DSG Support Multi Solution

Unit – 1

History Of Operating System

ऑपरेटिंग सिस्टम एक प्रकार का सॉफ्टवेयर है जो उपयोगकर्ता और हार्डवेयर के बीच इंटरफेस के रूप में कार्य करता है। यह कंप्यूटर के विभिन्न महत्वपूर्ण कार्यों को संभालने और संसाधनों का बहुत कुशलता से उपयोग करने के लिए जिम्मेदार है, इसलिए ऑपरेटिंग सिस्टम को संसाधन प्रबंधक के रूप में भी जाना जाता है। ऑपरेटिंग सिस्टम एक सरकार की तरह भी काम करता है क्योंकि जिस तरह सरकार के पास हर चीज पर अधिकार होता है, उसी तरह ऑपरेटिंग सिस्टम के पास सभी संसाधनों पर अधिकार होता है। OS द्वारा संभाले जाने वाले विभिन्न कार्य हैं फ़ाइल प्रबंधन, कार्य प्रबंधन, कचरा प्रबंधन, मेमोरी प्रबंधन, प्रक्रिया प्रबंधन, डिस्क प्रबंधन, I/O प्रबंधन, परिधीय प्रबंधन, आदि।

ऑपरेटिंग सिस्टम का कार्य

- स्मृति प्रबंधन
- प्रक्रिया प्रबंधन
- फ़ाइल प्रबंधन
- डिवाइस प्रबंधन
- गतिरोध निवारण
- इनपुट/आउटपुट डिवाइस प्रबंधन
- संसाधन प्रबंधक के रूप में कार्य करें
- समय प्रबंधन

ऑपरेटिंग सिस्टम की पीढ़ी

नीचे ऑपरेटिंग सिस्टम की चार पीढ़ियाँ दी गई हैं।

- पहली पीढ़ी
- दूसरी पीढ़ी
- तीसरी पीढ़ी
- चौथी पीढ़ी

1. पहली पीढ़ी (1940 से 1950 के दशक के प्रारंभ तक)

1940 में, पहले इलेक्ट्रिकल कंप्यूटर के निर्माण में ऑपरेटिंग सिस्टम को शामिल नहीं किया गया था। शुरुआती कंप्यूटर उपयोगकर्ताओं के पास डिवाइस पर पूरा नियंत्रण था और वे हर कार्य के लिए शुद्ध मशीन भाषा में प्रोग्राम लिखते थे । कंप्यूटर पीढ़ी के दौरान, एक प्रोग्रामर केवल बुनियादी गणितीय गणनाओं को निष्पादित और हल कर सकता है। इन गणनाओं के लिए ऑपरेटिंग सिस्टम की आवश्यकता नहीं होती है।

2. दूसरी पीढ़ी (1955 – 1965)

GMOSIS, पहला ऑपरेटिंग सिस्टम (OS) 1950 के दशक की शुरुआत में विकसित किया गया था। IBM कंप्यूटर के लिए, जनरल मोटर्स ने ऑपरेटिंग सिस्टम बनाया है। क्योंकि यह सभी संबंधित कार्यों को समूहों या बैचों में इकट्ठा करता है और फिर उन्हें सभी को पूरा करने के लिए पंच कार्ड का उपयोग करके ऑपरेटिंग सिस्टम में सबमिट करता है , दूसरी पीढ़ी का ऑपरेटिंग सिस्टम सिंगल-स्ट्रीम बैच प्रोसेसिंग सिस्टम पर बनाया गया था।

3. तीसरी पीढ़ी (1965 – 1980)

क्योंकि यह सभी समान कार्यों को समूहों या बैचों में इकट्ठा करता है और फिर उन्हें मशीन में सभी कार्यों को पूरा करने के लिए पंच कार्ड का उपयोग करके दूसरी पीढ़ी के ऑपरेटिंग सिस्टम में सबमिट करता है, दूसरी पीढ़ी का ऑपरेटिंग सिस्टम एकल स्ट्रीम बैच प्रोसेसिंग सिस्टम पर आधारित था । प्रत्येक कार्य के पूरा होने पर नियंत्रण ऑपरेटिंग सिस्टम को स्थानांतरित कर दिया जाता है, चाहे वह नियमित रूप से हो या अप्रत्याशित रूप से। ऑपरेटिंग सिस्टम प्रत्येक कार्य समाप्त होने के बाद पंच कार्ड पर अगले कार्य को पढ़ने और शुरू करने से पहले साफ करता है। इसके बाद मेनफ्रेम के रूप में जानी जाने वाली बड़ी, पेशेवर रूप से संचालित मशीनें पेश की गईं। ऑपरेटिंग सिस्टम डिजाइनर 1960 के दशक के अंत में एक नया ऑपरेटिंग सिस्टम बनाने में सक्षम थे

ऐसे ऑपरेटिंग सिस्टम बनाने के लिए जो CPU को एक ही समय में कंप्यूटर पर कई काम करके हर समय सक्रिय रखने में सक्षम बनाते हैं, मल्टीप्रोग्रामिंग को पेश किया जाना चाहिए। 1961 में DEC PDP-1 के रिलीज़ होने के साथ, मिनीकंप्यूटर की तीसरी पीढ़ी ने विकास और वृद्धि का एक नया चरण देखा।

4. चौथी पीढ़ी (1980 - वर्तमान दिन)

पर्सनल कंप्यूटर की चौथी पीढ़ी इन पीडीपी का परिणाम है। चौथी पीढ़ी (1980-वर्तमान)पर्सनल कंप्यूटर का विकास ऑपरेटिंग सिस्टम की चौथी पीढ़ी से जुड़ा हुआ है। फिर भी, तीसरी पीढ़ी के मिनी कंप्यूटर और पर्सनल कंप्यूटर में कई समानताएँ हैं। उस समय, मिनी कंप्यूटर पर्सनल कंप्यूटर की तुलना में थोड़े ही ज़्यादा महंगे थे, जो बहुत महंगे थे।

पर्सनल कंप्यूटर के निर्माण में माइक्रोसॉफ्ट और विंडोज ऑपरेटिंग सिस्टम का विकास एक महत्वपूर्ण प्रभाव था। **1975** में, माइक्रोसॉफ्ट ने पहला विंडोज ऑपरेटिंग सिस्टम विकसित किया। बिल गेट्स और पॉल एलन को माइक्रोसॉफ्ट विंडोज ओएस जारी करने के बाद पर्सनल कंप्यूटर को आगे बढ़ाने का विचार आया। नतीजतन, 1981 में MS-DOS जारी किया गया, लेकिन उपयोगकर्ताओं को इसके जटिल आदेशों को समझना बेहद चुनौतीपूर्ण लगा। विंडोज अब सबसे व्यापक रूप से इस्तेमाल किया जाने वाला और लोकप्रिय ऑपरेटिंग सिस्टम है। उसके बाद, विंडोज ने कई ऑपरेटिंग सिस्टम जारी किए, जिनमें विंडोज 95, विंडोज 98, विंडोज एक्सपी और विंडोज 7, सबसे हालिया ऑपरेटिंग सिस्टम शामिल हैं। अधिकांश विंडोज उपयोगकर्ता वर्तमान में विंडोज 10 चला रहे हैं। विंडोज के अलावा एप्पल एक और प्रसिद्ध ऑपरेटिंग सिस्टम है।

ऑपरेटिंग सिस्टम के प्रकार

ऑपरेटिंग सिस्टम पिछले कुछ सालों में काफी विकसित हुआ है। अपने मूल स्वरूप को पाने से पहले इसमें कई बदलाव हुए। ऑपरेटिंग सिस्टम में होने वाले इन बदलावों को ऑपरेटिंग **सिस्टम का विकास** कहा जाता है । OS नई तकनीक

के आविष्कार के साथ खुद को बेहतर बनाता है। मूल रूप से, OS ने नई तकनीक की विशेषता को जोड़ा और खुद को और अधिक शक्तिशाली बनाया। आइए ऑपरेटिंग सिस्टम के विकास को वर्षवार विस्तार से देखें:

- **कोई ऑपरेटिंग सिस्टम नहीं – (0s से 1940s तक)**
- **बैच प्रोसेसिंग सिस्टम -(1940 से 1950 के दशक तक)**
- **मल्टीप्रोग्रामिंग सिस्टम -(1950 से 1960 के दशक)**
- **टाइम-शेयरिंग सिस्टम -(1960 से 1970 के दशक)**
- **GUI का परिचय - (1970 से 1980 के दशक)**
- **नेटवर्क प्रणालियाँ – (1980 से 1990 के दशक तक)**
- **मोबाइल ऑपरेटिंग सिस्टम – (1990 के दशक के अंत से 2000 के दशक के प्रारंभ तक)**
- **एआई एकीकरण – (2010 से अब तक)**

1. कोई ऑपरेटिंग सिस्टम नहीं – (0s से 1940s तक)

जैसा कि हम जानते हैं कि 1940 के दशक से पहले, OS का कोई उपयोग नहीं था। पहले, लोगों के कंप्यूटर सिस्टम में OS की कमी थी, इसलिए उन्हें प्रत्येक कार्य के लिए मशीन भाषा (0-1 आधारित भाषा) में मैन्युअल रूप से निर्देश टाइप करने पड़ते थे। और उस समय, उपयोगकर्ताओं के लिए एक सरल कार्य को भी लागू करना बहुत कठिन था। और यह बहुत समय लेने वाला था और उपयोगकर्ता के अनुकूल भी नहीं था। क्योंकि हर किसी के पास मशीन भाषा को समझने के लिए इतनी समझ नहीं थी और इसके लिए गहरी समझ की आवश्यकता थी।

2. बैच प्रोसेसिंग सिस्टम -(1940 से 1950 के दशक तक)

समय के साथ-साथ बैच प्रोसेसिंग सिस्टम बाजार में आया। अब उपयोगकर्ताओं के पास अपने प्रोग्राम को पंच कार्ड पर लिखने और उसे कंप्यूटर ऑपरेटर को लोड करने की सुविधा थी। और फिर ऑपरेटर समान प्रकार के जॉब के विभिन्न बैच बनाते थे और फिर अलग-अलग बैच (जॉब के समूह) को एक-एक करके CPU को सौंपते थे। CPU पहले एक बैच के जॉब को निष्पादित करता है और फिर दूसरे बैच के जॉब को क्रमबद्ध तरीके से आगे बढ़ाता है।

3. मल्टीप्रोग्रामिंग सिस्टम -(1950 से 1960 के दशक)

मल्टीप्रोग्रामिंग पहला ऑपरेटिंग सिस्टम था जहाँ वास्तविक क्रांति शुरू हुई। यह उपयोगकर्ता को मेमोरी में कई प्रोग्राम लोड करने की सुविधा प्रदान करता है और प्रत्येक प्रोग्राम को मेमोरी का एक विशिष्ट भाग प्रदान करता है। जब कोई प्रोग्राम किसी I/O ऑपरेशन (जिसमें बहुत समय लगता है) के लिए प्रतीक्षा कर रहा होता है, तो उस समय OS CPU को पिछले प्रोग्राम से दूसरे प्रोग्राम (जो तैयार कतार में सबसे पहले होता है) पर स्विच करने की अनुमति देता है ताकि प्रोग्राम को लगातार बाधित किए बिना निष्पादित किया जा सके ।

4. टाइम-शेयरिंग सिस्टम -(1960 से 1970 के दशक)

टाइम-शेयरिंग सिस्टम मल्टीप्रोग्रामिंग सिस्टम का विस्तारित संस्करण है। इसमें एक अतिरिक्त सुविधा जोड़ी गई है ताकि किसी एक प्रोग्राम द्वारा लंबे समय तक CPU का उपयोग न किया जा सके और एक निश्चित समय अंतराल के बाद हर प्रोग्राम को CPU का एक्सेस दिया जा सके। मूल रूप से OS एक निश्चित समय अंतराल के बाद एक प्रोग्राम से दूसरे प्रोग्राम पर स्विच करता है ताकि हर प्रोग्राम CPU का एक्सेस पा सके और अपना काम पूरा कर सके।

5. GUI का परिचय - (1970 से 1980 के दशक)

समय के साथ ग्राफिकल यूजर इंटरफेस (GUI) आए। पहली बार OS अधिक उपयोगकर्ता-अनुकूल बन गया और लोगों के कंप्यूटर से बातचीत करने के तरीके को बदल दिया। GUI कंप्यूटर सिस्टम को विज़ुअल एलिमेंट प्रदान करता है, जिससे कंप्यूटर के साथ उपयोगकर्ता की बातचीत अधिक आरामदायक और उपयोगकर्ता-अनुकूल हो जाती है। उपयोगकर्ता कमांड टाइप करने के बजाय केवल विज़ुअल एलिमेंट पर क्लिक कर सकता है। यहाँ Microsoft के विंडोज़ आइकन, मेनू और विंडोज़ में GUI की कुछ विशेषताएँ दी गई हैं।

6. नेटवर्क सिस्टम – (1980 से 1990 के दशक)

1980 के दशक में कंप्यूटर नेटवर्क का क्रेज अपने चरम पर था। नेटवर्क संचार को प्रबंधित करने के लिए एक विशेष प्रकार के ऑपरेटिंग सिस्टम की आवश्यकता थी। नेटवर्क संचार को प्रबंधित करने के लिए नोवेल नेटवेयर और विंडोज एनटी जैसे ऑपरेटिंग सिस्टम विकसित किए गए थे जो उपयोगकर्ताओं को सहयोगी वातावरण में काम करने की सुविधा प्रदान करते थे और फ़ाइल शेयरिंग और रिमोट एक्सेस को बहुत आसान बनाते थे।

7. मोबाइल ऑपरेटिंग सिस्टम – (1990 के दशक के अंत से 2000 के दशक के प्रारंभ तक)

स्मार्टफोन के आविष्कार ने सॉफ्टवेयर उद्योग में एक बड़ी क्रांति ला दी है। स्मार्टफोन के संचालन को संभालने के लिए एक विशेष प्रकार के ऑपरेटिंग सिस्टम विकसित किए गए थे। उनमें से कुछ हैं: iOS और Android आदि। इन ऑपरेटिंग सिस्टम को समय के साथ अनुकूलित किया गया और अधिक शक्तिशाली बनाया गया।

8. एआई एकीकरण – (2010 से अब तक)

समय के साथ आर्टिफिशियल इंटेलिजेंस का चलन बढ़ा। ऑपरेटिंग सिस्टम में सिरी, गूगल असिस्टेंट और एलेक्सा जैसी AI तकनीक की खूबियाँ शामिल हैं और यह कई मायनों में ज़्यादा शक्तिशाली और कुशल बन गया है। ऑपरेटिंग सिस्टम के साथ ये AI खूबियाँ वॉयस कमांड, प्रेडिक्टिव टेक्स्ट और पर्सनलाइज्ड सिफ़ारिशें जैसी पूरी नई सुविधाएँ बनाती हैं।

नोट: उपर्युक्त OS मूल रूप से यह बताता है कि समय के साथ OS में किस प्रकार नए फीचर्स जुड़ते गए, लेकिन इसका मतलब यह नहीं है कि केवल नई पीढ़ी के OS ही उपयोग में हैं और पहले के OS सिस्टम उपयोग में नहीं हैं, आवश्यकतानुसार ये सभी OS अभी भी सॉफ्टवेयर उद्योग में उपयोग किए जाते हैं।

ऑपरेटिंग सिस्टम के लाभ

- ऑपरेटिंग सिस्टम बाह्य और आंतरिक उपकरणों का प्रबंधन करता है, उदाहरण के लिए, प्रिंटर, स्कैनर और अन्य।
- ऑपरेटिंग सिस्टम, सिस्टम और हार्डवेयर उपकरणों के बीच उचित संचार के लिए इंटरफेस और ड्राइवर प्रदान करता है।
- एकाधिक अनुप्रयोगों को एक साथ चलाने की अनुमति देता है।
- प्रक्रियाओं के निष्पादन का प्रबंधन करता है, यह सुनिश्चित करता है कि सिस्टम उत्तरदायी बना रहे।
- भंडारण डिवाइस पर फ़ाइलों को व्यवस्थित और प्रबंधित करता है।
- ऑपरेटिंग सिस्टम विभिन्न अनुप्रयोगों को संसाधन आवंटित करता है और उनका कुशल उपयोग सुनिश्चित करता है।

ऑपरेटिंग सिस्टम के नुकसान

- यदि आपके सिस्टम में कोई त्रुटि हुई है, तो संभावना है कि आपका डेटा नष्ट हो सकता है, इसलिए हमेशा अपने डेटा का बैकअप रखें।
- खतरे और वायरस किसी भी समय हमारे ऑपरेटिंग सिस्टम पर हमला कर सकते हैं, जिससे ऑपरेटिंग सिस्टम के लिए सिस्टम को इन खतरों से सुरक्षित रखना चुनौतीपूर्ण हो जाता है।
- नए ऑपरेटिंग सिस्टम के बारे में सीखना समय लेने वाला और चुनौतीपूर्ण हो सकता है, खासकर उन लोगों के लिए जो विशेष ऑपरेटिंग सिस्टम का उपयोग कर रहे हैं उदाहरण के लिए विंडोज ओएस से लिनक्स पर स्विच करना मुश्किल है।
- ऑपरेटिंग सिस्टम को अद्यतन रखने के लिए नियमित रखरखाव की आवश्यकता होती है, जो समय लेने वाला हो सकता है।
- ऑपरेटिंग सिस्टम CPU, मेमोरी और स्टोरेज सहित सिस्टम संसाधनों का उपभोग करते हैं, जो अन्य अनुप्रयोगों के प्रदर्शन को प्रभावित कर सकते हैं।

System Call

सिस्टम कॉल एक ऐसी विधि है जिसके द्वारा कंप्यूटर प्रोग्राम उस ऑपरेटिंग सिस्टम के कर्नेल से सेवा का अनुरोध करता है जिस पर वह चल रहा है। सिस्टम कॉल प्रोग्राम के माध्यम से ऑपरेटिंग सिस्टम के साथ बातचीत करने का एक तरीका है। सिस्टम कॉल कंप्यूटर सॉफ़्टवेयर से ऑपरेटिंग सिस्टम के कर्नेल के लिए एक अनुरोध है।

एप्लीकेशन **प्रोग्राम इंटरफ़ेस (API)** ऑपरेटिंग सिस्टम के कार्यों को उपयोगकर्ता प्रोग्राम से जोड़ता है। यह ऑपरेटिंग सिस्टम और प्रक्रिया के बीच एक कड़ी के रूप में कार्य करता है, जिससे उपयोगकर्ता-स्तरीय प्रोग्राम ऑपरेटिंग सिस्टम सेवाओं का अनुरोध कर सकते हैं। कर्नेल सिस्टम तक केवल सिस्टम कॉल का उपयोग करके ही पहुँचा जा सकता है। संसाधनों का उपयोग करने वाले किसी भी प्रोग्राम के लिए सिस्टम कॉल की आवश्यकता होती है।

उपयोगकर्ता बनाम कर्नेल मोड: उपयोगकर्ता मोड में, प्रोग्राम सीधे हार्डवेयर संसाधनों तक नहीं पहुंच सकते हैं; वे कर्नेल मोड में संचालन के लिए सिस्टम कॉल के माध्यम से अनुरोध करते हैं।

सिस्टम कॉल सिस्टम की सुरक्षा को उसके एक्सेसिबिलिटी संसाधनों के माध्यम से बनाए रखने के लिए है, जिसे एक ओएस प्राप्त कर सकता है। ऐसा इसलिए है क्योंकि सिस्टम कॉल मुख्य रूप से सीधे हार्डवेयर नियंत्रण को ओएस तक सीमित रखते हैं।

Why do you need system calls in Operating System?

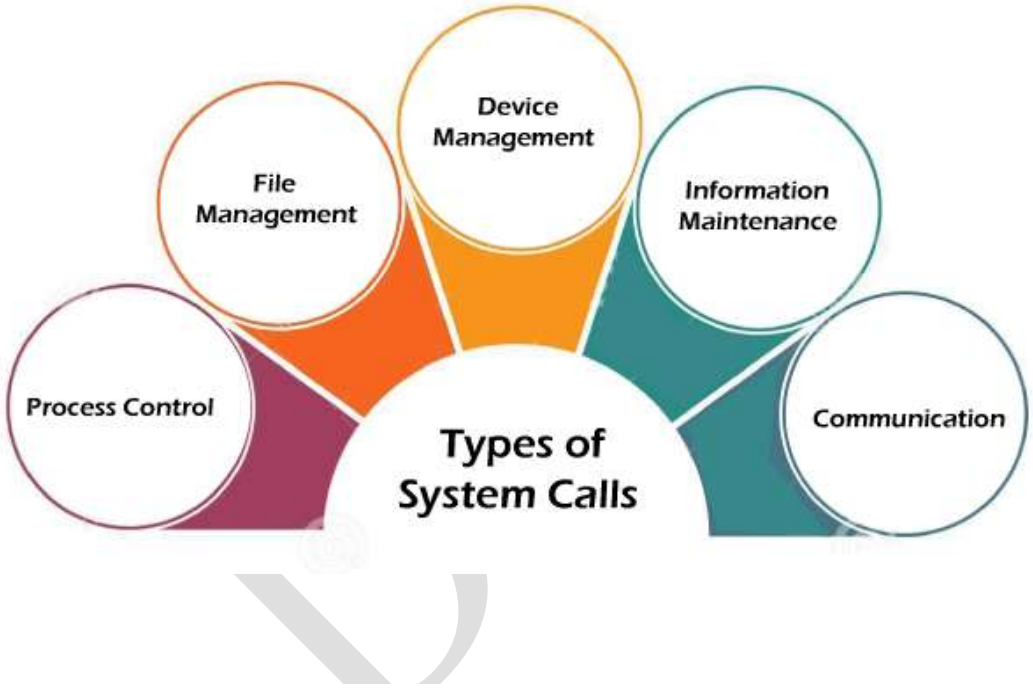
ऐसी कई स्थितियाँ हैं जहाँ आपको ऑपरेटिंग सिस्टम में सिस्टम कॉल की आवश्यकता होती है। निम्नलिखित स्थितियाँ इस प्रकार हैं:

1. जब फ़ाइल सिस्टम किसी फ़ाइल को बनाना या हटाना चाहता है तो इसकी आवश्यकता होती है।
2. नेटवर्क कनेक्शन को डेटा पैकेट भेजने और प्राप्त करने के लिए सिस्टम कॉल की आवश्यकता होती है।
3. यदि आप कोई फ़ाइल पढ़ना या लिखना चाहते हैं, तो आपको सिस्टम कॉल की आवश्यकता होगी।

4. यदि आप प्रिंटर, स्कैनर सहित हार्डवेयर उपकरणों तक पहुंचना चाहते हैं, तो आपको सिस्टम कॉल की आवश्यकता होगी।
5. सिस्टम कॉल का उपयोग नई प्रक्रियाओं को बनाने और प्रबंधित करने के लिए किया जाता है।
6. वे सिस्टम प्रोग्रामों को हार्डवेयर उपकरणों पर नियंत्रण रखने की अनुमति देते हैं - उदाहरण के लिए, पैरामीटर सेट करना या स्थिति पढ़ना।
7. वे सिस्टम संसाधनों की सुरक्षा के लिए पहुँच नियंत्रण और अनुमतियाँ लागू करते हैं।
8. अंतर-प्रक्रिया संचार और समन्वय के लिए उपयोग किया जाता है।
9. सिस्टम कॉल सिस्टम के बारे में स्थिति जानकारी और कॉन्फ़िगरेशन डेटा तक पहुंचने के लिए तंत्र प्रदान करते हैं।

सिस्टम कॉल के प्रकार

सामान्यतः सिस्टम कॉल के पाँच प्रकार होते हैं। ये इस प्रकार हैं:



1. **Process Control**
2. **File Management**
3. **Device Management**
4. **Information Maintenance**
5. **Communication**

1. प्रक्रिया नियंत्रण

प्रक्रिया नियंत्रण वह सिस्टम कॉल है जिसका उपयोग प्रक्रियाओं को निर्देशित करने के लिए किया जाता है। प्रक्रिया नियंत्रण के कुछ उदाहरणों में बनाना, लोड करना, निरस्त करना, समाप्त करना, निष्पादित करना, प्रक्रिया करना, प्रक्रिया समाप्त करना आदि शामिल हैं।

- **प्रक्रिया निर्माण:** जब भी आप कोई नया एप्लिकेशन शुरू करते हैं, तो ऑपरेटिंग सिस्टम उसे सक्रिय करता है और उसे एक प्रक्रिया सौंपता है। फिर यह उसे एक अद्वितीय प्रक्रिया आईडी प्रदान करता है और उसे सभी आवश्यक संसाधन, जैसे मेमोरी, देता है, जो उसे अपना स्टार्ट-अप शुरू करने में सक्षम करेगा।

- **प्रक्रियाओं का शेड्यूलिंग:** ऑपरेटिंग सिस्टम को यह तय करना होता है कि CPU के साथ किस प्रक्रिया को अपनी बारी मिलेगी। यह शेड्यूलिंग एल्गोरिदम के साथ ऐसा करता है - जैसे यह सुनिश्चित करना कि चेकआउट पर लाइन में लगे हर व्यक्ति को अपनी बारी मिले। यह बिना इंतज़ार किए काम को आगे बढ़ाता है, चाहे वह पहले आओ-पहले पाओ का तरीका हो या विशिष्ट कार्यों को प्राथमिकता देना।
- **प्रक्रिया निष्पादन:** प्रक्रिया के जन्म के बाद, प्रोग्राम को चलाने का समय आता है। OS प्रक्रियाओं के बीच अदला-बदली करता है, जिससे यह सुनिश्चित होता है कि आपका कंप्यूटर मल्टीटास्किंग कर रहा है। कल्पना कीजिए कि बाजीगर गेंदें उछाल रहे हैं: हर बार जब कोई गेंद फर्श पर गिरती है, तो दूसरी गेंद हवा में ही वापस आ जाती है। इसी तरह CPU सभी तरह की प्रक्रियाओं का प्रबंधन करता है!
- **निलंबन और पुनः आरंभ:** अन्य समय में, संसाधन मुक्त करने या अन्य अधिक महत्वपूर्ण प्रक्रियाओं को चलने देने के लिए प्रक्रियाओं को निलंबित कर दिया जाता है। सिस्टम अपनी स्थिति को सहेजता है ताकि पुनः आरंभ होने पर यह वहीं से शुरू हो सके जहाँ से इसे छोड़ा गया था, ठीक वैसे ही जैसे वीडियो गेम में पॉज़ बटन दबाना।
- **प्रक्रिया समाप्ति:** हर प्रक्रिया को अंततः समाप्त होने की आवश्यकता होती है-या तो यह अपना काम पूरा कर लेती है या कोई त्रुटि हो जाती है। OS के मामले में, यह बस उस संसाधन को मेमोरी से हटा देता है और उससे छुटकारा पा लेता है, जिससे सब कुछ साफ-सुथरा और स्वच्छ हो जाता है।
- **अंतर-प्रक्रिया संचार (आईपीसी):** प्रक्रियाओं को कभी-कभी अन्य प्रक्रियाओं के साथ संवाद करने की आवश्यकता होती है। इसे प्राप्त करने के लिए, ओएस में संदेश पासिंग या साझा मेमोरी जैसी सुविधाएं होती हैं। दो दोस्तों के बारे में सोचें जो एक दूसरे को संदेश भेज रहे हैं; यही वह तरीका है जिससे प्रक्रियाएं संवाद करती हैं।
- **सिंक्रोनाइजेशन:** कभी-कभी, प्रक्रियाएं संसाधनों को साझा करना चाहती हैं - उदाहरण के लिए फ़ाइलें या मेमोरी। हालाँकि, अगर वे सभी एक साथ किसी संसाधन तक पहुँचना चाहते हैं, तो चीजें कुछ हद तक उलझ सकती हैं। इसलिए, इस तरह की अराजकता को रोकने के लिए, ओएस सेमाफोर और म्यूटेक्स जैसे उपकरणों का उपयोग करता है, जो सब कुछ सुचारू रूप से संचालित करने के लिए जिम्मेदार होते हैं ताकि कोई ऐसा संघर्ष उत्पन्न न हो।
- **डेडलॉक हैंडलिंग:** डेडलॉक वह स्थिति है जब प्रक्रियाएँ डेडलॉक हो जाती हैं, एक दूसरे के संसाधनों की प्रतीक्षा करती हैं, लोग एक दूसरे के लिए दरवाज़े खुले रखते हैं, और कोई भी आगे नहीं बढ़ पाता। OS के पास ऐसी स्थितियों से बचने या उन्हें हल करने की रणनीतियाँ हैं, जिससे यह सुनिश्चित होता है कि कोई भी प्रक्रिया अनिश्चित समय के लिए डेडलॉक न हो।

2. फ़ाइल प्रबंधन

तो, **फ़ाइल प्रबंधन** मूल रूप से यह दर्शाता है कि आपका कंप्यूटर फ़ाइलों के संगठन और हेरफेर को कैसे संभालता है। यह आपके भौतिक दस्तावेज़ों को दराज में संग्रहीत करने जैसा है: प्रत्येक फ़ाइल को बनाना, खोलना, संपादित करना और शायद कभी-कभी ट्रैश में भी डालना होता है। ओएस पर्दे के पीछे सब कुछ संभालता है।

ठीक है, चलिए इसे थोड़ा विस्तार से समझते हैं:

- **फ़ाइलें बनाना:** जब भी आप कोई नया दस्तावेज़ या प्रोग्राम सेव करते हैं, तो आपका कंप्यूटर एक फ़ाइल बनाता है। यह उसे हार्ड ड्राइव या अन्य स्टोरेज पर एक स्थान देता है - ठीक उसी तरह जैसे आप किसी कागज़ी फ़ाइल को फ़ोल्डर में रख सकते हैं।
- **फ़ाइलें खोलना:** किसी फ़ाइल को देखने या संपादित करने के लिए, आप उसे खोलते हैं। ऑपरेटिंग सिस्टम उसे पुनः प्राप्त कर लेगा ताकि आप उसे देख सकें या उस पर काम कर सकें।
- **फाइल पढ़ना:** एक बार फाइल खुल जाने के बाद, कोई भी व्यक्ति अंदर क्या है, इसे पढ़ सकता है। चाहे वह टेक्स्ट फाइल हो, इमेज हो या कुछ और, ओएस स्टोरेज से कॉल करके उसे देखेगा और उस पर काम करेगा।

- **फ़ाइलों में लिखना:** जब भी आप कोई परिवर्तन करते हैं - चाहे वह किसी दस्तावेज़ में टाइप करके हो या किसी चित्र में सुधार करके - ऑपरेटिंग सिस्टम उन परिवर्तनों को फ़ाइल में लिख देता है।
- **फ़ाइलें बंद करना:** जब आप इसे पूरा कर लेते हैं, तो फ़ाइल को बंद करने से सब कुछ सुरक्षित हो जाता है और इसे उपयोग में नहीं के रूप में चिह्नित किया जाता है। यह दस्तावेज़ को पढ़ने या संशोधित करने के बाद उसे वापस उसके फ़ोल्डर में रखने जैसा है।
- **फ़ाइल हटाना:** जब भी आपको लगे कि फ़ाइलें आपके सिस्टम में दोबारा नहीं होनी चाहिए, तो आप उन्हें हटा दें। ऑपरेटिंग सिस्टम उन्हें आपके स्टोरेज से हटा देगा, जिससे अन्य चीज़ों के लिए ज़्यादा जगह बन जाएगी।
- **अनुमतियों को नियंत्रित करना:** आपको कभी-कभी यह नियंत्रित करने की आवश्यकता हो सकती है कि कौन कुछ फ़ाइलों तक पहुँच सकता है और उन्हें संपादित कर सकता है। OS आपको प्रत्येक फ़ाइल के लिए अनुमतियाँ सेट करने की अनुमति देता है ताकि आप चुन सकें कि कोई और इसे चला सकता है, बदल सकता है या खोल सकता है।

3. डिवाइस प्रबंधन

डिवाइस प्रबंधन एक सिस्टम कॉल है जिसका उपयोग डिवाइस से निपटने के लिए किया जाता है। डिवाइस प्रबंधन के कुछ उदाहरणों में रीड, डिवाइस, राइट, डिवाइस विशेषताएँ प्राप्त करना, डिवाइस रिलीज़ करना आदि शामिल हैं। डिवाइस प्रबंधन क्या करता है;

- **डिवाइस से पढ़ना:** जब भी कंप्यूटर डिवाइस से डेटा पढ़ना चाहता है, चाहे वह आपकी हार्ड ड्राइव से फ़ाइल पढ़ना हो या कीबोर्ड से इनपुट, ऑपरेटिंग सिस्टम उस अनुरोध को संभालता है। यह सुनिश्चित करता है कि डेटा सही तरीके से पढ़ा जाए और फिर सही जगह पर भेजा जाए।
- **डिवाइस पर लिखना:** जब भी आपको किसी डिवाइस पर डेटा निर्यात करने की आवश्यकता होती है, जैसे कि अपनी रिपोर्ट की फाइल को यूएसबी स्टिक में सहेजना या बस इसे प्रिंट करना, तो ओएस इसका ध्यान रखता है और सुनिश्चित करता है कि सभी डेटा सही तरीके से डिवाइस पर लिखा जाए।
- **डिवाइस विशेषताएँ प्राप्त करना:** OS किसी डिवाइस के बारे में ऐसी जानकारी प्राप्त कर सकता है, जैसे कि उसकी स्थिति या सेटिंग। उदाहरण के लिए, यह जाँच सकता है कि डिस्क पर क्या उपलब्ध है या वायरलेस माउस की बैटरी का स्तर कितना है।
- **संसाधनों का आवंटन:** ऑपरेटिंग सिस्टम यह तय करता है कि किसी डिवाइस को कितने सिस्टम संसाधन, जैसे मेमोरी या प्रोसेसिंग पावर, मिलेंगे, ताकि किसी भी डिवाइस को उसके हिस्से से अधिक संसाधन न मिलें, और पूरी व्यवस्था बिना किसी समस्या के चलती रहे।
- **डिवाइस रिलीज़ करें:** OS किसी डिवाइस को रिलीज़ करता है, यानी उस डिवाइस के लिए आवंटित सिस्टम संसाधनों को मुक्त करता है, जब वह अपना सारा काम पूरा कर लेता है। अनिवार्य रूप से, यह उतना ही सुरक्षित है जितना कि आपके द्वारा काम पूरा करने के बाद कंप्यूटर से USB ड्राइव को निकालना।
- **व्यवधानों को संभालना:** डिवाइस अक्सर CPU को सिग्नल भेजते हैं, जिन्हें व्यवधान कहते हैं, जिससे उसका ध्यान आकर्षित होता है। OS इन व्यवधानों को संभालता है ताकि उसे पता चल सके कि कब किसी डिवाइस को ध्यान देने की आवश्यकता है और सही क्रम में उनसे निपट सके।

4. सूचना रखरखाव

सूचना रखरखाव एक सिस्टम कॉल है जिसका उपयोग सूचना को बनाए रखने के लिए किया जाता है। सूचना रखरखाव के कुछ उदाहरण हैं, जिनमें सिस्टम डेटा प्राप्त करना, समय या तिथि निर्धारित करना, समय या तिथि प्राप्त करना, सिस्टम डेटा सेट करना आदि शामिल हैं। रोजमर्रा की जिंदगी की गतिविधियों में सामान्य सूचना रखरखाव हैं;

- **सिस्टम डेटा प्राप्त करें:** यह सिस्टम के बारे में जानकारी प्राप्त करता है कि वह वास्तव में क्या है, जैसे कि सीपीयू या मेमोरी उपयोग, नेटवर्क स्थिति, आदि। यह उपयोग में आने वाले सभी संसाधनों का रिकॉर्ड रखता है।
- **सिस्टम डेटा सेट करें:** ऑपरेटिंग सिस्टम, जब भी आवश्यक हो, सिस्टम से जुड़े कुछ कॉन्फ़िगरेशन को बदल सकता है, जैसे सिस्टम व्यवहार को बदलना, कॉन्फ़िगरेशन को अपडेट करना, उपयोगकर्ता प्राथमिकताएं सेट करना आदि।
- **समय या दिनांक की पुनर्प्राप्ति:** ओएस सिस्टम के समय और दिनांक को पुनः प्राप्त कर सकता है, जिसका उपयोग फ़ाइल टाइमस्टैम्प, लॉगिंग और शेड्यूलिंग जैसे कई अनुप्रयोगों में किया जाता है।
- **समय या तिथि निर्धारित करना:** सिस्टम आवश्यक होने पर वर्तमान समय और तिथि को अद्यतन कर सकता है; इस प्रकार, सटीक समय और तिथियों को फाइलों या प्रक्रियाओं में जोड़ा जा सकता है या बाहरी समय घड़ी के साथ सिंक्रनाइज़ किया जा सकता है।
- **सिस्टम लॉग और ऑडिट:** ऑपरेटिंग सिस्टम त्रुटियों, सिस्टम प्रदर्शन या उपयोगकर्ता क्रियाओं का इवेंट लॉग और गतिविधि रिकॉर्ड बनाए रखता है। इसका उपयोग मुख्य रूप से सिस्टम को डीबग करने और मॉनिटर करने के लिए किया जाता है।

5. संचार

संचार एक सिस्टम कॉल है जिसका उपयोग संचार के लिए किया जाता है। संचार के कुछ उदाहरण हैं, जिनमें संचार कनेक्शन बनाना, हटाना, संदेश भेजना, प्राप्त करना आदि शामिल हैं। संचार में शामिल प्रमुख गतिविधियाँ इस प्रकार हैं:

- **संचार लिंक स्थापित करना:** ऑपरेटिंग सिस्टम दो प्रक्रियाओं या उपकरणों को संचार चैनल के माध्यम से जोड़ता है। कनेक्शन सॉकेट, पाइप या साझा मेमोरी का उपयोग करके किया जा सकता है; इस प्रकार, यह इन दो प्रक्रियाओं या उपकरणों के बीच आसान संचार की सुविधा प्रदान करता है।
- **संचार लिंक को बंद करना:** यदि ऑपरेटिंग सिस्टम की संचार प्रक्रिया पूरी हो जाती है, तो वह लिंक को हटा देता है या नष्ट कर देता है; संसाधन मुक्त हो जाते हैं, और सिस्टम प्रभावित नहीं होता।
- ऑपरेटिंग सिस्टम यह सुनिश्चित करता है कि एक प्रक्रिया या सिस्टम द्वारा भेजा गया संदेश या डेटा स्थानीय रूप से या नेटवर्क पर लक्षित प्राप्तकर्ता तक पहुंचे।
- **संदेश प्राप्त करना:** प्राप्त करने वाली प्रक्रिया किसी अन्य प्रक्रिया द्वारा भेजे गए डेटा को प्राप्त करती है। OS आने वाले संदेशों का प्रबंधन करता है और सुनिश्चित करता है कि इसे सही लक्ष्य तक पहुंचाया जाए।
- **संचार प्रोटोकॉल का प्रबंधन करता है:** ऑपरेटिंग सिस्टम नियमों के साथ-साथ संचार प्रोटोकॉल का भी प्रबंधन करता है, जो डेटा को उचित प्रारूपों और अनुक्रमों में भेजने और प्राप्त करने में सक्षम बनाता है।

विंडोज़ और यूनिक्स सिस्टम कॉल के उदाहरण

विंडोज और यूनिक्स सिस्टम कॉल के कई उदाहरण हैं। ये नीचे दी गई तालिका में सूचीबद्ध हैं:

Process	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	Fork() Exit() Wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	Open() Read() Write() Close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	Ioctl() Read() Write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	Getpid() Alarm() Sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	Pipe() Shmget() Mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup()	Chmod() Umask() Chown()

open()

ओपन() कॉल प्रक्रिया को फ़ाइल सिस्टम में संग्रहीत फ़ाइल तक पहुँचना शुरू करने की अनुमति देता है। इस प्रकार, यह मांग पर संसाधन आवंटित कर सकता है और एक फ़ाइल डिस्क्रिप्टर लौटा सकता है, जो फ़ाइल तक पहुँचने या पढ़ने के लिए उपयोग किया जाने वाला हैंडल बन जाता है।

कई प्रक्रियाएँ एक ही फ़ाइल खोल सकती हैं, या फ़ाइल तक पहुँच केवल एक प्रक्रिया के लिए ही हो सकती है, जो फ़ाइल सिस्टम और अनुमतियों के डिज़ाइन पर निर्भर करता है। फ़ाइल पर कोई भी रीड या राइट करने से पहले फ़ाइल ओपन() सिस्टम कॉल करने की आवश्यकता होती है।

read()

इसका उपयोग फ़ाइल सिस्टम पर किसी फ़ाइल से डेटा प्राप्त करने के लिए किया जाता है। यह सामान्यतः तीन तर्क स्वीकार करता है:

- एक फ़ाइल डिस्क्रिप्टर.
- पढ़े गए डेटा को संग्रहीत करने के लिए एक बफर.
- फ़ाइल से पढ़ने के लिए बाइट्स की संख्या.

पढ़ी जाने वाली फ़ाइल के फ़ाइल डिस्क्रिप्टर का उपयोग उसे पहचानने और पढ़ने से पहले **open() का उपयोग करके उसे खोलने के लिए किया जा सकता है।**

wait()

कुछ सिस्टम में, किसी प्रक्रिया को आगे बढ़ने से पहले किसी दूसरी प्रक्रिया के निष्पादन को पूरा करने के लिए प्रतीक्षा करनी पड़ सकती है। जब कोई पैरेंट प्रक्रिया चाइल्ड प्रक्रिया बनाती है, तो पैरेंट प्रक्रिया का निष्पादन तब तक निलंबित रहता है जब तक कि चाइल्ड प्रक्रिया समाप्त नहीं हो जाती। पैरेंट प्रक्रिया को निलंबित करने के लिए **wait()** सिस्टम कॉल का उपयोग किया जाता है। एक बार चाइल्ड प्रक्रिया ने अपना निष्पादन पूरा कर लिया, तो नियंत्रण पैरेंट प्रक्रिया को वापस कर दिया जाता है।

close()

इसका उपयोग फ़ाइल सिस्टम एक्सेस को समाप्त करने के लिए किया जाता है। जब इस सिस्टम कॉल को लागू किया जाता है, तो इसका मतलब है कि प्रोग्राम को अब फ़ाइल की आवश्यकता नहीं है, और बफ़र्स फ्लश हो जाते हैं, फ़ाइल जानकारी बदल जाती है, और परिणामस्वरूप फ़ाइल संसाधन डी-आबंटित हो जाते हैं।

GUI

GUI का पूरा नाम **Graphical User Interface** हैं यह **1970** के दशक में अस्तित्व में आया इस सिद्धांत में कंप्यूटर एवं यूजर के मध्य अंत: क्रिया ग्राफिकल विधि से होती हैं यह एक प्रकार का यूजर इंटरफ़ेस है जिसमे ग्राफिकल तत्व शामिल होते है जैसे - **Icon, button, window.** इसमें कंप्यूटर स्क्रीन पर विभिन्न इकाईयों को चित्रित रूप में दिखाया जाता हैं | उपयोगकर्ता माउस की सहायता से स्क्रीन पर बनी चित्रित इकाइयों को केवल क्लिक करके काम में ले सकता हैं विंडोज इस सिद्धांत पर आधारित हैं | **GUI** के कारण ही आज कंप्यूटर चलाना आसान हैं अगर आज **GUI** नहीं होता तो कंप्यूटर चलाना भी कठिन होता |

आज हम माउस का उपयोग करके जो फोटो या विडियो एडिट (**Edit**) कर पाते है, वो इसलिए क्योकि हम उन्हें देख पाते है, जिससे मन मुताबिक बदलाव करना संभव हो पाता है, इसी प्रणाली को हम जीयूआई (ग्राफिकल यूजर इंटरफ़ेस) के नाम से जानते है |

DSG Support Multi Solution

GUI History

GUI (Graphical user Interface) का Concept बहुत पुराना है **सबसे पहले Xerox PARC Company ने GUI को बनाया था** और उसका डेमोंसट्रेशन दिया था उस Time Mouse का उपयोग GUI में हुआ था वह Mouse लकड़ी का एक बड़ा ब्लाक था जिसमें पहिए लगे हुए थे.

वह **User Interface** उस Time से बहुत आगे था परंतु वह ज्यादा चला नहीं क्योंकि वह बहुत ज्यादा महँगा था क्योंकि उस Time Computer में बहुत बड़े-बड़े कमरों के बराबर होते थे और Normal User के Access से बहुत दूर थे.

Xerox PARC कंपनी ने **Apple के Founder Steve Jobs** से मिलकर एक और **GUI User Interface** बनाया जिसे उन्होंने अपने **Fist Macintosh Personal Computer** में add करा जिससे यह Worldwide दुनिया भर में बहुत प्रसिद्ध हुआ और User Interface main stream बन गया.

Apple की सफलता के बाद **Microsoft** जोकि CUI पर काम करने बाले MS-Dos OS का उपयोग करती थी उन्होंने भी **GUI (Graphical user Interface)** का उपयोग करने लगी और उनका प्रोडक्ट Windows 98 पूरी दुनिया भर में बहुत ज्यादा प्रसिद्ध हुआ और सस्ता होने के कारण यहां पूरी दुनिया भर में फैल गया था और पुराने तरीके के Command पर चलने बाले कम्पयूटर काम होते गए.

GUI पुराने Interface से बहुत आसान था और कोई भी इसे आसानी से Use कर सकता था आप इसे virtually देख सकते थे आपको कोई Command याद करने की जरूरत नहीं थी और आराम से कोई बच्चा भी इसे use कर सकता था.

How It's Work GUI

GUI को समझना बहुत आसान है **GUI** बहुत Simple तरीके से काम करता है GUI बहुत सी image से मिलकर बनता है जिसकी मदद से आप Windows open कर सकते हैं Icon पर Click कर सकते हैं Drag Drop कर सकते हैं Copy Pest कर सकते हैं Wallpaper लगा सकते हैं और भी बहुत कुछ कर सकते हैं GUI का main मकसद होता है कोई भी Text को Graphical way में बताना.

Example :- अगर आपको एक पैराग्राफ दे दिया जाए जिसमें एक Seen के बारे में बताया गया है और एक Photo दे दी जाए जिससे वह Seen है तो आप किस को आसानी से याद रख पाओगे सिंपल सी बात है आप एक Photo को आसानी से याद रख पाओगे क्योंकि हमारी **Virtual memory** ज्यादा powerful होती है.

Command line में आपको अगर कोई Folder Open करना है उसमें से कोई File copy करके दूसरे Folder में Pest करनी है तो आपको यह सब एक Black box में Type करके Computer को बताना पड़ेगा और फिर वह Command Computer पढ़कर आपका काम करेगा परंतु **Graphical Interface** में आप normal सा काम जैसे कि copy pest करने के लिए एक file को click करके दूसरे folder में जाकर छोड़ देते हैं तो वह फाइल copy हो जाती है.

GUI Interface का इस्तेमाल कोई बच्चा भी आराम से कर सकता है और कंप्यूटर को सीख सकता है इसलिए कंप्यूटर आज दुनियाभर में प्रचलित हो गया है हर को इसे use कर रहा है.

Difference Between GUI & CUI

CUI (Character user Interface) :- मैं आप **Computer** को **Keyboard** के जरिये **command** देखकर कोई भी काम करवा सकते हैं यह **Normal use** जिनको कंप्यूटर की **Knowledge** कम होती है वह आसानी से **use** नहीं कर सकते **CUI** में

Graphical Interface नहीं होता है सिर्फ **Plan Text** होते हैं **Computer** में जो **Advance user** होते हैं वह इस **Interface** को पसंद करते हैं बड़े-बड़े **Data center** और छोटे-छोटे **Electronics** में **CUI** का ही उपयोग होता है.

CUI use करने बाले **Operating System :-**

1. **Linux Unix Operating System**
2. **Microsoft MS-Dos**

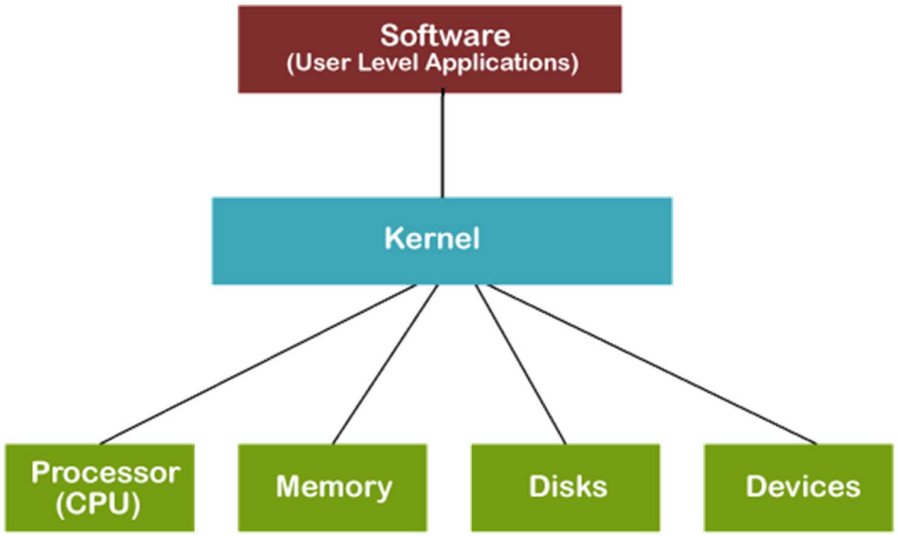
GUI (Graphical user Interface) :- में आपको **Keyboard** की जरूरत नहीं होती है आप आराम से **Mouse** का उपयोग करके **Graphical Interface** में **DATA** को **Access** कर सकते हैं **GUI** वाले **Operating System** को कोई भी **Normal user** आसानी से **use** कर सकता है इसमें आपको कोई **command** देने की जरूरत नहीं होती है यह अधिकतर **Personal Computer** में **use** होता है और इसके अलावा **Android Mobile** और **Apple Mobile** में भी **GUI Interface use** होता है.

GUI use करने वाले **Operating System :-**

1. **Microsoft Operating System**
2. **Linux Ubuntu Operating System**
3. **Apple Mac Operating System**
4. **Android Operating System**

The Kernel

- कर्नेल एक OS (ऑपरेटिंग सिस्टम) का मुख्य भाग है; इसलिए सिस्टम में हर चीज़ पर इसका पूरा नियंत्रण होता है। हार्डवेयर और सॉफ़्टवेयर का प्रत्येक ऑपरेशन कर्नेल द्वारा प्रबंधित और प्रशासित किया जाता है।
- यह हार्डवेयर स्तर पर किए जाने वाले अनुप्रयोगों और डेटा प्रोसेसिंग के बीच एक पुल के रूप में कार्य करता है। यह OS का केंद्रीय घटक है।
- यह ऑपरेटिंग सिस्टम का वह भाग है जो हमेशा कंप्यूटर मेमोरी में रहता है और सॉफ्टवेयर और हार्डवेयर घटकों के बीच संचार को सक्षम बनाता है।
- यह कंप्यूटर प्रोग्राम है जो सिस्टम के स्टार्ट-अप पर सबसे पहले लोड होता है (बूटलोडर के बाद)। लोड होने के बाद, यह बाकी स्टार्ट-अप को मैनेज करता है। यह सॉफ्टवेयर से मेमोरी, पेरिफेरल और I/O रिक्वेस्ट को भी मैनेज करता है। इसके अलावा, यह सभी I/O रिक्वेस्ट को CPU के लिए डेटा प्रोसेसिंग इंस्ट्रक्शन में ट्रांसलेट करता है। यह **मेमोरी मैनेजमेंट, टास्क मैनेजमेंट और डिस्क मैनेजमेंट जैसे** दूसरे काम भी मैनेज करता है ।
- कर्नेल को अलग मेमोरी स्पेस में रखा जाता है और आमतौर पर लोड किया जाता है, जिसे **प्रोटेक्टेड कर्नेल स्पेस के नाम से जाना जाता है।** इसे एप्लीकेशन प्रोग्राम या OS के कम महत्वपूर्ण भागों द्वारा एक्सेस किए जाने से सुरक्षित रखा जाता है।
- अन्य अनुप्रयोग प्रोग्राम जैसे ब्राउज़र, वर्ड प्रोसेसर, ऑडियो और वीडियो प्लेयर अलग मेमोरी स्पेस का उपयोग करते हैं जिसे **यूजर-स्पेस के रूप में जाना जाता है।**
- इन दो अलग-अलग स्थानों के कारण, उपयोगकर्ता डेटा और कर्नेल डेटा एक दूसरे के साथ हस्तक्षेप नहीं करते हैं और किसी भी अस्थिरता और धीमेपन का कारण नहीं बनते हैं।



Functions of a Kernel

किसी ऑपरेटिंग सिस्टम का कर्नेल विभिन्न कार्यों को करने के लिए जिम्मेदार होता है और सिस्टम पर नियंत्रण रखता है। कर्नेल की कुछ मुख्य जिम्मेदारियाँ नीचे दी गई हैं:

डिवाइस प्रबंधन

विभिन्न क्रियाएं करने के लिए, प्रक्रियाओं को परिधीय उपकरणों जैसे कि माउस, कीबोर्ड आदि तक पहुंच की आवश्यकता होती है, जो कंप्यूटर से जुड़े होते हैं। डिवाइस ड्राइवर का उपयोग करके इन उपकरणों को नियंत्रित करने के लिए कर्नेल जिम्मेदार होता है। यहां, डिवाइस ड्राइवर एक कंप्यूटर प्रोग्राम है जो ओएस को किसी भी हार्डवेयर डिवाइस के साथ संचार करने में मदद करता है या सक्षम बनाता है।

कर्नेल सभी उपलब्ध उपकरणों की एक सूची बनाए रखता है, और यह सूची पहले से ही ज्ञात हो सकती है, उपयोगकर्ता द्वारा कॉन्फ़िगर की जा सकती है, या रनटाइम पर ओएस द्वारा पता लगाई जा सकती है।

मेमोरी प्रबंधन

कर्नेल के पास कंप्यूटर की मेमोरी तक पहुँचने का पूरा नियंत्रण होता है। प्रत्येक प्रक्रिया को काम करने के लिए कुछ मेमोरी की आवश्यकता होती है, और कर्नेल प्रक्रियाओं को सुरक्षित रूप से मेमोरी तक पहुँचने में सक्षम बनाता है। मेमोरी आवंटित करने के लिए, पहला चरण वर्चुअल एड्रेसिंग के रूप में जाना जाता है, जो पेजिंग या सेगमेंटेशन द्वारा किया जाता है। वर्चुअल एड्रेसिंग प्रक्रियाओं को वर्चुअल एड्रेस स्पेस प्रदान करने की एक प्रक्रिया है। यह एप्लिकेशन को एक दूसरे से टकराने से रोकता है।

संसाधन प्रबंधन

कर्नेल की महत्वपूर्ण कार्यक्षमताओं में से एक है विभिन्न प्रक्रियाओं के बीच संसाधनों को साझा करना। इसे संसाधनों को इस तरह से साझा करना चाहिए कि प्रत्येक प्रक्रिया समान रूप से संसाधन तक पहुँच सके। कर्नेल सिंक्रोनाइज़ेशन और अंतर-प्रक्रिया संचार (IPC) के लिए एक तरीका भी प्रदान करता है। यह प्रक्रियाओं के बीच संदर्भ स्विचिंग के लिए जिम्मेदार है।

कंप्यूटर संसाधनों तक पहुँचना

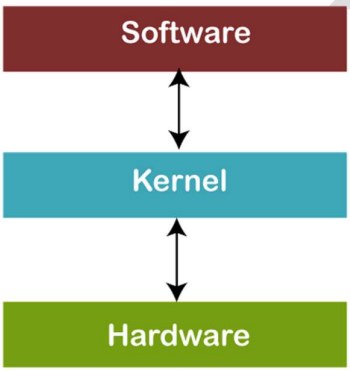
कर्नेल, RAM और I/O डिवाइस जैसे कंप्यूटर संसाधनों तक पहुँचने के लिए जिम्मेदार होता है। RAM या रैंडम-एक्सेस मेमोरी का उपयोग डेटा और निर्देश दोनों को रखने के लिए किया जाता है। प्रत्येक प्रोग्राम को निष्पादित करने के लिए मेमोरी तक पहुँचने की आवश्यकता होती है और अधिकांशतः उपलब्ध मेमोरी से अधिक मेमोरी की आवश्यकता होती है। ऐसे मामले में, कर्नेल अपनी भूमिका निभाता है और यह तय करता है कि प्रत्येक प्रक्रिया किस मेमोरी का उपयोग करेगी और यदि आवश्यक मेमोरी उपलब्ध नहीं है तो क्या करना है।

कर्नेल, कीबोर्ड, माइक्रोफ़ोन, प्रिंटर इत्यादि जैसे I/O डिवाइस का उपयोग करने के लिए अनुप्रयोगों से अनुरोध भी आवंटित करता है ।

कर्नेल के प्रकार :- कर्नेल मुख्यतः पांच प्रकार के होते हैं, जो नीचे दिए गए हैं:

1. मोनोलिथिक कर्नेल

मोनोलिथिक कर्नेल में, **उपयोगकर्ता सेवाओं और कर्नेल सेवाओं को कार्यान्वित करने के लिए समान मेमोरी स्थान का उपयोग किया जाता है।**



इसका अर्थ यह है कि इस प्रकार के कर्नेल में उपयोगकर्ता सेवाओं और कर्नेल सेवाओं के लिए अलग-अलग मेमोरी का उपयोग नहीं किया जाता है।जैसे-जैसे यह समान मेमोरी स्पेस का उपयोग करता है, कर्नेल का आकार बढ़ता जाता है, जिससे OS का समग्र आकार भी बढ़ जाता है।

प्रक्रियाओं का निष्पादन भी अन्य कर्नेल प्रकारों की तुलना में अधिक तेज है क्योंकि यह अलग-अलग उपयोगकर्ता और कर्नेल स्थान का उपयोग नहीं करता है। मोनोलिथिक कर्नेल के **उदाहरण हैं यूनिक्स, लिनक्स, ओपन वीएमएस, एक्सटीएस-400 आदि।**

लाभ:

1.प्रक्रियाओं का निष्पादन भी तेजी से होता है क्योंकि इसमें अलग से उपयोगकर्ता स्थान और कर्नेल स्थान नहीं होता है तथा कम सॉफ्टवेयर शामिल होता है।

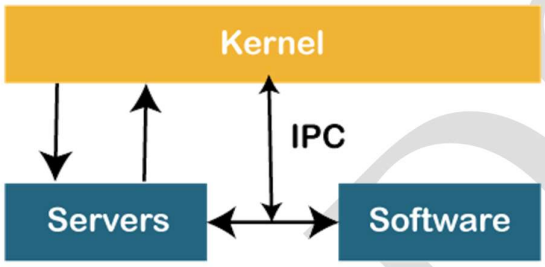
2. चूंकि यह एक ही सॉफ्टवेयर है, इसलिए इसके स्रोत और संकलित रूप दोनों छोटे हैं।

नुकसान:

1. यदि कोई भी सेवा कोई त्रुटि उत्पन्न करती है, तो इससे संपूर्ण सिस्टम क्रैश हो सकता है।
2. ये कर्नेल पोर्टेबल नहीं हैं, जिसका अर्थ है कि प्रत्येक नई आर्किटेक्चर के लिए इन्हें पुनः लिखना होगा।
3. आकार में बड़े होने के कारण इनका प्रबंधन कठिन हो जाता है।
4. नई सेवा जोड़ने के लिए सम्पूर्ण ऑपरेटिंग सिस्टम को संशोधित करना होगा।

2.माइक्रोकर्नेल

माइक्रोकर्नेल को **μK** भी कहा जाता है , और यह पारंपरिक कर्नेल या मोनोलिथिक कर्नेल से अलग है। इसमें ***उपयोगकर्ता सेवाओं और कर्नेल सेवाओं को दो अलग-अलग एड्रेस स्पेस में लागू किया जाता है: उपयोगकर्ता स्थान और कर्नेल स्थान***। चूंकि यह दोनों सेवाओं के लिए अलग-अलग स्पेस का उपयोग करता है, इसलिए माइक्रोकर्नेल का आकार कम हो जाता है, और इससे OS का आकार भी कम हो जाता है।



मोनोलिथिक कर्नेल की तुलना में माइक्रोकर्नेल को प्रबंधित करना और बनाए रखना आसान है। फिर भी, यदि सिस्टम कॉल और संदर्भ स्विचिंग की संख्या अधिक होगी, तो यह सिस्टम को धीमा करके उसके प्रदर्शन को कम कर सकता है।

ये कर्नेल एक सर्वर से दूसरे सर्वर तक अनुरोध को संभालने के लिए संदेश पासिंग प्रणाली का उपयोग करते हैं।

माइक्रोकर्नेल द्वारा केवल कुछ आवश्यक सेवाएँ प्रदान की जाती हैं, जैसे मेमोरी एड्रेस स्पेस को परिभाषित करना, IPC (इंटरप्रोसेस कम्युनिकेशन), और प्रक्रिया प्रबंधन। नेटवर्किंग जैसी अन्य सेवाएँ कर्नेल द्वारा प्रदान नहीं की जाती हैं और **सर्वर** के रूप में जाने जाने वाले उपयोगकर्ता-स्पेस प्रोग्राम द्वारा नियंत्रित की जाती हैं ।

मोनोलिथिक कर्नेल का एक मुख्य नुकसान यह है कि कर्नेल में कोई त्रुटि पूरे सिस्टम को क्रैश कर सकती है, जिसे माइक्रोकर्नेल में दूर किया जा सकता है। माइक्रोकर्नेल की तरह, यदि कर्नेल प्रक्रिया क्रैश हो जाती है, तो त्रुटि-कारण सेवाओं को पुनः आरंभ करके पूरे सिस्टम को क्रैश होने से रोका जा सकता है।

माइक्रोकर्नेल के **उदाहरण हैं L4, AmigaOS, Minix, K42** आदि।

लाभ

1. माइक्रोकर्नेल को आसानी से प्रबंधित किया जा सकता है।
2. संपूर्ण ऑपरेटिंग सिस्टम को संशोधित किए बिना एक नई सेवा आसानी से जोड़ी जा सकती है।
3. माइक्रोकर्नेल में, यदि कोई कर्नेल प्रक्रिया क्रैश हो जाती है, तो भी पूरे सिस्टम को क्रैश होने से बचाना संभव है।

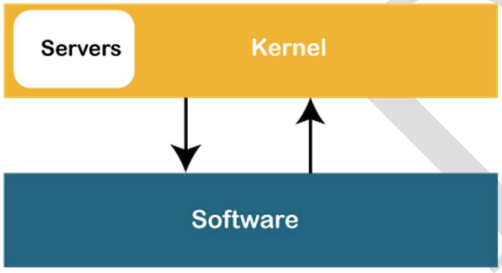
नुकसान

1. इंटरफेसिंग के लिए अधिक सॉफ्टवेयर की आवश्यकता होती है, जिससे सिस्टम का प्रदर्शन कम हो जाता है।
2. प्रक्रिया प्रबंधन बहुत जटिल है।
3. संदेश भेजने संबंधी त्रुटियों को ठीक करना कठिन है।

3. हाइब्रिड कर्नेल

हाइब्रिड कर्नेल को **मॉड्यूलर कर्नेल** के रूप में भी जाना जाता है , और यह मोनोलिथिक और माइक्रोकर्नेल दोनों का संयोजन है। यह मोनोलिथिक कर्नेल की गति और माइक्रोकर्नेल की मॉड्यूलरिटी का लाभ उठाता है।

हाइब्रिड कर्नेल को मोनोलिथिक कर्नेल के अतिरिक्त गुणों के साथ माइक्रोकर्नेल के विस्तारित संस्करण के रूप में समझा जा सकता है। इन कर्नेल का व्यापक रूप से वाणिज्यिक ओएस में उपयोग किया जाता है, जैसे कि एमएस विंडोज के विभिन्न संस्करण।



यह माइक्रोकर्नेल के समान ही है, लेकिन इसमें सिस्टम के प्रदर्शन को बढ़ाने के लिए कर्नेल स्पेस में कुछ अतिरिक्त कोड भी शामिल होता है।

हाइब्रिड कर्नेल कुछ सेवाओं जैसे **नेटवर्क स्टैक को कर्नेल स्पेस में** चलाने की अनुमति देता है , जिससे पारंपरिक माइक्रोकर्नेल की तुलना में प्रदर्शन कम हो जाता है, लेकिन यह अभी भी कर्नेल कोड (जैसे डिवाइस ड्राइवर) को उपयोगकर्ता-स्पेस में सर्वर के रूप में चलाने की अनुमति देता है।

हाइब्रिड कर्नेल के उदाहरण हैं विंडोज़ एनटी, नेटवेयर, बीओएस आदि।

लाभ:

- परीक्षण के लिए रीबूट की कोई आवश्यकता नहीं है।
- तीसरे पक्ष की प्रौद्योगिकी को तेजी से एकीकृत किया जा सकता है।

नुकसान:

- अधिक इंटरफेस से गुजरने पर अधिक बग होने की सम्भावना रहती है।
- कुछ प्रशासकों के लिए मॉड्यूलों का रखरखाव करना एक भ्रामक कार्य हो सकता है, विशेष रूप से जब प्रतीक अंतर जैसे मुद्दों से निपटना हो।

4. नैनोकर्नेल

नैनोकर्नेल में, कर्नेल का पूरा कोड बहुत छोटा होता है, जिसका अर्थ है कि हार्डवेयर के विशेषाधिकार प्राप्त मोड में निष्पादित कोड बहुत छोटा होता है। यहाँ नैनो शब्द एक कर्नेल को परिभाषित करता है जो नैनोसेकंड क्लॉक रिज़ॉल्यूशन का समर्थन करता है।
नैनोकर्नेल के उदाहरण **हैं EROS आदि।**
लाभ :- यह बहुत छोटे आकार के साथ भी हार्डवेयर अमूर्तता प्रदान करता है।

नुकसान:- नैनोकर्नेल में सिस्टम सेवाओं का अभाव है।

5. एक्सोकर्नेल

एक्सोकर्नेल अभी भी विकासशील है और यह ऑपरेटिंग सिस्टम डिजाइन करने के लिए एक प्रयोगात्मक दृष्टिकोण है। इस प्रकार का कर्नेल अन्य कर्नेल से भिन्न है क्योंकि इसमें संसाधन संरक्षण को प्रबंधन से अलग रखा जाता है, जो हमें अनुप्रयोग-विशिष्ट अनुकूलन करने की अनुमति देता है।

लाभ:- एक्सोकर्नेल-आधारित सिस्टम कई लाइब्रेरी ऑपरेटिंग सिस्टम को शामिल कर सकता है। प्रत्येक लाइब्रेरी एक अलग एपीआई निर्यात करती है, जैसे कि एक का उपयोग उच्च-स्तरीय यूआई विकास के लिए किया जा सकता है, और दूसरे का उपयोग वास्तविक समय नियंत्रण के लिए किया जा सकता है।

नुकसान:- एक्सोकर्नेल का डिज़ाइन बहुत जटिल है।

Booting

बूटिंग कंप्यूटर को शुरू करने की प्रक्रिया है। इसे हार्डवेयर द्वारा शुरू किया जा सकता है जैसे कि बटन प्रेस या सॉफ़्टवेयर कमांड द्वारा। स्विच ऑन होने के बाद, CPU की मुख्य मेमोरी में कोई सॉफ़्टवेयर नहीं होता है, इसलिए कुछ प्रक्रियाओं को निष्पादन से पहले मेमोरी में सॉफ़्टवेयर लोड करना पड़ता है। यह CPU में हार्डवेयर या फ़र्मवेयर द्वारा या कंप्यूटर सिस्टम में एक अलग प्रोसेसर द्वारा किया जा सकता है।

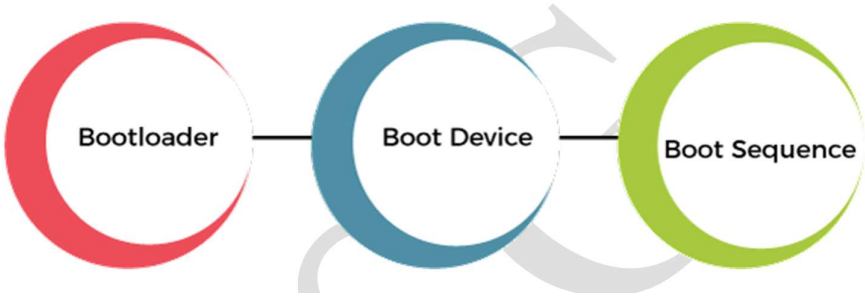
कंप्यूटर को पुनः आरंभ करना भी रीबूटिंग कहलाता है, जो "***हार्ड***" हो सकता है, उदाहरण के लिए, CPU को विद्युत शक्ति बंद से चालू करने के बाद, या "***सॉफ्ट***", जहाँ बिजली नहीं काटी जाती है। कुछ सिस्टम पर, सॉफ्ट बूट वैकल्पिक रूप से RAM को शून्य पर साफ़ कर सकता है। हार्ड और सॉफ्ट बूटिंग को बटन प्रेस या सॉफ़्टवेयर कमांड जैसे हार्डवेयर द्वारा आरंभ किया जा सकता है। बूटिंग तब पूरी होती है जब ऑपरेटिव रनटाइम सिस्टम, आमतौर पर ऑपरेटिंग सिस्टम और कुछ एप्लिकेशन, प्राप्त हो जाते हैं।

कंप्यूटर को स्लीप अवस्था से वापस लाने की प्रक्रिया में बूट करना शामिल नहीं है; हालाँकि, इसे हाइबरनेशन अवस्था से पुनर्स्थापित करना शामिल है। कम से कम, कुछ एम्बेडेड सिस्टम को काम करना शुरू करने के लिए ध्यान देने योग्य बूट अनुक्रम की आवश्यकता नहीं होती है और चालू होने पर, वे ROM में संग्रहीत परिचालन प्रोग्राम चला सकते हैं। सभी कंप्यूटर सिस्टम स्टेट मशीन हैं और रीबूट एक अनपेक्षित, लॉक की गई स्थिति से निर्दिष्ट शून्य-स्थिति में लौटने का एकमात्र तरीका हो सकता है।

ऑपरेटिंग सिस्टम या स्टैंड-अलोन यूटिलिटी को लोड करने के अतिरिक्त , बूट प्रक्रिया ऑपरेटिंग सिस्टम में समस्याओं के निदान के लिए स्टोरेज डंप प्रोग्राम को भी लोड कर सकती है।

Sequencing of Booting

बूटिंग एक स्टार्ट-अप अनुक्रम है जो कंप्यूटर के ऑपरेटिंग सिस्टम को चालू करने पर शुरू करता है। बूट अनुक्रम संचालन का प्रारंभिक सेट है जो कंप्यूटर चालू होने पर करता है। हर कंप्यूटर का एक बूट अनुक्रम होता है।



1. बूट लोडर: सेंट्रल प्रोसेसिंग यूनिट द्वारा संचालित कंप्यूटर केवल सिस्टम की मेमोरी में पाए जाने वाले कोड को निष्पादित कर सकते हैं। आधुनिक ऑपरेटिंग सिस्टम और एप्लिकेशन प्रोग्राम कोड और डेटा नॉनवोलेटाइल मेमोरी पर संग्रहीत होते हैं। जब कंप्यूटर को पहली बार चालू किया जाता है, तो उसे शुरू में सिस्टम की मेमोरी के नॉनवोलेटाइल भागों में संग्रहीत कोड और डेटा पर ही निर्भर रहना पड़ता है। ऑपरेटिंग सिस्टम वास्तव में बूट समय पर लोड नहीं होता है, और कंप्यूटर का हार्डवेयर कई जटिल सिस्टम क्रियाएँ नहीं कर सकता है।

वह प्रोग्राम जो चेन रिएक्शन शुरू करता है और जिसका अंत पूरे ऑपरेटिंग सिस्टम के लोड होने के साथ होता है, बूट लोडर या बूटस्ट्रैप लोडर कहलाता है। बूट लोडर का एकमात्र काम ऑपरेटिंग सिस्टम को शुरू करने के लिए अन्य सॉफ़्टवेयर को लोड करना होता है।

2. बूट डिवाइस: बूट डिवाइस वह डिवाइस है जिससे ऑपरेटिंग सिस्टम लोड किया जाता है। एक आधुनिक पीसी BIOS (बेसिक इनपुट/आउटपुट सिस्टम) विभिन्न डिवाइस से बूट करने का समर्थन करता है। इनमें स्थानीय हार्ड डिस्क ड्राइव, ऑप्टिकल ड्राइव, फ्लॉपी ड्राइव, एक नेटवर्क इंटरफ़ेस कार्ड और एक USB डिवाइस शामिल हैं। BIOS उपयोगकर्ता को बूट ऑर्डर कॉन्फ़िगर करने की अनुमति देगा। यदि बूट ऑर्डर इस पर सेट है:

- सीडी ड्राइव
- हार्ड डिस्क ड्राइव
- नेटवर्क

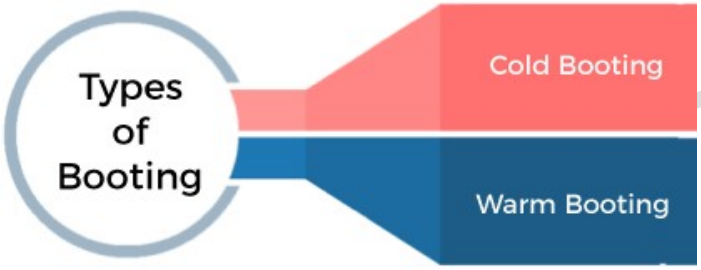
BIOS पहले CD ड्राइव से बूट करने का प्रयास करेगा, और यदि वह असफल हो जाता है, तो वह हार्ड डिस्क ड्राइव से बूट करने का प्रयास करेगा , और यदि वह असफल हो जाता है, तो वह नेटवर्क से बूट करने का प्रयास करेगा, और यदि वह असफल हो जाता है, तो वह बिल्कुल भी बूट नहीं होगा।

3. बूट अनुक्रम: एक मानक बूट अनुक्रम है जिसका उपयोग सभी व्यक्तिगत कंप्यूटर करते हैं। सबसे पहले, CPU BIOS के लिए मेमोरी में एक निर्देश चलाता है। उस निर्देश में एक जंप निर्देश होता है जो BIOS स्टार्ट-अप प्रोग्राम में स्थानांतरित होता है। यह प्रोग्राम यह जांचने के लिए पावर-ऑन सेल्फ-टेस्ट (POST) चलाता है कि कंप्यूटर जिस डिवाइस पर निर्भर करेगा वह ठीक से काम कर रहा है। फिर, BIOS कॉन्फ़िगर किए गए बूट अनुक्रम से तब तक गुजरता है जब तक कि उसे बूट करने योग्य डिवाइस नहीं मिल जाती। एक बार जब BIOS को बूट करने योग्य डिवाइस मिल जाती है, तो BIOS बूटसेक्टर को लोड करता है और निष्पादन को बूट सेक्टर में स्थानांतरित करता है। यदि बूट डिवाइस एक हार्ड ड्राइव है, तो यह एक मास्टर बूट रिकॉर्ड (MBR) होगा।

एमबीआर कोड सक्रिय विभाजन के लिए विभाजन तालिका की जाँच करता है। यदि कोई सक्रिय विभाजन पाया जाता है, तो एमबीआर कोड उस विभाजन के बूट सेक्टर को लोड करता है और उसे निष्पादित करता है। बूट सेक्टर अक्सर ऑपरेटिंग सिस्टम विशिष्ट होता है, और हालाँकि, अधिकांश ऑपरेटिंग सिस्टम में, इसका मुख्य कार्य ऑपरेटिंग सिस्टम कर्नेल को लोड और निष्पादित करना होता है, जो स्टार्ट-अप जारी रखता है। मान लीजिए कि कोई सक्रिय विभाजन नहीं है, या सक्रिय विभाजन का बूट सेक्टर अमान्य है। उस स्थिति में, एमबीआर एक द्वितीयक बूट लोडर लोड कर सकता है जो एक विभाजन का चयन करेगा और उसके बूट सेक्टर को लोड करेगा, जो आमतौर पर संबंधित ऑपरेटिंग सिस्टम कर्नेल को लोड करता है।

बूटिंग के प्रकार

ऑपरेटिंग सिस्टम में बूटिंग दो प्रकार की होती है।



1. **कोल्ड बूटिंग:** जब कंप्यूटर पहली बार स्टार्ट होता है या शट-डाउन अवस्था में होता है और सिस्टम को स्टार्ट करने के लिए पावर बटन को ऑन किया जाता है, तो कंप्यूटर को स्टार्ट करने की इस तरह की प्रक्रिया को कोल्ड बूटिंग कहते हैं। कोल्ड बूटिंग के दौरान, सिस्टम ROM (BIOS) से सभी निर्देशों को पढ़ेगा और ऑपरेटिंग सिस्टम अपने आप सिस्टम में लोड हो जाएगा। इस बूटिंग में हॉट या वार्म बूटिंग से ज़्यादा समय लगता है।
2. **वार्म बूटिंग:** वार्म या हॉट बूटिंग प्रक्रिया तब होती है जब कंप्यूटर सिस्टम नो रिस्पॉन्स या हैंग स्टेट में आ जाता है और फिर सिस्टम को चालू अवस्था में रीस्टार्ट करने की अनुमति दी जाती है। इसे रीबूटिंग भी कहा जाता है। इस स्थिति के कई कारण हैं और इसका एकमात्र समाधान कंप्यूटर को रीबूट करना है। जब हम नया सॉफ़्टवेयर या हार्डवेयर इंस्टॉल करते हैं तो रीबूटिंग की आवश्यकता हो सकती है। सॉफ़्टवेयर या हार्डवेयर कॉन्फ़िगरेशन में बदलाव करने के लिए सिस्टम को रीबूट करने की आवश्यकता होती है या कभी-कभी सिस्टम असामान्य रूप से व्यवहार कर सकता है या ठीक से प्रतिक्रिया नहीं दे सकता है। ऐसे मामले में, सिस्टम को फ़ोर्स रीस्टार्ट करना पड़ता है। सिस्टम को रीबूट करने के लिए सबसे आम तौर पर ***Ctrl+Alt+Del*** बटन का उपयोग किया जाता है। इसके अलावा, कुछ सिस्टम में, सिस्टम को रीबूट करने के लिए बाहरी रीसेट बटन उपलब्ध हो सकता है।

Booting Process in Operating System

जब हमारा कंप्यूटर चालू होता है, तो इसे हार्डवेयर द्वारा शुरू किया जा सकता है जैसे कि बटन प्रेस, या सॉफ़्टवेयर कमांड द्वारा, कंप्यूटर की सेंट्रल प्रोसेसिंग यूनिट (सीपीयू) की मुख्य मेमोरी में कोई सॉफ़्टवेयर नहीं होता है, कुछ प्रक्रिया होती है जिसे निष्पादित करने से पहले सॉफ़्टवेयर को मुख्य मेमोरी में लोड करना पड़ता है। ऑपरेटिंग सिस्टम में बूट प्रक्रिया का वर्णन करने के लिए नीचे छह चरण दिए गए हैं, जैसे:

चरण 1: एक बार जब कंप्यूटर सिस्टम चालू हो जाता है, तो ***BIOS*** (बेसिक इनपुट/आउटपुट सिस्टम) ROM में संग्रहीत प्रोग्रामों पर गतिविधियों या कार्यक्षमता परीक्षणों की एक श्रृंखला करता है, जिसे ***POST*** (पावर-ऑन सेल्फ टेस्ट) कहा जाता है, जो यह जांचता है कि सिस्टम में बाह्य उपकरण सही क्रम में हैं या नहीं।

चरण 2: BIOS द्वारा प्री-बूट गतिविधियों या कार्यक्षमता परीक्षण के बाद, यह ***CMOS (कॉमन मेटल ऑक्साइड सेमीकंडक्टर) से बूट करने योग्य अनुक्रम को पढ़ता है और CMOS*** में निर्दिष्ट बूट डिवाइस अनुक्रम के अनुसार बूट करने योग्य डिस्क के पहले भौतिक क्षेत्र में मास्टर बूट रिकॉर्ड की तलाश करता है। उदाहरण के लिए, यदि बूट डिवाइस अनुक्रम है:

- फ्लॉपी डिस्क
- हार्ड डिस्क
- सीडीरॉम

चरण 3: इसके बाद, मास्टर बूट रिकॉर्ड सबसे पहले फ्लॉपी डिस्क ड्राइव में खोजेगा। अगर नहीं मिला, तो हार्ड डिस्क ड्राइव मास्टर बूट रिकॉर्ड की खोज करेगा। लेकिन अगर हार्ड डिस्क पर मास्टर बूट रिकॉर्ड मौजूद ही नहीं है, तो CDROM ड्राइव खोजेगा। अगर सिस्टम इनमें से किसी भी स्रोत से मास्टर बूट रिकॉर्ड नहीं पढ़ पाता है, तो ROM ***" कोई बूट डिवाइस नहीं मिला"*** प्रदर्शित करता है और सिस्टम को रोक देता है। किसी विशेष बूट करने योग्य डिस्क ड्राइव से मास्टर बूट रिकॉर्ड मिलने पर, ऑपरेटिंग सिस्टम लोडर, जिसे बूटस्ट्रैप लोडर भी कहा जाता है, उस बूट करने योग्य ड्राइव के बूट सेक्टर से मेमोरी में लोड हो जाता है। बूटस्ट्रैप लोडर एक विशेष प्रोग्राम है जो बूट करने योग्य ड्राइव के बूट सेक्टर में मौजूद होता है।

चरण 4: बूटस्ट्रैप लोडर सबसे पहले ***IO.SYS*** फ़ाइल लोड करता है। इसके बाद ***MSDOS.SYS*** फ़ाइल लोड होती है, जो DOS ऑपरेटिंग सिस्टम की कोर फ़ाइल है।

चरण 5: इसके बाद, ***MSDOS.SYS फ़ाइल CONFIG.SYS*** फ़ाइल में कमांड इंटरप्रेटर को खोजती है , और जब उसे मिल जाता है, तो वह मेमोरी में लोड हो जाती है। यदि ***CONFIG.SYS*** फ़ाइल में कोई कमांड इंटरप्रेटर निर्दिष्ट नहीं है, तो ***COMMAND.COM*** फ़ाइल को DOS ऑपरेटिंग सिस्टम के डिफ़ॉल्ट कमांड इंटरप्रेटर के रूप में लोड किया जाता है।

चरण 6: अंतिम फ़ाइल जिसे लोड और निष्पादित किया जाना है वह ***AUTOEXEC.BAT*** फ़ाइल है जिसमें DOS कमांड का एक क्रम होता है। इसके बाद, प्रॉम्प्ट प्रदर्शित होता है। हम कंप्यूटर सिस्टम पर प्रदर्शित बूट करने योग्य ड्राइव का ड्राइव अक्षर देख सकते हैं, जो दर्शाता है कि ऑपरेटिंग सिस्टम उस ड्राइव से सिस्टम पर सफलतापूर्वक चला गया है।

DOS

डिस्क ऑपरेटिंग सिस्टम (DOS) सबसे पहले IBM कंपनी द्वारा पर्सनल कंप्यूटर्स के लिए प्रयुक्त पहला ऑपरेटिंग सिस्टम था। यह मूल रूप से दो संस्करणों में उपलब्ध था जो एक समान थे, लेकिन दो अलग-अलग नामों के तहत उपलब्ध कराए गए था। PC-DOS आईबीएम द्वारा विकसित संस्करण था एवं MS-DOS, माइक्रोसॉफ्ट द्वारा उपलब्ध कराया गया था। कम्प्यूटर अपने ऑपरेटिंग सिस्टम के नियंत्रण में ही काम करता है। डिस्क ऑपरेटिंग सिस्टम (DOS) एक कंप्यूटर ऑपरेटिंग सिस्टम है जो डिस्क स्टोरेज डिवाइस जैसे कि फ्लॉपी डिस्क, हार्ड डिस्क ड्राइव का उपयोग कर सकता है। डिस्क ऑपरेटिंग सिस्टम कम्प्यूटर सिस्टम में फाइल स्टोरेज एवं फाइल मेनेजमेंट, कम्प्यूटर सिस्टम एवं हार्डवेयर डिवाइस को कंट्रोल करने के लिए प्रयोग किया जाता है।

DOS का महत्व

डिस्क ऑपरेटिंग सिस्टम (DOS) कमांड लाइन इंटरफ़ेस (CLI) का उपयोग करता है, जो उपयोगकर्ता (Users) को कमांड टाइप करने की अनुमति देता है। copy और cd (चेंज डायरेक्टरी) जैसे सरल निर्देशों को टाइप करके उपयोगकर्ता हार्ड ड्राइव पर फाइलें ब्राउज़ कर सकते हैं, फाइलें खोल सकते हैं और प्रोग्राम चला सकते हैं। डिस्क ऑपरेटिंग सिस्टम (DOS) की कमांड्स टाइप करने के लिए आसान एवं सरल हैं। लेकिन इस ऑपरेटिंग सिस्टम की कमांड्स को नए सीखने वालों के लिए उपयोग करना मुश्किल होता है, इसी वजह से बाद में माइक्रोसॉफ्ट ने ग्राफिक आधारित विंडोज ऑपरेटिंग सिस्टम को डिस्क ऑपरेटिंग सिस्टम (DOS) जोड़ दिया।

डिस्क ऑपरेटिंग सिस्टम (DOS) का उपयोग किसी सिस्टम या कंप्यूटर को संचालित करने के लिए किया जाता है। यह कंप्यूटर प्रोग्राम का एक सेट है जिसके मुख्य कार्य डिस्क फ़ाइलों को मैनेज करना, आवश्यकता के अनुसार सिस्टम संसाधनों को आवंटित करना है। डिस्क ऑपरेटिंग सिस्टम (DOS) कीबोर्ड, स्क्रीन, डिस्क डिवाइस, प्रिंटर, मोडेम और प्रोग्राम जैसे हार्डवेयर उपकरणों को नियंत्रित करने के लिए आवश्यक सुविधाएँ प्रदान करता है।

डिस्क ऑपरेटिंग सिस्टम (DOS) वह माध्यम है जिसके माध्यम से सिस्टम से जुड़े उपयोगकर्ता और बाहरी उपकरण कंप्यूटर सिस्टम के साथ संवाद (Communication) करते हैं। DOS कंप्यूटर द्वारा समझे जाने वाले लैंग्वेज में उपयोगकर्ता द्वारा जारी किए गए आदेश का अनुवाद करता है और कंप्यूटर को उसी के अनुसार काम करने का निर्देश देता है। यह उपयोगकर्ता को समझने योग्य परिणाम और किसी भी त्रुटि संदेश का भी अनुवाद करता है।

डिस्क ऑपरेटिंग सिस्टम के कार्य

एक ऑपरेटिंग सिस्टम कंप्यूटर प्रोग्राम का एक समूह है जो कंप्यूटर हार्डवेयर उपकरणों के बीच सभी गतिविधियों का समन्वय करता है। यह एक बूट प्रोग्राम द्वारा कंप्यूटर में लोड किया गया पहला प्रोग्राम है और हर समय मेमोरी में बना रहता है।

डिस्क ऑपरेटिंग सिस्टम के मूल कार्य हैं:

1. कंप्यूटर को बूट करना (Booting up Computer)
2. हार्डवेयर कॉन्फ़िगर करना (Configure Hardware)
3. यूजर इंटरफेस उपलब्ध कराना (Provide User Interface)
4. सिस्टम रिसोर्सेज का उपयोग (Using System Resources)
5. फ़ाइल मैनेजमेंट (File Management)

डॉस की विशेषताएं (Features of DOS)

- यह फ़ाइल प्रबंधन को बेहतर बनाने में सहायक है। फाइलें बनाना, संपादित करना, हटाना आदि।
- यह उपयोगकर्ता ऑपरेटिंग सिस्टम है। कोई भी उपयोगकर्ता इस ऑपरेटिंग सिस्टम में एक समय पर काम कर सकता है।
- यह अचरट्रैक्टर बेस्डफ़ोर्टफेस सिस्टम है। हम पत्र (या इस ऑपरेटिंग सिस्टम में वर्ण) टाइप कर सकते हैं।
- MS-DOS 16 बिट ऑपरेटिंग सिस्टम है।
- डॉस सरल टेक्स्ट कमांड ऑपरेटिंग सिस्टम है, यह ग्राफिकल इंटरफ़ेस का समर्थन नहीं करता है
- डॉस टेक्स्ट आधारित इंटरफ़ेस का उपयोग करता है और इसे संचालित करने के लिए टेक्स्ट और कोड की आवश्यकता होती है|
- DOS में इनपुट बेसिक सिस्टम कमांड्स के माध्यम से होता है, अर्थात इसे संचालित करने के लिए माउस का उपयोग नहीं किया जा सकता है|
- डॉस इस बात का बहुविकल्पी समर्थन नहीं करता है कि इसका रैम में एक बार में केवल एक ही प्रक्रिया हो सकती है|
- उपलब्ध स्टोरेज स्पेस की उच्चतम मात्रा 2 जीबी है|

डॉस कमांड

डॉस कमांड ऐसे निर्देश हैं जो विंडोज उपयोगकर्ताओं को फ़ाइलों और फ़ोल्डरों पर कार्य करने की अनुमति देते हैं। जैसा कि आप जानते होंगे, एक फ़ाइल सूचना या डेटा के संग्रह के लिए एक कंटेनर है, और एक निर्देशिका फ़ाइलों का एक संग्रह है। ये केस असंवेदनशील हैं।

MS-DOS में फ़ाइल का नाम 8dot फ़ॉर्मेट में होता है और इसे दो भागों में विभाजित किया जाता है - प्राथमिक और द्वितीयक नाम। प्राथमिक नाम आठ वर्णों तक लंबा हो सकता है, जबकि द्वितीयक नाम एक बिंदु के साथ चार वर्णों तक हो सकता है। उदाहरण के लिए, Logo.jpg फ़ाइल नाम में, प्राथमिक नाम Logo है, जबकि द्वितीयक नाम .jpg है। प्रत्येक प्रकार की फ़ाइल के लिए द्वितीयक नाम सेट किए जाते हैं, इसलिए सिस्टम फ़ाइलों के लिए, द्वितीयक नाम **.sys** है , और टेक्स्ट फ़ाइलों के लिए, यह **.txt** है , आदि। फ़ाइल या निर्देशिका नामों में विशेष वर्ण जैसे **>, ., /, *, ?, |, और** स्पेस की

अनुमति नहीं है। यहाँ सबसे आम फ़ाइल प्रकारों की सूची दी गई है, साथ ही उनके डिफ़ॉल्ट द्वितीयक नाम भी दिए गए हैं:

File name	Extension
Text file	.txt
Program file	.prg
Library file	.lib
Command file	.com
Batch file	.bat
System file	.sys
Executable file	.exe
Database file	.dbm

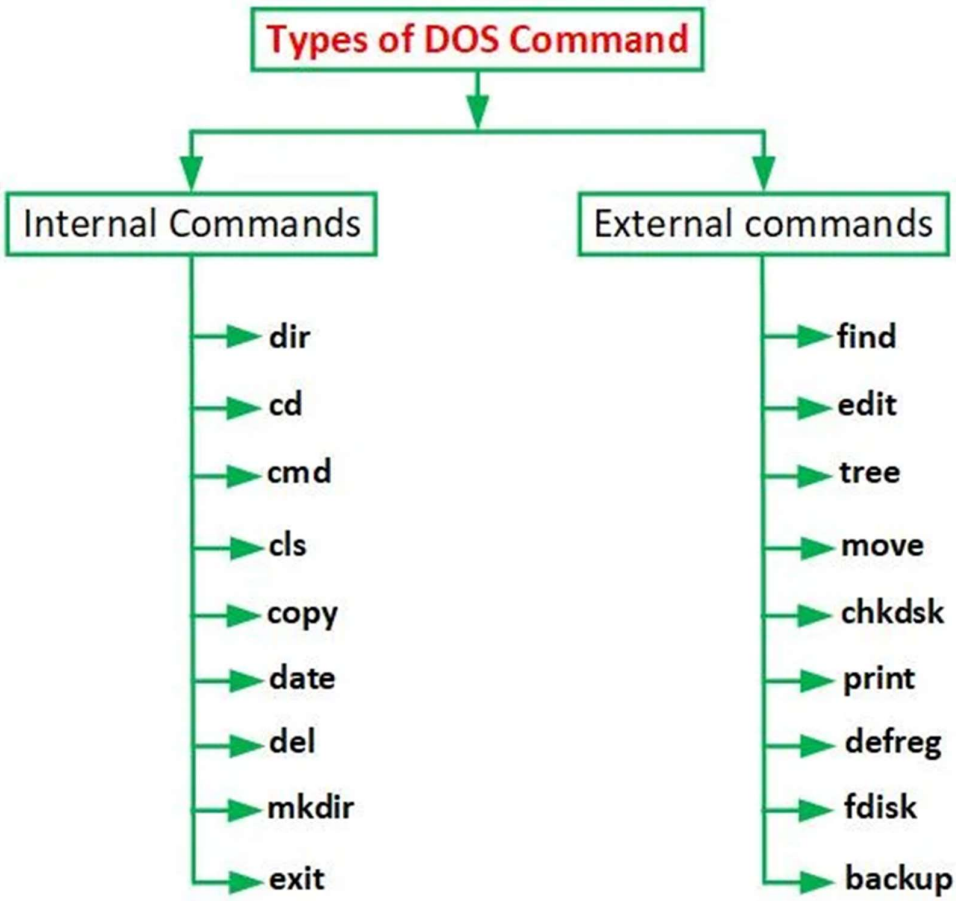
DOS कमांड के प्रकार

डॉस कमांड मुख्यतः दो प्रकार के होते हैं। ये इस प्रकार हैं:

1. **Internal Commands**
2. **External Commands**

1. **Internal Commands:- आंतरिक कमांड** सिस्टम बूटिंग के दौरान सिस्टम मेमोरी में लोड किए जाते हैं। इन आंतरिक कमांड को देखा, संशोधित, पहचाना या हटाया नहीं जा सकता क्योंकि वे फ़ाइल के रूप में मौजूद नहीं होते हैं। एक आंतरिक कमांड सिस्टम मेमोरी में संग्रहीत MS-DOS कमांड है और command.com या cmd.exe से लोड किया जाता है। आंतरिक कमांड में **MD, DATE, COPY, CD, TIME, COPR CON, TYPE** आदि शामिल हैं।

2. **External Commands :- बाहरी कमांड** सिस्टम बूट होने के बाद सिस्टम मेमोरी में स्टोर हो जाते हैं। इन बाहरी कमांड को आसानी से देखा, बदला, डिलीट या कॉपी किया जा सकता है क्योंकि ये फ़ाइल के रूप में मौजूद होते हैं। बाहरी कमांड में FORMAT , **SYS, EDIT, PROMPT, COPY, PRINT, TREE, SORT** आदि शामिल हैं।



Why are internal and external commands needed?

आंतरिक कमांड विंडोज ओएस के कामकाज और उपयोग के लिए अधिक सामान्य और आवश्यक हैं। उन्हें command.com फ़ाइल में एम्बेड करके MS-DOS और Windows में तुरंत सुलभ और हमेशा उपलब्ध बनाया जा सकता है।

बाहरी कमांड वास्तव में प्रभावी हैं। वे समस्याओं के समाधान, प्रदर्शन में वृद्धि और अन्य कार्यों के निष्पादन में मदद करते हैं। बाहरी कमांड को आमतौर पर आंतरिक कमांड की तुलना में अधिक संसाधनों की आवश्यकता होती है। ये कमांड आंतरिक कमांड से अलग रहते हैं और विंडोज पर लोड को कम करने में सहायता करते हैं। बाहरी कमांड वाली फ़ाइल को कंप्यूटर पर कॉपी करके उन्हें किसी भी समय विंडोज में जोड़ा जा सकता है।

DOS Internal Commands:-

1. Date Command –

इस कमाण्ड का प्रयोग कम्प्यूटर सिस्टम की दिनांक को देखने और बदलने के लिये किया जाता है।

Syntax: Date (mm-dd-yy)
उदाहरण:
C:\> Date↵
The current date is: 11-7-2001

MS-Dos commands in hindi

2. Time Command –

इस कमाण्ड का प्रयोग कम्प्यूटर सिस्टम के वर्तमान समय को देखने और बदलने के लिये किया जाता है।

Syntax: Time (time)
उदाहरण:
C:\> Time↵
The current time is: 17:36:49.78
Enter the new time:

MS-Dos commands in hindi

3. Copy Command –

इस कमाण्ड का प्रयोग किसी फाइल को अन्यत्र किसी स्थान पर copy करने के लिये किया जाता है।

Syntax: Copy (A: B:) Source
MS-Dos commands in hindi

4. Path Command –

यह कमाण्ड निष्पादन फाइल (exe file) के रास्ते (path) को निर्दिष्ट करता है। अर्थात इस कमाण्ड को पाथ प्रदर्शित करता है।

Syntax: path (drive) <path>=<path1; path2; ...>
MS-Dos commands in hindi

5. MKDIR / MD Command –

इस कमाण्ड का प्रयोग make directory के लिये किया जाता है अर्थात नई डायरेक्टरी बनाता है।

Syntax: mkdir (drive:) (path) <subdir name>

MS-Dos commands in hindi

6. RMDIR / RD Command –

इस कमाण्ड का प्रयोग डायरेक्टरी को हटाने के लिये किया जाता है। लेकिन delete की जाये वाली सब डायरेक्टरी खाली होनी चाहिये।

Syntax: rmdir (drive:) (path) <subdir name>

MS-Dos commands in hindi

7. CHDIR / CD Command –

यह कमाण्ड वर्तमान डायरेक्टरी को दर्शाने और बदलने के लिये काम आता है।

Syntax: chdir (drive:) (path)

MS-Dos commands in hindi

8) Echo Command –

यह कमांड का उपयोग संदेश को प्रदर्शित करने या कमांड के अनुसरण में आने वाले अंश को स्क्रीन में छपाने के लिए किया जाता है।

Syntax – Echo [on/off]

MS-Dos commands in hindi

9) Type Command –

इस कमांड का उपयोग टेक्स्ट फाइल के कंटेंट को प्रदर्शित या प्रिंट करने के लिए किया जाता है।

Syntax – type [drive:][path]filename

MS-Dos commands in hindi

10) Vol Command –

यह कमांड का उपयोग डिस्क लेबल एवं सीरियल नंबर को दिखाने के लिए किया जाता है।

Syntax – Vol [drive:]

MS-Dos commands in hindi

11) Del or Erase Command –

इस कमांड का उपयोग एक या एक से अधिक फाइलों को हटाने के लिए किया जाता है।

Syntax – Del [drive:][path]filename
Erase [drive:][path]filename

MS-Dos commands in hindi

12) Rename Command –

इस कमांड का उपयोग नाम को बदलने के लिए किया जाता है।

Syntax – RENAME [drive:][path]filename1 filename2

13) Pause Command –

इस कमांड का उपयोग बैच फाइल में प्रोसेस स्थाई रूप से को रोकने के लिए किया जाता है। यह निम्न संदेश दर्शाता है-

Press any key to continue……..
Syntax – Pause

DOS External Commands:-

1) Tree Command –

इस कमांड का उपयोग डायरेक्टरी की संरचना को ग्राफिक्स के रूप में प्रदर्शित करने के लिए किया जाता है।

Syntax – Tree
MS-Dos commands in hindi

2) CHKDSK Command (Check Disk) –

यह कमांड डिस्क को चेक करता है एवं उसकी स्थिति रिपोर्ट को प्रदर्शित करता है।

जैसे:- कूल हिडन फाइल की संख्या, डिस्क की क्षमता, खाली जगह, बेड सेक्टरों की संख्या इत्यादि

Syntax – CHKDSK [drive:][path]filename

MS-Dos commands in hindi

3) ATTRIB Command –

इस कमांड का उपयोग किसी फाइल के गुणों को प्रदर्शित करने अथवा बदलने के लिए किया जाता है।

Syntax – ATTRIB [attribute] [filename]

4) Undelete Command –

इस कमांड का उपयोग डिलीट की गई फाइल को पुनः प्राप्त करने के लिए किया जाता है।

Syntax – Undelete [drive:][path]filename

5) XCOPY Command –

इस कमांड का उपयोग फाइलों और डायरेक्टरी जिसके अंदर और सब डायरेक्टरी हो सकती है इत्यादि को कॉपी करने के लिए काम में लिया जाता है।

xcopy command अदृश्य और छुपी (Hidden) फाइलों की कॉपी नहीं कर सकता है।

Syntax – XCOPY [source][destination]

6) FDISK Command –

यह कमांड हार्ड डिस्क के भाग करने के लिए किया जाता है।

Syntax – FDISK

7) MORE Command –

यह कमांड आउटपुट को एक बार में एक स्क्रीन पर प्रदर्शित करता है जिससे बाद की स्क्रीन पर अगली स्क्रीन की प्रदर्शन होती है।

Syntax – MORE [filename]

8) DISKCOPY Command –

पूरी डिस्क को एक स्थान से दूसरे स्थान पर कॉपी करने के लिए किया जाता है।

Syntax – DISKCOPY [source][destination]

9) SYS Command –

इस कमांड का उपयोग निर्दिष्ट डिस्क में MS-DOS सिस्टम फाइलों की कॉपी करने के लिए किया जाता है जिससे बाद में निर्दिष्ट डिस्क बूटेबल बन जाती है।

Syntax – SYS [drive:]

10) SORT Command –

यह कमांड फाइल में दिखाए गए वाक्य को क्रम में लगाकर स्क्रीन पर प्रदर्शित करता है।

Syntax – SORT [filename]

(11) FORMAT Command:

इस कमांड का उपयोग किसी भी डिस्क की फॉरमॅटिंग के लिए किया जाता है। डिस्क की फॉरमॅटिंग के पश्चात डिस्क में एक फाइल सिस्टम का निर्माण हो जाता है। फॉरमॅटिंग के पश्चात डिस्क में पूर्व से संग्रहित समस्त डाटा नष्ट हो जाता है।

(12) DELTREE Command:

इस कमांड का उपयोग किसी डायरेक्टरी एवं उसके उपविभाग सहित सभी उपविभागों को समाप्त करने के लिए किया जाता है।

(13) Diskcomp Command:

इस कमांड का उपयोग दो डिस्क की अवयवों की तुलना करने के लिए किया जाता है।

(14) Backup Command:

इस कमांड का उपयोग एक या एक से अधिक फाइलों का बैकअप लेने में किया जाता है। बैकअप एवं कॉपी में अंतर यह होता है कि कॉपी की गई फाइल समान मेमोरी में स्थान गैरती है जबकि बैकअप ली गई फाइल बहुत कम स्थान गैरती है लेकिन बैकअप ली गई फाइल सीधे प्रोग्राम में नहीं की जा सकती जब तक बैकअप फाइल को रिस्टोर नहीं कर दिया जाता।

Context Switching in OS

ऑपरेटिंग सिस्टम एक प्रोग्राम है जो किसी सिस्टम या कंप्यूटर में लोड किया जाता है। और उस ओएस प्रोग्राम पर चल रहे सभी अन्य प्रोग्राम का प्रबंधन करता है, यह अन्य सभी एप्लिकेशन प्रोग्राम का प्रबंधन करता है। या दूसरे शब्दों में, हम कह सकते हैं कि ओएस उपयोगकर्ता और कंप्यूटर हार्डवेयर के बीच एक इंटरफेस है।

What is Context Switching in an Operating System?

ऑपरेटिंग सिस्टम में संदर्भ स्विचिंग में किसी चल रही प्रक्रिया के संदर्भ या स्थिति को सहेजना शामिल होता है, ताकि बाद में उसे पुनः स्थापित किया जा सके, तथा फिर किसी अन्य प्रक्रिया के संदर्भ या स्थिति को लोड करके उसे चलाया जा सके।

संदर्भ स्विचिंग से तात्पर्य उस प्रक्रिया/विधि से है जिसका उपयोग सिस्टम द्वारा अपने कार्य को करने के लिए सिस्टम में मौजूद सीपीयू का उपयोग करके प्रक्रिया को एक अवस्था से दूसरी अवस्था में बदलने के लिए किया जाता है।

Example of Context Switching

मान लीजिए कि OS में (N) संख्या में प्रोसेस एक प्रोसेस कंट्रोल ब्लॉक (PCB) में स्टोर किए गए हैं। जैसे प्रक्रिया अपना काम करने के लिए CPU का उपयोग करके चल रही है। जब कोई प्रक्रिया चल रही होती है, तो सर्वोच्च प्राथमिकता वाली अन्य प्रक्रियाएँ अपना काम पूरा करने के लिए CPU का उपयोग करने के लिए कतार में लग जाती हैं।

CPU को किसी दूसरी प्रक्रिया में स्विच करने के लिए वर्तमान प्रक्रिया की स्थिति को सहेजना और किसी दूसरी प्रक्रिया की स्थिति को पुनर्स्थापित करना आवश्यक है। इस कार्य को संदर्भ स्विच के रूप में जाना जाता है। जब संदर्भ स्विच होता है, तो कर्नेल अपने PCB में पुरानी प्रक्रिया के संदर्भ को

सहेजता है और चलाने के लिए शेड्यूल की गई नई प्रक्रिया के सहेजे गए संदर्भ को लोड करता है। संदर्भ-स्विच समय शुद्ध ओवरहेड है क्योंकि स्विच करते समय सिस्टम कोई उपयोगी कार्य नहीं करता है। स्विचिंग गति मशीन से मशीन में भिन्न होती है, जो मेमोरी की गति, कॉपी किए जाने वाले रजिस्टरों की संख्या और विशेष निर्देशों (जैसे सभी रजिस्टरों को लोड या संग्रहीत करने के लिए एक ही निर्देश) के अस्तित्व पर निर्भर करती है। एक सामान्य गति कुछ मिलीसेकंड होती है। संदर्भ-स्विच समय हार्डवेयर समर्थन पर अत्यधिक निर्भर होते हैं। उदाहरण के लिए, कुछ प्रोसेसर (जैसे सन अल्ट्रास्पार्क) रजिस्टरों के कई सेट प्रदान करते हैं। यहां संदर्भ स्विच के लिए बस पॉइंटर को वर्तमान रजिस्टर सेट में बदलने की आवश्यकता होती है। बेशक, अगर रजिस्टर सेट की तुलना में अधिक सक्रिय प्रक्रियाएँ हैं, तो सिस्टम पहले की तरह मेमोरी से रजिस्टर डेटा को कॉपी करने का सहारा लेता है। इसके अलावा, ऑपरेटिंग सिस्टम जितना अधिक जटिल होगा, संदर्भ स्विच के दौरान उतना ही अधिक काम करना होगा

Need of Context Switching

संदर्भ स्विचिंग सभी प्रक्रियाओं को उनके निष्पादन को पूरा करने और सिस्टम के कार्यों की स्थिति को संग्रहीत करने के लिए एक ही CPU साझा करने में सक्षम बनाता है। प्रक्रिया का निष्पादन उसी स्थान से शुरू होता है जहाँ प्रक्रिया को सिस्टम में पुनः लोड किए जाने पर संघर्ष होता है।
ऑपरेटिंग सिस्टम की संदर्भ स्विचिंग की आवश्यकता को नीचे सूचीबद्ध कारणों से समझाया गया है।

- सिस्टम के अंदर एक प्रक्रिया सीधे दूसरे पर स्विच नहीं करती है। संदर्भ स्विचिंग ऑपरेटिंग सिस्टम के लिए अपने कार्यों को पूरा करने के लिए CPU के संसाधनों का उपयोग करना और कई प्रक्रियाओं के बीच स्विच करते समय अपने संदर्भ को संग्रहीत करना आसान बनाता है।
- संदर्भ स्विचिंग सभी प्रक्रियाओं को उनके निष्पादन को पूरा करने और सिस्टम के कार्यों की स्थिति को संग्रहीत करने के लिए एक ही CPU साझा करने में सक्षम बनाता है। प्रक्रिया का निष्पादन उसी स्थान से शुरू होता है जहाँ प्रक्रिया को सिस्टम में पुनः लोड किए जाने पर संघर्ष होता है।
- संदर्भ स्विचिंग किसी एकल CPU को किसी अतिरिक्त प्रोसेसर की आवश्यकता के बिना, कई प्रक्रिया अनुरोधों को समानांतर रूप से संभालने की अनुमति देता है।

Context switching triggers

संदर्भ-स्विचिंग ट्रिगर्स की तीन अलग-अलग श्रेणियां इस प्रकार हैं।

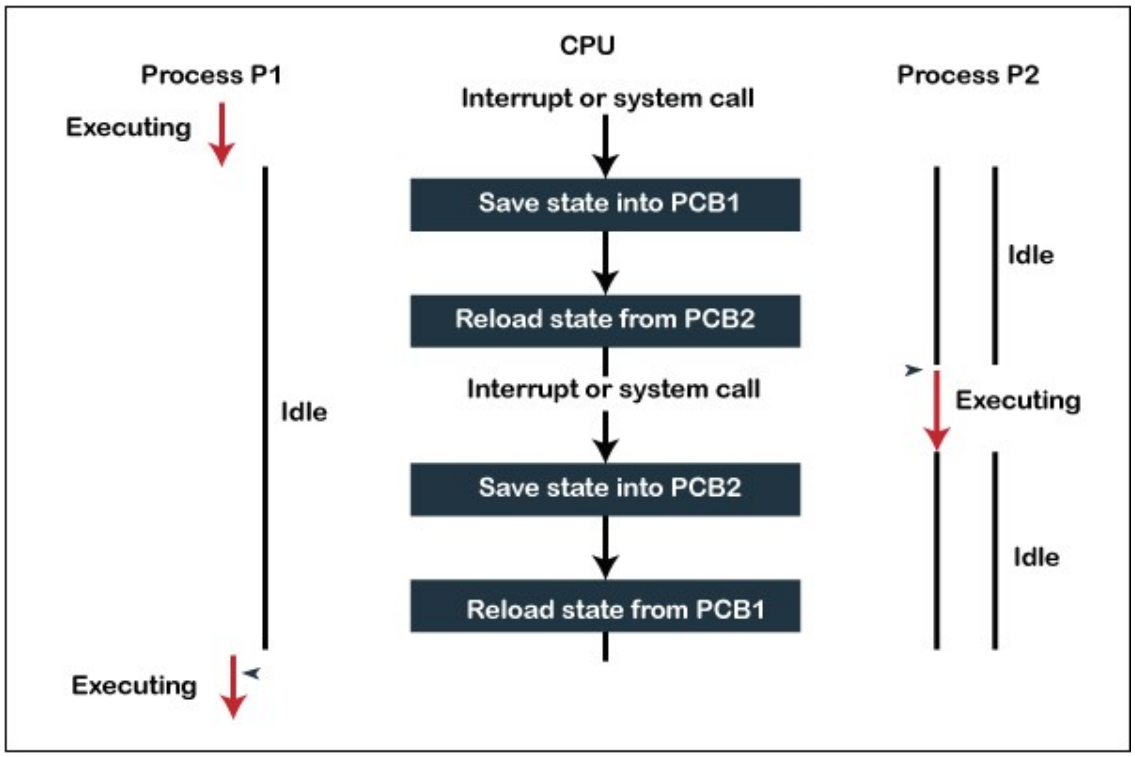
- बीच में आता है
- बहु कार्यण
- उपयोगकर्ता/कर्नेल स्विच

व्यवधान: जब सीपीयू अनुरोध करता है कि डिस्क से डेटा पढ़ा जाए, तो यदि कोई व्यवधान उत्पन्न होता है, तो संदर्भ स्विचिंग स्वचालित रूप से हार्डवेयर के उस घटक पर स्विच हो जाता है जो व्यवधानों को अधिक तेजी से संभाल सकता है।

मल्टीटास्किंग: किसी प्रक्रिया को CPU से स्विच करने की क्षमता ताकि दूसरी प्रक्रिया चल सके, संदर्भ स्विचिंग के रूप में जानी जाती है। जब किसी प्रक्रिया को स्विच किया जाता है, तो पिछली स्थिति को बनाए रखा जाता है ताकि प्रक्रिया सिस्टम में उसी स्थान पर चलती रहे।

कर्नेल/उपयोगकर्ता स्विच: इस ट्रिगर का उपयोग तब किया जाता है जब ओएस को उपयोगकर्ता मोड और कर्नेल मोड के बीच स्विच करने की आवश्यकता होती है ।

जब उपयोगकर्ता मोड और कर्नेल/उपयोगकर्ता मोड के बीच स्विच करना आवश्यक होता है, तो ऑपरेटिंग सिस्टम कर्नेल/उपयोगकर्ता स्विच का उपयोग करते हैं।



Working Process Context Switching

इसलिए दो प्रक्रियाओं का संदर्भ स्विचिंग, प्राथमिकता-आधारित प्रक्रिया प्रक्रिया नियंत्रण ब्लॉक की तैयार कतार में होती है। ये निम्नलिखित चरण हैं।

- वर्तमान प्रक्रिया की स्थिति को पुनर्निर्धारण के लिए सहेजा जाना चाहिए।
- प्रक्रिया स्थिति में पीसीबी या स्विच पर संग्रहीत रिकॉर्ड, क्रेडेंशियल और ऑपरेटिंग सिस्टम-विशिष्ट जानकारी शामिल होती है।
- पीसीबी को कर्नेल मेमोरी में एकल परत में या कस्टम ओएस फ़ाइल में संग्रहीत किया जा सकता है।

- सिस्टम को चलाने के लिए पीसीबी में एक हैंडल जोड़ा गया है।
- ऑपरेटिंग सिस्टम वर्तमान प्रक्रिया के निष्पादन को निरस्त कर देता है और अपने PCB को ट्यून करके प्रतीक्षा सूची से एक प्रक्रिया का चयन करता है।
- पीसीबी के प्रोग्राम काउंटर को लोड करें और चयनित प्रक्रिया में निष्पादन जारी रखें।
- प्रक्रिया/थ्रेड मान इस बात को प्रभावित कर सकते हैं कि कतार से कौन सी प्रक्रियाएं चुनी जाएंगी, यह महत्वपूर्ण हो सकता है।

Process Control Block (PCB)

ऑपरेटिंग सिस्टम में एक प्रक्रिया सॉफ्टवेयर का एक चालू ***उदाहरण*** है । यह ऑपरेटिंग सिस्टम में एक मौलिक अवधारणा है जो कार्य की एक इकाई या एक कार्य का प्रतिनिधित्व करती है जिसे सीपीयू शेड्यूल और निष्पादित कर सकता है।

- ***प्रोग्राम बनाम प्रक्रिया***: एक प्रोग्राम निर्देशों का एक संग्रह है जो सेकेंडरी स्टोरेज (उदाहरण के लिए, एक हार्ड डिस्क) में संग्रहीत होता है, जबकि एक प्रक्रिया मुख्य मेमोरी में उन निर्देशों का निष्पादन है। जब कोई प्रोग्राम मेमोरी में डाला जाता है और चलना शुरू होता है, तो उसे एक प्रक्रिया माना जाता है।
- ***प्रक्रिया स्थितियाँ***: एक प्रक्रिया किसी भी समय कई चरणों में हो सकती है, जैसे " ***नया," "तैयार," "चल रहा है," "प्रतीक्षा कर रहा है***," और " ***समाप्त***।" ये चरण प्रक्रिया के विकास और ऑपरेटिंग सिस्टम के साथ उसके संचार को दर्शाते हैं।
- ***प्रोसेस कंट्रोल ब्लॉक:*** जैसा कि पहले बताया गया है, प्रत्येक प्रक्रिया एक प्रोसेस कंट्रोल ब्लॉक (PCB) के साथ युग्मित होती है। PCB में प्रक्रिया की महत्वपूर्ण जानकारी होती है जैसे कि प्रक्रिया की स्थिति, प्रोग्राम काउंटर, CPU रजिस्टर, मेमोरी प्रबंधन विवरण, I/O स्थिति, और बहुत कुछ।
- ***प्रक्रिया निर्माण:*** प्रक्रियाएं ऑपरेटिंग सिस्टम द्वारा बनाई जाती हैं। जब कोई उपयोगकर्ता कोई प्रोग्राम लॉन्च करता है, तो ऑपरेटिंग सिस्टम आवश्यक संसाधन आवंटित करता है, एक नया PCB बनाता है, और प्रारंभिक प्रक्रिया स्थिति स्थापित करता है। नई बनाई गई प्रक्रिया को बनाने वाली मूल प्रक्रिया को "पैरेंट प्रक्रिया" कहा जाता है, और नई बनाई गई प्रक्रिया को "चाइल्ड प्रक्रिया" कहा जाता है।
- ***संदर्भ स्विचिंग:*** संदर्भ स्विचिंग के माध्यम से, ऑपरेटिंग सिस्टम CPU को एक प्रक्रिया से दूसरी प्रक्रिया में ले जाता है। संदर्भ संक्रमण के दौरान वर्तमान प्रक्रिया की स्थिति (इसके प्रोग्राम काउंटर और CPU रजिस्टर सहित) PCB पर सहेजी जाती है, और निष्पादित की जाने वाली अगली प्रक्रिया की स्थिति PCB से लोड की जाती है। परिणामस्वरूप, सॉफ़्टवेयर के कई भाग CPU को प्रभावी रूप से साझा कर सकते हैं।
- ***प्रक्रिया शेड्यूलिंग:*** ऑपरेटिंग सिस्टम में प्रक्रिया शेड्यूलर यह निर्धारित करता है कि कौन सी प्रक्रियाओं को CPU तक पहुँच मिलेगी और कितने समय के लिए। ***राउंड-रॉबिन, प्राथमिकता-आधारित, या सबसे छोटी नौकरी पहले*** जैसी शेड्यूलिंग तकनीकें समान CPU समय वितरण सुनिश्चित करती हैं और सिस्टम प्रदर्शन में सुधार करती हैं।
- ***अंतर-प्रक्रिया संचार (IPC):*** प्रक्रियाओं को एक दूसरे के साथ संवाद करने और डेटा साझा करने की आवश्यकता हो सकती है। पाइप, साझा मेमोरी और संदेश अग्रेषण IPC विधियों के उदाहरण हैं जो प्रक्रिया संचार और सिंक्रनाइज़ेशन में सहायता करते हैं।

मल्टीटास्किंग में प्रक्रियाएँ आवश्यक हैं क्योंकि वे ऑपरेटिंग सिस्टम को एक ही समय में कई गतिविधियों को प्रबंधित करने और निष्पादित करने की अनुमति देती हैं। प्रोग्राम निष्पादन को कई प्रक्रियाओं में विभाजित करके, ऑपरेटिंग सिस्टम सिस्टम ***स्थिरता, संसाधन उपयोग*** और उपयोगकर्ता अनुरोधों के प्रति जवाबदेही सुनिश्चित करता है।

कुछ ऑपरेटिंग सिस्टम में, ***प्रोसेस कंट्रोल ब्लॉक (PCB),*** जिसे कभी-कभी ***टास्क कंट्रोल ब्लॉक (TCB) के रूप में भी जाना जाता है,*** एक डेटा संरचना है जो महत्वपूर्ण प्रक्रिया जानकारी को संग्रहीत और बनाए रखती है। यह किसी विशिष्ट प्रक्रिया से संबंधित जानकारी के लिए एक केंद्रीय भंडार के रूप में कार्य करता है, जिससे ऑपरेटिंग सिस्टम प्रक्रिया निष्पादन को अधिक प्रभावी ढंग से प्रबंधित और नियंत्रित कर सकता है।

प्रोसेस कंट्रोल ब्लॉक प्रक्रिया से संबंधित डेटा और जानकारी का संग्रह है। हालाँकि ऑपरेटिंग सिस्टम के आधार पर इसकी विशिष्टताएँ अलग-अलग होती हैं, लेकिन एक सामान्य PCB में निम्नलिखित घटक होते हैं:

1. **प्रक्रिया आईडी (पीआईडी):** पीसीबी पर एक विशिष्ट प्रक्रिया आईडी (पीआईडी) ऑपरेटिंग सिस्टम के भीतर प्रक्रिया की पहचानकर्ता के रूप में कार्य करती है। ऑपरेटिंग सिस्टम प्रक्रियाओं का ट्रैक रखने, उन्हें प्रबंधित करने और उनके बीच अंतर करने के लिए इस आईडी का उपयोग करता है।
2. **प्रक्रिया की स्थिति:** प्रक्रिया की स्थिति, जैसे कि चल रही है, प्रतीक्षा कर रही है, तैयार है, या समाप्त हो गई है, इंगित की जाती है। ऑपरेटिंग सिस्टम संचालन को शेड्यूल और प्रबंधित करने के लिए इस डेटा का उपयोग करता है।
3. **प्रोग्राम काउंटर (पीसी):** प्रोग्राम काउंटर मान, जो प्रक्रिया में निष्पादित किए जाने वाले निम्नलिखित निर्देश का पता इंगित करता है, पीसीबी पर संग्रहीत होता है। प्रोग्राम काउंटर को संदर्भ स्विच के दौरान चल रही प्रक्रिया के पीसीबी में सहेजा जाता है और फिर निष्पादन को जारी रखने के लिए पुनर्स्थापित किया जाता है जहां इसे छोड़ा गया था।
4. **CPU रजिस्टर:** यह देखता है कि प्रक्रिया से जुड़े CPU रजिस्टर अब कैसे काम कर रहे हैं। उदाहरणों में ***स्टैक पॉइंटर्स***, सामान्य-उद्देश्य रजिस्टर और ***प्रोग्राम स्थिति फ़्लैग*** शामिल हैं । प्रक्रियाएँ रजिस्टर मानों को सहेजकर और पुनर्स्थापित करके संदर्भ परिवर्तनों के दौरान निर्बाध रूप से काम करना जारी रख सकती हैं।
5. **मेमोरी प्रबंधन जानकारी:** इसमें प्रक्रिया की मेमोरी आवंटन जानकारी शामिल होती है, जैसे कि ***आधार और सीमा रजिस्टर या पेज टेबल***। यह जानकारी ऑपरेटिंग सिस्टम को प्रक्रिया की मेमोरी आवश्यकताओं को उचित रूप से प्रबंधित करने की अनुमति देती है।
6. **प्राथमिकता:** कुछ ऑपरेटिंग सिस्टम प्रत्येक प्रक्रिया को एक प्राथमिकता मान प्रदान करते हैं ताकि यह तय किया जा सके कि प्रक्रिया को CPU समय किस क्रम में प्राप्त होगा। PCB में एक प्राथमिकता फ़ील्ड हो सकती है जो प्रक्रिया के प्राथमिकता स्तर को निर्धारित करती है, जिससे शेड्यूलर CPU संसाधनों को उचित रूप से वितरित कर सकता है।
7. **पैरेंट प्रोसेस आईडी (पीपीआईडी):** पैरेंट प्रोसेस का पीआईडी जिसने वर्तमान प्रोसेस को जन्म दिया। यह डेटा प्रोसेस प्रबंधन और प्रोसेस लिंकेज को ट्रैक करने के लिए महत्वपूर्ण है, खासकर उन परिदृश्यों में जिनमें प्रोसेस पदानुक्रम या प्रोसेस ट्री आर्किटेक्चर की आवश्यकता होती है।
8. **I/O स्थिति:** PCB I/O डिवाइस और प्रक्रिया से संबंधित डेटा के बारे में जानकारी रखता है। ओपन ***फ़ाइल डिस्क्रिप्टर, I/O बफ़र्स*** और लंबित I/O अनुरोध सभी शामिल हैं। इस जानकारी को संग्रहीत करने से ऑपरेटिंग सिस्टम I/O संचालन को प्रबंधित करने और इनपुट/आउटपुट अनुरोधों को कुशलतापूर्वक संभालने में सक्षम होता है।
9. **लेखांकन जानकारी:** प्रक्रिया के संसाधन उपयोग डेटा, जैसे ***CPU समय, मेमोरी उपयोग और I/O गतिविधियों*** का ट्रैक रखता है । यह डेटा प्रदर्शन मूल्यांकन और संसाधन आवंटन विकल्पों में सहायता करता है।
10. **अंतर-प्रक्रिया संचार (IPC) जानकारी:** यदि कोई प्रक्रिया अन्य प्रक्रियाओं के साथ संचार करती है, तो PCB में संचार चैनल, संदेश कतार, साझा मेमोरी क्षेत्र या सिंक्रोनाइज़ेशन प्राइमेटिव के लिए फ़ील्ड या पॉइंटर्स हो सकते हैं। यह प्रक्रियाओं को सफलतापूर्वक संचार करने और डेटा साझा करने की अनुमति देता है।

Process Control Block

Process ID (PID)

Process State

Program Counter(PC)

CPU Registers

Memory Management Information

Priority

Parent Process ID (PID)

I/O Status Information

Accounting Information

Inter-process Communication Information (IPC)

Advantages of Using Process Control Block

- पीसीबी प्रक्रिया के बारे में सभी जानकारी संग्रहीत करता है, इसलिए यह ऑपरेटिंग सिस्टम को प्रक्रिया शेड्यूलिंग, संदर्भ स्विचिंग आदि जैसे विभिन्न कार्यों को निष्पादित करने देता है।
- पीसीबी का उपयोग प्रक्रियाओं को शेड्यूल करने में मदद करता है और यह सुनिश्चित करता है कि सीपीयू संसाधनों का कुशलतापूर्वक आवंटन किया जाए।
- जब किसी प्रक्रिया के बारे में विभिन्न संसाधन उपयोग जानकारी का उपयोग पीसीबी से किया जाता है तो वे कुशल संसाधन उपयोग और संसाधन साझाकरण में मदद करते हैं।
- सीपीयू रजिस्टर और स्टैक पॉइंटर्स की जानकारी ओएस को प्रक्रिया स्थिति को सहेजने में मदद करती है जो संदर्भ स्विचिंग में मदद करती है।
- प्रक्रिया तालिका और PCB का उपयोग ऑपरेटिंग सिस्टम में प्रक्रियाओं को सिंक्रनाइज करने के लिए किया जा सकता है।

Disadvantages of using Process Control Block

- प्रत्येक प्रक्रिया के लिए PCB को संग्रहीत करने के लिए मेमोरी का महत्वपूर्ण उपयोग होता है क्योंकि OS में एक साथ कई प्रक्रियाएँ उपलब्ध हो सकती हैं। इसलिए PCB का उपयोग करने से अतिरिक्त मेमोरी उपयोग बढ़ जाता है।
- पीसीबी का उपयोग करने से ऑपरेटिंग सिस्टम में प्रक्रिया की मापनीयता कम हो जाती है, क्योंकि पीसीबी का उपयोग करने की पूरी प्रक्रिया उपयोगकर्ता के लिए कुछ जटिलताएं जोड़ती है, जिससे सिस्टम को आगे बढ़ाना कठिन हो जाता है।

- प्रक्रिया तालिका और पीसीबी ओवरहेड पेश कर सकते हैं और सिस्टम प्रदर्शन को कम कर सकते हैं। ऑपरेटिंग सिस्टम को प्रत्येक प्रक्रिया के लिए प्रक्रिया तालिका और पीसीबी बनाए रखना चाहिए, जो सिस्टम संसाधनों का उपभोग कर सकता है।

Paging in Operating System

पेजिंग एक मेमोरी प्रबंधन योजना है जो भौतिक मेमोरी के निरंतर आवंटन की आवश्यकता को समाप्त करती है। सेकेंडरी स्टोरेज से मुख्य मेमोरी में पेज के रूप में प्रक्रियाओं को पुनः प्राप्त करने की प्रक्रिया को **पेजिंग** के रूप में जाना जाता है । पेजिंग का मूल उद्देश्य प्रत्येक प्रक्रिया को पेजों में अलग करना है। इसके अतिरिक्त, मुख्य मेमोरी को विभाजित करने के लिए फ़्रेम का उपयोग किया जाएगा। यह योजना किसी प्रक्रिया के भौतिक पता स्थान को गैर-निरंतर होने की अनुमति देती है।
पेजिंग में, भौतिक मेमोरी को **पेज फ़्रेम** नामक निश्चित आकार के ब्लॉक में विभाजित किया जाता है , जो प्रक्रिया द्वारा उपयोग किए जाने वाले पृष्ठों के समान आकार के होते हैं। प्रक्रिया के तार्किक पता स्थान को भी पेज नामक निश्चित आकार के ब्लॉक में विभाजित किया जाता है, जो पेज फ़्रेम के समान आकार के होते हैं। जब कोई प्रक्रिया मेमोरी का अनुरोध करती है, तो ऑपरेटिंग सिस्टम प्रक्रिया को एक या अधिक पेज फ़्रेम आवंटित करता है और प्रक्रिया के **तार्किक पृष्ठों को** भौतिक पेज फ़्रेम में मैप करता है।
तार्किक पृष्ठों और भौतिक पृष्ठ फ़्रेमों के बीच मैपिंग को पृष्ठ तालिका द्वारा बनाए रखा जाता है, जिसका उपयोग **मेमोरी प्रबंधन इकाई** द्वारा तार्किक पतों को भौतिक पतों में बदलने के लिए किया जाता है। पृष्ठ तालिका प्रत्येक तार्किक पृष्ठ संख्या को एक भौतिक पृष्ठ फ़्रेम संख्या से मैप करती है।

Terminologies Associated with Memory Control

- **लॉजिकल एड्रेस या वर्चुअल एड्रेस:** यह एक डील है जो CPU के माध्यम से उत्पन्न होती है और मेमोरी तक पहुँचने के लिए एक तकनीक द्वारा उपयोग की जाती है। इसे लॉजिकल या डिजिटल डील के रूप में जाना जाता है क्योंकि यह मेमोरी में हमेशा एक भौतिक स्थान नहीं होता है, बल्कि डिवाइस के लॉजिकल एड्रेस लोकेशन के अंदर किसी स्थान के साथ कनेक्शन के लिए एक अवसर होता है।
- **लॉजिकल एड्रेस स्पेस या वर्चुअल एड्रेस स्पेस:** यह एक सॉफ्टवेयर प्रोग्राम के माध्यम से उत्पन्न सभी लॉजिकल एड्रेस का सेट है। इसे आम तौर पर वाक्यांशों या बाइट्स में दर्शाया जाता है और पेजिंग स्कीम में नियमित अवधि के पेजों में विभाजित किया जाता है।
- **भौतिक पता:** यह एक पता है जो स्मृति में एक भौतिक स्थान से मेल खाता है। यह वास्तविक पता है जो मेमोरी यूनिट पर उपलब्ध है और मेमोरी नियंत्रक द्वारा स्मृति तक पहुँचने के लिए उपयोग किया जाता है।

- **भौतिक पता स्थान:** यह उन सभी भौतिक पतों का समूह है जो मार्ग के तार्किक पता स्थान के अंदर तार्किक पतों के अनुरूप होते हैं। इसे आमतौर पर शब्दों या बाइट्स में दर्शाया जाता है और पेजिंग योजना में निश्चित आकार के फ़्रेम में विभाजित किया जाता है।

पेजिंग योजना में, क्षेत्र के तार्किक पते को स्थिर-अवधि वाले पृष्ठों में विभाजित किया जाता है, और प्रत्येक इंटरनेट वेब पेज को भौतिक पते के भीतर एक संगत निकाय में मैप किया जाता है। चलने वाला उपकरण प्रत्येक विधि के लिए एक वेब इंटरनेट वेब पेज डेस्क रखता है, जो सिस्टम के तार्किक पतों को उसके संगत भौतिक पतों पर मैप करता है। जब कोई विधि मेमोरी तक पहुँचती है, तो CPU एक तार्किक पता उत्पन्न करता है, जिसे नेट पेज तालिका का उपयोग करके एक भौतिक पते में अनुवादित किया जाता है। मेमोरी नियंत्रक तब मेमोरी तक पहुँचने के लिए भौतिक पते का उपयोग करता है।

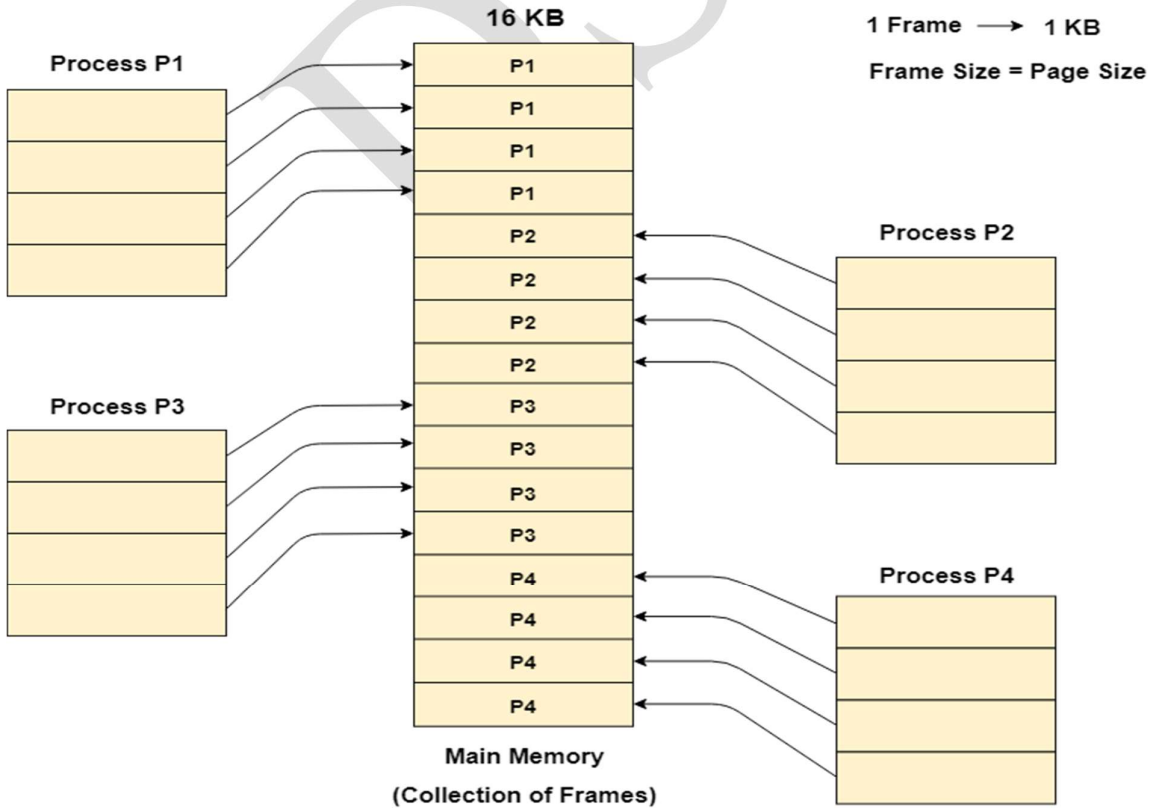
Example:-

मान लीजिए कि मुख्य मेमोरी का आकार 16 Kb है तथा फ्रेम का आकार 1 KB है, इसलिए मुख्य मेमोरी को 1 KB प्रत्येक के 16 फ्रेम के संग्रह में विभाजित किया जाएगा।

सिस्टम में 4 प्रक्रियाएँ हैं, अर्थात् P1, P2, P3 और P4, जिनमें से प्रत्येक का आकार 4 KB है। प्रत्येक प्रक्रिया को 1 KB के पृष्ठों में विभाजित किया गया है, ताकि एक पृष्ठ को एक फ़्रेम में संग्रहीत किया जा सके।

प्रारंभ में, सभी फ्रेम खाली होते हैं इसलिए प्रक्रियाओं के पृष्ठ सन्निहित तरीके से संग्रहीत हो जाएंगे।

फ्रेम, पृष्ठ और दोनों के बीच मैपिंग नीचे दी गई छवि में दिखाई गई है।



Paging

Segmentation in Operating System

एक प्रक्रिया को खंडों में विभाजित किया जाता है। एक प्रोग्राम को जिन खंडों में विभाजित किया जाता है, जो जरूरी नहीं कि सभी सटीक आकार के हों, उन्हें खंड कहा जाता है। सेगमेंटेशन प्रक्रिया के बारे में उपयोगकर्ता का दृष्टिकोण देता है जो पेजिंग प्रदान नहीं करता है। यहां उपयोगकर्ता के दृष्टिकोण को भौतिक मेमोरी में मैप किया जाता है।

सेगमेंटेशन के प्रकार

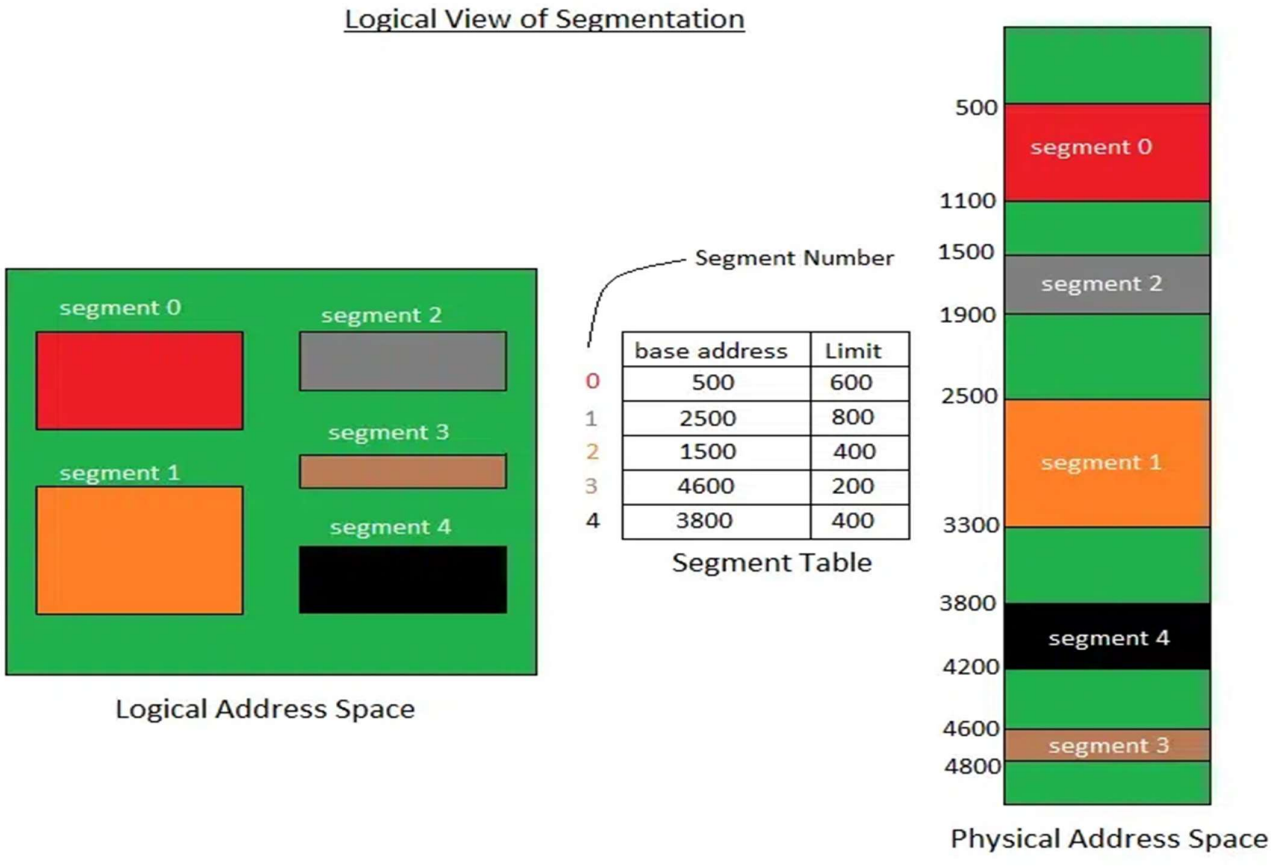
- **वर्चुअल मेमोरी सेगमेंटेशन:** प्रत्येक प्रक्रिया को कई खंडों में विभाजित किया जाता है, लेकिन विभाजन एक साथ नहीं किया जाता है। यह विभाजन प्रोग्राम के रन टाइम पर हो भी सकता है और नहीं भी।
- **सरल विभाजन:** प्रत्येक प्रक्रिया को कई खंडों में विभाजित किया जाता है, जिनमें से सभी को रन टाइम पर मेमोरी में लोड किया जाता है, हालांकि जरूरी नहीं कि वे लगातार हों।

सेगमेंटेशन में तार्किक पतों और भौतिक पतों के बीच कोई सरल संबंध नहीं होता। एक तालिका ऐसे सभी खंडों के बारे में जानकारी संग्रहीत करती है और इसे सेगमेंट तालिका कहा जाता है।

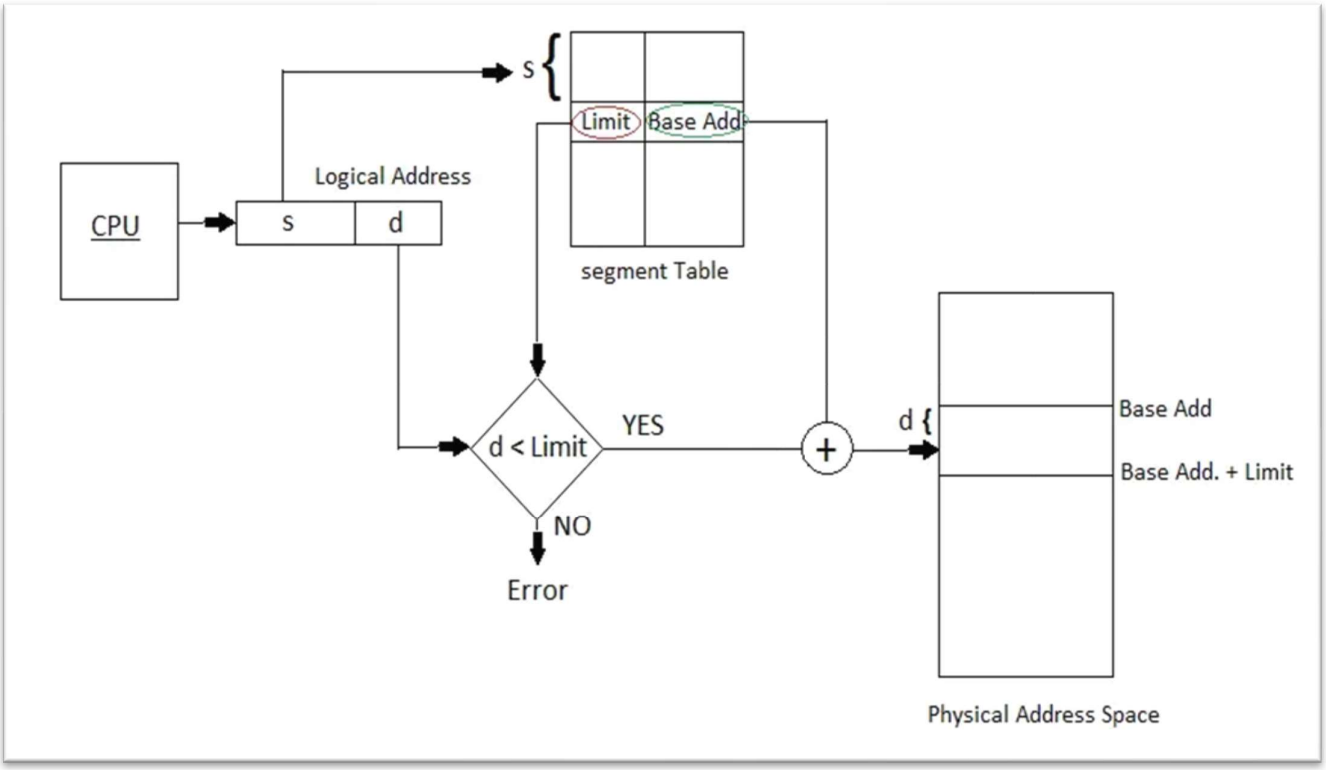
सेगमेंट तालिका

यह दो-आयामी तार्किक पते को एक-आयामी भौतिक पते में मैप करता है। इसकी प्रत्येक तालिका प्रविष्टि में यह है:

- **आधार पता:** इसमें प्रारंभिक भौतिक पता होता है जहां खंड मेमोरी में रहते हैं।
- **सेगमेंट सीमा:** इसे सेगमेंट ऑफ़सेट के नाम से भी जाना जाता है। यह सेगमेंट की लंबाई को निर्दिष्ट करता है।



द्वि-आयामी तार्किक पते का आयामी भौतिक पते में अनुवाद।



सीपीयू द्वारा उत्पन्न पता निम्न प्रकार से विभाजित होता है:

- **खंड संख्या (संख्या):** खंड को दर्शाने के लिए आवश्यक बिट्स की संख्या।
- **सेगमेंट ऑफसेट (d):** किसी सेगमेंट के भीतर डेटा की स्थिति को दर्शाने के लिए आवश्यक बिट्स की संख्या।

सेगमेंटेशन के लाभ

- **आंतरिक विखंडन में कमी** : सेगमेंटेशन निश्चित आकार की पेजिंग की तुलना में आंतरिक विखंडन को कम कर सकता है, क्योंकि सेगमेंट को प्रक्रिया की वास्तविक आवश्यकताओं के अनुसार आकार दिया जा सकता है। हालाँकि, आंतरिक विखंडन तब भी हो सकता है जब किसी सेगमेंट को उसके वास्तविक उपयोग से ज़्यादा जगह आवंटित की जाती है।
- पेजिंग में सेगमेंट टेबल, पेज टेबल की तुलना में कम स्थान लेती है।
- चूंकि पूरा मॉड्यूल एक साथ लोड किया जाता है, इसलिए विभाजन से CPU उपयोग में सुधार होता है।
- भौतिक स्मृति के बारे में उपयोगकर्ता की धारणा विभाजन के समान ही है। उपयोगकर्ता विभाजन के माध्यम से उपयोगकर्ता प्रोग्राम को मॉड्यूल में विभाजित कर सकते हैं। ये मॉड्यूल अलग-अलग प्रक्रियाओं के कोड से ज़्यादा कुछ नहीं हैं।
- उपयोगकर्ता खंड का आकार निर्दिष्ट करता है, जबकि पेजिंग में हार्डवेयर पृष्ठ का आकार निर्धारित करता है।
- विभाजन एक ऐसी विधि है जिसका उपयोग सुरक्षा परिचालनों से डेटा को अलग करने के लिए किया जा सकता है।
- **लचीलापन:** सेगमेंटेशन पेजिंग की तुलना में अधिक लचीलापन प्रदान करता है। सेगमेंट परिवर्तनशील आकार के हो सकते हैं, और प्रक्रियाओं को कई सेगमेंट रखने के लिए डिज़ाइन किया जा सकता है, जिससे अधिक बारीक मेमोरी आवंटन की अनुमति मिलती है।

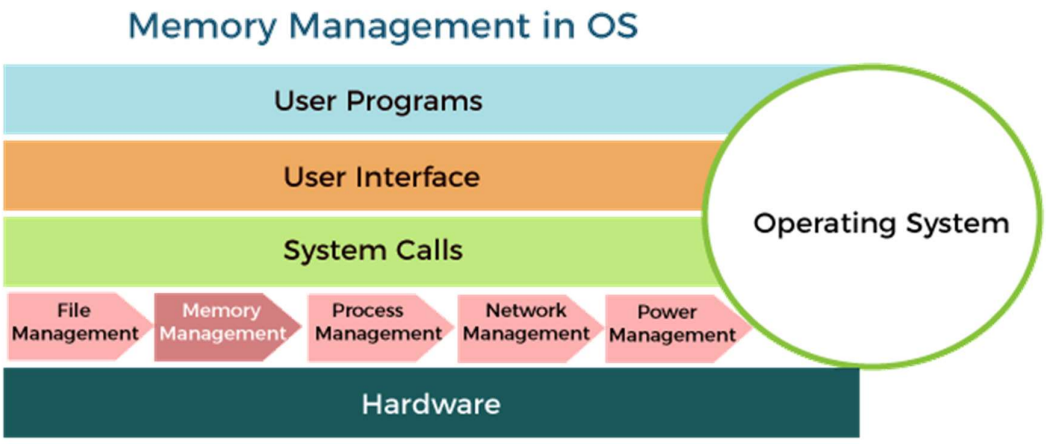
- **साझाकरण:** विभाजन प्रक्रियाओं के बीच मेमोरी खंडों को साझा करने की अनुमति देता है। यह अंतर-प्रक्रिया संचार या कोड लाइब्रेरी साझा करने के लिए उपयोगी हो सकता है।
- **सुरक्षा:** सेगमेंटेशन सेगमेंट के बीच सुरक्षा का एक स्तर प्रदान करता है, जो एक प्रक्रिया को दूसरी प्रक्रिया के मेमोरी सेगमेंट तक पहुँचने या उसे संशोधित करने से रोकता है। इससे सिस्टम की सुरक्षा और स्थिरता बढ़ाने में मदद मिल सकती है।

सेगमेंटेशन के नुकसान

- **बाह्य विखंडन** : जैसे-जैसे प्रक्रियाएँ लोड होती हैं और मेमोरी से हटाई जाती हैं, मुक्त मेमोरी स्पेस छोटे-छोटे टुकड़ों में टूट जाता है, जिससे बाह्य विखंडन होता है। यह पेजिंग से एक उल्लेखनीय अंतर है, जहाँ बाह्य विखंडन काफी कम होता है।
- ओवरहेड प्रत्येक गतिविधि के लिए एक सेगमेंट तालिका रखने से जुड़ा हुआ है।
- दो मेमोरी एक्सेस की आवश्यकता के कारण, एक सेगमेंट तालिका के लिए और दूसरा मुख्य मेमोरी के लिए, निर्देश प्राप्त करने के लिए एक्सेस समय बढ़ जाता है।
- **विखंडन:** जैसा कि बताया गया है, विभाजन से बाह्य विखंडन हो सकता है क्योंकि मेमोरी छोटे-छोटे खंडों में विभाजित हो जाती है। इससे मेमोरी बर्बाद हो सकती है और प्रदर्शन में कमी आ सकती है।
- **ओवरहेड:** सेगमेंट टेबल का उपयोग करने से ओवरहेड बढ़ सकता है और प्रदर्शन कम हो सकता है। प्रत्येक सेगमेंट टेबल प्रविष्टि के लिए अतिरिक्त मेमोरी की आवश्यकता होती है, और मेमोरी स्थानों को पुनः प्राप्त करने के लिए टेबल तक पहुँचने से मेमोरी संचालन के लिए आवश्यक समय बढ़ सकता है।
- **जटिलता:** पेजिंग की तुलना में सेगमेंटेशन को लागू करना और प्रबंधित करना अधिक जटिल हो सकता है। विशेष रूप से, प्रति प्रक्रिया कई सेगमेंट को प्रबंधित करना चुनौतीपूर्ण हो सकता है, और परिणामस्वरूप सेगमेंटेशन दोषों की संभावना बढ़ सकती है।

Memory Management in Operating System

मेमोरी कंप्यूटर का वह महत्वपूर्ण हिस्सा है जिसका उपयोग डेटा को स्टोर करने के लिए किया जाता है। इसका प्रबंधन कंप्यूटर सिस्टम के लिए महत्वपूर्ण है क्योंकि कंप्यूटर सिस्टम में उपलब्ध मुख्य मेमोरी की मात्रा बहुत सीमित होती है। किसी भी समय, कई प्रक्रियाएँ इसके लिए प्रतिस्पर्धा कर रही होती हैं। इसके अलावा, प्रदर्शन को बढ़ाने के लिए, कई प्रक्रियाओं को एक साथ निष्पादित किया जाता है। इसके लिए, हमें मुख्य मेमोरी में कई प्रक्रियाओं को रखना चाहिए, इसलिए उन्हें प्रभावी ढंग से प्रबंधित करना और भी महत्वपूर्ण है।



Role of Memory management

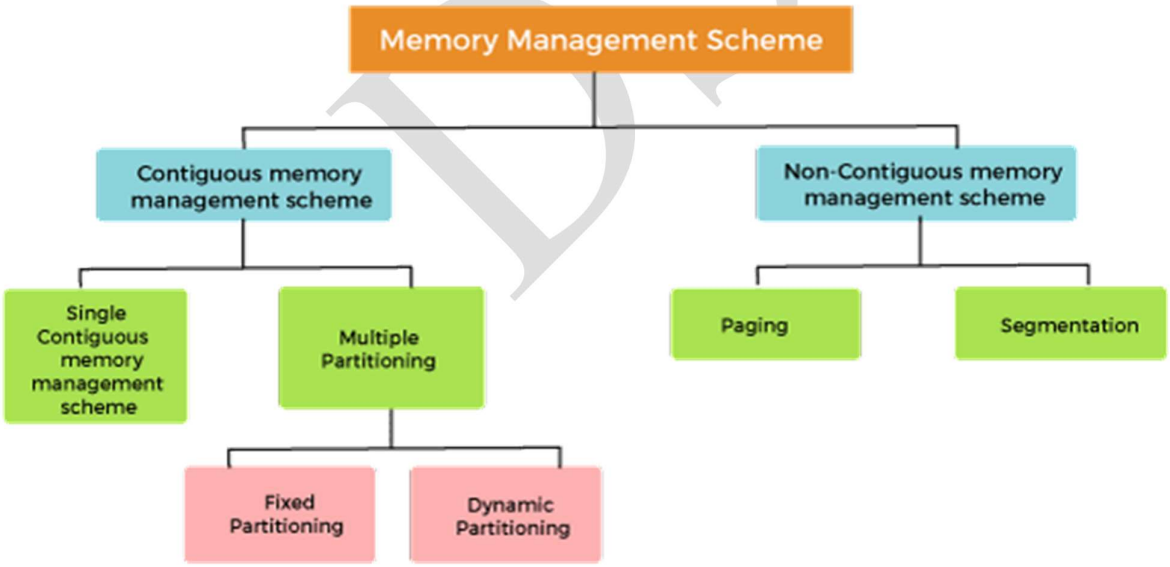
कंप्यूटर सिस्टम में मेमोरी प्रबंधन की महत्वपूर्ण भूमिकाएँ निम्नलिखित हैं:

- मेमोरी मैनेजर का उपयोग मेमोरी स्थानों की स्थिति पर नज़र रखने के लिए किया जाता है, चाहे वह खाली हो या आवंटित। यह अमूर्तता प्रदान करके प्राथमिक मेमोरी को संबोधित करता है ताकि सॉफ़्टवेयर को लगे कि उसे एक बड़ी मेमोरी आवंटित की गई है।
- मेमोरी मैनेजर, कम मात्रा में मुख्य मेमोरी वाले कंप्यूटर को उपलब्ध मेमोरी के आकार या मात्रा से बड़े प्रोग्राम निष्पादित करने की अनुमति देता है। यह स्वैपिंग की अवधारणा का उपयोग करके प्राथमिक मेमोरी और द्वितीयक मेमोरी के बीच सूचना को आगे-पीछे ले जाकर ऐसा करता है।
- मेमोरी मैनेजर प्रत्येक प्रक्रिया को आवंटित मेमोरी को किसी अन्य प्रक्रिया द्वारा दूषित होने से बचाने के लिए जिम्मेदार होता है। यदि यह सुनिश्चित नहीं किया जाता है, तो सिस्टम अप्रत्याशित व्यवहार प्रदर्शित कर सकता है।
- मेमोरी मैनेजर को प्रक्रियाओं के बीच मेमोरी स्पेस को साझा करने में सक्षम होना चाहिए। इस प्रकार, दो प्रोग्राम एक ही मेमोरी लोकेशन पर अलग-अलग समय पर रह सकते हैं।

Memory Management Techniques:

स्मृति प्रबंधन तकनीकों को निम्नलिखित मुख्य श्रेणियों में वर्गीकृत किया जा सकता है:

- **Contiguous memory management schemes**
- **Non-Contiguous memory management schemes**



Classification of memory management schemes

Contiguous memory management schemes:

सन्निहित स्मृति प्रबंधन योजना में, प्रत्येक प्रोग्राम भंडारण स्थानों के एक सन्निहित ब्लॉक पर कब्जा करता है, अर्थात, क्रमिक पतों के साथ स्मृति स्थानों का एक सेट।

Single contiguous memory management schemes:

एकल सन्निहित मेमोरी प्रबंधन योजना कंप्यूटर सिस्टम की शुरुआती पीढ़ी में इस्तेमाल की जाने वाली सबसे सरल मेमोरी प्रबंधन योजना है। इस योजना में, मुख्य मेमोरी को दो सन्निहित क्षेत्रों या विभाजनों में विभाजित किया जाता है। ऑपरेटिंग सिस्टम स्थायी रूप से एक विभाजन में रहते हैं, आमतौर पर निचली मेमोरी में, और उपयोगकर्ता प्रक्रिया को दूसरे विभाजन में लोड किया जाता है।

Advantages of Single contiguous memory management schemes:

- कार्यान्वयन में सरल.
- प्रबंधन और डिजाइन करने में आसान.
- एकल सन्निहित मेमोरी प्रबंधन योजना में, एक बार प्रक्रिया लोड हो जाने पर, उसे प्रोसेसर का पूरा समय दिया जाता है, तथा कोई अन्य प्रोसेसर उसे बाधित नहीं करेगा।

Disadvantages of Single contiguous memory management schemes:

- अप्रयुक्त मेमोरी के कारण मेमोरी स्थान की बर्बादी होती है, क्योंकि प्रक्रिया द्वारा समस्त उपलब्ध मेमोरी स्थान का उपयोग करना संभव नहीं होता।
- सीपीयू निष्क्रिय रहता है, तथा डिस्क द्वारा बाइनरी इमेज को मुख्य मेमोरी में लोड करने की प्रतीक्षा करता है।
- यदि प्रोग्राम इतना बड़ा हो कि वह सम्पूर्ण उपलब्ध मुख्य मेमोरी स्थान में समा न सके तो इसे निष्पादित नहीं किया जा सकता।
- यह मल्टीप्रोग्रामिंग का समर्थन नहीं करता है, अर्थात यह एक साथ कई प्रोग्रामों को संभाल नहीं सकता है।

Multiple Partitioning:

एकल सन्निहित मेमोरी प्रबंधन योजना अक्षम है क्योंकि यह कंप्यूटर को एक समय में केवल एक प्रोग्राम निष्पादित करने तक सीमित करती है जिसके परिणामस्वरूप मेमोरी स्पेस और CPU समय की बर्बादी होती है। अकुशल CPU उपयोग की समस्या को मल्टीप्रोग्रामिंग का उपयोग करके दूर किया जा सकता है जो एक से अधिक प्रोग्राम को एक साथ चलाने की अनुमति देता है। दो प्रक्रियाओं के बीच स्विच करने के लिए, ऑपरेटिंग सिस्टम को दोनों प्रक्रियाओं को मुख्य मेमोरी में लोड करने की आवश्यकता होती है। ऑपरेटिंग सिस्टम को मुख्य मेमोरी में कई प्रक्रियाओं को लोड करने के लिए उपलब्ध मुख्य मेमोरी को कई भागों में विभाजित करने की आवश्यकता होती है। इस प्रकार कई प्रक्रियाएँ एक साथ मुख्य मेमोरी में रह सकती हैं।

The multiple partitioning schemes can be of two types:

- Fixed Partitioning
- Dynamic Partitioning

Fixed Partitioning

मुख्य मेमोरी को एक निश्चित विभाजन मेमोरी प्रबंधन योजना या स्थिर विभाजन में कई निश्चित आकार के विभाजनों में विभाजित किया जाता है। ये विभाजन एक ही आकार या अलग-अलग आकार के हो सकते हैं। प्रत्येक विभाजन एक ही प्रक्रिया को पकड़ सकता है। विभाजनों की संख्या मल्टीप्रोग्रामिंग की डिग्री निर्धारित करती है, यानी मेमोरी में प्रक्रियाओं की अधिकतम संख्या। ये विभाजन सिस्टम जनरेशन के समय बनाए जाते हैं और उसके बाद स्थिर रहते हैं।

Advantages of Fixed Partitioning memory management schemes:

- कार्यान्वयन में सरल.
- प्रबंधन और डिजाइन करने में आसान.

Disadvantages of Fixed Partitioning memory management schemes:

- यह योजना आंतरिक विखंडन से ग्रस्त है।
- विभाजनों की संख्या सिस्टम निर्माण के समय निर्दिष्ट की जाती है।

Dynamic Partitioning

गतिशील विभाजन को एक निश्चित विभाजन योजना की समस्याओं को दूर करने के लिए डिज़ाइन किया गया था। गतिशील विभाजन योजना में, प्रत्येक प्रक्रिया केवल उतनी ही मेमोरी लेती है जितनी उसे प्रसंस्करण के लिए लोड किए जाने पर चाहिए। अनुरोधित प्रक्रियाओं को तब तक मेमोरी आवंटित की जाती है जब तक कि संपूर्ण भौतिक मेमोरी समाप्त न हो जाए या शेष स्थान अनुरोध करने वाली प्रक्रिया को रखने के लिए अपर्याप्त न हो जाए। इस योजना में उपयोग किए जाने वाले विभाजन परिवर्तनशील आकार के होते हैं, और सिस्टम निर्माण के समय विभाजनों की संख्या परिभाषित नहीं की जाती है।

Advantages of Dynamic Partitioning memory management schemes:

- कार्यान्वयन में सरल.
- प्रबंधन और डिजाइन करने में आसान.

Disadvantages of Dynamic Partitioning memory management schemes:

- यह योजना भी आंतरिक विखंडन से ग्रस्त है।
- सिस्टम विभाजन के समय विभाजनों की संख्या निर्दिष्ट की जाती है।

Non-Contiguous memory management schemes:

नॉन-कॉन्टीग्यूअस मेमोरी मैनेजमेंट स्कीम में, प्रोग्राम को अलग-अलग ब्लॉक में विभाजित किया जाता है और मेमोरी के अलग-अलग हिस्सों में लोड किया जाता है, जो जरूरी नहीं कि एक दूसरे से सटे हों। इस स्कीम को ब्लॉक के आकार और ब्लॉक मुख्य मेमोरी में रहते हैं या नहीं, के आधार पर वर्गीकृत किया जा सकता है।

1. **Paging**
2. **Sagmentaion**

Process Management

एकल कार्य या बैच प्रोसेसिंग सिस्टम के लिए प्रक्रिया प्रबंधन आसान है क्योंकि एक समय में केवल एक प्रक्रिया सक्रिय होती है। कई प्रक्रियाओं (मल्टीप्रोग्रामिंग या मल्टीटास्किंग) के सक्रिय होने के कारण, प्रक्रिया प्रबंधन जटिल हो जाता है क्योंकि CPU को कई प्रक्रियाओं द्वारा कुशलतापूर्वक उपयोग करने की आवश्यकता होती है। कई सक्रिय प्रक्रियाएँ मेमोरी जैसे संसाधनों को साझा कर सकती हैं और एक दूसरे के साथ संवाद कर सकती हैं। यह चीजों को और जटिल बनाता है क्योंकि ऑपरेटिंग सिस्टम को प्रक्रिया सिंक्रनाइज़ेशन करना पड़ता है।

कृपया याद रखें कि मल्टीप्रोग्रामिंग के मुख्य लाभ सिस्टम की प्रतिक्रियाशीलता और बेहतर CPU उपयोग हैं। हम एक ही CPU पर कई प्रक्रियाओं को आपस में जोड़कर चला सकते हैं। उदाहरण के लिए, जब मौजूदा प्रक्रिया IO में व्यस्त हो रही होती है, तो हम CPU को किसी दूसरी प्रक्रिया को सौंप देते हैं।

CPU-Bound vs I/O-Bound Processes

CPU-बाउंड प्रक्रिया को अधिक CPU समय की आवश्यकता होती है या वह रनिंग अवस्था में अधिक समय बिताती है। I/O-बाउंड प्रक्रिया को अधिक I/O समय और कम CPU समय की आवश्यकता होती है। I/O-बाउंड प्रक्रिया को प्रतीक्षा अवस्था में अधिक समय बिताना पड़ता है।

प्रक्रिया नियोजन प्रक्रिया प्रबंधन ऑपरेटिंग सिस्टम का एक अभिन्न अंग है। यह ऑपरेटिंग सिस्टम द्वारा उपयोग की जाने वाली उस प्रणाली को संदर्भित करता है जो यह निर्धारित करती है कि आगे कौन सी प्रक्रिया चलानी है। प्रक्रिया शेड्यूलिंग का लक्ष्य CPU उपयोग को अधिकतम करके, थ्रूपुट समय को कम करके और सिस्टम प्रतिक्रिया समय में सुधार करके समग्र सिस्टम प्रदर्शन में सुधार करना है।

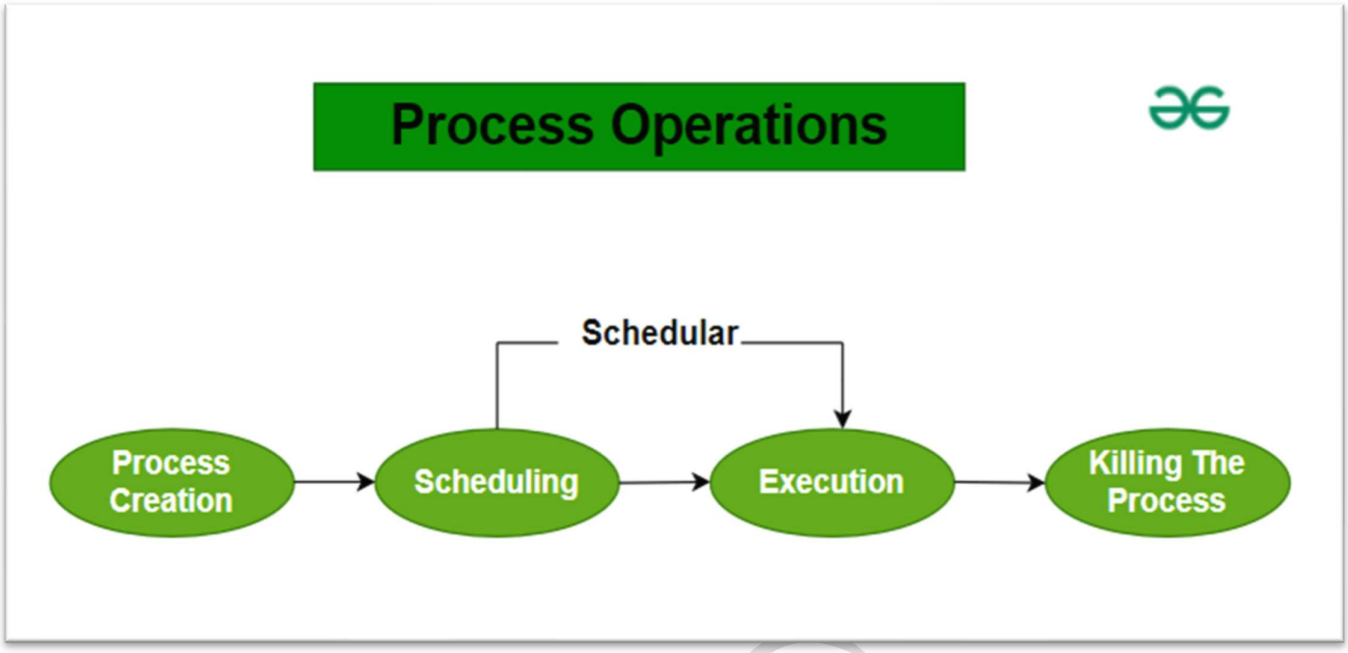
Process Management Tasks

मल्टी-प्रोग्रामिंग या मल्टीटास्किंग वाले ऑपरेटिंग सिस्टम में प्रक्रिया प्रबंधन एक महत्वपूर्ण हिस्सा है।

- प्रक्रिया निर्माण और समाप्ति : प्रक्रिया निर्माण में प्रक्रिया आईडी बनाना, प्रक्रिया नियंत्रण ब्लॉक सेट करना आदि शामिल है। किसी प्रक्रिया को ऑपरेटिंग सिस्टम या पैरेंट प्रक्रिया द्वारा समाप्त किया जा सकता है। प्रक्रिया समाप्ति में इसके लिए आवंटित सभी संसाधनों को साफ़ करना शामिल है।
- CPU शेड्यूलिंग : मल्टीप्रोग्रामिंग सिस्टम में, कई प्रक्रियाओं को CPU की आवश्यकता होती है। कई प्रक्रियाओं के सुचारू और कुशल निष्पादन को सुनिश्चित करना ऑपरेटिंग सिस्टम का काम है।
- डेडलॉक प्रबंधन : यह सुनिश्चित करना कि सिस्टम ऐसी स्थिति में न पहुंच जाए जहां दो या अधिक प्रक्रियाएं एक दूसरे पर निर्भरता के कारण आगे नहीं बढ़ सकतीं।
- अंतर-प्रक्रिया संचार : ऑपरेटिंग सिस्टम सहयोगी प्रक्रियाओं के बीच संचार हेतु साझा मेमोरी और संदेश पासिंग जैसी सुविधाएं प्रदान करता है।
- प्रक्रिया तुल्यकालन : प्रक्रिया तुल्यकालन एक बहु-प्रोग्रामिंग प्रणाली में एकाधिक प्रक्रियाओं के निष्पादन का समन्वय है, ताकि यह सुनिश्चित किया जा सके कि वे नियंत्रित और पूर्वानुमेय तरीके से साझा संसाधनों (जैसे मेमोरी) तक पहुंच सकें।

Process Operations

प्रक्रिया समाप्ति से पहले विभिन्न अवस्थाओं से गुजरती है और इन अवस्था परिवर्तनों के लिए ऑपरेटिंग सिस्टम द्वारा प्रक्रियाओं पर विभिन्न संचालन की आवश्यकता होती है। इन संचालनों में प्रक्रिया निर्माण, प्रक्रिया शेड्यूलिंग, निष्पादन और प्रक्रिया को समाप्त करना शामिल है। यहाँ मुख्य प्रक्रिया संचालन दिए गए हैं:



प्रक्रिया निर्माण

ऑपरेटिंग सिस्टम (OS) में प्रक्रिया निर्माण एक नई प्रक्रिया उत्पन्न करने का कार्य है। यह नई प्रक्रिया एक प्रोग्राम का उदाहरण है जो स्वतंत्र रूप से निष्पादित हो सकती है।

निर्धारण

एक बार जब कोई प्रक्रिया चलने के लिए तैयार हो जाती है, तो वह "तैयार कतार" में प्रवेश करती है। शेड्यूलर का काम इस कतार से एक प्रक्रिया चुनना और उसका निष्पादन शुरू करना है।

कार्यान्वयन

निष्पादन का मतलब है कि CPU प्रक्रिया पर काम करना शुरू कर देता है। इस दौरान, प्रक्रिया निम्न कार्य कर सकती है:

- यदि इसे I/O ऑपरेशन करने की आवश्यकता है तो इसे प्रतीक्षा कतार में ले जाएं।
- यदि उच्च प्राथमिकता वाली प्रक्रिया को CPU की आवश्यकता हो तो ब्लॉक हो जाएँ।

प्रक्रिया को ख़त्म करना

प्रक्रिया द्वारा अपना कार्य समाप्त करने के बाद, ऑपरेटिंग सिस्टम उसे समाप्त कर देता है और उसके प्रोसेस कंट्रोल ब्लॉक (पीसीबी) को हटा देता है।

प्रक्रिया प्रबंधन के लाभ

- **एकाधिक प्रोग्राम चलाना:** प्रक्रिया प्रबंधन आपको एक ही समय में एकाधिक एप्लिकेशन चलाने की सुविधा देता है, उदाहरण के लिए, वेब ब्राउज़ करते समय संगीत सुनना।
- **प्रक्रिया पृथक्करण:** यह सुनिश्चित करता है कि विभिन्न प्रोग्राम एक-दूसरे के साथ हस्तक्षेप न करें, ताकि एक प्रोग्राम में समस्या आने पर दूसरा प्रोग्राम क्रैश न हो जाए।
- **उचित संसाधन उपयोग:** यह सुनिश्चित करता है कि CPU समय और मेमोरी जैसे संसाधन प्रोग्रामों के बीच समान रूप से साझा किए जाएं, ताकि निम्न प्राथमिकता वाले प्रोग्रामों को भी चलने का मौका मिले।

- **सुचारू स्विचिंग:** यह प्रोग्रामों के बीच स्विचिंग को कुशलतापूर्वक संभालता है, सिस्टम को प्रतिक्रियाशील बनाए रखने और देरी को न्यूनतम करने के लिए उनकी स्थिति को शीघ्रता से सहेजता और लोड करता है।

प्रक्रिया प्रबंधन के नुकसान

- **ओवरहेड:** प्रक्रिया प्रबंधन सिस्टम संसाधनों का उपयोग करता है क्योंकि ओएस को विभिन्न डेटा संरचनाओं और शेड्यूलिंग कतारों का ट्रैक रखने की आवश्यकता होती है। इसके लिए CPU समय और मेमोरी की आवश्यकता होती है, जो सिस्टम के प्रदर्शन को प्रभावित कर सकती है।
- **जटिलता:** जटिल शेड्यूलिंग एल्गोरिदम और संसाधन आवंटन विधियों की आवश्यकता के कारण OS का डिजाइन और रखरखाव जटिल है।
- **डेडलॉक:** प्रक्रियाओं को सुचारू रूप से एक साथ चलाने के लिए, ओएस सेमाफोर और म्यूटेक्स लॉक जैसे तंत्र का उपयोग करता है। हालाँकि, ये डेडलॉक की ओर ले जा सकते हैं, जहाँ प्रक्रियाएँ अनिश्चित काल तक एक-दूसरे की प्रतीक्षा में अटकी रहती हैं।
- **बढ़ी हुई संदर्भ स्विचिंग:** मल्टीटास्किंग सिस्टम में , ऑपरेटिंग सिस्टम अक्सर प्रक्रियाओं के बीच स्विच करता है। प्रत्येक प्रक्रिया की स्थिति को संग्रहीत करने और लोड करने (संदर्भ स्विचिंग) में समय और कंप्यूटिंग शक्ति लगती है, जो सिस्टम को धीमा कर सकती है।

Parallel Processing

समानांतर कंप्यूटिंग कई प्रोसेसर द्वारा एक साथ किसी एप्लिकेशन या गणना को निष्पादित करने की प्रक्रिया को संदर्भित करता है। आम तौर पर, यह एक प्रकार की कंप्यूटिंग वास्तुकला है जहाँ बड़ी समस्याएँ स्वतंत्र, छोटे, आमतौर पर समान भागों में टूट जाती हैं जिन्हें एक बार में संसाधित किया जा सकता है। यह साझा मेमोरी के माध्यम से संचार करने वाले कई CPU द्वारा किया जाता है, जो पूरा होने पर परिणामों को जोड़ता है। यह बड़ी गणना करने में मदद करता है क्योंकि यह बड़ी समस्या को एक से अधिक प्रोसेसर के बीच विभाजित करता है।

समानांतर कंप्यूटिंग सिस्टम की उपलब्ध कम्प्यूटेशन शक्ति को बढ़ाकर तेज़ एप्लिकेशन प्रोसेसिंग और टास्क रिज़ॉल्यूशन में भी मदद करती है। समानांतर कंप्यूटिंग सिद्धांतों का उपयोग अधिकांश सुपरकंप्यूटर द्वारा संचालित करने के लिए किया जाता है। जिन परिचालन परिदृश्यों में भारी प्रोसेसिंग पावर या कम्प्यूटेशन की आवश्यकता होती है, आमतौर पर, समानांतर प्रोसेसिंग का उपयोग आमतौर पर किया जाता है।

आम तौर पर, यह बुनियादी ढांचा वहां रखा जाता है जहां विभिन्न प्रोसेसर एक सर्वर रैक में स्थापित होते हैं; एप्लिकेशन सर्वर कम्प्यूटेशनल अनुरोधों को छोटे-छोटे हिस्सों में वितरित करता है, फिर प्रत्येक सर्वर पर अनुरोधों को एक साथ संसाधित किया जाता है। सबसे शुरुआती कंप्यूटर सॉफ़्टवेयर सीरियल कम्प्यूटेशन के लिए लिखे गए हैं क्योंकि वे एक समय में एक ही निर्देश को निष्पादित करने में सक्षम हैं, लेकिन समानांतर कंप्यूटिंग अलग है जहां यह एक समय में कई प्रोसेसर एक एप्लिकेशन या कम्प्यूटेशन को निष्पादित करता है।

समानांतर कंप्यूटिंग का उपयोग करने के कई कारण हैं, जैसे समय और पैसा बचाना, समवर्तीता प्रदान करना, बड़ी समस्याओं को हल करना आदि। इसके अलावा, समानांतर कंप्यूटिंग जटिलता को कम करती है। समानांतर कंप्यूटिंग के वास्तविक जीवन के उदाहरण में, किसी भी चीज़ का टिकट पाने के लिए दो कतारें होती हैं; यदि दो कैशियर एक साथ 2 व्यक्तियों को टिकट दे रहे हैं, तो यह समय बचाने के साथ-साथ जटिलता को कम करने में भी मदद करता है।

Types of parallel computing (Processing)

ओपन-सोर्स और स्वामित्व वाले समानांतर कंप्यूटिंग विक्रेताओं से, आम तौर पर तीन प्रकार की समानांतर कंप्यूटिंग उपलब्ध हैं, जिनकी चर्चा नीचे की गई है:

1. **Bit-level parallelism:** समानांतर कंप्यूटिंग का वह रूप जिसमें हर कार्य प्रोसेसर वर्ड साइज़ पर निर्भर होता है। बड़े आकार के डेटा पर कार्य करने के मामले में, यह प्रोसेसर द्वारा निष्पादित किए जाने वाले निर्देशों की संख्या को कम करता है। ऑपरेशन को निर्देशों की श्रृंखला में विभाजित करने की आवश्यकता है। उदाहरण के लिए, एक 8-बिट प्रोसेसर है, और आप 16-बिट संख्याओं पर एक ऑपरेशन करना चाहते हैं। पहले, इसे 8 निचले-क्रम बिट्स और फिर 8 उच्च-क्रम बिट्स को संचालित करना होगा। इसलिए, ऑपरेशन को निष्पादित करने के लिए दो निर्देशों की आवश्यकता होती है। ऑपरेशन को 16-बिट प्रोसेसर द्वारा एक निर्देश के साथ निष्पादित किया जा सकता है।
2. **Instruction-level parallelism:** एक एकल CPU क्लॉक चक्र में, प्रोसेसर निर्देश-स्तर समानांतरता में तय करता है कि एक ही समय में कितने निर्देश लागू किए जाएं। प्रत्येक क्लॉक चक्र चरण के लिए, निर्देश-स्तर समानांतरता में एक प्रोसेसर में एक से कम निर्देश को संबोधित करने की क्षमता हो सकती है। निर्देश-स्तर समानांतरता में सॉफ़्टवेयर दृष्टिकोण स्थिर समानांतरता पर कार्य करता है, जहाँ कंप्यूटर तय करता है कि किन निर्देशों को एक साथ निष्पादित करना है।
3. **Task Parallelism:** कार्य समानांतरता समानांतरता का वह रूप है जिसमें कार्यों को उप-कार्यों में विघटित किया जाता है। फिर, प्रत्येक उप-कार्य को निष्पादन के लिए आवंटित किया जाता है। और, उप-कार्यों का निष्पादन प्रोसेसर द्वारा समवर्ती रूप से किया जाता है।

Applications of Parallel Computing (Processing)

समानांतर कंप्यूटिंग के विभिन्न अनुप्रयोग हैं, जो इस प्रकार हैं:

- समानांतर कंप्यूटिंग के प्राथमिक अनुप्रयोगों में से एक डेटाबेस और डेटा माइनिंग है।
- प्रणालियों का वास्तविक समय सिमुलेशन समानांतर कंप्यूटिंग का एक अन्य उपयोग है।
- नेटवर्क वीडियो और मल्टीमीडिया जैसी प्रौद्योगिकियां।
- विज्ञान और इंजीनियरिंग.
- सहयोगात्मक कार्य वातावरण.
- समानांतर कंप्यूटिंग की अवधारणा का उपयोग संवर्धित वास्तविकता, उन्नत ग्राफिक्स और आभासी वास्तविकता द्वारा किया जाता है।

Advantages of Parallel computing (Processing)

समानांतर कंप्यूटिंग के लाभों की चर्चा नीचे की गई है:

- समानांतर कंप्यूटिंग में, कार्य को पूरा करने के लिए अधिक संसाधनों का उपयोग किया जाता है जिससे समय कम होता है और संभावित लागत में कटौती होती है। इसके अलावा, समानांतर क्लस्टर बनाने के लिए सस्ते घटकों का उपयोग किया जाता है।

- सीरियल कंप्यूटिंग की तुलना में, समानांतर कंप्यूटिंग कम समय में बड़ी समस्याओं को हल कर सकती है।
- जटिल, वास्तविक दुनिया की घटनाओं के अनुकरण, मॉडलिंग और समझने के लिए, सीरियल कंप्यूटिंग की तुलना में समानांतर कंप्यूटिंग अधिक उपयुक्त है।
- जब स्थानीय संसाधन सीमित हों, तो यह आपको गैर-स्थानीय संसाधनों की तुलना में अधिक लाभ प्रदान कर सकता है।
- ऐसी अनेक समस्याएं हैं जो बहुत बड़ी हैं और उन्हें एक ही कंप्यूटर पर हल करना अव्यावहारिक या असंभव हो सकता है; समानांतर कंप्यूटिंग की अवधारणा इस प्रकार की समस्याओं को दूर करने में मदद करती है।
- समानांतर कंप्यूटिंग का सबसे अच्छा लाभ यह है कि यह आपको एकाधिक कंप्यूटिंग संसाधनों का उपयोग करके एक समय में कई काम करने की अनुमति देता है।
- इसके अलावा, समानांतर कंप्यूटिंग हार्डवेयर के लिए उपयुक्त है क्योंकि सीरियल कंप्यूटिंग संभावित कंप्यूटिंग शक्ति को बर्बाद कर देती है।

Disadvantages of Parallel Computing (Processing)

समानांतर कंप्यूटिंग की कई सीमाएँ हैं, जो इस प्रकार हैं:

- यह समानांतर वास्तुकला को संबोधित करता है जिसे प्राप्त करना कठिन हो सकता है।
- क्लस्टरों के मामले में, समानांतर कंप्यूटिंग में बेहतर शीतलन प्रौद्योगिकियों की आवश्यकता है।
- इसके लिए प्रबंधित एल्गोरिदम की आवश्यकता होती है, जिसे समानांतर तंत्र में संचालित किया जा सकता है।
- मल्टी-कोर आर्किटेक्चर उच्च विद्युत खपत करते हैं।
- समानांतर कंप्यूटिंग प्रणाली को कम युग्मन और उच्च सामंजस्य की आवश्यकता होती है, जिसे बनाना कठिन है।
- समानांतरवाद-आधारित प्रोग्राम के लिए कोड का निर्माण सर्वाधिक तकनीकी रूप से कुशल और विशेषज्ञ प्रोग्रामर द्वारा किया जा सकता है।
- तुल्यकालन, थ्रेड निर्माण, डेटा स्थानांतरण आदि के कारण, अतिरिक्त लागत कभी-कभी काफी बड़ी हो सकती है; यहां तक कि यह समानांतरीकरण के कारण होने वाले लाभ से भी अधिक हो सकती है।
- इसके अलावा, प्रदर्शन में सुधार के लिए, समानांतर कंप्यूटिंग प्रणाली को विभिन्न लक्ष्य आर्किटेक्चर के लिए अलग-अलग कोड ट्वीकिंग की आवश्यकता होती है

Distributed Computing (Processing)

वितरित कंप्यूटिंग एक ऐसी प्रणाली को संदर्भित करता है जहाँ प्रसंस्करण और डेटा भंडारण को एक एकल केंद्रीय डिवाइस द्वारा नियंत्रित किए जाने के बजाय कई डिवाइस या सिस्टम में वितरित किया जाता है। एक वितरित प्रणाली में, प्रत्येक डिवाइस या सिस्टम की अपनी प्रसंस्करण क्षमताएँ होती हैं और वे अपना डेटा संग्रहीत और प्रबंधित भी कर सकते हैं। ये डिवाइस या सिस्टम कार्य करने और संसाधनों को साझा करने के लिए एक साथ काम करते हैं, जिसमें कोई भी डिवाइस केंद्रीय हब के रूप में काम नहीं करता है।

वितरित कंप्यूटिंग सिस्टम का एक उदाहरण क्लाउड कंप्यूटिंग सिस्टम है, जहाँ कंप्यूटिंग पावर, स्टोरेज और नेटवर्किंग जैसे संसाधन इंटरनेट पर वितरित किए जाते हैं और मांग पर एक्सेस किए जाते हैं। इस प्रकार की प्रणाली में, उपयोगकर्ता वेब ब्राउज़र या अन्य क्लाइंट सॉफ़्टवेयर के माध्यम से साझा संसाधनों तक पहुँच सकते हैं और उनका उपयोग कर सकते हैं।

Components

वितरित कंप्यूटिंग सिस्टम के कई प्रमुख घटक हैं

- **उपकरण या प्रणालियाँ:** वितरित प्रणाली में उपकरणों या प्रणालियों की अपनी प्रसंस्करण क्षमताएं होती हैं और वे अपना डेटा संग्रहीत और प्रबंधित भी कर सकते हैं।
- **नेटवर्क:** नेटवर्क वितरित प्रणाली में उपकरणों या प्रणालियों को जोड़ता है, जिससे उन्हें संचार और डेटा का आदान-प्रदान करने की अनुमति मिलती है।
- **संसाधन प्रबंधन:** वितरित प्रणालियों में अक्सर साझा संसाधनों, जैसे कंप्यूटिंग शक्ति, भंडारण और नेटवर्किंग को आवंटित करने और प्रबंधित करने के लिए किसी प्रकार की संसाधन प्रबंधन प्रणाली मौजूद होती है।

वितरित कंप्यूटिंग प्रणाली की वास्तुकला आम तौर पर पीयर-टू-पीयर वास्तुकला होती है, जहां डिवाइस या सिस्टम क्लाइंट और सर्वर दोनों के रूप में कार्य कर सकते हैं और एक दूसरे के साथ सीधे संवाद कर सकते हैं।

विशेषताएँ

वितरित कंप्यूटिंग सिस्टम को परिभाषित करने वाली कई विशेषताएं हैं

- **एकाधिक डिवाइस या प्रणालियाँ:** प्रसंस्करण और डेटा भंडारण एकाधिक डिवाइस या प्रणालियों में वितरित किया जाता है।
- **पीयर-टू-पीयर आर्किटेक्चर:** वितरित प्रणाली में उपकरण या प्रणालियां क्लाइंट और सर्वर दोनों के रूप में कार्य कर सकती हैं, क्योंकि वे नेटवर्क में अन्य उपकरणों या प्रणालियों से सेवाओं का अनुरोध और उन्हें प्रदान दोनों कर सकते हैं।
- **साझा संसाधन:** कंप्यूटिंग शक्ति, भंडारण और नेटवर्किंग जैसे संसाधन नेटवर्क में उपकरणों या प्रणालियों के बीच साझा किए जाते हैं।
- **क्षैतिज स्केलिंग:** वितरित कंप्यूटिंग सिस्टम को स्केल करने में आम तौर पर प्रोसेसिंग और स्टोरेज क्षमता बढ़ाने के लिए नेटवर्क में अधिक डिवाइस या सिस्टम जोड़ना शामिल होता है। यह हार्डवेयर अपग्रेड के माध्यम से या नेटवर्क में अतिरिक्त डिवाइस या सिस्टम जोड़कर किया जा सकता है।

फायदे और नुकसान

वितरित कंप्यूटिंग कई मशीनों में डेटा को संसाधित करने की अनुमति देता है, जो DevOps में अनुप्रयोगों को स्केल करने के लिए महत्वपूर्ण है। क्लाउड वातावरण में वितरित सिस्टम को कैसे तैनात और प्रबंधित किया जाता है, इस पर गहन जानकारी के लिए, **DevOps इंजीनियरिंग - प्लानिंग टू प्रोडक्शन** कोर्स उत्पादन में वितरित कंप्यूटिंग का लाभ उठाने में व्यावहारिक अंतर्दृष्टि प्रदान करता है।

वितरित कंप्यूटिंग प्रणाली के कुछ लाभ हैं:

- **मापनीयता:** वितरित प्रणालियाँ आमतौर पर केंद्रीकृत प्रणालियों की तुलना में अधिक मापनीय होती हैं, क्योंकि वे प्रसंस्करण और भंडारण क्षमता बढ़ाने के लिए नेटवर्क में आसानी से नए उपकरण या प्रणालियाँ जोड़ सकती हैं।
- **विश्वसनीयता:** वितरित प्रणालियाँ अक्सर केंद्रीकृत प्रणालियों की तुलना में अधिक विश्वसनीय होती हैं, क्योंकि वे एक डिवाइस या प्रणाली के खराब हो जाने पर भी काम करना जारी रख सकती हैं।
- **लचीलापन:** वितरित प्रणालियाँ आम तौर पर केंद्रीकृत प्रणालियों की तुलना में अधिक लचीली होती हैं, क्योंकि उन्हें बदलती कंप्यूटिंग आवश्यकताओं को पूरा करने के लिए अधिक आसानी से कॉन्फ़िगर और पुन: कॉन्फ़िगर किया जा सकता है।

वितरित कंप्यूटिंग प्रणाली की कुछ सीमाएँ हैं

- **जटिलता:** वितरित प्रणालियाँ केंद्रीकृत प्रणालियों की तुलना में अधिक जटिल हो सकती हैं, क्योंकि उनमें अनेक डिवाइस या प्रणालियाँ शामिल होती हैं, जिन्हें समन्वित और प्रबंधित करने की आवश्यकता होती है।
- **सुरक्षा:** वितरित प्रणाली को सुरक्षित करना अधिक चुनौतीपूर्ण हो सकता है, क्योंकि संपूर्ण प्रणाली की सुरक्षा सुनिश्चित करने के लिए प्रत्येक डिवाइस या प्रणाली पर सुरक्षा उपायों को लागू किया जाना चाहिए।
- **प्रदर्शन:** वितरित प्रणालियाँ, केंद्रीकृत प्रणालियों के समान प्रदर्शन स्तर की पेशकश नहीं कर सकती हैं, क्योंकि प्रसंस्करण और डेटा भंडारण कई उपकरणों या प्रणालियों में वितरित होता है।

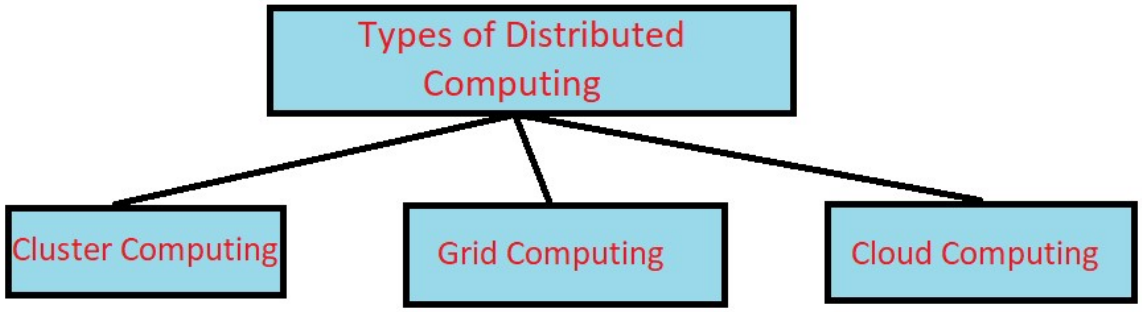
अनुप्रयोग

वितरित कंप्यूटिंग प्रणालियों के कई अनुप्रयोग हैं, जिनमें शामिल हैं:

- **क्लाउड कंप्यूटिंग:** क्लाउड कंप्यूटिंग सिस्टम एक प्रकार की वितरित कंप्यूटिंग प्रणाली है जिसका उपयोग इंटरनेट पर कंप्यूटिंग शक्ति, भंडारण और नेटवर्किंग जैसे संसाधनों को वितरित करने के लिए किया जाता है।
- **पीयर-टू-पीयर नेटवर्क:** पीयर-टू-पीयर नेटवर्क एक प्रकार का वितरित कंप्यूटिंग सिस्टम है जिसका उपयोग उपयोगकर्ताओं के बीच फ़ाइलों और कंप्यूटिंग शक्ति जैसे संसाधनों को साझा करने के लिए किया जाता है।
- **वितरित आर्किटेक्चर:** कई आधुनिक कंप्यूटिंग सिस्टम, जैसे कि माइक्रोसर्विस आर्किटेक्चर, कई डिवाइसों या प्रणालियों में प्रसंस्करण और डेटा भंडारण को वितरित करने के लिए वितरित आर्किटेक्चर का उपयोग करते हैं।

Types Of Distributed Processing

- इसके तीन प्रकार होते है:-



1- Cluster Computing (क्लस्टर कंप्यूटिंग)

क्लस्टर कंप्यूटिंग बहुत से जुड़े हुए कम्पूटरो का एक set है जिसमे सभी कंप्यूटर एक साथ काम करते है ताकि वह single entity के रूप में कार्य कर सके। इसमें जुड़े हुए सभी कंप्यूटर एक साथ किसी कार्य को perform करते है और उसे पूरा करते है। क्लस्टर कंप्यूटिंग में कंप्यूटर को जोड़ने के लिए LAN (लोकल एरिया नेटवर्क) का उपयोग किया जाता है।

Advantages of Cluster Computing

1- इसकी performance काफी अच्छी है।
2- इसे expand किया जा सकता है।
3- यह flexible है।
4- यह scalable है।
5- क्लस्टर कंप्यूटिंग आसानी से मिल जाता है।

Disadvantages of Cluster Computing

1- क्लाउड कंप्यूटिंग काफी महंगा है।
2- इसमें समस्याओ को ढूढ़ना मुश्किल है
3- इसे ज्यादा space की आवश्यकता पड़ती है।

2- Grid computing (ग्रिड कंप्यूटिंग)

ग्रिड कंप्यूटिंग एक प्रकार का सिस्टम है जिसे अक्सर कंप्यूटर सिस्टम के नेटवर्क के रूप में स्थापित (establish) किया जाता है। इसमें प्रत्येक सिस्टम अलग अलग administrative domain से संबंधित होता है। इस कंप्यूटिंग का इस्तेमाल अलग अलग बिज़नेस को आपस में जोड़ने के लिए किया जाता है। इसमें अलग अलग विभाग (department) के पास अलग अलग कंप्यूटर होता है जिसके कारण कंप्यूटर एक दूसरे के साथ कम्यूनिकेट कर पाते है।

Advantages of Grid Computing

1- यह कम समय में जटील (complex) और बड़ी समस्याओ को हल करने में सक्ष्म है।

2- इसमें दुसरे organization के साथ कम्यूनिकेट करना आसान है।

3- इसमें हार्डवेयर का उपयोग अच्छे से किया जा सकता है।

Disadvantages of Grid Computing

1- ग्रिड सॉफ्टवेयर अभी पूरी तरह से विकसित (develop) नहीं हुए है।

2- इसमें कंप्यूटर संसाधनों के बिच तेज़ कनेक्शन की आवश्यकता होती है।

3- Cloud Computing (क्लाउड कंप्यूटिंग)

क्लाउड कंप्यूटिंग एक ऐसी तकनीक है जिसमें data और program को इंटरनेट में स्टोर और एक्सेस किया जाता है। सरल शब्दो में कहे तो, क्लाउड कंप्यूटिंग एक तकनीक है जिसके द्वारा data या information को इन्टरनेट की सहायता से स्टोर, मैनेज और retrieve किया जाता है। इसमें इंटरनेट की होस्ट की गई सेवाएं (services) शामिल है।

Advantages of Cloud Computing

1- इसमें डेटा को स्टोर करना और उसका backup लेना आसान है।

2- इसमें जानकारी को शेयर करना आसान होता है।

3- क्लाउड कंप्यूटिंग में यूजर दुनिया के किसी कोने से जानकारी को एक्सेस कर सकता है।

4- इसमें उपयोग किये जाने वाले सॉफ्टवेयर और हार्डवेयर को मेन्टेन करना आसान है।

Disadvantages of Cloud Computing

1- इसमें यदि यूजर के पास अच्छा इंटरनेट कनेक्शन नहीं है तो डेटा और फाइलों को एक्सेस करना मुश्किल हो जाता है।

2- इसमें अपनी सेवाओं को एक customer से दुसरे customer के पास ट्रांसफर करने के लिए organization या कंपनी को कई समस्याओ का सामना करना पड़ता है।

3- क्लाउड कंप्यूटिंग सुरक्षा के मामले में अच्छा होता है लेकिन इसके बावजूद इसमें डेटा को ट्रांसफर करते वक़्त hackers डेटा की चोरी कर सकते है।

Linux Operating System

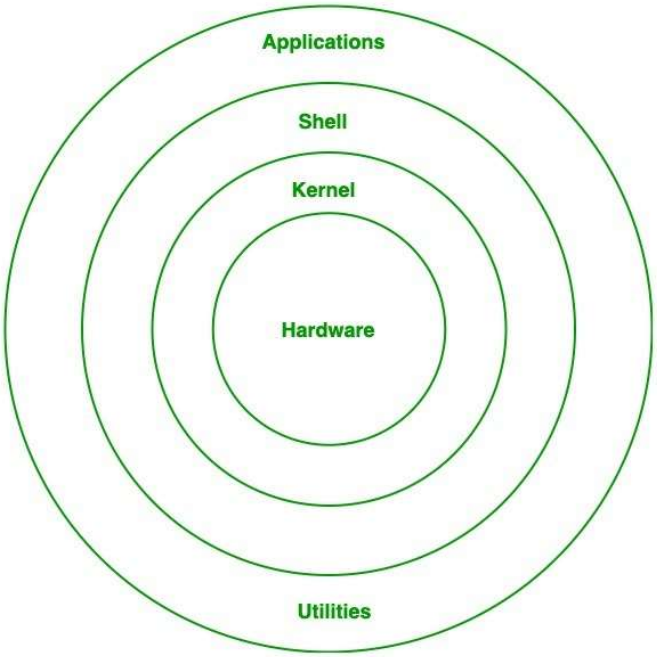
लिनक्स ऑपरेटिंग सिस्टम का एक स्वतंत्र और ओपन-सोर्स परिवार है जो लचीला और लचीला है। 1991 में, लिनस टोरवाल्ड्स नाम के एक व्यक्ति ने इसका निर्माण किया। सिस्टम का सोर्स कोड सभी के लिए सुलभ है, जिसे कोई भी देख सकता है और बदल सकता है, जिससे यह अच्छा लगता है कि कोई भी देख सकता है कि सिस्टम कैसे काम करता है। दुनिया भर के लोगों से एक साथ काम करने और लिनक्स के खुलेपन के कारण इसे विकसित करने का आग्रह किया जाता है। शुरुआत से ही, लिनक्स एक भरोसेमंद और सुरक्षित ओएस के रूप में विकसित हुआ है जिसका उपयोग पीसी, सेल फोन और विशाल सुपरकंप्यूटर सहित कई गैजेट में किया जाता है।

यह लागत प्रभावी होने के लिए प्रसिद्ध है, जिसका अर्थ है कि इसे इस्तेमाल करने में बहुत अधिक खर्च नहीं होता है, और कुशल है, जिसका अर्थ है कि यह बहुत सारे काम जल्दी से पूरा कर सकता है।

1991 में लिनस टोरवाल्ड्स द्वारा विकसित, लिनक्स ऑपरेटिंग सिस्टम एक शक्तिशाली और लचीला ओपन-सोर्स सॉफ़्टवेयर प्लेटफ़ॉर्म है। यह कई तरह के उपकरणों, जैसे एम्बेडेड सिस्टम, सेल फ़ोन, सर्वर और पर्सनल कंप्यूटर के लिए आधार के रूप में कार्य करता है। लिनक्स, जो अपनी विश्वसनीयता, सुरक्षा और लचीलेपन के लिए जाना जाता है, उपयोगकर्ताओं को विशिष्ट आवश्यकताओं के अनुरूप अपने वातावरण को अनुकूलित और बेहतर बनाने की अनुमति देता है। इसका समर्थन करने वाले एक व्यापक और सक्रिय समुदाय के साथ, लिनक्स अपने संसाधनों और निरंतर विकास के कारण लोगों के साथ-साथ कंपनियों के लिए भी एक आकर्षक विकल्प है।

Architecture of Linux

लिनक्स आर्किटेक्चर में निम्नलिखित घटक हैं:



लिनक्स आर्किटेक्चर

1. **कर्नेल:** कर्नेल लिनक्स आधारित ऑपरेटिंग सिस्टम का मूल है। यह प्रत्येक प्रक्रिया को उसके वर्चुअल संसाधन प्रदान करने के लिए कंप्यूटर के सामान्य हार्डवेयर संसाधनों को वर्चुअलाइज़ करता है। इससे प्रक्रिया ऐसी प्रतीत होती है जैसे कि यह मशीन पर चलने वाली एकमात्र प्रक्रिया है। कर्नेल विभिन्न प्रक्रियाओं के बीच संघर्ष को रोकने और कम करने के लिए भी जिम्मेदार है। कर्नेल के विभिन्न प्रकार हैं:
   - मोनोलिथिक कर्नेल
   - संकर गुठली
   - एक्सो कर्नेल
   - माइक्रो कर्नेल

2. **सिस्टम लाइब्रेरी:** लिनक्स ऑपरेटिंग सिस्टम की विभिन्न कार्यक्षमताओं को लागू करने के लिए सिस्टम लाइब्रेरी का उपयोग करता है, जिसे साझा लाइब्रेरी के रूप में भी जाना जाता है। इन लाइब्रेरी में पहले से लिखा हुआ कोड होता है जिसका उपयोग एप्लिकेशन विशिष्ट कार्यों को करने के लिए कर सकते हैं। इन लाइब्रेरी का उपयोग करके, डेवलपर्स समय और प्रयास बचा सकते हैं, क्योंकि उन्हें बार-बार एक ही कोड लिखने की आवश्यकता नहीं होती है। सिस्टम लाइब्रेरी एप्लिकेशन और कर्नेल के बीच एक इंटरफ़ेस के रूप में कार्य करती हैं, जो एप्लिकेशन को अंतर्निहित सिस्टम के साथ बातचीत करने के लिए एक मानकीकृत और कुशल तरीका प्रदान करती हैं।
3. **शेल:** शेल लिनक्स ऑपरेटिंग सिस्टम का यूजर इंटरफेस है। यह उपयोगकर्ताओं को कमांड दर्ज करके सिस्टम के साथ इंटरैक्ट करने की अनुमति देता है, जिसे शेल व्याख्या करता है और निष्पादित करता है। शेल उपयोगकर्ता और कर्नेल के बीच एक पुल के रूप में कार्य करता है, उपयोगकर्ता के अनुरोधों को प्रसंस्करण के लिए कर्नेल को अग्रेषित करता है। यह उपयोगकर्ताओं को विभिन्न कार्य करने के लिए एक सुविधाजनक तरीका प्रदान करता है, जैसे प्रोग्राम चलाना, फ़ाइलों का प्रबंधन करना और सिस्टम को कॉन्फ़िगर करना।
4. **हार्डवेयर परत:** हार्डवेयर परत में कंप्यूटर के सभी भौतिक घटक शामिल होते हैं, जैसे RAM (रैंडम एक्सेस मेमोरी), HDD (हार्ड डिस्क ड्राइव), CPU (सेंट्रल प्रोसेसिंग यूनिट), और इनपुट/आउटपुट डिवाइस। यह परत लिनक्स ऑपरेटिंग सिस्टम के साथ बातचीत करने और सिस्टम और एप्लिकेशन को ठीक से काम करने के लिए आवश्यक संसाधन प्रदान करने के लिए जिम्मेदार है। लिनक्स कर्नेल और सिस्टम लाइब्रेरी इन हार्डवेयर घटकों पर संचार और नियंत्रण सक्षम करते हैं, यह सुनिश्चित करते हुए कि वे एक साथ सामंजस्यपूर्ण रूप से काम करते हैं।
5. **सिस्टम यूटिलिटी:** सिस्टम यूटिलिटीज लिनक्स ऑपरेटिंग सिस्टम द्वारा सिस्टम के विभिन्न पहलुओं को प्रबंधित और कॉन्फ़िगर करने के लिए प्रदान किए जाने वाले आवश्यक उपकरण और प्रोग्राम हैं। ये यूटिलिटीज सॉफ्टवेयर इंस्टॉल करने, नेटवर्क सेटिंग कॉन्फ़िगर करने, सिस्टम के प्रदर्शन की निगरानी करने, उपयोगकर्ताओं और अनुमतियों को प्रबंधित करने और बहुत कुछ जैसे कार्य करती हैं। सिस्टम यूटिलिटीज सिस्टम एडमिनिस्ट्रेशन कार्यों को सरल बनाती हैं, जिससे उपयोगकर्ताओं के लिए अपने लिनक्स सिस्टम को कुशलतापूर्वक बनाए रखना आसान हो जाता है।

Advantages of Linux

- लिनक्स का मुख्य लाभ यह है कि यह एक ओपन-सोर्स ऑपरेटिंग सिस्टम है। इसका मतलब है कि सोर्स कोड सभी के लिए आसानी से उपलब्ध है और आप बिना किसी अनुमति के कोड में योगदान, संशोधन और वितरण कर सकते हैं।
- सुरक्षा के मामले में, लिनक्स किसी भी अन्य ऑपरेटिंग सिस्टम की तुलना में अधिक सुरक्षित है। इसका मतलब यह नहीं है कि लिनक्स 100 प्रतिशत सुरक्षित है, इसके लिए कुछ मैलवेयर हैं लेकिन यह किसी भी अन्य ऑपरेटिंग सिस्टम की तुलना में कम असुरक्षित है। इसलिए, इसे किसी एंटी-वायरस सॉफ़्टवेयर की आवश्यकता नहीं है।
- लिनक्स में सॉफ्टवेयर अपडेट आसान और लगातार होते हैं।
- विभिन्न लिनक्स वितरण उपलब्ध हैं ताकि आप उन्हें अपनी आवश्यकताओं या अपनी पसंद के अनुसार उपयोग कर सकें।
- लिनक्स इंटरनेट पर उपयोग के लिए स्वतंत्र रूप से उपलब्ध है।

- इसे बड़े पैमाने पर सामुदायिक समर्थन प्राप्त है।
- यह उच्च स्थिरता प्रदान करता है। यह शायद ही कभी धीमा या रुकता है और थोड़े समय के बाद इसे रीबूट करने की आवश्यकता नहीं होती है।
- यह उपयोगकर्ता की गोपनीयता बनाए रखता है।
- लिनक्स सिस्टम का प्रदर्शन अन्य ऑपरेटिंग सिस्टम की तुलना में बहुत अधिक है। यह एक ही समय में बड़ी संख्या में लोगों को काम करने की अनुमति देता है और उन्हें कुशलतापूर्वक संभालता है।
- यह नेटवर्क अनुकूल है।
- लिनक्स का लचीलापन बहुत अधिक है। पूरा लिनक्स सूट इंस्टॉल करने की आवश्यकता नहीं है; आपको केवल आवश्यक घटक ही इंस्टॉल करने की अनुमति है।

Disadvantages of Linux

- यह बहुत उपयोगकर्ता-अनुकूल नहीं है। इसलिए, यह शुरुआती लोगों के लिए भ्रामक हो सकता है।
- इसमें विंडोज़ की तुलना में छोटे परिधीय हार्डवेयर ड्राइवर हैं।

Linux File System

ऑपरेटिंग सिस्टम, वह सॉफ़्टवेयर जो आपके कंप्यूटर को शक्ति प्रदान करता है, फ़ाइल सिस्टम नामक एक महत्वपूर्ण तत्व पर निर्भर करता है। इसे एक आभासी संगठनात्मक उपकरण के रूप में सोचें जो आपके डेटा को कुशलतापूर्वक प्रबंधित, संग्रहीत और पुनर्प्राप्त करता है। लिनक्स की दुनिया में, फ़ाइल सिस्टम की एक विविध श्रेणी उभरी है, जिनमें से प्रत्येक को विशिष्ट आवश्यकताओं और प्राथमिकताओं को संबोधित करने के लिए तैयार किया गया है। इस लेख का उद्देश्य लिनक्स फ़ाइल सिस्टम की पेचीदगियों को सरल बनाना है, शुरुआती लोगों को उनकी परतों, विशेषताओं और कार्यान्वयन के माध्यम से मार्गदर्शन करना है। इन बारीकियों पर प्रकाश डालकर, हम उपयोगकर्ताओं को लिनक्स ऑपरेटिंग सिस्टम के गतिशील परिदृश्य को नेविगेट करने में सूचित विकल्प बनाने के लिए सशक्त बनाते हैं।

What is the Linux File System

लिनक्स फ़ाइल सिस्टम एक बहुआयामी संरचना है जिसमें तीन आवश्यक परतें शामिल हैं। इसकी नींव में, लॉजिकल फ़ाइल सिस्टम उपयोगकर्ता अनुप्रयोगों और फ़ाइल सिस्टम के बीच इंटरफ़ेस के रूप में कार्य करता है, जो फ़ाइलों को खोलने, पढ़ने और बंद करने जैसे कार्यों का प्रबंधन करता है। इसके ऊपर, वर्चुअल फ़ाइल सिस्टम कई भौतिक फ़ाइल सिस्टम के समवर्ती संचालन की सुविधा प्रदान करता है, जो संगतता के लिए एक मानकीकृत इंटरफ़ेस प्रदान करता है। अंत में, भौतिक फ़ाइल सिस्टम डिस्क पर भौतिक मेमोरी ब्लॉकों के ठोस प्रबंधन और भंडारण के लिए जिम्मेदार है, जो कुशल डेटा आवंटन और पुनर्प्राप्ति सुनिश्चित करता है। साथ में, ये परतें एक सुसंगत वास्तुकला बनाती हैं, जो लिनक्स ऑपरेटिंग सिस्टम में डेटा के संगठित और कुशल संचालन को व्यवस्थित करती हैं।

Linux File System Structure

एक फ़ाइल सिस्टम में मुख्यतः तीन परतें होती हैं। ऊपर से नीचे तक:

1. तार्किक फ़ाइल सिस्टम:

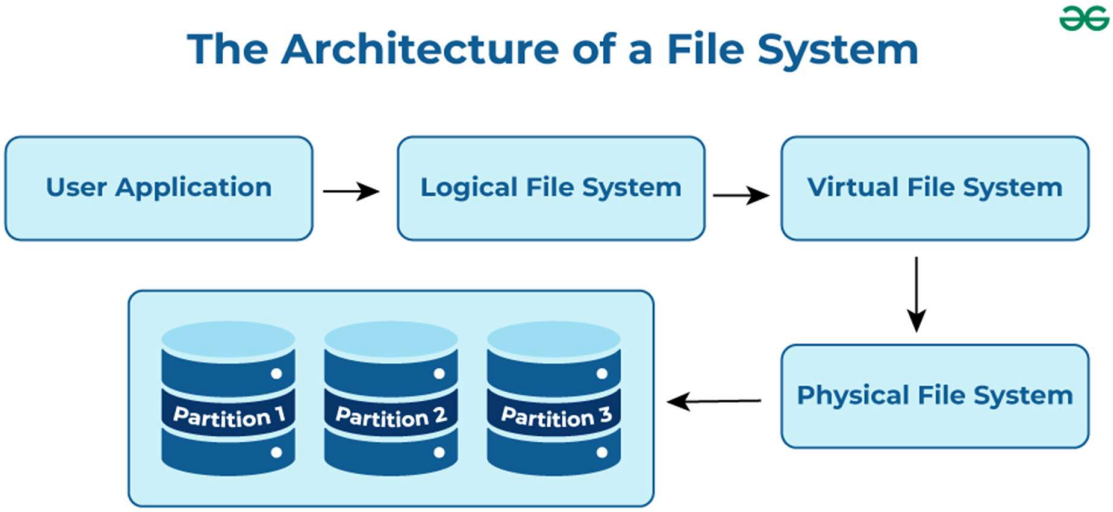
लॉजिकल फ़ाइल सिस्टम उपयोगकर्ता अनुप्रयोगों और फ़ाइल सिस्टम के बीच इंटरफ़ेस के रूप में कार्य करता है। यह फ़ाइलों को खोलने, पढ़ने और बंद करने जैसे आवश्यक कार्यों को सुविधाजनक बनाता है। अनिवार्य रूप से, यह उपयोगकर्ता के अनुकूल फ्रंट-एंड के रूप में कार्य करता है, यह सुनिश्चित करता है कि अनुप्रयोग फ़ाइल सिस्टम के साथ इस तरह से बातचीत कर सकें जो उपयोगकर्ता की अपेक्षाओं के अनुरूप हो।

2. वर्चुअल फ़ाइल सिस्टम:

वर्चुअल फ़ाइल सिस्टम (VFS) एक महत्वपूर्ण परत है जो भौतिक फ़ाइल सिस्टम के कई उदाहरणों के समवर्ती संचालन को सक्षम बनाती है। यह एक मानकीकृत इंटरफ़ेस प्रदान करता है, जिससे विभिन्न फ़ाइल सिस्टम एक साथ सह-अस्तित्व में रह सकते हैं और एक साथ काम कर सकते हैं। यह परत अंतर्निहित जटिलताओं को दूर करती है, विभिन्न फ़ाइल सिस्टम कार्यान्वयनों के बीच संगतता और सामंजस्य सुनिश्चित करती है।

3. भौतिक फ़ाइल सिस्टम:

भौतिक फ़ाइल सिस्टम डिस्क पर भौतिक मेमोरी ब्लॉक के ठोस प्रबंधन और भंडारण के लिए जिम्मेदार है। यह हार्डवेयर घटकों के साथ सीधे बातचीत करते हुए डेटा को संग्रहीत करने और पुनर्प्राप्त करने के निम्न-स्तरीय विवरणों को संभालता है। यह परत भौतिक भंडारण संसाधनों के कुशल आवंटन और उपयोग को सुनिश्चित करती है, जो फ़ाइल सिस्टम के समग्र प्रदर्शन और विश्वसनीयता में योगदान देती है।



फ़ाइल सिस्टम की वास्तुकला

फ़ाइल सिस्टम की विशेषताएँ

- **स्पेस मैनेजमेंट** : स्टोरेज डिवाइस पर डेटा कैसे स्टोर किया जाता है। मेमोरी ब्लॉक और उसमें लागू विखंडन प्रथाओं से संबंधित।
- **फ़ाइल नाम** : फ़ाइल सिस्टम में फ़ाइल नाम पर कुछ प्रतिबंध हो सकते हैं जैसे नाम की लंबाई, विशेष वर्णों का उपयोग, और केस संवेदनशीलता।

- **निर्देशिका** : निर्देशिकाएं/फ़ोल्डर फ़ाइलों को रैखिक या श्रेणीबद्ध तरीके से संग्रहीत कर सकते हैं, जबकि उस निर्देशिका या उपनिर्देशिका में निहित सभी फ़ाइलों की एक अनुक्रमणिका तालिका बनाए रख सकते हैं।
- **मेटाडेटा** : संग्रहीत प्रत्येक फ़ाइल के लिए, फ़ाइल सिस्टम उस फ़ाइल के अस्तित्व के बारे में विभिन्न जानकारी संग्रहीत करता है जैसे कि इसकी डेटा लंबाई, इसकी पहुँच अनुमतियाँ, डिवाइस प्रकार, संशोधित दिनांक-समय और अन्य विशेषताएँ। इसे मेटाडेटा कहा जाता है।
- **उपयोगिताएँ** : फ़ाइल सिस्टम फ़ाइलों और फ़ोल्डरों को आरंभ करने, हटाने, नाम बदलने, स्थानांतरित करने, प्रतिलिपि बनाने, बैकअप लेने, पुनर्प्राप्ति और उन तक पहुंच को नियंत्रित करने के लिए सुविधाएँ प्रदान करते हैं।
- **डिजाइन** : उनके कार्यान्वयन के कारण, फ़ाइल सिस्टम में संग्रहीत डेटा की मात्रा पर सीमाएं होती हैं।

कुछ महत्वपूर्ण शब्द:

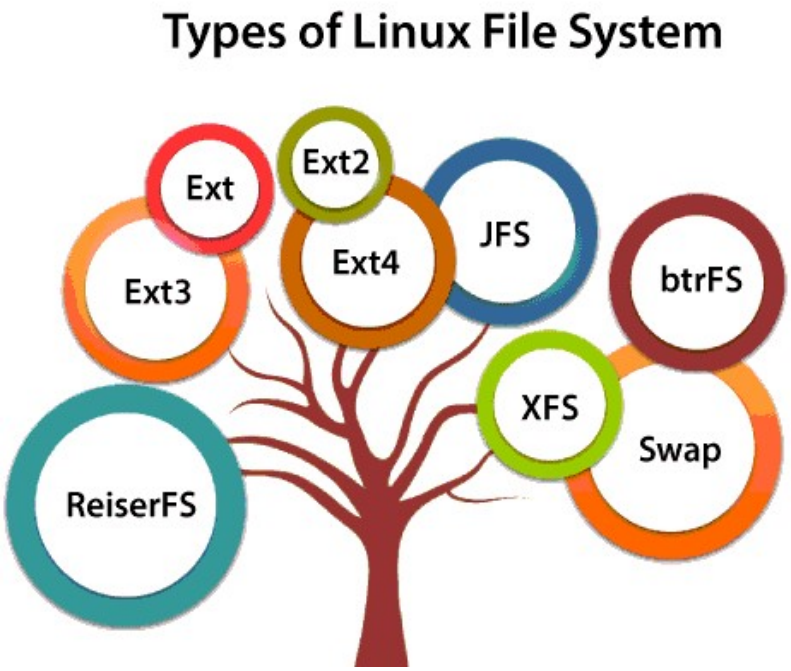
1) जर्नलिंग: जर्नलिंग फ़ाइल सिस्टम जर्नल नामक एक लॉग रखता है, जो फ़ाइल में किए गए परिवर्तनों का ट्रैक रखता है, लेकिन अभी तक डिस्क पर स्थायी रूप से प्रतिबद्ध नहीं है, ताकि सिस्टम विफलता के मामले में खोए गए परिवर्तनों को वापस लाया जा सके।

2) संस्करण: संस्करण फ़ाइल सिस्टम किसी फ़ाइल के पहले से सहेजे गए संस्करणों को संग्रहीत करता है, अर्थात, बैकअप बनाने के लिए फ़ाइल की प्रतियां डिस्क पर पिछले कमिट के आधार पर प्रति मिनट या प्रति घंटे के हिसाब से संग्रहीत की जाती हैं।

3) इनोड: इंडेक्स नोड किसी भी फ़ाइल या निर्देशिका का प्रतिनिधित्व करता है जो मापदंडों - फ़ाइल और निर्देशिका के आकार, अनुमति, स्वामित्व और स्थान पर आधारित होता है।

अब हम उस भाग पर आते हैं जहां हम डिस्क भंडारण उपकरणों के लिए लिनक्स में फ़ाइल सिस्टम के विभिन्न कार्यान्वयनों पर चर्चा करते हैं।

लिनक्स फ़ाइल सिस्टम:



1) ext (विस्तारित फ़ाइल सिस्टम) : 1992 में लागू किया गया यह पहला फ़ाइल सिस्टम है जिसे विशेष रूप से Linux के लिए डिज़ाइन किया गया है। यह फ़ाइल सिस्टम के ext परिवार का पहला सदस्य है।

2) ext2 : दूसरा एक्सट 1993 में विकसित किया गया था। यह एक गैर-जर्नलिंग फ़ाइल सिस्टम है जिसे फ्लैश ड्राइव और एसएसडी के साथ उपयोग करना पसंद किया जाता है। इसने एक्सेस, इनोड संशोधन और डेटा संशोधन के लिए अलग-अलग टाइमस्टैम्प की समस्याओं को हल किया। जर्नल न होने के कारण, यह बूट समय पर लोड होने में धीमा है।

3) ज़ियाफ़्स : 1993 में विकसित यह फ़ाइल सिस्टम ext2 की तुलना में कम शक्तिशाली और कार्यात्मक था और अब कहीं भी उपयोग में नहीं है।

4) ext3 : 1999 में विकसित तीसरा एक्सट एक जर्नलिंग फ़ाइल सिस्टम है। यह विश्वसनीय है और एक्सट2 के विपरीत, यह सिस्टम बूट में लंबी देरी को रोकता है यदि फ़ाइल सिस्टम अशुद्ध शटडाउन के बाद असंगत स्थिति में है। अन्य कारक जो इसे एक्सट2 से बेहतर और अलग बनाते हैं, वे हैं ऑनलाइन फ़ाइल सिस्टम ग्रोथ और बड़ी निर्देशिकाओं के लिए HTree इंडेक्सिंग।

5) जेएफएस (जर्नल फाइल सिस्टम) : 1990 में IBM द्वारा पहली बार निर्मित, मूल JFS को 1999 में Linux के लिए क्रियान्वित करने हेतु ओपन सोर्स में ले जाया गया था। JFS विभिन्न प्रकार के लोड के तहत अच्छा प्रदर्शन करता है, लेकिन 2006 में ext4 के जारी होने के कारण अब इसका उपयोग आम तौर पर नहीं किया जाता है, जो बेहतर प्रदर्शन देता है।

6) रीसरएफएस : यह 2001 में विकसित एक जर्नल फ़ाइल सिस्टम है। इसके पहले के मुद्दों के बावजूद, इसमें आंतरिक विखंडन को कम करने की योजना के रूप में टेल पैकिंग है । यह एक B+ ट्री का उपयोग करता है जो निर्देशिका लुकअप और अपडेट में रैखिक समय से कम देता है। यह SUSE Linux में संस्करण 6.4 तक डिफ़ॉल्ट फ़ाइल सिस्टम था, 2006 में संस्करण 10.2 के लिए ext3 पर स्विच करने तक।

7) एक्सएफएस : XFS एक 64-बिट जर्नलिंग फ़ाइल सिस्टम है और इसे 2001 में Linux में पोर्ट किया गया था। यह अब कई Linux डिस्ट्रीब्यूशन के लिए डिफ़ॉल्ट फ़ाइल सिस्टम के रूप में कार्य करता है। यह स्नैपशॉट, ऑनलाइन डीफ़्रैगमेंटेशन, विरल फ़ाइलें, परिवर्तनशील ब्लॉक आकार और उत्कृष्ट क्षमता जैसी सुविधाएँ प्रदान करता है। यह समानांतर I/O संचालन में भी उत्कृष्ट है।

8) स्क्वैशएफएस : 2002 में विकसित यह फाइल सिस्टम केवल पढ़ने के लिए है और इसका उपयोग केवल एम्बेडेड सिस्टम के साथ किया जाता है जहां कम ओवरहेड की आवश्यकता होती है।

9) रीसर4 : यह ReiserFS का एक वृद्धिशील मॉडल है। इसे 2004 में विकसित किया गया था। हालाँकि, यह कई लिनक्स वितरणों पर व्यापक रूप से अनुकूलित या समर्थित नहीं है।

10) एक्सटी4 : 2006 में विकसित चौथा ext एक जर्नलिंग फ़ाइल सिस्टम है। इसमें ext3 और ext2 के साथ पश्चगामी संगतता है और यह कई अन्य सुविधाएँ प्रदान करता है, जिनमें से कुछ हैं लगातार पूर्व-आवंटन, उपनिर्देशिकाओं की असीमित संख्या, मेटाडेटा चेकसमिंग और बड़ी फ़ाइल आकार। ext4 कई लिनक्स वितरणों के लिए डिफ़ॉल्ट फ़ाइल सिस्टम है और यह विंडोज और मैकिन्टोश के साथ भी संगतता रखता है।

11) btrfs (बेटर/बटर/बी-ट्री एफएस) : इसे 2007 में विकसित किया गया था। यह स्नैपशॉटिंग, ड्राइव पूलिंग, डेटा स्क्रबिंग, सेल्फ-हीलिंग और ऑनलाइन डीफ़्रैगमेंटेशन जैसी कई सुविधाएँ प्रदान करता है। यह फेडोरा वर्कस्टेशन के लिए डिफ़ॉल्ट फ़ाइल सिस्टम है।

12) बीकैशफ्स: यह एक कॉपी-ऑन-राइट फ़ाइल सिस्टम है जिसे पहली बार 2015 में btrfs और ext4 से बेहतर प्रदर्शन करने के लक्ष्य के साथ घोषित किया गया था। इसकी विशेषताओं में पूर्ण फ़ाइल सिस्टम एन्क्रिप्शन, मूल संपीड़न, स्नैपशॉट और 64-बिट चेक समिंग शामिल हैं।

13) अन्य : लिनक्स में NTFS और exFAT जैसे ऑपरेटिंग सिस्टम के फ़ाइल सिस्टम के लिए भी समर्थन है, लेकिन ये मानक यूनिक्स अनुमति सेटिंग्स का समर्थन नहीं करते हैं। इनका उपयोग ज़्यादातर दूसरे ऑपरेटिंग सिस्टम के साथ अंतरसंचालनीयता के लिए किया जाता है।

Linux Kernel

लिनक्स ***कर्नेल*** एक ओपन-सोर्स, मुफ़्त, मल्टीटास्किंग, मॉड्यूलर, मोनोलिथिक और यूनिक्स जैसा ओएस कर्नेल है। 1991 में, इसे मूल रूप से ***लिनस टोरवाल्ड्स*** ने अपने i386-आधारित पीसी के लिए शुरू किया था और जल्द ही इसे कर्नेल के रूप में GNU ऑपरेटिंग सिस्टम के लिए अपनाया गया था, जिसे यूनिक्स के लिए एक लिब्रे (मुफ़्त) प्रतिस्थापन के रूप में लिखा गया था।

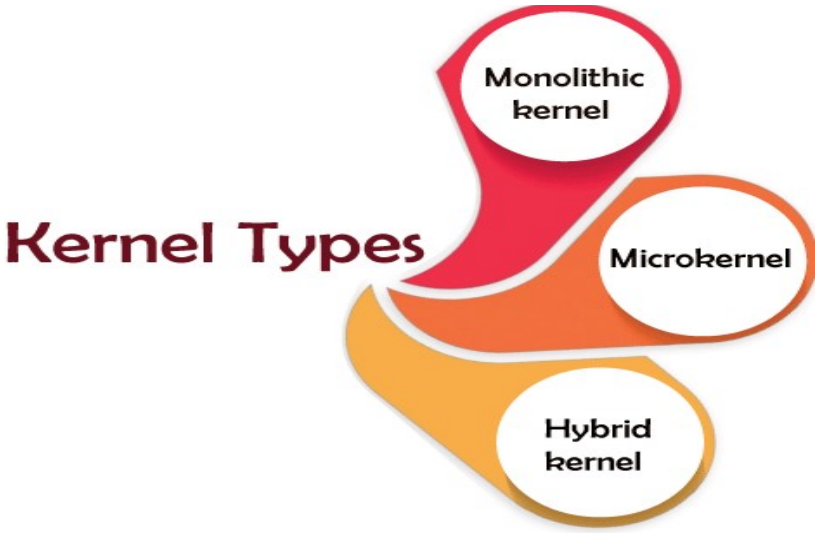
लिनक्स केवल GNU जनरल पब्लिक लाइसेंस v2 के अंतर्गत दिया जाता है, लेकिन इसमें अन्य संगत लाइसेंसों में फ़ाइलें शामिल हैं। इसे 1990 के अंत से कई ऑपरेटिंग सिस्टम वितरणों के हिस्से के रूप में जोड़ा गया है, जिनमें से कई को आम तौर पर लिनक्स के रूप में भी जाना जाता है।

- लिनक्स का उपयोग विभिन्न प्रकार की कंप्यूटिंग प्रणालियों पर किया जाता है, जैसे सुपर कंप्यूटर, मेनफ्रेम, सर्वर, पर्सनल कंप्यूटर, मोबाइल डिवाइस और एम्बेडेड डिवाइस।
- इसे आसान कमांडों के परिवार के साथ कई उपयोग परिदृश्यों और आर्किटेक्चर के लिए अनुकूलित किया जा सकता है; विशेषाधिकार प्राप्त उपयोगकर्ता रनटाइम के दौरान कर्नेल पैरामीटर्स को भी ठीक कर सकते हैं।
- लिनक्स कर्नेल का लगभग हर कोड C प्रोग्रामिंग भाषा के GCC GNU एक्सटेंशन के साथ लिखा गया है।
- यह कार्य निष्पादन समय और मेमोरी स्थान का उपयोग करने के लिए एक अत्यधिक ***vmlinux (अनुकूलित निष्पादन योग्य) उत्पन्न करता है।***
- दिन-प्रतिदिन विकास चर्चाएँ ***LKML*** (लिनक्स कर्नेल मेलिंग सूची) पर विचार करती हैं।
- संशोधनों को गिट संस्करण नियंत्रण प्रणाली के साथ ट्रैक किया जाता है, जिसे मूल रूप से टोरवाल्ड्स द्वारा ***बिटकीपर*** के लिए एक मुफ्त सॉफ्टवेयर प्रतिस्थापन के रूप में शुरू किया गया था।

DSG Support Multi Solution

Kernel Types

The kernel has three types in general, which are listed and explained below:



मोनोलिथिक कर्नेल

यह OSes द्वारा व्यापक रूप से उपयोग किया जाने वाला कर्नेल है। कर्नेल कई मॉड्यूल से बना होता है जिन्हें मोनोलिथिक आर्किटेक्चर में गतिशील रूप से लोड और अनलोड किया जा सकता है। इस प्रकार की आर्किटेक्चर ऑपरेटिंग सिस्टम की क्षमताओं को बढ़ाएगी और कर्नेल के लिए आसान एक्सटेंशन की अनुमति देगी। मोनोलिथिक आर्किटेक्चर के साथ कर्नेल रखरखाव सुविधाजनक हो जाता है क्योंकि यह किसी विशिष्ट मॉड्यूल में बग को हल करने की आवश्यकता होने पर संबंधित मॉड्यूल को लोड और अनलोड करने की अनुमति देता है।

माइक्रोकर्नेल

माइक्रोकर्नेल को मोनोलिथिक कर्नेल के प्रतिस्थापन के रूप में विकसित किया गया है ताकि लगातार बढ़ते कर्नेल कोड आकार की समस्या को परिभाषित किया जा सके, जिसे मोनोलिथिक कर्नेल करने में असफल रहा। यह कुछ बुनियादी सेवाओं, जैसे फ़ाइल सिस्टम, डिवाइस ड्राइवर प्रबंधन, प्रोटोकॉल स्टैक, आदि को यूजरस्पेस में निष्पादित करने की अनुमति देता है। यह बेहतर सुरक्षा और न्यूनतम कोड के साथ ऑपरेटिंग सिस्टम की क्षमता को बढ़ा सकता है और स्थिरता सुनिश्चित करता है।

यह सिस्टम के बाकी हिस्सों को बिना किसी रुकावट के सही ढंग से काम करने के लिए प्रेरित करके प्रभावित क्षेत्रों को होने वाले नुकसान को कम करता है। ओएस की हर बुनियादी सेवा IPC (इंटरप्रोसेस कम्युनिकेशन) द्वारा प्रोग्राम के लिए होती है। माइक्रोकर्नेल आर्किटेक्चर में। यह हार्डवेयर और डिवाइस ड्राइवरों के बीच सीधे संपर्क की अनुमति देता है।

हाइब्रिड कर्नेल

यह निर्धारित कर सकता है कि वह पर्यवेक्षक मोड और उपयोगकर्ता मोड में क्या निष्पादित करना चाहता है। डिवाइस ड्राइवरों की तरह, हाइब्रिड कर्नेल वातावरण में, फ़ाइल सिस्टम I/O उपयोगकर्ता मोड में निष्पादित होगा, हालाँकि IPC और सर्वर कॉल पर्यवेक्षक मोड में रहते हैं।

DSG Support Multi Solution

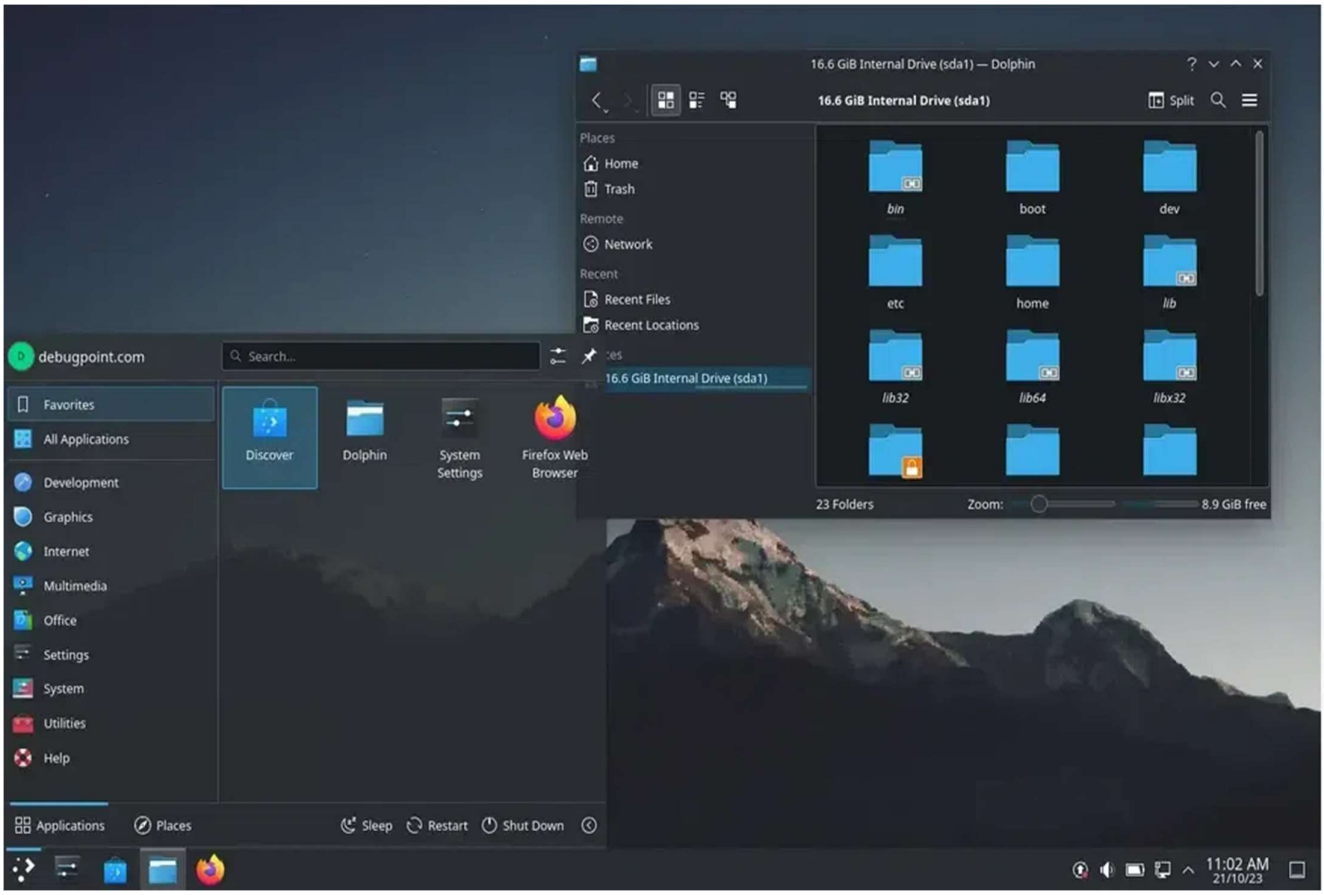
KDE & Gnome Interface

Desktop Environments :- **डेस्कटॉप वातावरण एक ग्राफिकल यूजर इंटरफेस (GUI) है जो लिनक्स कर्नेल के ऊपर स्थित होता है।** इसमें टास्कबार, विंडो, लॉन्चर मेनू आदि जैसे घटक शामिल हैं, साथ ही समग्र वर्कफ़्लो प्रबंधन भी शामिल है। KDE और GNOME दो व्यापक रूप से उपयोग किए जाने वाले DE हैं जो दर्शन और सुविधा समृद्धि में भिन्न हैं।

Types Of Desktop Enviorments

1. **KDE**
2. **GNOME**

1. KDE :- लिनक्स डेस्कटॉप के लिए, ओपन-सोर्स **KDE (K डेस्कटॉप एनवायरनमेंट)** प्रोजेक्ट एक शक्तिशाली ग्राफिकल यूजर इंटरफेस (GUI) प्रदान करता है। तथ्य यह है कि थीम, लेआउट और यहां तक कि प्रोग्राम व्यवहार को अनुकूलित किया जा सकता है, जो इसे उन व्यक्तियों के लिए आदर्श बनाता है जो निजीकरण का आनंद लेते हैं। अपनी कई विशेषताओं, जीवंत डेवलपर समुदाय और आविष्कारशील प्रकृति के कारण, KDE को **Kubuntu** जैसे वितरण पर अत्यधिक पसंद किया जाता है ।



2. GNOME :- **GNOME (GNU Network Object Model Environment) एक** निःशुल्क सॉफ्टवेयर है, जिसकी विशेषता इसकी सरलता और आकर्षण है। डेवलपर्स ने इसे अच्छा दिखाने के लिए हर संभव प्रयास किया; इसलिए वे फ़ॉन्ट रेंडरिंग या आइकन डिज़ाइन जैसे विवरणों पर ध्यान देते हैं। KDE के विपरीत जो पावर उपयोगकर्ताओं या उत्साही लोगों के लिए अंतहीन अनुकूलन विकल्पों की अनुमति देता है, GNOME एक सरल नियम का पालन करता है: कम ही अधिक है! यही कारण है कि Gnome शेल बुनियादी कार्यक्षमता को बढ़ाने वाले न्यूनतम लेकिन पर्याप्त संख्या में एक्सटेंशन प्रदान करता है। डिफ़ॉल्ट रूप से Ubuntu यूनिटी का उपयोग करता है, लेकिन संस्करण **18.04 LTS** से Gnome को भी अपनाया है जबकि **Fedora वर्कस्टेशन** संस्करण स्टॉक Gnome इंस्टॉलेशन के साथ आता है।

KDE vs GNOME: A Detailed Breakdown by Aspect

यहां विभिन्न व्यक्तिगत पहलुओं में **केडीई और गनोम की स्थिति पर करीब से नजर डाली गई है:**

1. अनुकूलन

केडीई: अनुकूलन के लिए कई विकल्प प्रदान करता है। ***कीबोर्ड शॉर्टकट, गतिविधियाँ (वर्चुअल डेस्कटॉप), लेआउट, विजेट, पैनल*** और ***थीम*** जैसे कई तत्वों को संशोधित करना संभव है। नए उपयोगकर्ताओं को अनुकूलन की यह डिग्री चुनौतीपूर्ण लग सकती है, लेकिन यह अत्यधिक अनुकूलित अनुभव को सक्षम बनाता है।

GNOME: अधिक सुव्यवस्थित दृष्टिकोण प्रदान करता है। थीम और एक्सटेंशन समर्थन कुछ वैयक्तिकरण की अनुमति देता है, लेकिन KDE के समान सीमा तक नहीं। व्यक्तिगत उपयोगकर्ता के बदलावों पर कम जोर देते हुए पूर्व-निर्धारित वर्कफ़्लो पर ध्यान केंद्रित करता है।

2. अनुप्रयोग

केडीई: केडीई ऐप सूट में कई तरह के ऐप शामिल हैं, जिनमें ***एडवांस्ड टेक्स्ट एडिटर केट, फाइल मैनेजर डॉल्फिन, पिक्चर व्यूअर ग्वेनव्यू, डिजिटल पेंटर क्रिटा*** और ***टर्मिनल एमुलेटर कोनसोल*** शामिल हैं। इसमें गनोम की तुलना में ऐप का अधिक चयन है।

गनोम: गनोम एप्लीकेशन सूट में गनोम नामक एक टेक्स्ट एडिटर, ***फोटो नामक एक इमेज व्यूअर***, ***फाइल्स नामक एक फाइल ऑर्गनाइजर***, ***टर्मिनल नामक एक टर्मिनल एमुलेटर*** और बहुत कुछ शामिल है। क्योंकि प्रोग्राम गनोम के डिजाइन सिद्धांतों का पालन करते हैं, इसलिए उनका लुक और फील एक जैसा होता है। केडीई की तुलना में, इसमें कम एप्लीकेशन होते हैं, लेकिन यह क्लोज डेस्कटॉप एनवायरनमेंट इंटीग्रेशन पर अधिक जोर देता है।

3. प्रदर्शन

केडीई:

ऐतिहासिक रूप से, **KDE को संसाधन-गहन होने की प्रतिष्ठा प्राप्त थी।** हालाँकि, हाल ही में किए गए अनुकूलन ने इसकी दक्षता में उल्लेखनीय सुधार किया है। इसे अभी भी GNOME की तुलना में थोड़ा अधिक शक्तिशाली हार्डवेयर की आवश्यकता हो सकती है, खासकर सहज एनिमेशन और जटिल थीम के लिए।

केडीई प्रदर्शन को बेहतर बनाने के लिए विकल्प प्रदान करता है, जैसे कि कुछ विजेट या प्रभाव को अक्षम करना। उन्नत ज्ञान वाले उपयोगकर्ता अपने विशिष्ट हार्डवेयर के लिए डेस्कटॉप को और अधिक अनुकूलित कर सकते हैं।

गनोम:

अपने हल्के वजन और संसाधनों के कुशल उपयोग के लिए जाना जाने वाला **GNOME पुराने या कम शक्तिशाली कंप्यूटरों पर अच्छी तरह से चलता है** । यह इसे सीमित हार्डवेयर वाले लोगों के लिए एक अच्छा विकल्प बनाता है।

गनोम स्वच्छ और प्रतिक्रियाशील अनुभव को प्राथमिकता देता है, जिसमें न्यूनतम एनिमेशन और प्रभाव होते हैं जो प्रदर्शन को प्रभावित कर सकते हैं।

4. उपयोग में आसानी

केडीई:

शुरुआती लोगों के लिए यह भारी लग सकता है क्योंकि इसमें बहुत सारी सुविधाएँ और कस्टमाइज़ करने के तरीके हैं। आपको तब तक चीज़ों के साथ खेलना होगा जब तक कि आप उन्हें अपनी इच्छानुसार न पा लें। *शुरुआत में सीखने की प्रक्रिया थोड़ी कठिन होती है, हालाँकि गाइड और ट्यूटोरियल उपलब्ध हैं।*

गनोम:

नए उपयोगकर्ताओं को साफ इंटरफ़ेस और कॉन्फ़िगरेशन सेटिंग्स की कमी के साथ शुरुआत करना आसान लगता है। वर्कफ़्लो सरल है; इसका उपयोग करने में सक्षम होने से पहले आपको बहुत अधिक सेटअप की आवश्यकता नहीं है। इस कारण से, यदि कोई आउट-ऑफ-द-बॉक्स उपयोगकर्ता-अनुकूल डेस्कटॉप वातावरण चाहता है, तो GNOME बहुत बढ़िया है।

5. कार्यप्रवाह

केडीई:

उन शक्तिशाली उपयोगकर्ताओं की ज़रूरतों को पूरा करता है जो अपने वर्कफ़्लो को सूक्ष्म विवरण में कॉन्फ़िगर और वैयक्तिकृत करना पसंद करते हैं। डेस्कटॉप के हर पहलू पर बारीक नियंत्रण उपयोगकर्ताओं को एक अत्यधिक अनुकूलित वातावरण तैयार करने की अनुमति देता है जो उनकी ज़रूरतों के हिसाब से पूरी तरह से अनुकूल है। चाहे आप प्रत्येक स्क्रीन के लिए विशिष्ट गतिविधियों

के साथ एक **मल्टी-मॉनीटर सेटअप** , अक्सर उपयोग किए जाने वाले अनुप्रयोगों के साथ एक सावधानीपूर्वक संगठित पैनल लेआउट, या अनुकूलित उत्पादकता के लिए एक अद्वितीय कीबोर्ड शॉर्टकट योजना पसंद करते हों, KDE आपको इसे प्राप्त करने में सक्षम बनाता है।

गनोम:

एक केंद्रित और विकर्षण-मुक्त वातावरण को प्राथमिकता देता है। सीमित अनुकूलन विकल्पों के साथ पूर्व-निर्धारित वर्कफ़्लो एक सहज और सुसंगत उपयोगकर्ता अनुभव सुनिश्चित करता है। **GNOME शेल को सहज विंडो प्रबंधन सुविधाओं के साथ एकल कार्यक्षेत्र के आसपास डिज़ाइन किया गया था** जो उपयोगकर्ताओं को उनके वर्तमान कार्य पर केंद्रित रखता है। यह सरलता उन्नत उपयोगकर्ताओं के खिलाफ काम करती है जो सेटिंग्स पर अधिक नियंत्रण चाहते हैं लेकिन नौसिखियों के लिए एक साफ जगह प्रदान करता है जो बुनियादी सेटअप पसंद करते हैं।

6. समुदाय

केडीई:

KDE का समुदाय बहुत **बड़ा** और **सक्रिय** है और नवाचार तथा निरंतर विकास के लिए जाना जाता है। यह समूह खास तौर पर डेस्कटॉप वातावरण के प्रति अपनी प्रतिबद्धता के लिए जाना जाता है जो सीमाओं से परे जाकर कुछ नया लेकर आता है। उनके पास ऑनलाइन बहुत सारे दस्तावेज़, ट्यूटोरियल, फ़ोरम आदि हैं जिनका उपयोग समस्या निवारण के साथ-साथ अनुकूलन युक्तियों या KDE से अधिक लाभ उठाने के लिए किया जा सकता है।

गनोम:

केडीई की तरह, गनोम में भी एक बड़ा और सक्रिय समुदाय है जो इसके अनुप्रयोगों के साथ-साथ गनोम शेल के विकास और रखरखाव में योगदान देता है। **इस परियोजना के पीछे मुख्य फोकस बिंदु एक ऐसा वातावरण बनाना है जहाँ सब कुछ साफ और उपयोगकर्ता के अनुकूल दिखे ताकि लोगों को इसका उपयोग करना आसान लगे।** यदि आपको किसी भी सहायता की आवश्यकता है या गनोम से संबंधित किसी भी चीज़ के बारे में अधिक जानना चाहते हैं तो कई सहायता संसाधन हैं जैसे कि व्यापक दस्तावेज़ीकरण या यहाँ तक कि कुछ फ़ोरम ब्राउज़ करने से आपको गनोम की दुनिया में क्या चल रहा है, इसकी अच्छी जानकारी मिल जाएगी।

7. अंतर्निहित प्रौद्योगिकी

केडीई:

वीएलसी मीडिया प्लेयर जैसे लोकप्रिय ऐप भी **क्यूटी टूलकिट का** उपयोग करते हैं जो केडीई द्वारा उपयोग किया जाने वाला क्रॉस-प्लेटफ़ॉर्म आधारित सिस्टम है। अन्य प्रणालियों की तुलना में इसे इसलिए चुना गया क्योंकि यह लचीलापन है जो केडीई के **प्लाज्मा डेस्कटॉप वातावरण** के तहत दृश्य प्रभावों के भीतर व्यापक रेंज की कार्यक्षमता की अनुमति देता है।

गनोम:

जीटीके टूलकिट लिनक्स वितरण पर चलने वाले अधिकांश ओपन सोर्स प्रोग्रामों की नींव में निहित है, जिसमें बीएसडी भी शामिल है, इसके अलावा इसका उपयोग मुख्य रूप से ग्नोम अनुप्रयोगों में किया जाता है। इसे दक्षता को ध्यान में रखते हुए डिज़ाइन किया गया था, जबकि डेवलपर्स के बीच लोकप्रिय एक आकर्षक डिज़ाइन भाषा को बनाए रखा गया था, इसलिए यह ग्नोम के पीछे समग्र दर्शन के साथ पूरी तरह से फिट बैठता है

8. बॉक्स से बाहर डिफ़ॉल्ट

केडीई:

Kubuntu और KDE नियॉन जैसे लोकप्रिय Linux डिस्ट्रीब्यूशन के लिए डिफ़ॉल्ट डेस्कटॉप वातावरण KDE प्लाज्मा है। ये ऑपरेटिंग सिस्टम उन लोगों के लिए बनाए गए हैं जो अपने कंप्यूटर को कस्टमाइज़ करने के लिए कई विकल्प चाहते हैं।

गनोम:

गनोम पहले से ही फेडोरा और ***उबंटू*** के साथ स्थापित है (जिन्हें स्थापना के दौरान चुना जा सकता है); जो लोग एक सरल, सहज डेस्कटॉप वातावरण चाहते हैं और बहुत अधिक अनुकूलन नहीं चाहते हैं, उनके लिए यह आदर्श विकल्प है।

Difference Between KDE and GNOME

Aspect	KDE	GNOME
Customization	Highly customizable with many options	Limited customization, focused on simplicity
Applications	KDE apps suite with a variety of applications	GNOME app suite with fewer but well-integrated apps
Performance	Historically resource-intensive, but recent optimizations have improved efficiency	Lightweight, runs well on older hardware
Ease of Use	Can be overwhelming for beginners	Simple and user-friendly out of the box
Workflow	Highly customizable for power users	Streamlined workflow for a focused experience

DSG Support Multi Solution

Aspect	KDE	GNOME
Community	Large and active community with a focus on innovation and development	Active community focused on clean and user-friendly design
Technology	Uses Qt toolkit	Uses GTK toolkit
Defaults	Default in distributions like Kubuntu and KDE neon	Default in Fedora and Ubuntu
Suitable for	Power users who want extensive customization	New users or those looking for a simpler interface
Hardware	May require more powerful hardware	Lightweight and suitable for weaker hardware

Linux Commands

लिनक्स एक स्वतंत्र और ओपन-सोर्स सॉफ्टवेयर है जो अपने स्वयं के ऑपरेटिंग सिस्टम पर काम करता है। ' लिनक्स' शब्द का अर्थ है GNU + Linux । शुरुआत में लिनस टोरवाल्ड्स द्वारा विकसित , इसे यूनिक्स के स्रोत कोड के साथ बनाया गया था। जबकि लिनक्स का उपयोग विभिन्न उद्देश्यों के लिए बड़े पैमाने पर किया जाता है, इसके अनुप्रयोग कई लोगों के लिए अच्छी तरह से जाने जाते हैं।

लिनक्स कमांड का उपयोग

लिनक्स कमांड एक प्रकार का यूनिक्स कमांड या शेल प्रक्रिया है। वे व्यक्तिगत स्तर पर लिनक्स के साथ बातचीत करने के लिए उपयोग किए जाने वाले बुनियादी उपकरण हैं। लिनक्स कमांड का उपयोग विभिन्न प्रकार के कार्यों को करने के लिए किया जाता है, जिसमें फ़ाइलों और निर्देशिकाओं के बारे में जानकारी प्रदर्शित करना शामिल है।

लिनक्स ऑपरेटिंग सिस्टम का इस्तेमाल सर्वर, डेस्कटॉप और शायद आपके स्मार्टफोन पर भी किया जाता है। इसमें बहुत सारे कमांड लाइन टूल हैं जिनका इस्तेमाल सिस्टम पर लगभग हर चीज़ के लिए किया जा सकता है। सभी उपयोगकर्ताओं को इनमें से अधिकांश कमांडों से परिचित होना चाहिए क्योंकि अधिकांश ऑपरेटिंग सिस्टम कार्यों और कंप्यूटर प्रोग्रामिंग के लिए इनकी आवश्यकता होती है।

DSG Support Multi Solution

Linux Commands	Functions
1. ls command in Linux	Displays information about files in the current directory.
2. pwd command in Linux	Displays the current working directory.
3. mkdir command in Linux	Creates a directory.
4. cd command in Linux	To navigate between different folders.
5. rmdir command in Linux	Removes empty directories from the directory lists.
6. cp command in Linux	Copy files from one directory to another.
7. mv command in Linux	Rename and Replace the files
8. rm command in Linux	Delete files
9. uname command in Linux	Command to get basic information about the OS
10. locate command in Linux	Find a file in the database.
11. touch command in Linux	Create empty files
12. ln command in Linux	Create shortcuts to other files
13. cat command in Linux	Display file contents on terminal
14. clear command in Linux	Clear terminal
15. ps command in Linux	Display the processes in terminal
16. man command in Linux	Access manual for all Linux commands
17. grep command in Linux	Search for a specific string in an output

18. echo command in Linux	Print string or text to the terminal
19. wget command in Linux	download files from the internet.
20. whoami command in Linux	Displays the current users name
21. sort command in Linux	sort the file content
22. cal command in Linux	View Calendar in terminal
23. whereis command in Linux	View the exact location of any command typed after this command
24. df command in Linux	Check the details of the file system
25. wc command in Linux	Check the lines, word count, and characters in a file using different options

1. लिनक्स में ls कमांड

ls कमांड का इस्तेमाल आम तौर पर वर्किंग डायरेक्टरी में मौजूद फ़ाइलों और डायरेक्टरीज़ को पहचानने के लिए किया जाता है। यह कमांड उन कई बार इस्तेमाल किए जाने वाले लिनक्स कमांड में से एक है, जिन्हें आपको जानना चाहिए।

इस कमांड का इस्तेमाल बिना किसी तर्क के किया जा सकता है और यह हमें मौजूदा वर्किंग डायरेक्टरी में मौजूद फ़ाइलों और डायरेक्टरीज़ के बारे में सभी विवरणों के साथ आउटपुट प्रदान करेगा। आउटपुट में डेटा प्रदर्शित करने के मामले में इस कमांड द्वारा बहुत लचीलापन प्रदान किया जाता है। आउटपुट के लिए नीचे दी गई छवि देखें।

2. लिनक्स में pwd कमांड

pwd कमांड का इस्तेमाल ज़्यादातर आपके टर्मिनल पर मौजूदा वर्किंग डायरेक्टरी को प्रिंट करने के लिए किया जाता है। यह सबसे ज़्यादा इस्तेमाल की जाने वाली कमांड में से एक है।

अब, आपके टर्मिनल प्रॉम्प्ट में आमतौर पर पूरी डायरेक्टरी शामिल होनी चाहिए। अगर ऐसा नहीं है, तो यह एक त्वरित कमांड है जिससे पता चलता है कि आप किस डायरेक्टरी में हैं। इस कमांड का

एक और उद्देश्य स्क्रिप्ट बनाते समय है क्योंकि यह हमें उस डायरेक्टरी को खोजने में मदद कर सकता है जिसमें स्क्रिप्ट सहेजी गई थी। नीचे दी गई तस्वीरें कमांड के साथ आउटपुट हैं।

आज्ञा:

```
1  pwd
```

आउटपुट:

```
/home/cg/root/63b31828bde9b
```

3. लिनक्स में mkdir कमांड

यह mkdir कमांड आपको टर्मिनल में ही नई डायरेक्टरी बनाने की अनुमति देता है। डिफ़ॉल्ट सिंटैक्स **mkdir <डायरेक्टरी नाम>** है और नई डायरेक्टरी बनाई जाएगी।

उदाहरण के लिए, यदि आप "GeeksforGeeks" के रूप में एक निर्देशिका बनाना चाहते हैं तो मूल सिंटैक्स होगा:

```
mkdir गीक्सफॉरगीक्स
```

यदि आप प्रोजेक्ट्स को संग्रहीत करने के लिए मुख्य निर्देशिका GeeksforGeeks के अंदर एक और निर्देशिका बनाना चाहते हैं, तो आप ऐसा करने के लिए निम्नलिखित कमांड का उपयोग कर सकते हैं। **mkdir GeeksforGeeks/projects**

आज्ञा:

```
1  ls
2  mkdir GeeksForGeeks
3  ls
```

आउटपुट:

```
main.sh
GeeksForGeeks  main.sh
```

आप देख सकते हैं कि हमने वहां मौजूद निर्देशिकाओं को देखने के लिए पहले ls का उपयोग किया और फिर एक अन्य निर्देशिका बनाने के लिए **mkdir का** उपयोग किया, उसके बाद बनाई गई निर्देशिकाओं को देखने के लिए **ls का उपयोग किया।**

4. लिनक्स में cd कमांड

cd **कमांड का** उपयोग डायरेक्टरी के बीच नेविगेट करने के लिए किया जाता है। इसके लिए या तो पूरा पथ या डायरेक्टरी नाम की आवश्यकता होती है, जो आपकी वर्तमान कार्यशील डायरेक्टरी पर निर्भर करता है। यदि आप इस कमांड को बिना किसी विकल्प के चलाते हैं, तो यह आपको आपके

होम फ़ोल्डर में ले जाएगा। ध्यान रखें कि इसे केवल **sudo** विशेषाधिकार वाले उपयोगकर्ता ही निष्पादित कर सकते हैं।

आज्ञा:

```
pwd
cd GeeksForGeeks
pwd
```

आउटपुट:

```
/home/cg/root/63b3a4b346875
/home/cg/root/63b3a4b346875/GeeksForGeeks
```

यहां हमने संदर्भ के लिए वर्तमान निर्देशिका को देखने के लिए **pwd का** उपयोग किया और फिर हमने निर्देशिका को स्विच करने के लिए **cd *GeeksforGeeks* का उपयोग किया और फिर से pwd** कमांड के साथ हम देख सकते हैं कि आउटपुट स्विच की गई निर्देशिका है, यानी - ***GeeksforGeeks***

5. लिनक्स में rmdir कमांड

rmdir कमांड का उपयोग किसी खाली डायरेक्टरी को स्थायी रूप से हटाने के लिए किया जाता है। इस कमांड को निष्पादित करने के लिए इस कमांड को चलाने वाले उपयोगकर्ता के पास पैरेंट डायरेक्टरी में **sudo विशेषाधिकार होना चाहिए।**

आज्ञा:

```
ls
rmdir GeeksForGeeks
ls
```

निर्देशिका को हटाने के लिए आदेश

आउटपुट:

```
GeeksForGeeks  main.sh
main.sh
```

यहां हमने वहां मौजूद निर्देशिकाओं की जांच करने के लिए **ls** कमांड का उपयोग किया और निर्देशिका को हटाने के लिए **rmdir <निर्देशिका नाम> का** उपयोग किया और निर्देशिकाओं को हटाने के बाद उन्हें देखने के लिए पुनः **ls कमांड का उपयोग किया।**

6. लिनक्स में cp कमांड

लिनक्स का cp **कमांड** विंडोज़ के कॉपी-पेस्ट और कट-पेस्ट के समतुल्य है।

आज्ञा:

```
ls
cp first.txt second.txt
ls
```

आउटपुट:

```
first.txt   main.sh
first.txt   main.sh second.txt
```

यहां हमने फ़ाइलों को देखने के लिए **ls का उपयोग किया और फिर *first.txt*** की फ़ाइलों को ***second.txt*** में कॉपी करने के लिए **cp का** उपयोग किया और अपडेट की गई फ़ाइलों को देखने के लिए फिर से **ls कमांड का उपयोग किया।**

7. लिनक्स में mv कमांड

mv **कमांड का** प्रयोग सामान्यतः लिनक्स में फ़ाइलों का नाम बदलने के लिए किया जाता है।

आज्ञा:

```
ls
mv first.txt renamed.txt
ls
```

आउटपुट:

```
first.txt   main.sh
main.sh   renamed.txt
```

यहां हमने निर्देशिकाओं की जांच करने के लिए **ls** कमांड का उपयोग किया और फिर फ़ाइलों का नाम बदलने के लिए **mv <फ़ाइल नाम> <पुनःनामांकित फ़ाइल नाम> का उपयोग किया, और फिर पुनः हमने पुनःनामांकित फ़ाइल को देखने के लिए ls** कमांड का उपयोग किया जैसा कि आप आउटपुट स्क्रीनशॉट में देख सकते हैं।

8. लिनक्स में rm कमांड

लिनक्स में rm कमांड का प्रयोग सामान्यतः डायरेक्टरी में बनाई गई फाइलों को हटाने के लिए किया जाता है।

आज्ञा:

```
1  ls
2  rm renamed.txt
3  ls
```

आउटपुट:

```
main.sh   renamed.txt
main.sh
```

आप देख सकते हैं कि हमने टर्मिनल में फ़ाइलें देखने के लिए **ls** कमांड लिखा और फिर फ़ाइलों को हटाने के लिए **rm <फ़ाइल नाम>** लिखा और फिर से हमारे पास अपडेट की जांच करने के लिए **ls कमांड था।**

9. लिनक्स में uname कमांड

uname **कमांड** का उपयोग सिस्टम की पूरी OS जानकारी जाँचने के लिए किया जाता है। नीचे कमांड और आउटपुट देखें

आज्ञा:

```
1  uname
```

आउटपुट:

```
SMP Sun Dec 04 08:06:28 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

10. लिनक्स में लोकेट कमांड

लोकेट **कमांड का** इस्तेमाल आम तौर पर डेटाबेस में फ़ाइलों का पता लगाने के लिए किया जाता है। दो या उससे ज़्यादा शब्दों वाली सामग्री को खोजने के लिए तारांकन चिह्न (*) का इस्तेमाल करें। उदाहरण के लिए: **लोकेट फ़र्स्ट*फ़ाइल।** यह कमांड डेटाबेस में उन फ़ाइलों को खोजेगा जिनमें **फ़र्स्ट** और **फ़ाइल ये दो नाम हैं।**

आज्ञा:

```
1  rm first.txt
2  locate first.txt
```

आउटपुट:

```
locate -e first.txt
```

हमने पहले फ़ाइल को हटाने के लिए **rm कमांड का उपयोग किया और फिर डेटाबेस में फ़ाइल को खोजने के लिए लोकेट कमांड का उपयोग किया, जिसके परिणामस्वरूप -e** के साथ आउटपुट मिला क्योंकि फ़ाइल हटा दी गई थी।

11. लिनक्स में टच कमांड

टच कमांड टर्मिनल में इस प्रारूप में डालने पर एक खाली फ़ाइल बनाता है जैसे touch **<फ़ाइल नाम>**

आज्ञा:

```
ls
touch GeeksforGeeks.txt
ls
```

आउटपुट:

```
main.sh
GeeksforGeeks.txt   main.sh
```

हमने टर्मिनल में वर्तमान निर्देशिकाओं की जांच करने के लिए **ls** कमांड का उपयोग किया और फिर एक खाली फ़ाइल बनाने के लिए **टच** कमांड का उपयोग किया और फिर हमने टर्मिनल में बनाई गई फ़ाइल का पता लगाने के लिए फिर से **ls का उपयोग किया।**

12. लिनक्स में ln कमांड

ln कमांड का उपयोग किसी अन्य फ़ाइल के लिए शॉर्टकट लिंक बनाने के लिए किया जाता है। यदि आप लिनक्स एडमिनिस्ट्रेटर के रूप में काम करना चाहते हैं तो यह सबसे महत्वपूर्ण लिनक्स कमांड में से एक है।

आज्ञा:

```
mkdir Demo
mkdir Linked
ln -s Demo Linked
```

आउटपुट:

```
Linked/Demo
```

यहां हमने दो निर्देशिकाएं बनाने के लिए **mkdir का** उपयोग किया और फिर इसमें एक सॉफ्ट लिंक बनाने के लिए **-s** के साथ **ln का उपयोग किया।**

13. लिनक्स में cat कमांड

जब आप किसी विशेष फ़ाइल की सामग्री देखना चाहते हैं तो cat कमांड का उपयोग करना सबसे सरल कमांड है। एकमात्र समस्या यह है कि यह पूरी फ़ाइल को आपके टर्मिनल पर अनलोड कर देता है। यदि आप किसी बड़ी फ़ाइल के चारों ओर नेविगेट करना चाहते हैं, तो आपको वैकल्पिक रूप **से less** कमांड का उपयोग करना चाहिए।

आज्ञा:

```
cat files.txt
```

आउटपुट:

```
this is a File
```

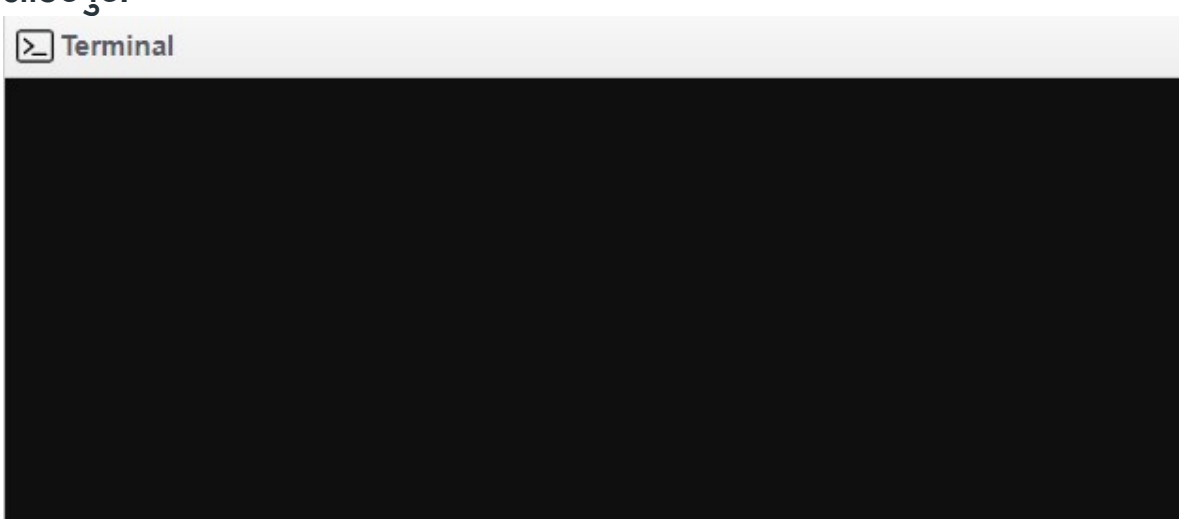
14. लिनक्स में क्लियर कमांड

क्लियर **कमांड** टर्मिनल स्क्रीन को साफ़ करने के लिए एक मानक कमांड है।

कमांड: *यह कमांड से पहले का टर्मिनल था।

```
$ ls
Demo
files.txt Linked main.sh NewFile Second
$ pwd
/home/cg/root/638c34db4d98e
$ cp Linked Non-Linked
cp: -r not specified; omitting directory 'Linked'
$ clear
```

आउटपुट:



15. लिनक्स में ps कमांड

लिनक्स में ps कमांड का उपयोग टर्मिनल में सक्रिय प्रक्रियाओं की जांच करने के लिए किया जाता है।

आज्ञा:

```
ps
```

आउटपुट:

```
PID TTY          TIME CMD
8454 pts/521  00:00:00 bash
11982 pts/521  00:00:00 bash
11983 pts/521  00:00:00 ps
```

16. लिनक्स में मैन कमांड

मैन **कमांड** टर्मिनल में उपलब्ध किसी भी कमांड या उपयोगिता के लिए उपयोगकर्ता मैनुअल प्रदर्शित करता है, जिसमें उनका नाम, विवरण और विकल्प शामिल होते हैं।

संपूर्ण मैनुअल देखने के लिए कमांड:

```
आदमी <कमांड नाम>
```

उदाहरण के लिए, मान लीजिए कि आप ls कमांड के लिए मैनुअल देखना चाहते हैं: **man ls**

आज्ञा:

```
1  man -f ls
```

आउटपुट:

```
ls (1) - list directory contents
```

17. लिनक्स में grep कमांड

grep **कमांड का** उपयोग आउटपुट की श्रृंखला में एक विशिष्ट स्ट्रिंग को खोजने के लिए किया जाता है। उदाहरण के लिए, यदि आप किसी फ़ाइल में स्ट्रिंग ढूँढना चाहते हैं, तो आप सिंटैक्स का उपयोग कर सकते हैं: **<आउटपुट के साथ कोई भी कमांड> | grep "<ढ़ूंढ़ना करने के लिए स्ट्रिंग> "**

उदाहरण के लिए:

```
cat फ़ाइलें.txt | grep "नया"
```

आज्ञा:

```
1  cat file.txt
2  cat file.txt | grep "GeeksforGeeks"
```

आउटपुट:

```
Hello World
Welcome to GeeksforGeeks
Welcome to GeeksforGeeks
```

इस कमांड में, हमने पहले फ़ाइल की सामग्री देखने के लिए **cat <file name> का** उपयोग किया, और फिर हमने इसमें स्ट्रिंग की जांच करने के लिए **cat <file name> | grep "string" का उपयोग किया।**

18. लिनक्स में इको कमांड

लिनक्स में इको कमांड का उपयोग विशेष रूप से टर्मिनल में कुछ प्रिंट करने के लिए किया जाता है

आज्ञा:

```
1  echo "Hello World"
```

आउटपुट:

```
Hello World
```

19. लिनक्स में wget कमांड

लिनक्स कमांड लाइन में wget **कमांड** आपको इंटरनेट से फ़ाइलें डाउनलोड करने की अनुमति देता है। यह बैकग्राउंड में चलता है और अन्य प्रक्रियाओं में हस्तक्षेप नहीं करता है।

यहाँ मूल सिंटैक्स है: **wget [विकल्प] [url]**

आज्ञा:

```
wget http://sample.com/sample-menu.php
```

आउटपुट:

20. लिनक्स में whoami कमांड

Whoami **कमांड** बुनियादी जानकारी प्रदान करता है जो कई सिस्टम पर काम करते समय बेहद उपयोगी है। सामान्य तौर पर, यदि आप एक ही कंप्यूटर के साथ काम कर रहे हैं, तो आपको नेटवर्क एडमिनिस्ट्रेटर की तरह इसकी अक्सर आवश्यकता नहीं होगी।

आज्ञा:

```
1  whoami
```

आउटपुट:

```
acer
```

21. लिनक्स में सॉर्ट कमांड

सॉर्ट कमांड **का** इस्तेमाल आम तौर पर फ़ाइल के आउटपुट को सॉर्ट करने के लिए किया जाता है। आइए कमांड का इस्तेमाल करें और आउटपुट देखें।

कमांड: (हम फ़ाइल सामग्री देखने के लिए cat कमांड का उपयोग कर रहे हैं)

```
1  cat multiple.txt
```

आउटपुट: (टर्मिनल में multiple.txt फ़ाइल की सामग्री)

```
Hello World
GeeksforGeeks
Thank you
```

अब हम सॉर्ट कमांड का उपयोग करके परिणाम को सॉर्ट करेंगे

आज्ञा:

```
1  sort multiple.txt
```

आउटपुट:

```
GeeksforGeeks
Hello World
Thank you
```

यहां सबसे पहले हमने cat कमांड का उपयोग करके फ़ाइल सामग्री की जांच की और फिर **sort** कमांड का उपयोग करके इसे वर्णानुक्रम में सॉर्ट किया ।

22. लिनक्स में cal कमांड

cal **कमांड** टर्मिनल में सबसे मशहूर कमांड नहीं है, लेकिन यह टर्मिनल में किसी खास महीने का कैलेंडर देखने के लिए काम करता है। आइए देखें कि यह कैसे काम करता है।

आज्ञा:

```
1  cal January 2023
```

आउटपुट:

```
January 2023
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

23. लिनक्स में whereis कमांड

लिनक्स में whereis कमांड का इस्तेमाल आम तौर पर इसके बाद टाइप किए गए किसी भी कमांड का सटीक स्थान देखने के लिए किया जाता है। आइए देखें कि यह कैसे काम करता है।

आज्ञा:

```
1  whereis printf
```

आउटपुट:

```
printf: /usr/bin/printf /usr/include/printf.h
```

24. लिनक्स में df कमांड

लिनक्स में df कमांड फ़ाइल सिस्टम का विवरण प्राप्त करता है।

आज्ञा:

```
1  df -h
```

आउटपुट:

```
Filesystem      Size  Used Avail Use% Mounted on
overlay         875G  120G  711G  15% /
tmpfs            63G     0   63G   0% /dev
tmpfs            63G     0   63G   0% /sys/fs/cgroup
/dev/nvme0n1p3  875G  120G  711G  15% /dev/init
shm              64M     0   64M   0% /dev/shm
tmpfs            63G     0   63G   0% /proc/acpi
tmpfs            63G     0   63G   0% /proc/scsi
tmpfs            63G     0   63G   0% /sys/firmware
```

यहां हमने **df -h का** प्रयोग किया है क्योंकि केवल **df** टाइप करने से आउटपुट बाइट्स में प्राप्त होगा जो पढ़ने योग्य नहीं है, इसलिए हम आउटपुट को अधिक पढ़ने योग्य और समझने योग्य बनाने के लिए **-h जोड़ते हैं।**

25. लिनक्स में wc कमांड

लिनक्स में wc कमांड विकल्पों के एक सेट का उपयोग करके शब्दों, वर्णों, पंक्तियों आदि की संख्या इंगित करता है।

- **wc -w** शब्दों की संख्या दर्शाता है
- **wc -l** पंक्तियों की संख्या दर्शाता है
- **wc -m** किसी फ़ाइल में मौजूद वर्णों की संख्या दिखाता है

आइये इन विकल्पों का एक उदाहरण देखें

आज्ञा:

```
1 touch file.txt
2 echo -e "This file has only six words" › file.txt
3 wc -w file.txt
```

आउटपुट:

```
6 file.txt
```

यहां हमने एक टेक्स्ट फ़ाइल बनाने के लिए **टच** कमांड का उपयोग किया और फिर छह शब्दों वाले वाक्य को इनपुट करने के लिए **इको** कमांड का उपयोग किया और इसमें शब्दों की संख्या की गणना करने के लिए हमने **wc -w कमांड का उपयोग किया।**

Thanks & Regards

DSG Support Multi Solution