

**AISHWARYA COLLEGE OF EDUCATION**  
**DEPARTMENT: COMPUTER SCIENCE**  
**FACULTY NAME: SHAILENDRA PUROHIT**

<b>Course</b>	<b>BCA</b>
<b>Class</b>	<b>BCA 2<sup>nd</sup> YEAR</b>
<b>Subject</b>	<b>OS</b>
<b>Unit</b>	<b>1 and 2</b>

## TABLE OF CONTENT

SNO	TOPICS
1.	WHAT IS AN OPERATING SYSTEM?
2.	FUNCTIONS OF OPERATING SYSTEM
3.	EVOLUTION OF OS
4.	OPERATING SYSTEM SERVICES
5.	SYSTEM CALL
6.	PROCESS CONCEPT
7.	PCB
8.	CPU SCHEDULING
9.	CPU SCHEDULING CRITERIA
10.	SCHEDULING ALGORITHM
11.	THREADS
12.	TYPES OF THREADS
13.	MULTITHREADING MODELS
14.	THREADS IN LINUX

### ऑपरेटिंग सिस्टम

ऑपरेटिंग सिस्टम (OS) कंप्यूटर यूजर और कंप्यूटर हार्डवेयर के बीच एक इंटरफ़ेस है। ऑपरेटिंग सिस्टम एक सॉफ्टवेयर है जो फ़ाइल प्रबंधन, मेमोरी प्रबंधन, प्रक्रिया प्रबंधन, इनपुट और आउटपुट को संभालने और पेरिफेरल डिवाइस जैसे कि डिस्क ड्राइव और प्रिंटर को नियंत्रित करने जैसे सभी बुनियादी कार्य करता है। कुछ लोकप्रिय ऑपरेटिंग सिस्टम में Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS आदि शामिल हैं।

ऑपरेटिंग सिस्टम सॉफ्टवेयर प्रोग्रामों का सेट होता है जो कंप्यूटर की समस्त क्रियाओं को संचालित व नियंत्रित करता है। कंप्यूटर सिस्टम के विभिन्न हार्डवेयर उपकरण स्वयं अपने बल पर कार्य नहीं कर सकते और न ही एक दूसरे से तालमेल स्थापित कर सकते हैं। ये सभी उपकरण ऑपरेटिंग सिस्टम द्वारा दिये जाने वाले इलेक्ट्रॉनिक सिग्नलों के द्वारा संचालित होते हैं, जिस प्रकार ऑर्केस्ट्रा में म्यूजिक आर्गेनाइजर के इशारे पर विभिन्न वादक वाद्य बजाते हैं और एक सामूहिक प्रस्तुति देते हैं, ठीक उसी प्रकार ऑपरेटिंग सिस्टम के द्वारा दिये जाने वाले सिग्नलों के अनुसार कंप्यूटर के उपकरण अपना अपना कार्य करते हुए सयुक्त रूप से किसी निश्चित कार्य को पूरा करते हैं।

ऑपरेटिंग सिस्टम को निम्न बिन्दुओं के आधार पर आसानी से समझा जा सकता है:-

1:- operating system (OS) एक सॉफ्टवेयर होता है जो कि कंप्यूटर तथा यूजर के मध्य interface की तरह कार्य करता है। इसे system software कहते हैं।

2:- ऑपरेटिंग सिस्टम निर्देशों का समूह होता है जो कि स्टोरेज डिवाइस में स्टोर रहता है। तथा यह programs का समूह होता है जो कि कंप्यूटर के resources तथा operations को manage करता है।

3:- OS कंप्यूटर में लोड होने वाला यह पहला प्रोग्राम होता है। इसे program of programs भी कहते हैं।

4:- OS कंप्यूटर के सभी operations को manage करता है।

5:- operating system दो प्रकार के होते हैं:-

\*character user interface (CUI)

\*Graphical user interface (GUI)

# ऑपरेटिंग सिस्टम की विशेषतायें

इसकी characteristics निम्नलिखित है:-

- **memory management (मैमोरी मैनेजमेंट) :-** OS मैमोरी को मैनेज करता है, यह primary memory की पूरी जानकारी रखता है और देखता है कि मैमोरी के कौन से भाग का use किस प्रोग्राम ने किया है. जब भी कोई प्रोग्राम request करता है तो उसे मैमोरी allocate करता है.
- **processor management (प्रोसेसर मैनेजमेंट) :-** यह प्रोग्राम को processor (CPU) allocate करता है और जब किसी प्रोग्राम को cpu की जरूरत खत्म हो जाती है तो इसे deallocate भी करता है.
- **device management (डिवाइस मैनेजमेंट) :-** OS सभी devices की जानकारी रखता है इसे I/O controller भी कहते है. तथा OS यह भी निर्णय लेता है कि किस program को कौन सी डिवाइस दी जाएँ, कब दी जाएँ तथा कितने समय के लिए दी जाएँ.
- **file management (फाइल मैनेजमेंट) :-** यह resources को allocate तथा deallocate करता है तथा यह निर्णय लेता है कि किस प्रोग्राम को resources दी जाएँ अर्थात् allocate की जाये.
- **security (सुरक्षा) :-** यह किसी भी प्रोग्राम या डेटा को unauthorized एक्सेस से बचाता है. इसमें password तथा अन्य तकनीकों का इस्तेमाल किया जाता है.
- **reliability (विश्वसनीय) :-** यह बहुत ही reliable होता है क्योंकि इसमें किसी भी virus तथा हानिकारक codes को detect किया जा सकता है.
- **cost (मूल्य):-** operating system का cost इसके features के आधार पर निर्धारित होता है जैसे:- windows की cost 100 \$ के आस पास होती है जबकि DOS तथा UNIX ऑपरेटिंग सिस्टम free है.
- **ease of use :-** इसे आसानी से use किया जा सकता है क्योंकि इसमें GUI इंटरफ़ेस भी होता है.

## ऑपरेटिंग सिस्टम के लाभ

इसके लाभ निम्नलिखित है:-

- इसे आसानी से use किया जा सकता है क्योंकि इसका ग्राफिकल यूजर इंटरफ़ेस होता है. और नए users इसके द्वारा कंप्यूटर को आसानी से चला सकते है.

- इसके द्वारा हम एक data को बहुत सारें users के साथ share कर सकते हैं.
- इसके द्वारा हम resources को share कर सकते हैं जैसे:- प्रिंटर.
- इन्हें आसानी से update किया जा सकता है.
- यह सुरक्षित (secure) होता है जैसे:- windows में windows defender होता है जो कि किसी भी प्रकार की हानिकारक files को detect कर लेता है और उन्हें remove कर देता है.
- इसके द्वारा हम कोई भी game या सॉफ्टवेर install सकते हैं और उन्हें चला सकते हैं.
- कुछ ऑपरेटिंग सिस्टम (जैसे:- LINUX) open source होते हैं इन्हें हम free में अपने computer पर run कर सकते हैं.

## ऑपरेटिंग सिस्टम के प्रकार

**इसके प्रकार निम्नलिखित हैं:-**

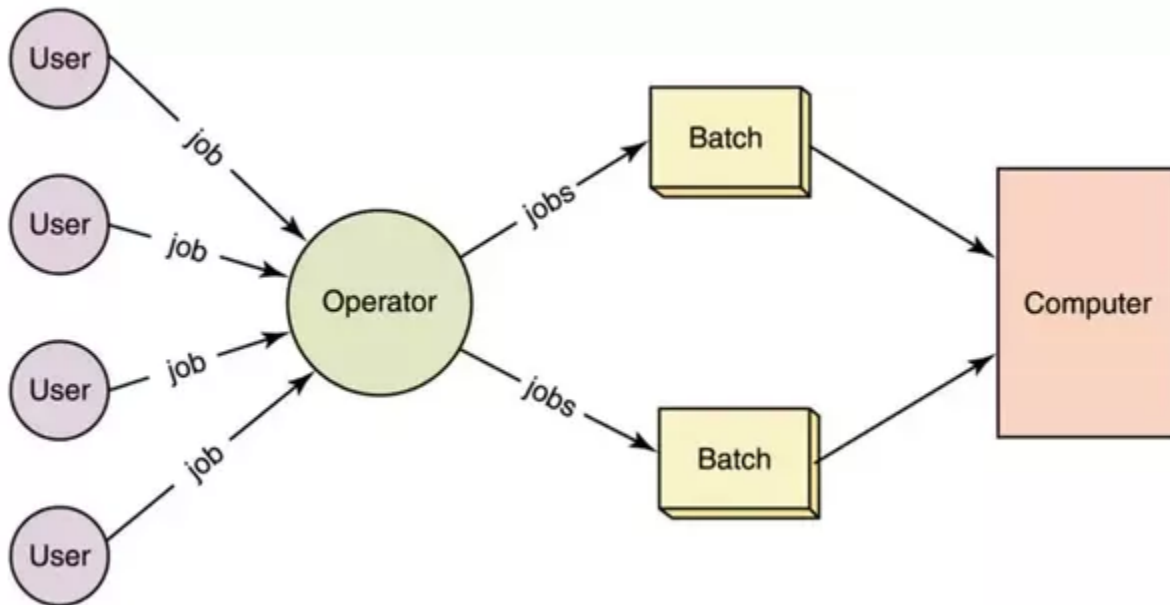
1. real time
2. batch
3. distributed
4. network

### **batch ऑपरेटिंग सिस्टम**

**batch ऑपरेटिंग सिस्टम** users के साथ सीधे इंटरैक्ट नहीं करता है.

इस ऑपरेटिंग सिस्टम में समान प्रकार के jobs का एक batch बना दिया जाता है और उस batch को punch card (यह डिजिटल डेटा को स्टोर करता है) में स्टोर कर दिया जाता है और उस पंच कार्ड को ऑपरेटर को दिया जाता है और वह ऑपरेटर punch card को कंप्यूटर को processing के लिए देता है और कंप्यूटर तब उस कार्ड में से क्रमबद्ध तरीके से jobs को पूरा करता है. इस ऑपरेटिंग सिस्टम की सबसे बड़ी परेशानी यह थी कि यह users के साथ सीधे इंटरैक्ट नहीं कर सकता था. जिससे अगर किसी job में कोई गड़बड़ी हो गयी तो सभी jobs इससे प्रभावित होते थे. और अन्य jobs को तक तब wait करना पड़ता था जब तक कि गड़बड़ी सही नहीं हो जाती थी.

दूसरी परेशानी यह थी कि एक batch में सभी समान प्रकार के job होने चाहिए और इसमें batch बनाने में अधिक समय लग जाता था.



### Multiprocessing Operating System

Multiprocessing Word का use एक Processing method को स्पष्ट करने के लिये किया जाता है, जहाँ पर दो या दो से अधिक Processor एक दूसरे से जुड़े रहते हैं। इस प्रकार के system में different and independent Program के निर्देश एक ही समय में एक से अधिक processor के द्वारा execute किये जाते हैं। अतः Processor द्वारा विभिन्न निर्देश का execution एक के बाद एक किया जाता है, जो कि एक ही Program से प्राप्त हुये हो।

### Multitasking Operating System

मल्टीटास्किंग में एक तरह से multiprogramming का ही concept है लेकिन इसके साथ ही इसमें time sharing भी मौजूद है. Multitasking operating system में प्रत्येक process को एक fixed time period के लिए execute किया जाता है. उसके बाद C.P.U मुख्य मेमोरी में पड़ी दूसरी प्रोसेस को निष्पादित करने लग जाता है. ऐसे ही वह कुछ निर्धारित समय अवधि के बाद सभी प्रोसेस को निष्पादित करते रहता है. सीपीयू की switching इतनी तेज होती है, कि उपयोगकर्ता illusion का शिकार हो जाता है.

उसे लगता है, कि सभी process एक साथ चल रही है. परंतु ऐसा नहीं हो सकता क्योंकि सीपीयू एक समय में एक ही प्रोसेस को चला सकता है. अब क्योंकि processor इतना शक्तिशाली है, कि वह सभी प्रक्रियाओं को एक साथ संभाल लेता है. इस निश्चित समयावधि को time quantum कहा जाता है. इस operating system को time sharing operating system भी कहते हैं.

### Time Sharing Operating System

एक time sharing operating system जिसमें प्रत्येक process को execute होने के लिए एक fixed time दिया जाता है. मान लीजिए एक system से कई user connect हैं, तो प्रत्येक उपयोगकर्ता C.P.U का इस्तेमाल करने के लिए आपस में time share करेंगे. यदि एक यूजर के लिए सीपीयू इस्तेमाल करने का समय दो सेकंड है, तो सिस्टम दो सेकंड बाद दूसरे यूजर को उपलब्ध हो जाएगा. ऐसे ही वह अन्य उपयोगकर्ता की तरफ बढ़ता जाएगा. इस तरह से उपयोगकर्ता बिना इंतजार के सीपीयू का उपयोग कर पायेंगे. इन operating system को हम multitasking OS भी कहते हैं.

### Distributed ऑपरेटिंग सिस्टम

distributed ऑपरेटिंग सिस्टम वे सिस्टम होते हैं जो डेटा को स्टोर करते हैं और उसे बहुत सारे locations पर डिस्ट्रीब्यूट कर देते हैं.

distributed ऑपरेटिंग सिस्टम में बहुत सारे central processors का प्रयोग किया जाता है और इन processors के मध्य डेटा प्रोसेसिंग jobs को डिस्ट्रीब्यूट कर दिया जाता है.

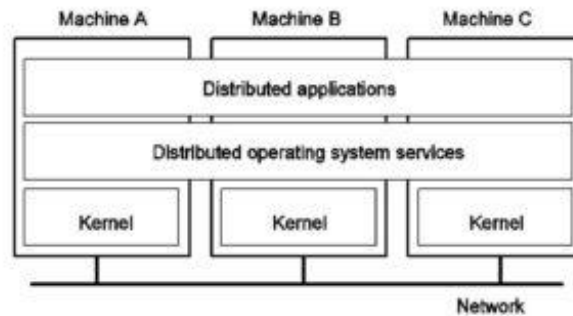
यह central processor कोई कंप्यूटर, साइट, नोड या फिर कोई अन्य डिवाइस हो सकता है. ये सभी processors आपस में कम्युनिकेशन लाइन्स के द्वारा आपस में एक दूसरे से connected रहते हैं.

### इसके लाभ

- distributed OS का एक फायदा यह है कि अगर users एक कंप्यूटर पर हैं तो वह अन्य किसी दूसरे कंप्यूटर का डेटा / रिसोर्स को एक्सेस कर सकता है.
- इससे डेटा का आदान प्रदान ईमेल के द्वारा भी किया जाता है जिससे डेटा आदान-प्रदान की गति बढ़ती है.

- distributed सिस्टम में अगर एक साइट या कंप्यूटर बंद भी पड़ जाए तो अन्य दूसरे कंप्यूटरों से काम किया जा सकता है.
- इससे डेटा प्रोसेसिंग का कार्य आसान हो जाता है.

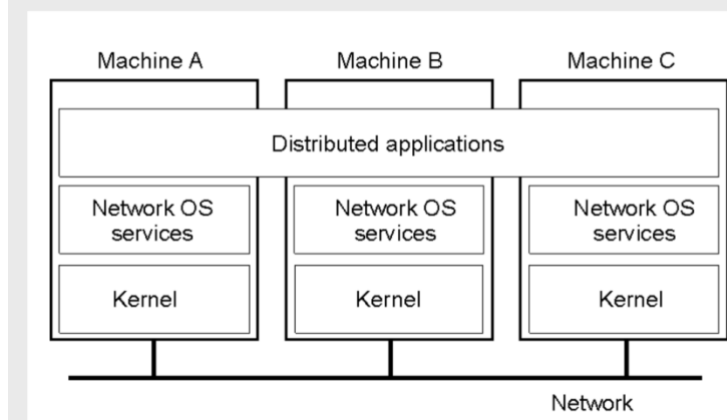
### Distributed Operating Systems (DOS)



### नेटवर्क ऑपरेटिंग सिस्टम

नेटवर्क ऑपरेटिंग सिस्टम एक ऐसा ऑपरेटिंग सिस्टम है जो कि [server](#) पर रन होता है. इस OS में ऐसे functions होते हैं जिससे कि कंप्यूटर, वर्क स्टेशन या अन्य डिवाइस को लोकल एरिया नेटवर्क से जोड़ा जाता है. इस OS का मुख्य उद्देश्य LAN में बहुत सारें computers के मध्य फाइल share करना, एप्लीकेशन share करना, डेटाबेस share करना, प्रिंटर share करना तथा security उपलब्ध करना आदि है. इस OS के उदाहरण हैं:- विंडोज NT, विंडोज सर्वर 2003, विंडोज सर्वर 2008, linux, unix, mac OS. आदि.

### Network Operating System





### Real time ऑपरेटिंग सिस्टम

real time ऑपरेटिंग सिस्टम वह ऑपरेटिंग सिस्टम होता है जो कि दिए गये समय में task (कार्य) को पूरा करता है. यह बहुत ही तेज ऑपरेटिंग सिस्टम होता है जिसमें कि समय बहुत कम होता है. यह real time एप्लीकेशन को सपोर्ट करता है और इनका प्रयोग industrial और scientific कार्य के लिए होता है.

यह दो प्रकार का होता है:-

**1:- hard real time ऑपरेटिंग सिस्टम:-** यह ऑपरेटिंग सिस्टम गारंटी देता है कि दिए गये समय में task को पूरा कर लिया जाएगा. यह बहुत ही strict होता है.

**2:- soft real time :-** यह ऑपरेटिंग सिस्टम भी पूरी कोशिश करता है कि दिए गये task को समय पर पूरा कर लिया जाए और जो highest priority task है उन्हें पहले पूरा कर लिया जाएँ परन्तु इसमें task के समय पर पूरा होने की कोई गारंटी नहीं होती. यह थोड़ा कम strict है.

### Real-Time Operating System के Advantage:

- डिवाइस और सिस्टम का अधिकतम उपयोग होता है.
- इस प्रकार के सिस्टम में memory allocation को बेहतर manage किया जाता है.
- इसके program का size कम होने के कारण RTOS का उपयोग transport जैसे embedded systems में भी किया जाता है.

### Real-Time Operating System के Disadvantage:

- एक समय में बहुत कम टास्क को चलाया जा सकता है.
- कभी-कभी सिस्टम संसाधन इतने अच्छे नहीं होते और वे महंगे भी होते हैं.
- इनकी algorithm को समझना एक डिज़ाइनर के लिए काफी जटिल और कठिन होता है.

## System call

हमें system call के कांसेप्ट को समझने से पहले CPU के kernel mode तथा user mode के मध्य difference को समझना पड़ेगा. हर [ऑपरेटिंग सिस्टम](#) में दो मोड्स होते हैं:-

1:- kernel mode

2:- user mode

kernel mode:-

जब सी.पी.यू. kernel mode में होता है तो execute होने वाला प्रोग्राम, मैमोरी एड्रेस तथा हार्डवेयर रिसोर्स को एक्सेस कर सकता है. यह सी.पी.यू. के किसी भी instructions को execute कर सकता है.

kernel mode बहुत ही शक्तिशाली मोड होता है क्योंकि इसमें ऑपरेटिंग सिस्टम के सबसे विश्वनीय प्रोग्राम execute होते हैं. अगर kernel mode में कोई प्रोग्राम crash हो गया तो पूरे सिस्टम ही ठप पड़ जाता है अर्थात् रुक जाता है.

user mode:-

जब सी.पी.यू. user mode में होता है तो program को मैमोरी एड्रेस तथा हार्डवेयर रिसोर्स को डायरेक्ट एक्सेस करने की अनुमति नहीं होती है.

user mode में अगर कोई प्रोग्राम crash भी हो जाएँ तो केवल वही प्रोग्राम बंद होगा. अर्थात् user mode में कोई प्रोग्राम बंद भी पड़ जाएँ तो भी सिस्टम सुरक्षित रहता है.

कंप्यूटर में सबसे ज्यादा प्रोग्राम user mode में ही run होते हैं.

### system call (सिस्टम कॉल):-

जब भी user mode में कोई प्रोग्राम perform किया जाता है तो यूजर द्वारा सिस्टम को उस प्रोग्राम को पूरा करने की request की जाती है. क्योंकि user mode में मैमोरी एड्रेस तथा हार्डवेयर रिसोर्स को डायरेक्ट एक्सेस नहीं किया जा सकता है.

तो यूजर के द्वारा यह request एक call के द्वारा की जाती है जिसे हम system call (सिस्टम कॉल) कहते हैं.

आसान शब्दों में कहें तो, “system call प्रोग्राम तथा ऑपरेटिंग सिस्टम के मध्य इंटरफ़ेस का कार्य करता है.”

जब यूजर द्वारा किसी प्रोग्राम को परफॉर्म करने के लिए system call की जाती है तो जो mode होता है वह user mode से kernel mode में बदल जाता है.

जब भी हम सिस्टम में कोई फोल्डर खोलने की request करते हैं या माउस को स्क्रीन में move करते हैं तो यह सिस्टम कॉल कहलाता है.

### प्रोसेस कंट्रोल ब्लॉक

PCB का पूरा नाम process control block (प्रोसेस कंट्रोल ब्लॉक) है। इसे task control block भी कहते हैं।

PCB एक डेटा स्ट्रक्चर है जिसे प्रत्येक प्रोसेस या टास्क के लिए [ऑपरेटिंग सिस्टम](#) के द्वारा maintain किया जाता है।

दूसरें शब्दों में कहें तो, “process control block (PCB) एक डेटा स्ट्रक्चर है जिसका प्रयोग processes की information को स्टोर करने के लिए किया जाता है।”

प्रोसेस की information का प्रयोग सी.पी.यू. के द्वारा runtime में किया जाता है।

PCB की पहचान एक integer process ID (PID) के द्वारा की जाती है।

process control block जो है वह multi-programming में बहुत महत्वपूर्ण हो जाता है क्योंकि multi programming में बहुत सारें processes एक साथ एक ही समय पर run करते हैं तो यह उन सभी processes की information को स्टोर करता है।

PCB निम्नलिखित इनफार्मेशन को स्टोर करती है:-

- प्रोसेस का नाम
- प्रोसेस की state, अर्थात् उसकी वर्तमान state क्या है जैसे:- running, waiting, ready आदि.
- प्रोसेस के रिसोर्स
- मैमोरी (जो प्रोसेस को प्रदान की गयी है)
- प्रोसेस के द्वारा प्रयोग की गयी इनपुट तथा आउटपुट डिवाइस
- process ID, (ऑपरेटिंग सिस्टम में प्रत्येक प्रोसेस को indentify करने के लिए एक यूनिक indentification होता है.)
- scheduling information
- [file management](#) information

description of process control block components:-

Process control block के कंपोनेंट्स निम्नलिखित हैं:-

**1:- Process ID:-** कंप्यूटर सिस्टम में एक समय में बहुत सारें processes एक साथ run करते हैं तो प्रत्येक प्रोसेस का एक यूनिक ID होता है। इस ID के मदद से सिस्टम processes की scheduling करता है। यह ID, PCB के द्वारा उपलब्ध की जाती है।

आसान शब्दों में कहें तो, “यह ऑपरेटिंग सिस्टम में प्रत्येक processes को identify करने के लिए एक indentification नंबर होता है।”

**2:- Process state:-** प्रत्येक प्रोसेस की कोई न कोई एक state होती है जैसे:- new, ready, running, waiting, halted इत्यादि.

process control block प्रत्येक प्रोसेस की state को डिफाइन करता है.

**3:- CPU registers:-** रजिस्टर कितने भी प्रकार के हो सकते हैं क्योंकि रजिस्टर की संख्या तथा प्रकार सी.पी.यू. के आर्किटेक्चर पर निर्भर करता है. इसके अंतर्गत accumulators, index registers, stack pointers तथा general purpose registers आदि आते हैं.

**4:- program counter:-** program counter, प्रोसेस के लिए execute होने वाले अगले instruction के address को स्टोर किये रहता है.

**5:- CPU scheduling information:-** [CPU scheduling](#) information का प्रयोग processes की priority को सेट करने के लिए किया जाता है.

**6:- memory management information:-** यह page table, segment table, base register तथा limit registers की इनफार्मेशन को स्टोर करता है.

**7:- accounting information:-** इसमें CPU utilization, प्रोसेस का execution time, प्रोसेस की संख्या तथा account no. आदि के बारे में सूचना स्टोर रहती है.

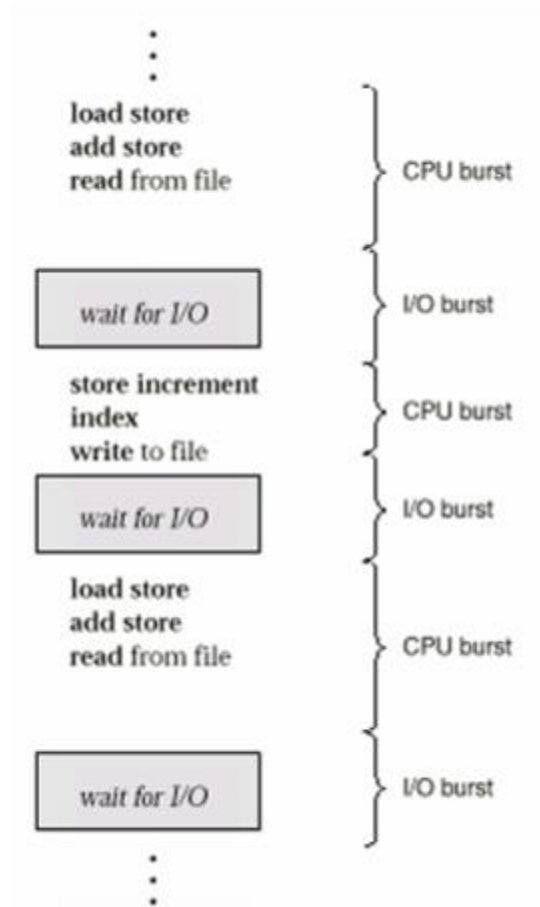
**8:- I/O status information:-** इसमें प्रोसेस को allocate की गयी इनपुट तथा आउटपुट डिवाइसों के बारे में सूचना स्टोर करती है.

## CPU scheduling

CPU scheduling एक ऐसी प्रक्रिया है जिसमें एक process को cpu का प्रयोग करने की आज्ञा दी जाती है जबकि दूसरे process की execution को होल्ड पर रखा जाता है क्योंकि कोई resource उपलब्ध नहीं होता है.

दूसरे शब्दों में कहें तो “बहुत से process के मध्य cpu को switch(बदलने) की प्रक्रिया को CPU scheduling कहते हैं.” CPU scheduler ऑपरेटिंग सिस्टम का एक हिस्सा है तथा इसका कार्य CPU scheduling करना है.

**उदाहरण के लिए:-** CPU scheduling की प्रक्रिया में, दो process A तथा B सी.पी.यू. के समय को share करते हैं. जब process A सी.पी.यू. का प्रयोग करता है तब process B I/O ऑपरेशन को परफॉर्म करता है और जब process B सी.पी.यू. का प्रयोग करता है तब process A I/O ऑपरेशन को परफॉर्म किया जाता है.



CPU scheduling जो है वह preemptive तथा non-preemptive दोनों प्रकार की हो सकती है.

“**preemptive scheduling** में, scheduler जो है वह cpu से running process को remove करता है जिससे कि कोई दूसरा process run हो सके.”

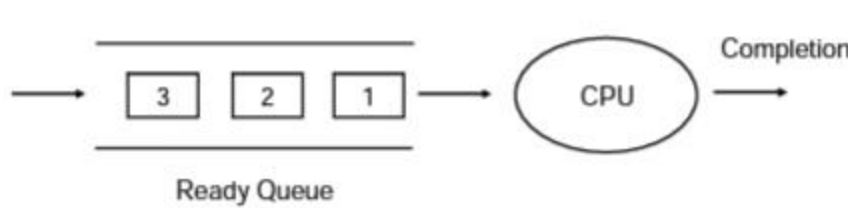
आसान शब्दों में कहें तो “preemptive scheduling में process को उसके ऑपरेशन के मध्य में भी interrupt किया जा सकता है.”

**non-preemptive scheduling** में, process जो है वह cpu का नियन्त्रण तभी दे सकती है जब उसका ऑपरेशन पूरा हो जायें, अर्थात इस scheduling में जबरदस्ती ऑपरेशन के बीच में process को interrupt नहीं किया जा सकता है.

cpu scheduling निम्नलिखित प्रकार की होती है.

### 1:-first come first serve(FCFS):-

यह सबसे सरल CPU scheduling अल्गोरिथम है. इसमें, जो process CPU को सबसे पहले request करेगा उसे सबसे पहले cpu allocate किया जायेगा. लेकिन इस अल्गोरिथम का नुकसान है कि इसमें औसत waiting time बहुत ज्यादा है.



### **First-Come, First-Served (FCFS) Scheduling**

- Example:

<u>Process</u>	<u>Burst Time (milliseconds)</u>
$P_1$	24
$P_2$	3
$P_3$	3

- Suppose that the processes arrive in the order:  $P_1, P_2, P_3$   
The Gantt Chart for the schedule is:



- Waiting time for  $P_1 = 0$ ;  $P_2 = 24$ ;  $P_3 = 27$
- Average waiting time:  $(0 + 24 + 27) / 3 = 17$  (milliseconds)

## 2:-Shortest-job-first(SJF) scheduling:-

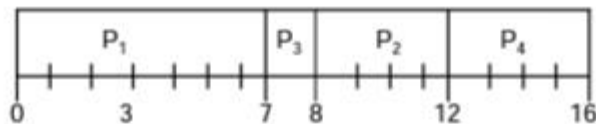
SJF को shortest-job-next(SJN) भी कहते हैं. तथा यह अल्गोरिथम FCFS से तेज है. इस अल्गोरिथम में जो process सबसे कम समय में पूरा हो जायेगा अर्थात जिस process को पूरा होने में सबसे कम समय लगेगा उसे सबसे पहले execute किया जायेगा. इस अल्गोरिथम की प्रकृति non-preemptive होती है.

### Example of Non-Preemptive SJF

• Example: Process      Arrival Time      Burst Time

$P_1$	0.0	7
$P_2$	2.0	4
$P_3$	4.0	1
$P_4$	5.0	4

• SJF (non-preemptive)



• Average waiting time =  $(0 + 6 + 3 + 7) / 4 = 4$

## 3:-Priority scheduling:-

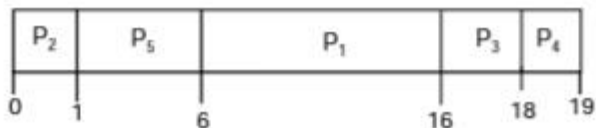
इस scheduling में, सभी process को एक priority दी जाती है तथा highest priority वाले process को सबसे पहले CPU allocate किया जाता है.

अगर दो process की priority समान होगी तो तब उनके मध्य FCFS scheduling की जाती है.

### Example of Nonpreemptive Priority

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

• Gantt Chart:



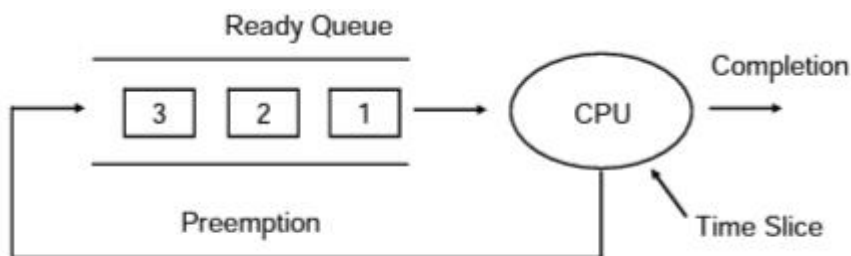
• Average waiting time =  $(0 + 1 + 6 + 16 + 18) / 5 = 8.2$

## 4:-Round-robin scheduling:-

यह scheduling FCFS की तरह ही समान है परन्तु round-robin में time-sharing कांसेप्ट का प्रयोग किया जाता है.

इसमें सभी processes को पहले से डिफाईड time के लिए CPU को allocate किया जाता है. इस predefined time को time-slice कहते हैं.

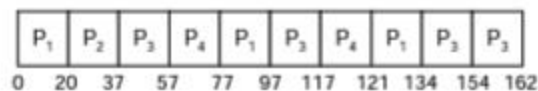
यदि process इस time-slice के अन्दर पूरा हो जाता है तो दूसरे process को execute किया जाता है. अगर process पूरा नहीं होता है तो वह preempted हो जाता है अर्थात इस process को वापस पीछे(rear end)) में भेज दिया जाता है और अगले process को execute किया जाता है.



### Example: RR with Time Quantum = 20

Process	Burst Time	Waiting Time of each Process
$P_1$	53	$0 + (77 - 20) + (121 - 97) = 81$
$P_2$	17	20
$P_3$	68	$37 + (97 - 57) + (134 - 117) = 94$
$P_4$	24	$57 + (117 - 77) = 97$

• The Gantt chart is:



• Average Waiting Time =  $(81 + 20 + 94 + 97) / 4 = 73$



“ऑपरेटिंग सिस्टम, प्रोग्राम्स का एक समूह (collection) है, जो सिस्टम रिसोर्सेस जैसे प्रोसेसर (processor) मेमोरी (memory), इनपुट/आउटपुट डिवाइसेस (I/O Devices) तथा फाइल सिस्टम (File System) को मैनेज (manage) करता है।” ये सभी सिस्टम रिसोर्सेस मूल्यवान (valuable) होते हैं, जिन्हें ऑपरेटिंग सिस्टम एक-दूसरे से सहयोगी (co-operative) ढंग से कार्य करने में सहायता करता है।

ऑपरेटिंग सिस्टम प्रत्येक रिसोर्स (resource) की स्थिति (status) की जानकारी रखता है और निश्चित पॉलिसी (certain policy) के आधार पर विभिन्न प्रोसेसेस के लिए रिसोर्सेस को बांटता (allocate) है तथा यह निर्णय भी लेता है कि कोई प्रोसेस कितनी देर तक रिसोर्सेस का प्रयोग करेगी। अन्त में यह बांटे (allocated) गए रिसोर्सेस को डिअलोकेट (de-allocate) भी करता है।

जहां तक प्रोसेसर मैनेजमेंट का सवाल है, तो ऑपरेटिंग सिस्टम द्वारा फिजिकल प्रोसेसर्स (CPUs) का प्रबन्धन (management) अर्थात् प्रोसेसेस या टास्कस या जॉब्स (process or tasks or task or jobs) के लिए प्रोसेसर (processor) को एलोकेट (allocate) करना ही प्रोसेसर मैनेजमेंट कहलाता है।

प्रोग्राम की एक्जक्यूट (execute) कर रही अवस्था ही प्रोसेस (process) कहलाती है। अर्थात् कोई भी प्रोग्राम जब चल रहा होता है तो वह एक प्रोसेस कहलाती है, जो किसी खास कार्य को सम्पन्न करता है। उदाहरणस्वरूप, UNIX ऑपरेटिंग सिस्टम में Shell या Command Interpreter भी एक प्रोसेस है, जो यूजर द्वारा दिए गए कमाण्ड को एक्सेप्ट (accept) करती है तथा इसे एक प्रोसेस के रूप में एक्सीक्यूट करती है। ज्ञातव्य हो कि जब भी कोई प्रोसेस क्रिएट (Create) होती है, तो उसे विभिन्न सिस्टम रिसोर्सेस, जैसे CPU का समय, मेमोरी (memory) रजिस्टर्स (registers), फाइलों (files) इत्यादि की आवश्यकता होती है।

कई ऑपरेटिंग सिस्टम में प्रत्येक प्रोसेस की इनफॉर्मेशन (information), प्रोसेस टेबल (process tables) में स्टोर होती है। ऑपरेटिंग सिस्टम प्रोसेस को मैनेज (manage) करने के लिए सिस्टम कॉल्स (system calls) भी प्रदान करती है। इन सिस्टम कॉल्स (system calls) में create और kill सिस्टम कॉल्स प्रमुख हैं।

प्रोग्राम जहां पैसिव इन्टिटी (passive entity) हैं, जैसे, डिस्क पर प्रोग्राम कोड्स को फाइल के रूप में स्टोर करना। वहीं प्रोसेस एक एक्टिव इन्टिटी (active entity) है, जिसके साथ एक प्रोग्राम काउन्टर (program counter) होता है, जो अगले इन्सट्रक्शन (next instruction) को एक्जक्यूट करने को निर्दिष्ट करता है तथा प्रोसेस से सम्बन्धित रिसोर्स का सेट (set) होता है। केवल एक प्रोसेसर (processor) को कई प्रोसेसेस (processes) के बीच शिड्यूलिंग नीति (Scheduling Policy) द्वारा शेयर किया जा सकता है।

जब कोई प्रोसेस क्रिएट (create) होती है, तो वह अन्य प्रोसेसेस को क्रिएट करती है और फिर नये प्रोसेसेस अन्य प्रोसेसेस को क्रिएट कर सकती हैं। जिस प्रोसेस ने अन्य प्रोसेस का निर्माण किया है उसे पैरेंट प्रोसेस (parent process) तथा नए प्रोसेसेस (new process) को उस प्रोसेस का चिल्ड्रेन प्रोसेस (children process) कहा जाता है। इस प्रकार एक प्रोसेस द्वारा अन्य प्रोसेसेस को क्रिएट (create) करने के कारण प्रोसेसेस ट्री (process tree) का निर्माण होता है।

### प्रोसेस स्थिति

निम्नलिखित में से कोई भी एक स्टेट (state) हो सकता है, अर्थात् एक प्रोसेस निम्नलिखित में किसी भी एक स्टेट (state) में हो सकती है।

#### 1. न्यू (New)

2. रेडी (Ready)
3. रनिंग (Running)
4. वेटिंग (Waiting)
5. टरमिनेटेड (Terminated)

### न्यू स्टेट (New State)

इस स्टेट में कोई भी प्रोसेस नयी बनती है।

### रेडी स्टेट (Ready State)

कोई भी प्रोसेस शुरू या चालू या क्रिएट (create) होने के पश्चात् शीघ्र ही (immediately) रेडी स्टेट (ready state) में आ जाती है। इस स्टेट में कोई भी प्रोसेस प्रोसेसर के एलोकेट (allocate) होने की प्रतीक्षा करती रहती है।

### रनिंग स्टेट (Running State)

जब किसी प्रोसेस के इन्सट्रक्शन्स (instructions) एक्जक्यूट करते रहते हैं, तो वह प्रोसेस रनिंग स्टेट (Running State) में कहलाती है। जब कोई प्रोसेस CPU तथा अन्य सिस्टम रिसोर्सेस से कंट्रोल (control) पाता है, तो वह एक्जक्यूट करना प्रारम्भ कर देती है। रनिंग प्रोसेस (running process) के एक्जक्यूट करने के दरम्यान इनपुट/आउटपुट (I/O) ऑपरेशन्स के क्रम से, रनिंग प्रोसेस (running process) किसी दूसरे प्रोसेस को एक्जक्यूट करने के लिए, कंट्रोल स्थानान्तरित कर सकती है। रनिंग प्रोसेस द्वारा इनपुट/आउटपुट (I/O) ऑपरेशन्स के दरम्यान कंट्रोल को किसी दूसरे प्रोसेस (जो रन करने के लिए रेडी (ready) होगा) या ऑपरेटिंग सिस्टम को स्थानान्तरित किया जाना, ऑपरेटिंग सिस्टम के शिड्यूलिंग नीति (scheduling policy) पर निर्भर करता है।

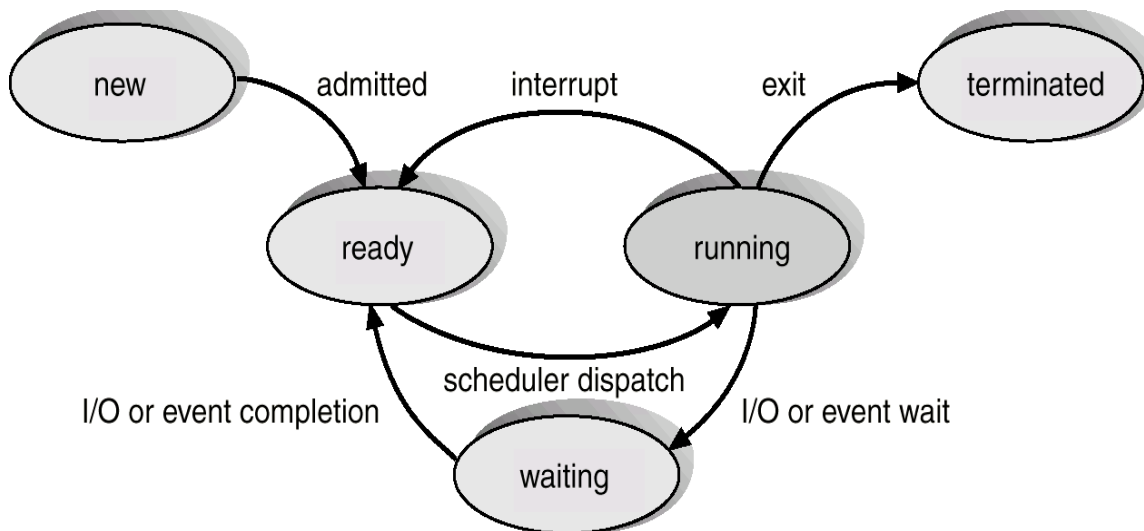
## OS NOTES : UNIT 1 AND 2

वेटिंग स्टेट (Waiting State)

इस स्टेट में कोई प्रोसेस किसी इवेंट (event) के घटित होने की प्रतीक्षा करती रहती है। जैसे, इनपुट/आउटपुट (I/O) ऑपरेशन्स के पूर्णरूपेण खत्म होने की प्रतीक्षा करना।

टरमिनेटेड (Terminated)

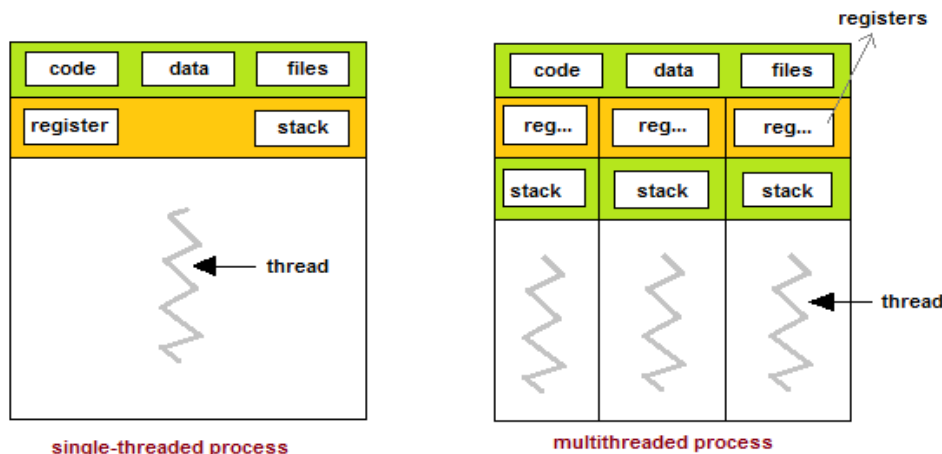
जब कोई प्रोसेस अपने अन्तिम स्टेटमेंट (last statement) को एक्जक्यूट कर लेती है तो वह ऑपरेटिंग सिस्टम द्वारा टरमिनेट कर दिया जाता है। प्रोसेस का यह स्टेट, टरमिनेटेड स्टेट (Terminated State) कहलाता है। टरमिनेटेड स्टेट में कोई प्रोसेस अपने पैरेंट (parent) प्रोसेस को डेटा भी रिटर्न (return) कर सकती है। किसी प्रोसेस को सिस्टम कॉल (system call) द्वारा क्रिएट किए गए दूसरे प्रोसेस से भी टरमिनेट



किया जा सकता है।

## ऑपरेटिंग सिस्टम में थ्रेड

किसी भी ऑपरेटिंग सिस्टम के प्रोसेस के अंदर थ्रेड execution का एक रास्ता होता है। इसके अलावे एक प्रोसेस के अंदर एक से ज्यादा थ्रेड भी हो सकते हैं। थ्रेड को लाइटवेट प्रोसेस भी कहा जाता है। किसी प्रोसेस को बहुत सारे थ्रेड में विभाजित कर देते हैं और स्मनान्तरीकरण को प्राप्त करते हैं। उदाहरण के लिए किसी ब्राउज़र में बहुत सारे टैब्स को थ्रेड की तरह देखा जा सकता है। एमएस वर्ड कई सारे थ्रेड का प्रयोग करता है- एक थ्रेड को टेक्स्ट को फॉर्मेट करने के लिए, दूसरे थ्रेड से प्रोसेस इनपुट करने के लिए, इत्यादि। मल्टीथ्रेडिंग के और भी फायदे हैं जिन्हें हम आपको बतायेंगे।



## प्रोसेस vs थ्रेड?

इन दोनों के बीच में सबसे बड़ा अंतर यही है कि किसी भी समान प्रोसेस के अंदर सारे थ्रेड शेयर्ड मेमोरी स्पेस में रन होता है जबकि प्रोसेसर अलग मेमोरी स्पेस में रन होते हैं।

थ्रेड्स प्रोसेस की तरह एक-दूसरे से स्वतंत्र नहीं होते और इसके परिणामस्वरूप थ्रेड्स कोड सेक्शन में दूसरे थ्रेड्स के साथ डाटा सेक्शन और ऑपरेटिंग सिस्टम रिसोर्सज जैसे कि ओपन फाइल्स और सिग्नल्स को शेयर करते हैं। लेकिन प्रोसेस की तरह ही थ्रेड का अपना अलग प्रोग्राम काउंटर (PC), रजिस्टर सेट और एक स्टैक स्पेस होता है।

### थ्रेड के प्रोसेस पर फायदे

1. *Responsiveness*: अगर प्रोसेस को एक से ज्यादा थ्रेड में विभाजित कर दिया जाए और एक थ्रेड अपना execution को पूरा कर ले तब इसके आउटपुट को जल्द से जल्द रिस्पांस दिया जा सकता है।

2. *Faster context switch*: थ्रेड्स के बीच कॉन्टेक्स्ट स्विच टाइम प्रोसेस कॉन्टेक्स्ट स्विच के टाइम से बहुत कम होता है। प्रोसेस कॉन्टेक्स्ट स्विच का अर्थ हुआ CPU के लिए ज्यादा ओवरहेड।

3. *Effective Utilization of Multiprocessor system*: अगर हमारे पास एक सिंगल प्रोसेस में एक से ज्यादा थ्रेड हो तब हम एक से ज्यादा प्रोसेसर में एक से ज्यादा थ्रेड को शेड्यूल कर सकते हैं। ये प्रोसेस के एक्सीक्यूट होने की गति को बढ़ा देता है।

4. *Resource sharing*: रिसोर्सज जैसे कि कोड, डाटा और फाइल को किसी प्रोसेस के अंदर सभी थ्रेड के बीच शेयर किया जा सकता है।

नोट: स्टैक और रजिस्टर को थ्रेड्स के बीच शेयर नहीं किया जा सकता। हर थ्रेड का अपना स्टैक और रजिस्टर होता है।

**5. Communication:** एक से ज्यादा थ्रेड्स के बीच का संवाद आसान होता है क्योंकि थ्रेड एक ही एड्रेस स्पेस को साझा करते हैं। जबकि हमें प्रोसेस में दो प्रोसेस के बीच कम्युनिकेशन के लिए कुछ खास कम्युनिकेशन तकनीक को फॉलो करना पड़ता है।

**6. Enhanced Throughput of the system:** अगर प्रोसेस को बहुत सारे थ्रेड्स में विभाजित किया गया है और हर थ्रेड फंक्शन को एक जॉब की तरह देखा जाता है तब एक यूनिट समय में पूरे होने वाले जॉब्स की संख्या बढ़ जाती है। इसीलिए सिस्टम का थ्रूपुट भी बढ़ता है।

## **यूजर लेवल थ्रेड और कर्नेल लेवल थ्रेड**

थ्रेड्स दो प्रकार के होते हैं- यूजर लेवल थ्रेड और कर्नेल लेवल थ्रेड। इन दोनों को आप नीचे दिए गये चार्ट की मदद से समझ सकते हैं:

<b>यूजर लेवल थ्रेड</b>	<b>कर्नेल लेवल थ्रेड</b>
यूजर थ्रेड को यूजर द्वारा इम्प्लेमेंट किया जाता है।	कर्नेल थ्रेड को ऑपरेटिंग सिस्टम द्वारा इम्प्लेमेंट किया जाता है।
ऑपरेटिंग सिस्टम यूजर लेवल थ्रेड को नहीं पहचान पाता।	कर्नेल थ्रेड को ऑपरेटिंग सिस्टम पहचान लेते हैं।
यूजर थ्रेड को आसानी से इम्प्लेमेंट किया जा सकता है।	कर्नेल थ्रेड का इम्प्लीमेंटेशन कठिन है।
कॉन्टेक्स्ट स्विच टाइम कम होता है।	कॉन्टेक्स्ट स्विच टाइम ज्यादा होता है।

## OS NOTES : UNIT 1 AND 2

कॉन्टेक्स्ट स्विच को किसी हार्डवेयर सपोर्ट की जरूरत नहीं पड़ती।	हार्डवेयर सपोर्ट की जरूरत पड़ती है।
अगर एक यूजर लेवल थ्रेड ब्लॉकिंग ऑपरेशन परफॉर्म करता है तब सारे प्रोसेस को ब्लाक कर दिया जाता है।	अगर एक कर्नेल थ्रेड ऑपरेशन ब्लॉकिंग ऑपरेशन परफॉर्म करता है तो बांकी थ्रेड execution जारी रख सकते हैं।
उदाहरण: जावा थ्रेड, POSIX थ्रेड्स इत्यादि।	उदाहरण: विंडो सोलारिस

### Question

#### PART A

1. What is OS
2. What are services of OS
3. Write types of OS
4. What are the functions of OS
5. What is system call
6. What is process
7. What is pcb
8. Why scheduling require
9. Write full form of FCFS
10. What is process state
11. Write any 3 names of OS
12. Write name of states
13. What are scheduling criteria
14. What is real time OS
15. What is thread
16. What is kernel
17. What is batch system
18. What is Primitive algo



## OS NOTES : UNIT 1 AND 2

---

19. What is linux
20. What is multithreading

### PART B

1. Explain types of OS
2. What is process ? explain its state
3. What is threads explain types
4. What is cpu scheduling? Explain
5. Explain PCB

### PART C

1. Explain following scheduling algorithm with example  
Fdfs Round robin Priority
2. What is OS explain various types and services in details
3. What is thread ? explain in details