होता है।
- **उदाहरण**: MS-DOS, Windows 98।

9. सिंगल-टास्किंग ऑपरेटिंग सिस्टम (Single-tasking Operating System):

- **परिभाषा**: सिंगल-टास्किंग ऑपरेटिंग सिस्टम वह ऑपरेटिंग सिस्टम होता है जो केवल एक कार्य (टास्क) को निष्पादित करने में सक्षम होता है। एक बार एक कार्य पूरा होने के बाद ही दूसरा कार्य शुरू किया जा सकता है।
- **विशेषताएँ**:
    - एक समय में केवल एक ही कार्य किया जा सकता है।
- **उदाहरण**: MS-DOS।

4.Give the view of OS as User Interface.

ANS:- **ऑपरेटिंग सिस्टम (OS) के दृष्टिकोण से यूज़र इंटरफ़ेस (User Interface)**

यूज़र इंटरफ़ेस (UI) वह माध्यम है, जिसके द्वारा उपयोगकर्ता (user) और ऑपरेटिंग सिस्टम (OS) के बीच संवाद (interaction) स्थापित होता है। यह उपयोगकर्ता को सिस्टम से आसानी से जुड़ने और निर्देश देने की अनुमति देता है। ऑपरेटिंग सिस्टम के दृष्टिकोण से, यूज़र इंटरफ़ेस का उद्देश्य उपयोगकर्ता को कंप्यूटर की कार्यप्रणाली को समझने और नियंत्रित करने में सहायता प्रदान करना है।

आइए, ऑपरेटिंग सिस्टम के यूज़र इंटरफ़ेस के मुख्य पहलुओं को समझते हैं:

1. ग्राफिकल यूज़र इंटरफ़ेस (GUI):

- **परिभाषा**: GUI एक प्रकार का यूज़र इंटरफ़ेस होता है जिसमें उपयोगकर्ता ग्राफिक्स, चित्र, और आइकन (icons) के माध्यम से ऑपरेटिंग सिस्टम के साथ इंटरएक्ट करता है। इसमें उपयोगकर्ता को टाइप करने की बजाय, माउस, कीबोर्ड या टच स्क्रीन का उपयोग करके कार्यक्रमों को चलाने का विकल्प मिलता है।
- **विशेषताएँ**:
  - इसमें विंडो, बटन, मेनू और आइकन होते हैं, जो उपयोगकर्ता को विभिन्न कार्यों को आसान तरीके से करने की सुविधा देते हैं।
  - GUI यूज़र्स के लिए आकर्षक और उपयोग में आसान होता है।
- **उदाहरण**: Windows, macOS, Linux (Ubuntu के GUI संस्करण)।

2. कमांड-लाइन इंटरफ़ेस (CLI):

- **परिभाषा**: CLI वह इंटरफ़ेस है जिसमें उपयोगकर्ता को कंप्यूटर को निर्देश देने के लिए टेक्स्ट कमांड (commands) टाइप करने होते हैं। इसमें ग्राफिकल तत्व नहीं होते हैं, और उपयोगकर्ता को आदेश देने के लिए पूरी तरह से कीबोर्ड का ही उपयोग करना पड़ता है।
- **विशेषताएँ**:
  - CLI का उपयोग तकनीकी विशेषज्ञों द्वारा किया जाता है, क्योंकि इसमें उपयोगकर्ता को टेक्स्ट कमांड का ज्ञान होना चाहिए।
  - यह पुराना, लेकिन शक्तिशाली और तेज़ तरीका है।
- **उदाहरण**: MS-DOS, Linux का Terminal, macOS का Terminal।

3. माउस और कीबोर्ड इंटरफ़ेस:

- **परिभाषा**: इस इंटरफ़ेस में उपयोगकर्ता माउस और कीबोर्ड का उपयोग करके ऑपरेटिंग सिस्टम के साथ संवाद करते हैं। माउस के द्वारा स्क्रीन पर स्थित आइकन, बटन, और मेनू पर क्लिक करके उपयोगकर्ता आदेश देते हैं, जबकि कीबोर्ड का उपयोग टाइपिंग और शॉर्टकट्स के लिए किया जाता है।
- **विशेषताएँ**:
  - यह यूज़र फ्रेंडली होता है और यूज़र्स को इंटरैक्शन के लिए दोनों इनपुट डिवाइसों (माउस और कीबोर्ड) की सुविधा प्रदान करता है।
  - अधिकतर ग्राफिकल यूज़र इंटरफेस में माउस और कीबोर्ड का संयोजन होता है।
- **उदाहरण**: Windows OS, macOS।

4. टच इंटरफ़ेस:

- **परिभाषा**: टच इंटरफ़ेस में उपयोगकर्ता स्क्रीन पर टच करके ऑपरेटिंग सिस्टम के साथ इंटरैक्ट करता है। इसमें स्क्रीन पर अलग-अलग ऑप्शन्स और आइकन होते हैं, जिन्हें उपयोगकर्ता अपनी अंगुली से या स्टाइलस से टैप करके नियंत्रित कर सकते हैं।
- **विशेषताएँ**:
    - यह स्मार्टफोन, टैबलेट और कुछ लैपटॉप्स पर सबसे सामान्य इंटरफ़ेस है।
    - उपयोगकर्ता के पास ज्यादा सहज और त्वरित नियंत्रण होता है, जैसे स्वाइप करना, टैप करना, ज़ूम इन या आउट करना।
- **उदाहरण**: Android, iOS, Windows 8/10 टैबलेट मोड।

5. वॉयस इंटरफ़ेस (Voice User Interface - VUI):

- **परिभाषा**: वॉयस इंटरफ़ेस में उपयोगकर्ता अपने वॉयस कमांड्स (आवाज़ के आदेश) द्वारा ऑपरेटिंग सिस्टम के साथ इंटरैक्ट करते हैं। इसमें उपयोगकर्ता बोलकर कंप्यूटर को निर्देश देते हैं, और सिस्टम उस निर्देश को समझकर कार्य करता है।
- **विशेषताएँ**:
    - यह उपयोगकर्ताओं को बिना हाथ लगाए और जल्दी से कार्यों को पूरा करने की सुविधा देता है।
    - आमतौर पर यह स्मार्ट डिवाइसों में पाया जाता है, जैसे स्मार्ट स्पीकर्स।
- **उदाहरण**: Google Assistant, Siri, Amazon Alexa।

ऑपरेटिंग सिस्टम में यूज़र इंटरफ़ेस के उद्देश्य:

- **सुलभता (Accessibility)**: ऑपरेटिंग सिस्टम का मुख्य उद्देश्य यह सुनिश्चित करना है कि उपयोगकर्ता बिना किसी कठिनाई के सिस्टम का उपयोग कर सके। सही UI डिजाइन उपयोगकर्ताओं को आसानी से अपने कार्यों को पूरा करने में मदद करता है।
- **सहजता (Usability)**: UI का उद्देश्य उपयोगकर्ता के लिए सरल और सहज होना चाहिए। उपयोगकर्ता को बिना अधिक प्रयास के काम करने का अनुभव प्राप्त करना चाहिए।
- **कार्यक्षमता (Efficiency)**: एक अच्छा UI उपयोगकर्ता को कार्यों को तेज़ और प्रभावी रूप से पूरा करने का अवसर देता है। इसके लिए शॉर्टकट्स, बेहतर नेविगेशन, और ऑटोमेशन जैसी सुविधाएँ जरूरी होती हैं।
- **सुरक्षा (Security)**: ऑपरेटिंग सिस्टम का UI यूज़र को सुरक्षित रूप से सिस्टम में लॉगिन करने और व्यक्तिगत जानकारी तक पहुंच प्रदान करने में मदद करता है। पासवर्ड एंटर करने, दो-चरण प्रमाणीकरण (two-factor authentication) जैसी सुरक्षा प्रक्रिया भी UI का हिस्सा होती है।

5.What is Operating System? Define its architecture?

ANS:- **ऑपरेटिंग सिस्टम (Operating System) :-**

ऑपरेटिंग सिस्टम (OS) एक सॉफ़्टवेयर है जो कंप्यूटर के हार्डवेयर और उपयोगकर्ता के बीच इंटरफेस के रूप में कार्य करता है। यह कंप्यूटर सिस्टम की संसाधनों (जैसे प्रोसेसर, मेमोरी, स्टोरेज, आदि) का प्रबंधन करता है और उपयोगकर्ताओं को विभिन्न एप्लिकेशन प्रोग्रामों को चलाने की अनुमति देता है। ऑपरेटिंग सिस्टम यह सुनिश्चित करता है कि सभी कार्य सुचारु रूप से चलते रहें और उपयोगकर्ता आसानी से सिस्टम का उपयोग कर सकें।

ऑपरेटिंग सिस्टम की वास्तुकला (Architecture)

ऑपरेटिंग सिस्टम की वास्तुकला का मतलब है ऑपरेटिंग सिस्टम के विभिन्न घटकों की संरचना और उनका आपस में कार्य करने का तरीका। ऑपरेटिंग सिस्टम की वास्तुकला आमतौर पर निम्नलिखित प्रमुख घटकों से मिलकर बनती है:

1. **कर्नल (Kernel):**
   कर्नल ऑपरेटिंग सिस्टम का मुख्य हिस्सा है। यह सिस्टम के सभी महत्वपूर्ण कार्यों को नियंत्रित करता है, जैसे कि प्रोसेस मैनेजमेंट, मेमोरी मैनेजमेंट, डिवाइस ड्राइवर्स, फ़ाइल सिस्टम, और नेटवर्किंग। कर्नल को ऑपरेटिंग सिस्टम का दिल कहा जा सकता है।
2. **मेमोरी मैनेजमेंट (Memory Management):**
   यह घटक कंप्यूटर की मेमोरी का प्रबंधन करता है। यह सुनिश्चित करता है कि प्रत्येक प्रक्रिया को आवश्यक मात्रा में मेमोरी मिले और मेमोरी के सीमित संसाधनों का सही उपयोग हो।
3. **प्रोसेस मैनेजमेंट (Process Management):**
   यह घटक विभिन्न प्रक्रियाओं (processes) का प्रबंधन करता है। यह तय करता है कि कौन सी प्रक्रिया कब और कितनी देर तक चलनी चाहिए, और यह सुनिश्चित करता है कि प्रक्रियाएँ एक-दूसरे से टकराएं नहीं।
4. **फाइल सिस्टम (File System):**
   फाइल सिस्टम का काम डेटा को संग्रहित करना और उसे सही तरीके से व्यवस्थित करना है। यह फाइलों को नाम, स्थिति और प्रकार के हिसाब से संगठित करता है और डेटा तक पहुंचने के लिए एक इंटरफेस प्रदान करता है।
5. **डिवाइस ड्राइवर्स (Device Drivers):**
   ऑपरेटिंग सिस्टम विभिन्न हार्डवेयर उपकरणों (जैसे कि प्रिंटर, माउस, कीबोर्ड, डिस्क ड्राइव आदि) के साथ संवाद करने के लिए डिवाइस ड्राइवर्स का उपयोग करता है। ये ड्राइवर्स हार्डवेयर और सॉफ़्टवेयर के बीच मध्यस्थ के रूप में काम करते हैं।
6. **यूज़र इंटरफेस (User Interface):**
   ऑपरेटिंग सिस्टम का यह हिस्सा उपयोगकर्ता और सिस्टम के बीच इंटरैक्शन प्रदान करता है। यह ग्राफिकल यूजर इंटरफेस (GUI) या कमांड-लाइन इंटरफेस (CLI) हो सकता है। यह उपयोगकर्ताओं को सिस्टम से संवाद करने की अनुमति देता है।

ऑपरेटिंग सिस्टम की वास्तुकला के प्रकार:

1. **मोनोलिथिक आर्किटेक्चर (Monolithic Architecture):**
   इस प्रकार में, सभी ऑपरेटिंग सिस्टम के घटक कर्नल में समाहित होते हैं। यह एक बड़े कोड बेस में होता है, जहां सभी कार्य एक साथ होते हैं। उदाहरण: UNIX, Linux
2. **माइक्रोकर्नल आर्किटेक्चर (Microkernel Architecture):**
   इसमें कर्नल का आकार छोटा होता है और केवल सबसे महत्वपूर्ण कार्य (जैसे प्रोसेस मैनेजमेंट) करता है। अन्य कार्य बाहरी मॉड्यूल्स द्वारा किए जाते हैं। उदाहरण: Minix, QNX
3. **हाइब्रिड आर्किटेक्चर (Hybrid Architecture):**
   यह मोनोलिथिक और माइक्रोकर्नल दोनों की विशेषताएँ मिलाकर बनाई जाती है। कर्नल में कुछ घटक एकजुट होते हैं, जबकि अन्य बाहरी होते हैं। उदाहरण: Windows NT, macOS

6.Explain different types of operating system?

ANS:- ऑपरेटिंग सिस्टम (OS) विभिन्न प्रकार के होते हैं, और हर प्रकार को विशेष कार्यों और उपयोगकर्ताओं की आवश्यकताओं के आधार पर डिज़ाइन किया जाता है। प्रत्येक ऑपरेटिंग सिस्टम का उद्देश्य हार्डवेयर संसाधनों का प्रबंधन करना और उपयोगकर्ता और एप्लिकेशन को सेवाएं प्रदान करना है। नीचे विभिन्न प्रकार के ऑपरेटिंग सिस्टम की व्याख्या दी गई है:

1. बैच ऑपरेटिंग सिस्टम (Batch Operating System)

- **परिभाषा**: बैच ऑपरेटिंग सिस्टम में समान प्रकार के कार्यों को एक साथ समूहित किया जाता है और बिना किसी उपयोगकर्ता इंटरएक्शन के क्रम में निष्पादित किया जाता है।
- **विशेषताएँ**:
  - कार्यों को बैचों में समूहित किया जाता है।
  - उपयोगकर्ता का कार्य निष्पादन के दौरान कोई हस्तक्षेप नहीं होता।
  - यह समय बचाता है और दोहराए जाने वाले कार्यों के लिए प्रभावी होता है।
- **उदाहरण**: IBM के पुराने मेनफ्रेम सिस्टम।

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

- **परिभाषा**: टाइम-शेयरिंग ऑपरेटिंग सिस्टम का उद्देश्य एक समय में एक से अधिक उपयोगकर्ताओं को कंप्यूटर संसाधनों का उपयोग करने की अनुमति देना है।
- **विशेषताएँ**:
  - एक से अधिक उपयोगकर्ता एक साथ सिस्टम का उपयोग कर सकते हैं।
  - CPU समय को छोटे-छोटे टुकड़ों में बाँट दिया जाता है।
  - उपयोगकर्ताओं को इंटरएक्टिव तरीके से काम करने की अनुमति मिलती है।
- **उदाहरण**: UNIX, Multics।

3. रियल-टाइम ऑपरेटिंग सिस्टम (Real-Time Operating System - RTOS)

- **परिभाषा**: रियल-टाइम ऑपरेटिंग सिस्टम उन अनुप्रयोगों के लिए डिज़ाइन किया गया है जिन्हें सटीक समय और उच्च विश्वसनीयता की आवश्यकता होती है।
- **विशेषताएँ**:
  - समय सीमा के भीतर कार्यों को निष्पादित करना।
  - हार्ड रियल-टाइम और सॉफ्ट रियल-टाइम में वर्गीकृत किया जा सकता है।
  - समय-संवेदनशील कार्यों के लिए उपयोगी।
- **उदाहरण**: VxWorks, FreeRTOS, RTEMS।

4. वितरित ऑपरेटिंग सिस्टम (Distributed Operating System)

- **परिभाषा**: वितरित ऑपरेटिंग सिस्टम एकाधिक कंप्यूटरों को जोड़ता है और इन्हें एक एकल सिस्टम के रूप में प्रस्तुत करता है।
- **विशेषताएँ**:
  - विभिन्न कंप्यूटरों को नेटवर्क के माध्यम से जोड़ा जाता है।
  - संसाधनों को साझा करने और समन्वय करने के लिए एक साथ काम करते हैं।
  - प्रणाली की विश्वसनीयता और संसाधन साझा करने में मदद करता है।
- **उदाहरण**: Google Android, Apache Hadoop, Plan 9।

5. नेटवर्क ऑपरेटिंग सिस्टम (Network Operating System - NOS)

- **परिभाषा**: नेटवर्क ऑपरेटिंग सिस्टम नेटवर्क संसाधनों का प्रबंधन करता है और उपयोगकर्ताओं को नेटवर्क के माध्यम से विभिन्न सेवाएं प्रदान करता है।
- **विशेषताएँ**:
  - नेटवर्क के भीतर संसाधनों को साझा करना।
  - फाइल शेयरिंग, प्रिंटर शेयरिंग और नेटवर्क सुरक्षा प्रदान करता है।
  - नेटवर्क के विभिन्न कंप्यूटरों के बीच संचार स्थापित करता है।
- **उदाहरण**: Microsoft Windows Server, Novell NetWare, Linux-based systems।

6. एंबेडेड ऑपरेटिंग सिस्टम (Embedded Operating System)

- **परिभाषा**: एंबेडेड ऑपरेटिंग सिस्टम विशिष्ट कार्यों को पूरा करने के लिए डिज़ाइन किए गए एंबेडेड सिस्टम्स में उपयोग होता है, जैसे कि स्मार्ट डिवाइसेज़ और IoT (Internet of Things) उपकरण।
- **विशेषताएँ**:
  - छोटे आकार में और न्यूनतम संसाधनों के साथ काम करता है।
  - विशेष हार्डवेयर के लिए अनुकूलित होता है।
  - अक्सर रियल-टाइम होती है।
- **उदाहरण**: Embedded Linux, Windows CE, QNX।

7. मोबाइल ऑपरेटिंग सिस्टम (Mobile Operating System)

- **परिभाषा**: मोबाइल ऑपरेटिंग सिस्टम स्मार्टफोन, टैबलेट और अन्य मोबाइल डिवाइसों के लिए डिज़ाइन किया गया है।
- **विशेषताएँ**:
  - टच स्क्रीन इंटरफेस के लिए अनुकूलित।
  - बैटरी जीवन और संसाधन प्रबंधन पर ध्यान केंद्रित करता है।
  - वायरलेस संचार, ऐप्स और लोकेशन सेवाओं का समर्थन करता है।
- **उदाहरण**: Android, iOS, Windows Phone।

8. सिंगल-यूज़र, सिंगल-टास्क ऑपरेटिंग सिस्टम (Single-User, Single-Task Operating System)

- **परिभाषा**: सिंगल-यूज़र, सिंगल-टास्क ऑपरेटिंग सिस्टम केवल एक उपयोगकर्ता को एक समय में एक कार्य करने की अनुमति देता है।
- **विशेषताएँ**:
  - केवल एक कार्य को एक समय में चलाने की अनुमति होती है।
  - सरल और उपयोग में आसान होता है।
- **उदाहरण**: MS-DOS, Windows 3.x के पहले संस्करण।

9. सिंगल-यूज़र, मल्टी-टास्क ऑपरेटिंग सिस्टम (Single-User, Multi-Task Operating System)

- **परिभाषा**: इस प्रकार का ऑपरेटिंग सिस्टम एक उपयोगकर्ता को एक समय में कई कार्यों को निष्पादित करने की अनुमति देता है।

- **विशेषताएँ**:
  - एक उपयोगकर्ता द्वारा एक साथ कई एप्लिकेशन चलाए जा सकते हैं।
  - इंटरएक्टिव कार्यों के लिए उपयुक्त।
- **उदाहरण**: Windows, macOS, Linux (पर्सनल उपयोग के लिए)।

10. मल्टी-यूज़र ऑपरेटिंग सिस्टम (Multi-User Operating System)

- **परिभाषा**: मल्टी-यूज़र ऑपरेटिंग सिस्टम एक साथ कई उपयोगकर्ताओं को कंप्यूटर संसाधनों का उपयोग करने की अनुमति देता है।
- **विशेषताएँ**:
  - एक समय में कई उपयोगकर्ता कंप्यूटर का उपयोग कर सकते हैं।
  - सुरक्षा और उपयोगकर्ता प्रबंधन के लिए विशेष तंत्र होते हैं।
- **उदाहरण**: UNIX, Linux, IBM mainframe systems (z/OS)।

11. हाइब्रिड ऑपरेटिंग सिस्टम (Hybrid Operating System)

- **परिभाषा**: हाइब्रिड ऑपरेटिंग सिस्टम मोनोलिथिक और माइक्रोकर्नल दोनों की विशेषताओं को मिलाकर बनाया जाता है।
- **विशेषताएँ**:
  - मोनोलिथिक सिस्टम की प्रदर्शन क्षमता और माइक्रोकर्नल की मॉड्यूलर संरचना का मिश्रण।
  - यह सिस्टम की स्थिरता और प्रदर्शन को बेहतर बनाता है।
- **उदाहरण**: Windows NT, macOS, और आधुनिक Linux संस्करण।

7.What is process? Define the five state process models with transition?

ANS:- प्रोसेस (Process) :-

प्रोसेस एक प्रोग्राम है जो कंप्यूटर की मेमोरी में चल रहा होता है। जब हम कोई प्रोग्राम (जैसे किसी सॉफ़्टवेयर एप्लिकेशन) चलाते हैं, तो वह प्रोग्राम एक प्रोसेस बन जाता है। एक प्रोसेस में कोड, डेटा और प्रोसेस संबंधित जानकारी जैसे रजिस्टर्स, स्टैक और अन्य संसाधन शामिल होते हैं। प्रोसेस की स्थिति समय-समय पर बदलती रहती है क्योंकि वह CPU का उपयोग करता है और विभिन्न कार्य करता है।

प्रोसेस की अवस्थाएँ (Process States)

प्रोसेस का जीवनकाल विभिन्न अवस्थाओं (States) से गुजरता है, और इन अवस्थाओं के बीच संक्रमण (Transitions) होते रहते हैं। इन अवस्थाओं को **प्रोसेस मॉडल** के रूप में समझा जा सकता है। यहाँ पाँच मुख्य अवस्थाएँ हैं जिनमें प्रोसेस हो सकता है:

1. New (नया)

- **परिभाषा**: जब एक प्रोसेस को सिस्टम में बनाया जाता है, तो वह "नई" अवस्था में होता है। इस समय प्रोसेस को मेमोरी में लोड किया जाता है और संसाधन आवंटित किए जाते हैं।
- **संक्रमण**: जब प्रोसेस को सही संसाधन मिल जाते हैं, तो वह "Ready" अवस्था में ट्रांसफर हो जाता है।

2. Ready (तैयार)

- **परिभाषा**: प्रोसेस "Ready" अवस्था में तब होता है जब वह CPU के लिए तैयार होता है, यानी वह चलने के लिए प्रतीक्षारत है, लेकिन अभी तक उसे CPU नहीं मिला है।
- **संक्रमण**: प्रोसेस CPU के लिए तैयार है, और जब CPU उपलब्ध होता है, तो प्रोसेस "Running" अवस्था में ट्रांसफर हो जाता है। या यदि CPU का उपयोग अन्य प्रोसेस द्वारा हो रहा हो, तो वह फिर से "Ready" अवस्था में लौट सकता है।

3. Running (चल रहा)

- **परिभाषा**: प्रोसेस "Running" अवस्था में तब होता है जब CPU उसे निष्पादित कर रहा होता है। इस समय प्रोसेस अपने निर्धारित कार्यों को करता है।
- **संक्रमण**:
  - यदि प्रोसेस को CPU की आवश्यकता समाप्त हो जाती है या यदि वह I/O ऑपरेशन की आवश्यकता को पूरा करता है, तो वह "Waiting" या "Blocked" अवस्था में जा सकता है।
  - यदि प्रोसेस का समय समाप्त हो जाता है (time slice), तो वह "Ready" अवस्था में वापस आ सकता है।

4. Waiting (प्रतीक्षारत) / Blocked

- **परिभाषा**: प्रोसेस "Waiting" या "Blocked" अवस्था में तब होता है जब उसे किसी बाहरी घटना, जैसे I/O ऑपरेशन, के पूरा होने का इंतजार करना होता है। इस अवस्था में प्रोसेस CPU का उपयोग नहीं कर सकता।
- **संक्रमण**: प्रोसेस फिर से "Ready" अवस्था में ट्रांसफर हो सकता है जब वह बाहरी घटना पूरी हो जाती है (जैसे डेटा प्राप्त करना या किसी डिवाइस से कनेक्शन स्थापित करना)।

5. Terminated (समाप्त)

- **परिभाषा**: जब प्रोसेस अपने कार्य को पूरा कर लेता है या उसे किसी कारणवश समाप्त किया जाता है, तो वह "Terminated" अवस्था में पहुँच जाता है।
- **संक्रमण**: यह अवस्था अंतिम होती है। प्रोसेस के समाप्त होने के बाद, उसकी मेमोरी को मुक्त कर दिया जाता है और अन्य संसाधनों को भी पुनः आवंटित किया जाता है।

प्रोसेस स्टेट ट्रांज़िशन (Process State Transitions)

प्रोसेस के जीवन में विभिन्न अवस्थाओं के बीच कई प्रकार के संक्रमण होते हैं। निम्नलिखित प्रमुख संक्रमण होते हैं:

1. **New → Ready**: जब प्रोसेस को बनाने के बाद संसाधन मिलते हैं और वह CPU के लिए तैयार हो जाता है।
2. **Ready → Running**: जब CPU उपलब्ध होता है और प्रोसेस निष्पादन के लिए चयनित होता है।
3. **Running → Waiting/Blocked**: जब प्रोसेस को कुछ I/O ऑपरेशन की प्रतीक्षा होती है और वह CPU से हटा दिया जाता है।
4. **Running → Ready**: यदि प्रोसेस का समय समाप्त हो जाता है (time slice) तो वह फिर से "Ready" अवस्था में लौट आता है।
5. **Waiting/Blocked → Ready**: जब I/O ऑपरेशन पूरा हो जाता है, तो प्रोसेस फिर से "Ready" अवस्था में लौट आता है।
6. **Running → Terminated**: जब प्रोसेस अपना कार्य पूरा कर लेता है, तो वह समाप्त हो जाता है।

8.What is CPU scheduling? Explain different types of CPU Scheduling Algorithms?

ANS:- CPU Scheduling :-

CPU Scheduling कंप्यूटर सिस्टम में CPU समय के प्रभावी उपयोग को सुनिश्चित करने की प्रक्रिया है। जब एक साथ कई प्रोग्राम या प्रक्रियाएँ चल रही होती हैं, तो ऑपरेटिंग सिस्टम यह तय करता है कि कौन सी प्रक्रिया CPU का उपयोग करेगी और कब। यह निर्णय CPU Scheduling Algorithms के माध्यम से लिया जाता है। CPU Scheduling मुख्य रूप से मल्टीटास्किंग और मल्टीप्रोग्रामिंग वातावरण में सिस्टम के प्रदर्शन को बेहतर बनाने के लिए उपयोग किया जाता है।

CPU Scheduling Algorithms के प्रकार:

1. **First-Come, First-Served (FCFS)**
   - यह सबसे सरल Scheduling Algorithm है।
   - जो प्रक्रिया सबसे पहले आती है, उसे सबसे पहले CPU का समय दिया जाता है।
   - **लाभ**: सरल और आसान।
   - **नुकसान**: Convoy Effect (लंबी प्रक्रिया के कारण छोटी प्रक्रियाएँ इंतजार करती हैं)।
2. **Shortest Job Next (SJN) या Shortest Job First (SJF)**
   - सबसे छोटी CPU बर्स्ट टाइम वाली प्रक्रिया को प्राथमिकता दी जाती है।
   - **लाभ**: न्यूनतम औसत प्रतीक्षा समय।
   - **नुकसान**: Starvation (लंबी प्रक्रियाओं को नजरअंदाज किया जा सकता है)।
   - यह **Non-Preemptive** और **Preemptive** दोनों हो सकता है।
3. **Priority Scheduling**
   - प्रत्येक प्रक्रिया को एक प्राथमिकता दी जाती है, और उच्च प्राथमिकता वाली प्रक्रिया को पहले CPU दिया जाता है।
   - **लाभ**: महत्वपूर्ण प्रक्रियाओं को प्राथमिकता।
   - **नुकसान**: Starvation (कम प्राथमिकता वाली प्रक्रियाएँ लंबे समय तक इंतजार कर सकती हैं)।
   - **Solution**: Aging (समय के साथ प्राथमिकता बढ़ाना)।
4. **Round Robin (RR)**
   - प्रत्येक प्रक्रिया को CPU का समय (Time Quantum) समान रूप से दिया जाता है।
   - जब Time Quantum समाप्त हो जाता है, तो प्रक्रिया को कतार के अंत में भेज दिया जाता है।
   - **लाभ**: समय-साझा (Time-Sharing) सिस्टम के लिए उपयुक्त।
   - **नुकसान**: अगर Time Quantum ठीक से सेट न हो, तो प्रदर्शन कम हो सकता है।
5. **Multilevel Queue Scheduling**
   - प्रक्रियाओं को उनकी प्राथमिकताओं या विशेषताओं (जैसे बैच प्रोसेस, इंटरएक्टिव प्रोसेस) के आधार पर विभिन्न कतारों में विभाजित किया जाता है।
   - हर कतार का अलग-अलग Scheduling Algorithm होता है।
   - **लाभ**: लचीलापन।
   - **नुकसान**: जटिलता।
6. **Multilevel Feedback Queue Scheduling**
   - यह Multilevel Queue Scheduling का एक बेहतर रूप है।
   - प्रक्रिया को उसके प्रदर्शन के आधार पर एक कतार से दूसरी कतार में स्थानांतरित किया जा सकता है।
   - **लाभ**: अधिकतम CPU उपयोग और बेहतर प्रतिक्रिया समय।
   - **नुकसान**: जटिलता।

9. What is memory management? Define the techniques and requirement of memory management?

Ans:- Memory Management :-

Memory Management कंप्यूटर सिस्टम में उपलब्ध मुख्य मेमोरी (RAM) के कुशल उपयोग को सुनिश्चित करने की प्रक्रिया है। यह सुनिश्चित करता है कि प्रक्रियाएँ मेमोरी का उपयोग प्रभावी ढंग से करें और सिस्टम का प्रदर्शन बेहतर बना रहे। ऑपरेटिंग सिस्टम मेमोरी मैनेजमेंट के लिए जिम्मेदार होता है, जो यह तय करता है कि कौन सी प्रक्रिया मेमोरी में कब और कितनी जगह लेगी।

Memory Management की आवश्यकता:

1. **प्रोसेस का निष्पादन (Process Execution)**
   - किसी भी प्रक्रिया को निष्पादित करने के लिए उसे मुख्य मेमोरी में लोड करना आवश्यक है।
2. **कुशल संसाधन उपयोग (Efficient Resource Utilization)**
   - सीमित मेमोरी को प्रभावी तरीके से वितरित करना।
3. **प्रोसेस आइसोलेशन (Process Isolation)**
   - एक प्रक्रिया की मेमोरी को दूसरी प्रक्रिया से अलग रखना ताकि डेटा सुरक्षा बनी रहे।
4. **मल्टीप्रोग्रामिंग को सक्षम करना (Enable Multiprogramming)**
   - एक साथ कई प्रक्रियाओं को मेमोरी में रखने की क्षमता।
5. **तेज़ प्रदर्शन (Fast Performance)**
   - मेमोरी प्रबंधन सिस्टम मेमोरी की कुशलता से पहुँच को सुनिश्चित करता है।

Memory Management Techniques:

1. **Partitioning (पार्टीशनिंग)**
   - मेमोरी को छोटे-छोटे हिस्सों में विभाजित किया जाता है।
   - दो प्रकार हैं:
     - **Fixed Partitioning**: मेमोरी को स्थायी आकार के भागों में विभाजित किया जाता है।
     - **Dynamic Partitioning**: मेमोरी को प्रक्रियाओं की आवश्यकता के अनुसार विभाजित किया जाता है।
   - **समस्या**: **External Fragmentation** (मेमोरी के छोटे-छोटे खाली हिस्से)।
2. **Paging (पेजिंग)**
   - मेमोरी को समान आकार के पेज फ्रेम्स में विभाजित किया जाता है।
   - प्रोसेस को भी छोटे पेजेज में विभाजित किया जाता है, जो फ्रेम्स में फिट होते हैं।
   - **लाभ**: External Fragmentation से बचाव।
   - **नुकसान**: **Internal Fragmentation** (फ्रेम में अप्रयुक्त मेमोरी)।
3. **Segmentation (सेगमेंटेशन)**
   - मेमोरी को लॉजिकल सेगमेंट्स में विभाजित किया जाता है, जैसे कोड, डेटा, और स्टैक।
   - यह प्रक्रिया की संरचना को बेहतर तरीके से दर्शाता है।
   - **समस्या**: External Fragmentation हो सकती है।
4. **Virtual Memory (वर्चुअल मेमोरी)**
   - यह तकनीक मुख्य मेमोरी की तुलना में बड़ी मेमोरी की प्रतीत प्रदान करती है।
   - डेटा का वह हिस्सा जो तत्काल आवश्यक नहीं है, उसे डिस्क (secondary memory) में संग्रहित किया जाता है।
   - **लाभ**: बड़ी प्रक्रियाओं को निष्पादित करना संभव बनाता है।

- **नुकसान**: **Thrashing** (प्रक्रिया का बार-बार पेज स्वैप करना)।

5. **Swapping (स्वैपिंग)**
   - एक प्रक्रिया को मेमोरी से हटा कर डिस्क में स्थानांतरित किया जाता है और आवश्यकता होने पर वापस लाया जाता है।
   - **लाभ**: मल्टीप्रोग्रामिंग में सहायक।
   - **नुकसान**: समय लगने वाला (I/O Overhead)।

Memory Management की आवश्यकताएँ:

1. **रिलायबिलिटी (Reliability)**
   - प्रक्रिया के डेटा को सुरक्षित रखना।
2. **Efficiency (दक्षता)**
   - मेमोरी का कुशल उपयोग सुनिश्चित करना।
3. **Isolation (आइसोलेशन)**
   - प्रक्रियाओं को एक-दूसरे के हस्तक्षेप से बचाना।
4. **Dynamic Allocation (डायनेमिक अलोकेशन)**
   - प्रक्रियाओं को उनकी आवश्यकता के अनुसार मेमोरी प्रदान करना।
5. **Fragmentation Management (फ्रैगमेंटेशन का प्रबंधन)**
   - Internal और External Fragmentation को कम करना।

10.What is different between contiguous and non-contiguous memory allocation?

Ans:- **Contiguous और Non-Contiguous Memory Allocation में अंतर:**

1. परिभाषा (Definition):

- **Contiguous Memory Allocation (सन्निहित मेमोरी आवंटन) :**
  - इसमें एक प्रक्रिया को मेमोरी में निरंतर (एक साथ) स्थान दिया जाता है।
  - प्रक्रिया का डेटा मेमोरी में लगातार ब्लॉक्स में स्टोर होता है।
- **Non-Contiguous Memory Allocation (असन्निहित मेमोरी आवंटन) :**
  - इसमें एक प्रक्रिया के लिए मेमोरी के विभिन्न स्थानों पर ब्लॉक्स आवंटित किए जा सकते हैं।
  - प्रक्रिया का डेटा मेमोरी में अलग-अलग जगहों पर स्टोर होता है।

---

2. मेमोरी फ्रैगमेंटेशन (Memory Fragmentation):

- **Contiguous Allocation:**
  - **External Fragmentation** (मेमोरी के छोटे-छोटे खाली हिस्से) की समस्या होती है।
  - यह मेमोरी उपयोग को सीमित कर सकता है।
- **Non-Contiguous Allocation:**
  - External Fragmentation की समस्या नहीं होती।
  - लेकिन **Internal Fragmentation** (ब्लॉक में अप्रयुक्त मेमोरी) हो सकती है।

---

3. प्रदर्शन (Performance):

- **Contiguous Allocation:**
  - मेमोरी एक्सेस तेज़ होता है क्योंकि डेटा निरंतर स्थान पर होता है।
  - **लाभ:** CPU और I/O डिवाइस के लिए आसान और कुशल।
- **Non-Contiguous Allocation:**
  - मेमोरी एक्सेस धीमा हो सकता है क्योंकि डेटा अलग-अलग स्थानों पर फैला होता है।
  - पेजिंग और सेगमेंटेशन के लिए मेमोरी मैनेजमेंट यूनिट (MMU) की आवश्यकता होती है।

---

4. जटिलता (Complexity):

- **Contiguous Allocation:**
  - सरल और कम जटिल।
  - मेमोरी आवंटन और डि-आवंटन करना आसान है।
- **Non-Contiguous Allocation:**
  - जटिल और अधिक प्रबंधन की आवश्यकता।
  - पेज टेबल और सेगमेंट टेबल जैसे अतिरिक्त डेटा संरचनाओं की आवश्यकता होती है।

---

5. उपयोग (Usage):

- **Contiguous Allocation:**
  - यह सिस्टम में सरल प्रक्रियाओं और फाइल स्टोरेज में उपयोगी है।
  - मुख्यतः पुरानी सिस्टम डिज़ाइन में देखा जाता है।
- **Non-Contiguous Allocation:**
  - मल्टीप्रोग्रामिंग, वर्चुअल मेमोरी, और बड़े डेटा सेट्स के लिए उपयुक्त।
  - आधुनिक ऑपरेटिंग सिस्टम (जैसे पेजिंग और सेगमेंटेशन) में अधिक उपयोग होता है।

---

6. उदाहरण (Examples):

- **Contiguous Allocation:**
  - Fixed Partitioning, Dynamic Partitioning।
- **Non-Contiguous Allocation:**
  - Paging, Segmentation, Virtual Memory।

11. What is device management? Write the main functions of device management in OS?

Ans:- **Device Management क्या है?**

Device Management ऑपरेटिंग सिस्टम (OS) का एक महत्वपूर्ण कार्य है, जो सिस्टम में उपलब्ध हार्डवेयर डिवाइस (जैसे प्रिंटर, हार्ड डिस्क, कीबोर्ड, माउस, आदि) का प्रबंधन करता है। यह सुनिश्चित करता है कि डिवाइस कुशलता से उपयोग किए जाएँ, और विभिन्न प्रक्रियाएँ हार्डवेयर संसाधनों को साझा कर सकें।

डिवाइस मैनेजमेंट का मुख्य उद्देश्य हार्डवेयर डिवाइस को प्रक्रियाओं के लिए उपलब्ध कराना, डेटा ट्रांसफर का प्रबंधन करना, और डिवाइस एक्सेस को नियंत्रित करना है।

Device Management के मुख्य कार्य:

1. **Device Communication (डिवाइस संचार) :**
   - OS यह सुनिश्चित करता है कि सिस्टम और इनपुट/आउटपुट (I/O) डिवाइस के बीच संचार सही ढंग से हो।
   - यह डेटा ट्रांसफर के लिए उचित प्रोटोकॉल और ड्राइवर्स का उपयोग करता है।
2. **Device Allocation (डिवाइस आवंटन) :**
   - OS यह तय करता है कि कौन सी प्रक्रिया को किस डिवाइस का उपयोग करना है।
   - जब कई प्रक्रियाएँ एक ही डिवाइस का उपयोग करना चाहती हैं, तो OS आवंटन का प्रबंधन करता है।
3. **Device Deallocation (डिवाइस डि-आवंटन) :**
   - जब कोई प्रक्रिया डिवाइस का उपयोग समाप्त कर लेती है, तो OS उस डिवाइस को रिलीज़ कर देता है, ताकि अन्य प्रक्रियाएँ इसका उपयोग कर सकें।
4. **Device Drivers (डिवाइस ड्राइवर्स) :**
   - OS डिवाइस ड्राइवर्स का उपयोग करके हार्डवेयर और सॉफ़्टवेयर के बीच संचार करता है।
   - यह ड्राइवर्स हार्डवेयर डिवाइस की विशिष्टता को छिपाते हैं और उपयोगकर्ता के लिए आसान इंटरफ़ेस प्रदान करते हैं।
5. **Buffering (बफरिंग) :**
   - डेटा को अस्थायी रूप से मेमोरी में संग्रहित किया जाता है (बफर में) ताकि I/O डिवाइस और CPU के बीच डेटा ट्रांसफर सुचारू रूप से हो।
   - यह गति के अंतर को संतुलित करता है।
6. **Spooling (स्पूलिंग) :**
   - एक डिवाइस को कई प्रक्रियाओं के बीच साझा करने के लिए स्पूलिंग का उपयोग किया जाता है।
   - उदाहरण: एक ही समय में कई प्रिंट जॉब्स को प्रिंटर कतार में डालना।
7. **Error Handling (त्रुटि प्रबंधन) :**
   - अगर कोई डिवाइस फेल होता है या डेटा ट्रांसफर में समस्या होती है, तो OS त्रुटियों का पता लगाता है और उन्हें ठीक करने की कोशिश करता है।
8. **I/O Scheduling (आई/ओ शेड्यूलिंग) :**
   - OS I/O डिवाइस के लिए प्रक्रियाओं की कतार बनाता है और उनका शेड्यूलिंग करता है।
   - यह सुनिश्चित करता है कि डिवाइस का उपयोग कुशल और प्रभावी ढंग से हो।
9. **Access Control (पहुंच नियंत्रण) :**
   - OS यह सुनिश्चित करता है कि केवल अधिकृत प्रक्रियाएँ ही डिवाइस का उपयोग कर सकें।
   - यह सुरक्षा और डेटा इंटिग्रिटी बनाए रखने में मदद करता है।

Device Management के लाभ:

1. सिस्टम और डिवाइस के बीच कुशल संचार।
2. संसाधनों का इष्टतम उपयोग।
3. कई प्रक्रियाओं के बीच हार्डवेयर का समुचित साझाकरण।
4. हार्डवेयर की त्रुटियों का बेहतर प्रबंधन।
5. उपयोगकर्ता और प्रक्रियाओं को हार्डवेयर के जटिल विवरण से मुक्त रखना।

12.Explain the disk scheduling algorithm?

Ans:- Disk Scheduling Algorithm क्या है?

Disk Scheduling Algorithm हार्ड डिस्क पर डेटा पढ़ने या लिखने के लिए I/O अनुरोधों (Requests) को अनुक्रमित करने की प्रक्रिया है।
डिस्क हेड (Disk Head) को ट्रैक्स पर स्थानांतरित करने में समय लगता है, जिसे **Seek Time** कहते हैं।
Disk Scheduling Algorithm का मुख्य उद्देश्य है:

1. **Seek Time** को कम करना।
2. **Disk Throughput** को अधिकतम करना।
3. **Latency** को कम करना।

Disk Scheduling Algorithms के प्रकार:

1. **First-Come, First-Served (FCFS):**
   - **विवरण:**
     अनुरोधों को उसी क्रम में पूरा किया जाता है, जैसे वे आते हैं।
   - **लाभ:**
     - सरल और आसान।
     - किसी भी प्रक्रिया के साथ भेदभाव नहीं।
   - **नुकसान:**
     - Seek Time अधिक हो सकता है।
     - डिस्क हेड अनावश्यक रूप से लंबी दूरी तय कर सकता है।

---

2. **Shortest Seek Time First (SSTF):**
   - **विवरण:**
     डिस्क हेड उस अनुरोध को पूरा करता है, जो सबसे कम Seek Time (नजदीकी ट्रैक) पर होता है।
   - **लाभ:**
     - Seek Time को कम करता है।
   - **नुकसान:**
     - Starvation की समस्या हो सकती है, क्योंकि दूर के अनुरोध लंबे समय तक इंतजार कर सकते हैं।

---

3. **SCAN (Elevator Algorithm):**
   - **विवरण:**
     डिस्क हेड एक दिशा में लगातार ट्रैक्स को स्कैन करता है और अनुरोधों को पूरा करता है। जब अंतिम ट्रैक तक पहुँचता है, तो दिशा बदलता है।
   - **लाभ:**
     - Starvation की समस्या कम होती है।
     - समान वितरण।
   - **नुकसान:**
     - Seek Time कभी-कभी अधिक हो सकता है।

4. **C-SCAN (Circular SCAN):**
   - **विवरण:**
     SCAN Algorithm का एक संस्करण। डिस्क हेड केवल एक दिशा में ट्रैक्स को स्कैन करता है, और अंतिम ट्रैक तक पहुँचने के बाद शुरुआत में वापस आ जाता है।
   - **लाभ:**
     - समान प्रतीक्षा समय।
     - Starvation को रोका जाता है।
   - **नुकसान:**
     - कुछ अतिरिक्त समय लग सकता है, क्योंकि डिस्क हेड को शुरू में वापस आना होता है।

---

5. **LOOK Algorithm:**
   - **विवरण:**
     यह SCAN Algorithm के समान है, लेकिन डिस्क हेड केवल उन ट्रैक्स तक जाता है, जहाँ अनुरोध होते हैं, अंतिम ट्रैक तक नहीं जाता।
   - **लाभ:**
     - अनावश्यक यात्रा को कम करता है।
   - **नुकसान:**
     - SCAN की तरह ही प्रदर्शन करता है।

---

6. **C-LOOK Algorithm:**
   - **विवरण:**
     यह C-SCAN का संस्करण है। डिस्क हेड केवल अनुरोधों के बीच चलता है और अंतिम ट्रैक तक जाने के बजाय पहले अनुरोध पर लौट आता है।
   - **लाभ:**
     - C-SCAN की तुलना में कम Seek Time।
   - **नुकसान:**
     - कुछ जटिलता।

12.What is file? How many types of file in Operating System?

Ans:- फ़ाइल (File) :-

फ़ाइल डेटा को स्थायी रूप से स्टोर करने का एक संग्रह है, जिसे ऑपरेटिंग सिस्टम द्वारा व्यवस्थित और प्रबंधित किया जाता है। फ़ाइलें उपयोगकर्ता द्वारा या सिस्टम द्वारा बनाई जाती हैं, और इनमें डेटा, जानकारी, या प्रोग्राम कोड हो सकता है।

ऑपरेटिंग सिस्टम फ़ाइल को नाम और स्थान के माध्यम से पहचानता है।

फ़ाइल के प्रकार (Types of File in Operating System):

ऑपरेटिंग सिस्टम में फ़ाइलों को उनके उपयोग और सामग्री के आधार पर विभिन्न श्रेणियों में विभाजित किया गया है। मुख्य प्रकार निम्नलिखित हैं:

---

1. Text File (पाठ फ़ाइल) :

- विवरण:
  - **यह फ़ाइल अक्षरों, अंकों, और अन्य वर्णों का एक क्रम होती है।**
  - **इसमें साधारण टेक्स्ट होता है जिसे उपयोगकर्ता पढ़ सकता है।**
- उदाहरण:
  - `.txt`, `.csv`, `.log`।
- उपयोग:
  - **नोट्स लिखने, लॉग फाइल, डेटा संग्रह।**

---

2. Binary File (बाइनरी फ़ाइल) :

- विवरण:
  - **इसमें डेटा बाइनरी प्रारूप (0 और 1) में संग्रहीत होता है।**
  - **इसे सीधे पढ़ा नहीं जा सकता, इसे विशेष प्रोग्राम के माध्यम से पढ़ा जाता है।**
- उदाहरण:
  - `.exe`, `.dll`, `.bin`।
- उपयोग:
  - **सॉफ़्टवेयर निष्पादन योग्य फ़ाइलें।**

---

3. Program File (प्रोग्राम फ़ाइल) :

- विवरण:
  - **इन फ़ाइलों में कोड या निर्देश होते हैं जिन्हें कंप्यूटर निष्पादित कर सकता है।**
- उदाहरण:
  - `.exe`, `.class`, `.py`, `.sh`।
- उपयोग:
  - **प्रोग्रामिंग और सॉफ्टवेयर डेवलपमेंट।**

---

4. Multimedia File (मल्टीमीडिया फ़ाइल) :

- विवरण:
  - **इन फ़ाइलों में चित्र, वीडियो, और ऑडियो जैसे मल्टीमीडिया डेटा होता है।**
- उदाहरण:
  - **चित्र**: `.jpg`, `.png`
  - **वीडियो**: `.mp4`, `.mkv`
  - **ऑडियो**: `.mp3`, `.wav`।
- उपयोग:
  - **मनोरंजन और सूचना प्रसार।**

---

5. Database File (डेटाबेस फ़ाइल) :

- विवरण:
  - **इन फ़ाइलों में संरचित डेटा होता है जिसे प्रोग्राम द्वारा व्यवस्थित और एक्सेस किया जा सकता है।**
- उदाहरण:
  - `.db`, `.mdb`, `.sql`।
- उपयोग:
  - **डेटा स्टोरेज और प्रबंधन।**

---

6. Directory File (डायरेक्टरी फ़ाइल) :

- विवरण:
  - **यह फ़ाइल अन्य फ़ाइलों और डायरेक्टरीज़ का एक संग्रह होती है।**
  - **इसे Folder भी कहा जाता है।**
- उदाहरण:
  - **किसी फोल्डर का स्थान।**
- उपयोग:
  - **फ़ाइलों को व्यवस्थित करना।**

---

7. Special File (स्पेशल फ़ाइल) :

- विवरण:
  - **यह फ़ाइलें हार्डवेयर डिवाइस से संबंधित होती हैं।**
  - **इनका उपयोग इनपुट/आउटपुट ऑपरेशंस के लिए होता है।**
- उदाहरण:
  - Block Files (**डिस्क से संबंधित**)
  - Character Files (**कीबोर्ड, माउस से संबंधित**)।
- उपयोग:
  - **हार्डवेयर और सॉफ़्टवेयर के बीच संचार।**

---

8. Temporary File (अस्थायी फ़ाइल) :

- विवरण:
  - **यह अस्थायी डेटा स्टोरेज के लिए उपयोग की जाती है और उपयोग के बाद हटा दी जाती है।**
- उदाहरण:
  - `.tmp`।
- उपयोग:
  - **ऑपरेशन्स के दौरान डेटा कैशिंग।**

14.**Explain the different between logical and physical file system?**

Ans:- Logical और Physical File System में अंतर

Logical File System और Physical File System फाइलों के प्रबंधन के लिए ऑपरेटिंग सिस्टम द्वारा उपयोग किए जाने वाले दो अलग-अलग स्तर हैं। दोनों का उद्देश्य फाइलों को स्टोर, व्यवस्थित, और एक्सेस करना है, लेकिन ये अलग-अलग कार्य करते हैं।

---

1. Logical File System (तार्किक फाइल सिस्टम) :

- **परिभाषा:**
  Logical File System फाइलों की संरचना और उपयोगकर्ता द्वारा उन्हें एक्सेस करने के तरीके को परिभाषित करता है। यह उपयोगकर्ता को फाइल नाम और निर्देशिकाओं के रूप में फाइलों तक पहुंचने की सुविधा देता है।
- **मुख्य कार्य:**
  - फाइलों का नामकरण (Naming)।
  - फाइल संरचना और संगठन।
  - फाइल सुरक्षा और अनुमति प्रबंधन।
  - फाइलों को उपयोगकर्ता से अमूर्त (abstract) रूप में प्रस्तुत करना।
- **उदाहरण:**
  - फाइलों को फोल्डर में व्यवस्थित करना।
  - फ़ाइल नाम, एक्सटेंशन और अनुमति (Permissions) को नियंत्रित करना।
- **उपयोगकर्ता दृष्टिकोण:**
  - उपयोगकर्ता फाइलों को नाम, पथ (Path), और निर्देशिकाओं के माध्यम से एक्सेस करता है।
- **लाभ:**
  - उपयोगकर्ता को फाइल सिस्टम की भौतिक जटिलता से मुक्त रखता है।
  - फाइल सुरक्षा और नियंत्रण के लिए आवश्यक।

---

2. Physical File System (भौतिक फाइल सिस्टम) :

- **परिभाषा:**
  Physical File System हार्ड डिस्क, SSD, या अन्य स्टोरेज डिवाइस पर फाइलों को भौतिक रूप से स्टोर और व्यवस्थित करता है।
- **मुख्य कार्य:**
  - डेटा को डिस्क पर ब्लॉक्स में स्टोर करना।
  - फ्री स्पेस और स्टोरेज का प्रबंधन।
  - डिस्क पर फाइल की लोकेशन को ट्रैक करना।
- **उदाहरण:**
  - हार्ड डिस्क पर फाइलों को क्लस्टर या सेक्टर में स्टोर करना।
  - डेटा को रीड/राइट ऑपरेशन्स के लिए हार्डवेयर तक पहुंचाना।
- **उपयोगकर्ता दृष्टिकोण:**
  - उपयोगकर्ता फाइलों की भौतिक लोकेशन से अवगत नहीं होता।
- **लाभ:**
  - डेटा को कुशलतापूर्वक स्टोर और एक्सेस करने में मदद करता है।
  - स्टोरेज स्पेस का प्रभावी उपयोग।

15.What is system call? Explain the types of system call.

Ans:- System Call :- **System Call** एक ऐसा इंटरफ़ेस है जो उपयोगकर्ता के प्रोग्राम और ऑपरेटिंग सिस्टम के बीच संचार स्थापित करता है। जब कोई प्रोग्राम किसी हार्डवेयर रिसोर्स (जैसे कि मेमोरी, फाइल, या डिवाइस) का उपयोग करना चाहता है, तो वह सीधे हार्डवेयर से

संपर्क नहीं कर सकता। इसके बजाय, वह **System Call** के माध्यम से ऑपरेटिंग सिस्टम से अनुरोध करता है।

System Calls का मुख्य उद्देश्य उपयोगकर्ता और ऑपरेटिंग सिस्टम के बीच एक सुरक्षित और नियंत्रित इंटरफ़ेस प्रदान करना है।

System Call के प्रकार

System Calls को उनके उपयोग और कार्यक्षमता के आधार पर कई श्रेणियों में विभाजित किया गया है:

---

1. Process Control (प्रक्रिया नियंत्रण) :

- विवरण:
  यह System Calls **प्रक्रिया** (Process) **के निर्माण, निष्पादन, और समाप्ति के लिए उपयोग की जाती हैं।**
- उदाहरण:
  - `fork()` - **नई प्रक्रिया बनाने के लिए।**
  - `exec()` - **एक प्रक्रिया को निष्पादित करने के लिए।**
  - `exit()` - **प्रक्रिया समाप्त करने के लिए।**
  - `wait()` - **प्रक्रिया के समाप्त होने का इंतजार करने के लिए।**
- उपयोग:
  - **नई प्रक्रिया बनाना।**
  - **प्रक्रिया को रोकना, पुनः आरंभ करना।**

---

2. File Management (फाइल प्रबंधन) :

- विवरण:
  यह System Calls **फाइलों को पढ़ने, लिखने, खोलने और बंद करने के लिए उपयोग की जाती हैं।**
- उदाहरण:
  - `open()` - **फ़ाइल खोलने के लिए।**
  - `read()` - **फ़ाइल से डेटा पढ़ने के लिए।**
  - `write()` - **फ़ाइल में डेटा लिखने के लिए।**
  - `close()` - **फ़ाइल बंद करने के लिए।**
- उपयोग:
  - **फाइलों पर संचालन** (Operations)**।**
  - **फाइल की अनुमति** (Permissions) **सेट करना।**

---

3. Device Management (डिवाइस प्रबंधन) :

- विवरण:
  यह System Calls **इनपुट/आउटपुट डिवाइस (जैसे कीबोर्ड, प्रिंटर) के साथ संचार करने के लिए उपयोग की जाती हैं।**

- उदाहरण:
  - `ioctl()` - **डिवाइस को नियंत्रित करने के लिए।**
  - `read()` - **डिवाइस से डेटा पढ़ने के लिए।**
  - `write()` - **डिवाइस पर डेटा लिखने के लिए।**
- उपयोग:
  - **हार्डवेयर से डेटा का आदान-प्रदान।**
  - **डिवाइस की स्थिति जांचना।**

---

4. Information Maintenance (जानकारी रखरखाव) :

- विवरण:
  यह System Calls **सिस्टम की जानकारी और सेटिंग्स को प्राप्त और संशोधित करने के लिए उपयोग की जाती हैं।**
- उदाहरण:
  - `getpid()` - **प्रक्रिया का पहचान संख्या** (PID) **प्राप्त करने के लिए।**
  - `getuid()` - **उपयोगकर्ता की पहचान संख्या** (UID) **प्राप्त करने के लिए।**
  - `setuid()` - **उपयोगकर्ता की पहचान संख्या सेट करने के लिए।**
- उपयोग:
  - **प्रक्रिया और उपयोगकर्ता जानकारी।**
  - **सिस्टम सेटिंग्स बदलना।**

---

5. Communication (संचार) :

- विवरण:
  यह System Calls **विभिन्न प्रक्रियाओं के बीच डेटा का आदान-प्रदान करने के लिए उपयोग की जाती हैं।**
- उदाहरण:
  - `pipe()` - **पाइप बनाना।**
  - `send()` - **डेटा भेजने के लिए।**
  - `receive()` - **डेटा प्राप्त करने के लिए।**
  - `shmget()` - **साझा मेमोरी प्राप्त करने के लिए।**
- उपयोग:
  - **इंटर-प्रोसेस कम्युनिकेशन** (IPC)**।**
  - **नेटवर्क पर डेटा ट्रांसफर।**

16.What is Process? Give the difference between a process and a program.

Ans:- Process:- Process एक प्रोग्राम का निष्पादन (Execution) है।
यह एक सक्रिय इकाई है जो प्रोग्राम कोड और इससे संबंधित सभी गतिविधियों (जैसे डेटा और संसाधन) को शामिल करती है।
ऑपरेटिंग सिस्टम के तहत, किसी प्रोग्राम को निष्पादित करने के लिए आवश्यक संसाधन जैसे CPU, मेमोरी, और इनपुट/आउटपुट डिवाइस, एक प्रक्रिया के रूप में प्रदान किए जाते हैं।

Process के प्रमुख तत्व:

1. प्रोग्राम कोड: **वह कोड जिसे निष्पादित किया जा रहा है।**
2. प्रोसेसर स्टेट: CPU **रजिस्टर और अन्य संसाधनों की स्थिति।**
3. संसाधन: **मेमोरी**, **फाइल्स**, **और इनपुट/आउटपुट डिवाइस।**
4. **Process Control Block (PCB): प्रक्रिया की जानकारी जैसे** ID, **स्थिति** (State), **और प्रायरिटी।**

Program और Process के बीच अंतर:

पैरामीटर	**Program** (प्रोग्राम)	**Process** (प्रक्रिया)
परिभाषा	**प्रोग्राम निर्देशों का स्थिर** (Static) **समूह है।**	**प्रक्रिया प्रोग्राम का सक्रिय** (Active) **रूप है।**
स्थिति (**State**)	**यह निष्क्रिय** (Inactive) **है।**	**यह निष्पादन के दौरान सक्रिय** (Active) **होती है।**
भंडारण (**Storage**)	**हार्ड डिस्क या स्टोरेज डिवाइस पर संग्रहीत।**	**मेमोरी** (RAM) **में लोड की जाती है।**
उपयोगकर्ता का दृष्टिकोण	**यह उपयोगकर्ता द्वारा लिखा और देखा जा सकता है।**	**यह ऑपरेटिंग सिस्टम द्वारा प्रबंधित की जाती है।**
लाइफ साइकिल	**एक स्थायी इकाई है**, **जब तक इसे हटाया न जाए।**	**अस्थायी है**, **जब प्रक्रिया समाप्त होती है**, **तो यह नष्ट हो जाती है।**
संसाधन उपयोग	**यह किसी भी सिस्टम संसाधन का उपयोग नहीं करता।**	**यह** CPU, **मेमोरी**, **और अन्य संसाधनों का उपयोग करता है।**
उदाहरण	`.exe`, `.txt`, **या अन्य फाइल।**	**फ़ाइल का निष्पादन**, **जैसे प्रोग्राम चलाना।**

17.What is paging? What is Page Table? Explain the conversion of Virtual Address to Physical Address in Paging with example.

Ans:- **Paging क्या है?**

Paging एक मेमोरी प्रबंधन तकनीक है जिसका उपयोग ऑपरेटिंग सिस्टम में मेमोरी को कुशलतापूर्वक प्रबंधित करने के लिए किया जाता है।
Paging के तहत, मेमोरी को समान आकार के छोटे ब्लॉक्स में विभाजित किया जाता है:

- **Logical Memory** (वर्चुअल मेमोरी) : **इसे Pages** (पृष्ठ) **में विभाजित किया जाता है।**
- **Physical Memory** (भौतिक मेमोरी) : **इसे Frames** (फ्रेम) **में विभाजित किया जाता है।**

हर पेज का आकार एक फ्रेम के आकार के बराबर होता है।

जब एक प्रोग्राम को निष्पादित किया जाता है, तो उसके पेज को फिजिकल मेमोरी के फ्रेम में लोड किया जाता है।

Page Table क्या है?

Page Table एक डेटा संरचना (Data Structure) है जो वर्चुअल एड्रेस (Virtual Address) और फिजिकल एड्रेस (Physical Address) के बीच का मैपिंग (Mapping) स्टोर करता है।

Page Table का मुख्य कार्य है:

- **यह बताना कि वर्चुअल मेमोरी का कौन सा पेज फिजिकल मेमोरी के किस फ्रेम में स्टोर है।**
- **यदि पेज मेमोरी में नहीं है, तो यह डिस्क पर पेज की लोकेशन का पता देता है।**

Page Table के तत्व:

1. **Page Number (**पृष्ठ संख्या**) : वर्चुअल एड्रेस का हिस्सा जो पेज को पहचानता है।**
2. **Frame Number (**फ्रेम संख्या**) : फिजिकल मेमोरी में पेज का स्थान।**
3. **Control Bits: जैसे कि वैलिड/इनवैलिड बिट।**

Virtual Address को Physical Address में बदलने की प्रक्रिया (Paging के साथ):

वर्चुअल एड्रेस का विभाजन:

1. **Page Number (**पृष्ठ संख्या**) : वर्चुअल एड्रेस का वह हिस्सा जो पेज को निर्दिष्ट करता है।**
2. **Page Offset (**पृष्ठ ऑफ़सेट**) : पेज के अंदर किसी डेटा की स्थिति को दर्शाता है।**

प्रक्रिया:

1. **Page Table Lookup (**पेज टेबल खोज**) :**
   - **वर्चुअल एड्रेस के Page Number का उपयोग करके**, Page Table **से संबंधित फ्रेम नंबर प्राप्त करें।**
2. **Frame Number** जोड़ें:
   - **फ्रेम नंबर में Page Offset जोड़कर फिजिकल एड्रेस प्राप्त करें।**

उदाहरण:

जानकारी:

- **वर्चुअल एड्रेस स्पेस:** 16 **पेज (**Page Size = 4 KB**)।**
- **फिजिकल मेमोरी स्पेस:** 8 **फ्रेम।**

वर्चुअल एड्रेस:

- 16 **पेज** = 242^424 **पेज** → Page Number **के लिए** 4 **बिट।**
- Page Offset **के लिए:** 4 KB=2124 \text{ KB} = 2^{12}4 KB=212, **इसलिए** Offset = 12 **बिट।**
- **कुल वर्चुअल एड्रेस** = 4+12=164 + 12 = 164+12=16 **बिट।**

प्रक्रिया:

1. **वर्चुअल एड्रेस: 0001 0100 0000 0000**
   - **Page Number:** `0001` **(पेज** 1**)।**

- **Offset:** `0100 0000 0000`।

2. Page Table **देखें**:
   - Page 1 → Frame 5।
3. **फिजिकल एड्रेस**:
   - Frame 5 = `0101`।
   - Offset = `0100 0000 0000`।
   - **फिजिकल एड्रेस = 0101 0100 0000 0000**।

18.Mention few Page Replacement Strategies

Ans:- **Page Replacement Strategies (पृष्ठ प्रतिस्थापन रणनीतियाँ)**

Page Replacement एक ऐसी तकनीक है जिसका उपयोग ऑपरेटिंग सिस्टम द्वारा किया जाता है जब फिजिकल मेमोरी में एक नई पेज को लोड करने के लिए कोई खाली स्थान नहीं होता है। ऐसी स्थिति में, सिस्टम को यह तय करना होता है कि किस पेज को हटाया जाए ताकि नया पेज मेमोरी में लोड किया जा सके।
यह प्रक्रिया विशेष रूप से **Demand Paging** और **Virtual Memory** सिस्टम में महत्वपूर्ण है।

नीचे कुछ प्रमुख **Page Replacement Strategies** दी जा रही हैं:

---

1. FIFO (First In First Out)

- **विवरण**:
  FIFO पेज प्रतिस्थापन रणनीति सबसे पुरानी पेज को हटाने का काम करती है। इसका मतलब है कि जो पेज सबसे पहले मेमोरी में लोड हुआ था, वह सबसे पहले निकाला जाएगा।
- **कार्यप्रणाली**:
  FIFO में एक पंक्ति (Queue) होती है जिसमें पेजों को लोड करने का क्रम रिकॉर्ड होता है। सबसे पहले जो पेज लोड होता है, उसे पहले बाहर किया जाता है।
- **लाभ**:
  - सरल और लागू करने में आसान।
- **नुकसान**:
  - यह रणनीति कभी-कभी प्रदर्शन में खराब हो सकती है क्योंकि पुराने पेजों को हटा दिया जाता है, भले ही वे अक्सर उपयोग हो रहे हों।

---

2. LRU (Least Recently Used)

- **विवरण**:
  LRU पेज प्रतिस्थापन रणनीति सबसे कम उपयोग किए गए पेज को हटाने का काम करती है। इसमें यह मान लिया जाता है कि यदि कोई पेज हाल ही में उपयोग नहीं किया गया है, तो उसे निकाला जाना चाहिए।
- **कार्यप्रणाली**:
  LRU यह ट्रैक करता है कि किस पेज को अंतिम बार कब उपयोग किया गया था और उसी के आधार पर सबसे पुराने (कम उपयोग किए गए) पेज को हटाता है।
- **लाभ**:
  - यह रणनीति काफी प्रभावी होती है क्योंकि यह यह सुनिश्चित करती है कि वे पेज जिनका उपयोग नहीं हो रहा है, वे हटाए जाते हैं।

- **नुकसान**:
    - इसे लागू करना जटिल हो सकता है और अतिरिक्त मेमोरी और संसाधनों की आवश्यकता होती है।

---

3. Optimal Page Replacement (OPT)

- **विवरण**:
  Optimal Page Replacement रणनीति में वह पेज हटाया जाता है जिसे भविष्य में सबसे देर से उपयोग किया जाएगा। यह रणनीति **थेoperिकली** सर्वोत्तम मानी जाती है, लेकिन वास्तविक दुनिया में इसे लागू करना कठिन होता है क्योंकि भविष्य का पूर्वानुमान करना असंभव है।
- **कार्यप्रणाली**:
  जब मेमोरी में एक नया पेज लोड करने की आवश्यकता होती है, तो यह उस पेज को हटाता है जिसे सबसे लंबी अवधि तक उपयोग नहीं किया जाएगा।
- **लाभ**:
    - यह सबसे आदर्श रणनीति मानी जाती है, जो मेमोरी का सर्वोत्तम उपयोग करती है।
- **नुकसान**:
    - यह रणनीति वास्तविक प्रणाली में लागू करना कठिन है क्योंकि यह भविष्य की गतिविधियों को पूर्वानुमान करने की आवश्यकता होती है।

---

4. Clock (Second Chance) Algorithm

- **विवरण**:
  Clock पेज प्रतिस्थापन रणनीति FIFO की तरह ही काम करती है, लेकिन इसमें पेजों को एक "second chance" दी जाती है। यह एक रिंग बफर जैसा होता है, जहां पेजों को एक घड़ी की सुई की तरह घुमाया जाता है।
- **कार्यप्रणाली**:
  जब एक पेज को बाहर निकालने की आवश्यकता होती है, तो इसे एक चेकिंग बिट (Reference Bit) के माध्यम से देखा जाता है। यदि बिट सेट है (यानी पेज हाल ही में उपयोग किया गया था), तो उसे अगली बार के लिए छोड़ा जाता है। अगर बिट सेट नहीं है, तो पेज को हटा दिया जाता है।
- **लाभ**:
    - FIFO की तुलना में अधिक प्रभावी।
    - सस्ते तरीके से LRU के लाभ प्राप्त करता है।
- **नुकसान**:
    - LRU जितना कुशल नहीं होता, और अधिक जटिल हो सकता है।

---

5. Least Frequently Used (LFU)

- **विवरण**:
  LFU पेज प्रतिस्थापन रणनीति उस पेज को हटाने का काम करती है जिसे सबसे कम बार उपयोग किया गया हो। यह पेज की "प्रायिकता" को ट्रैक करता है कि उसे कितनी बार उपयोग किया गया है।
- **कार्यप्रणाली**:
  LFU पेज के उपयोग की गणना करती है, और उस पेज को हटाती है जिसे सबसे कम बार एक्सेस किया गया हो।
- **लाभ**:
    - यदि पेज का उपयोग बार-बार किया जा रहा है, तो वह मेमोरी में बना रहता है।

- **नुकसान:**
    - इसे लागू करना थोड़ा जटिल हो सकता है, और पेज काउंटर के लिए अतिरिक्त संसाधनों की आवश्यकता होती है।

18.What is meant by Global Replacement and Local Replacement?

Ans:- **Global Replacement और Local Replacement क्या होते हैं?**

Global Replacement और **Local Replacement** पेज प्रतिस्थापन (Page Replacement) रणनीतियाँ हैं, जो ऑपरेटिंग सिस्टम में पृष्ठों (Pages) को लोड करने और हटाने के लिए उपयोग की जाती हैं, खासकर जब फिजिकल मेमोरी में जगह की कमी होती है। इन दोनों रणनीतियों का उद्देश्य यह तय करना है कि जब मेमोरी में नया पेज लोड करने की आवश्यकता हो, तो कौन से पेज को हटाया जाए।

---

1. Global Replacement (वैश्विक प्रतिस्थापन)

Global Replacement में सभी प्रक्रियाओं के पृष्ठों को एक ही पूल के रूप में माना जाता है। जब एक प्रक्रिया को नया पेज लोड करने की आवश्यकता होती है, तो यह किसी भी प्रक्रिया के पृष्ठ को बाहर निकाल सकती है।

कार्यप्रणाली:

- **जब मेमोरी में एक नया पेज लोड करना होता है, तो यह किसी भी प्रक्रिया के पृष्ठ को हटाने के लिए चयन कर सकती है।**
- **यह मेमोरी के सभी पृष्ठों को एक सामान्य सूची में देखता है, और जो पेज कम उपयोग हो रहा है, उसे बाहर निकाल देता है।**

लाभ:

- संसाधन का बेहतर उपयोग: **सभी प्रक्रियाओं के पृष्ठों को एक साथ देखना मेमोरी का अधिकतम उपयोग कर सकता है।**
- लचीलापन: **कोई भी पेज किसी भी प्रक्रिया से हटाया जा सकता है, जिससे लचीलापन बढ़ता है।**

नुकसान:

- प्रोसेस के असंतुलन का खतरा: **एक प्रक्रिया के पृष्ठ को अधिक बार हटाया जा सकता है, जिससे प्रदर्शन असंतुलित हो सकता है।**
- कम प्रभावशीलता: **कभी-कभी यह अन्य प्रक्रियाओं की मेमोरी को अनावश्यक रूप से प्रभावित कर सकता है।**

---

2. Local Replacement (स्थानीय प्रतिस्थापन)

Local Replacement में प्रत्येक प्रक्रिया के लिए एक अलग पेज पूल होता है। जब किसी प्रक्रिया को नया पेज लोड करना होता है, तो केवल उसी प्रक्रिया के पृष्ठों में से एक पेज को हटाया जाता है।

कार्यप्रणाली:

- **प्रत्येक प्रक्रिया का अपना पेज पूल होता है।**
- **जब एक प्रक्रिया को नया पेज लोड करने की आवश्यकता होती है, तो सिर्फ उसी प्रक्रिया के पृष्ठों में से एक पेज को हटाया जाता है, न कि किसी अन्य प्रक्रिया के पृष्ठों को।**

लाभ:

- प्रोसेस का संतुलित संसाधन वितरण: **प्रत्येक प्रक्रिया को मेमोरी में समान जगह मिलती है, जिससे मेमोरी का उपयोग अधिक नियंत्रित रहता है।**
- स्थिरता: **एक प्रक्रिया का पृष्ठ प्रतिस्थापन दूसरी प्रक्रिया को प्रभावित नहीं करता, जिससे सिस्टम में अधिक स्थिरता रहती है।**

नुकसान:

- संसाधनों का खराब उपयोग: **कभी-कभी, एक प्रक्रिया अधिक मेमोरी का उपयोग करती है जबकि दूसरी कम, जिससे मेमोरी का उचित उपयोग नहीं हो पाता।**
- कम लचीलापन: **चूंकि केवल उस प्रक्रिया के पृष्ठों को हटाया जा सकता है, अन्य पृष्ठों की सहायता नहीं ली जा सकती।**

Global Replacement और Local Replacement के बीच अंतर:

पैरामीटर	Global Replacement	Local Replacement
पृष्ठ पूल	**सभी प्रक्रियाओं के पृष्ठों को एक साथ देखा जाता है।**	**प्रत्येक प्रक्रिया का अपना पृष्ठ पूल होता है।**
पृष्ठ प्रतिस्थापन	**किसी भी प्रक्रिया के पृष्ठ को हटाया जा सकता है।**	**केवल उस प्रक्रिया के पृष्ठों को हटाया जाता है।**
लचीलापन	**अधिक लचीलापन होता है, क्योंकि कोई भी पृष्ठ हटाया जा सकता है।**	**कम लचीलापन, क्योंकि केवल उसी प्रक्रिया के पृष्ठों को हटाया जा सकता है।**
प्रोसेस के संसाधन	**यह एक प्रक्रिया को अधिक या कम संसाधन दे सकता है।**	**प्रत्येक प्रक्रिया को समान संसाधन मिलता है।**
प्रदर्शन	**कभी-कभी प्रदर्शन असंतुलित हो सकता है।**	**प्रदर्शन अधिक स्थिर होता है, लेकिन कुशल नहीं हो सकता।**

निष्कर्ष:

Global Replacement अधिक लचीला होता है, लेकिन यह कभी-कभी प्रदर्शन असंतुलन का कारण बन सकता है।
Local Replacement अधिक संतुलित होता है, लेकिन यह सभी प्रक्रियाओं को समान संसाधन देने के कारण कुशल नहीं हो सकता है।
इन दोनों रणनीतियों का उद्देश्य मेमोरी के पृष्ठों का कुशल प्रबंधन करना है, लेकिन इनके कार्यान्वयन के तरीके और परिणाम अलग होते हैं।