



---

# OPREATING SYSTEM

---

Paktia University



**Shafiullah Samoon**

**Mobile Num:0771003070**

**Gmail:Shafikhan5692@gmail.com**

OCTOBER 29, 2023

PAKTIA UNIVRSITY  
Logar Afghanistan

صفحه	عنوان
3-4	Software and Types
4	System Software
4	Operating System
4	Need For O/S
4	History of O/S
4-5	First second third and fourth Generation of O/S
4-5	Function Of O/S
5-6	Components Of O/S
5-7-8	Types Of O/S
8-10	Types of Structure
10-11	Services Of O/S
11	Program Execution
11	I/O Operation
11-12	File manipulation and management
12	Commination
12	Error handling
12	Resource Management
12	Security and Security
13	Networking
13	User Interface
13	What is process and O/S
13	Difference between process and program
13	Attribute or characteristics of process
13	Process id
13	Process state
13	Process counter
13	Priority
13	General purpose register
13-14	List of open files
14	Account information
14	Cpu scheduling information
14	Process control block, process life cycle states of process
15	Process scheduling, categories of scheduling, process scheduling
15-16	Types of scheduling, what is scheduling algorithm, purpose of scheduling
16-17	Scheduling strategy, types of cpu scheduling, FCFS, SJF, LGF
17-18	Priority scheduling, shortest remaining time first, round robin scheduling Algorithm
18-19	Multi-level scheduling algorithm, context switch, threads in O/S
19-20	Why do we need threads in O/S, threads VS process, components of thread
21-22	Types of threads, advantages of threads, parallelism
22-23	Concurrency, process in synchronization, types of synchronization
21-22	Conversance, synchronization پروسی دولونه
22-23	Race condition, critical solution, critical solution problem
23-24	Types of solution, software-based solution, hardware-based solution
24-25	Semaphore, dead lock, dead lock condition,
26	handling dead lock, Dead lock handling strategies, memory management

26-27	Memory management technique, memory management tech
27	Contiguous, non-contiguous, paging
28	Segmentation, paging, fragmentation
28-29	له منځه وړی شو internal fragmentation څنګه
29	Virtual memory
30	Demand paging, page replacement algorithm
30	FIFO, OPR, LRU, MRU, trashing
30	I/O management, device driver, device control
31	Synchronous VS asynchronous, Memory-mapped I/O , communication I/O devices
32	DMA, polling VS interrupts, storage management
32-33	Storage management key attributes, disk management
33	Goals of disk managements, task of disk management
33	Disk scheduling, disk scheduling algorithm
33-34	FCFS, SSTF, Secom, file system
34	Advantages and dis advantages of file system
34-35	Various file system, FAT, NTFS, EXT, HFS, APFS
35	Append, truncate
35	Sequential access, direct random access
35	Indexed sequential access
35	What is CIA triad, Confidentiality, Integrity, Availability
36	Threats, Physical Threats,
36	Non-physical threats

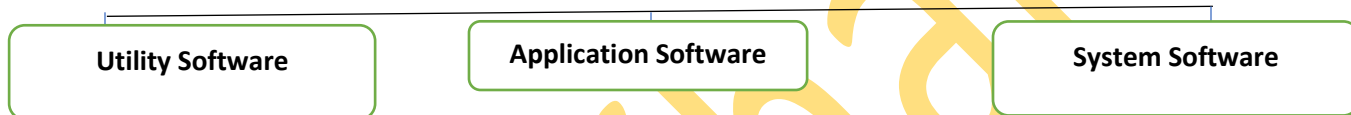
### سافټوېر څه شي ده؟

د هداياتو او لارښوونو مجموعه چې د انسانانو له خوا دندې هدايات ورکړل شوي ترڅو د user لپاره خاص وظيفي اجرا کړي. يا په بل عبارت کمپيوټر او د کمپيوټر اړوند ټولو ډيوایسونو ته لارښونه کوي. سافټوېر د باينري ژبه په اساس جوړ شوی لومړنیو پروگرامرانو د سافټوېر جوړولو له پاره له باينري يا ماشين (language) څخه کار اخيسته خو اوس پروگرامران دا کوشش کوي چې په high level ژبه کې پروگرام جوړ کړي. د هداياتو او لارښوونو مجموعه چې يوږز ته ددې اجازه ورکوي تر څو يوڅانگړی دنده تر سره کړي سافټوېر د کمپيوټر ټولو ډيوایسونو ته لارښونه کوي. د هداياتو او لارښوونو مجموعه ده چې کمپيوټر ته وايي چې تر څو څه تر څه تر سره کړي بيا ځه له سافټوېر څخه کمپيوټر د استعمال وړ ندی

### Classification of Software

دلته د مقصد (purpose) له مخې سافټوېر په درې ډوله دی.

#### Purpose



System Software : هغه سافټوېر ده چې د سیستم د چلولو دنده په غاړه لري .

Application Software : هغه سافټوېر ده چې د يوه خاص کار د ترسره کولو لپاره رامنځ ته کيږي.

Utility Software : هغه سافټوېر ته وايي چې د سیستم او اپليکشن سافټوېر سره مرسته کوي.

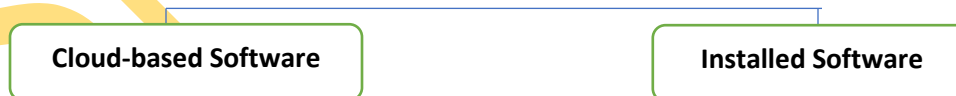
#### Platform



Native Software : هغه سافټوېر ته وايي چې د يو خاص اپريټينگ سيستم لپاره ډيزاين شوی وی او په هغه اپريټينگ سيستم باندې کار کوي .

Cross Platform : هغه سافټوېر ده چې په هر اپريټينگ سيستم باندې کار کوي.

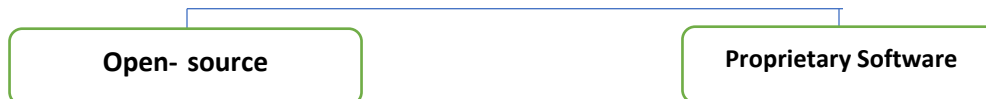
#### Deployment



Installed Software : هغه سافټوېر ته وايي چې په اپريټينگ باندې انسټال کيږي. لکه اکسيل او داسی نور.....

Cloud -based Software : هغه سافټوېر ته وايي چې د انټرنیټ په مرسته ورته لاس رسې پيدا کيږي. لکه پاڅی گيم.

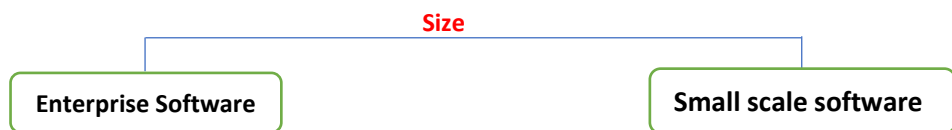
#### License



Proprietary Software : هغه سافټوېر ته وايي چې يو کمپنی يې جوړوی او بازار ته يې وړاندې کوي او هر څوک يې د

پيسو په مقابل کې اخيستلی شی

Open-Source: هغه سافټوېر ته وايي چې پرته له پيسو څخه يې اخيستلي شو لکه : لينکس .



**Small scale software**: هغه چې د يو شخص يا يو کوچني گروپ لپاره جوړ شوی وي.

**Enterprise software**: هغه سافټوېر چې د يوې لويې ادارې لپاره ډيزاين شوی وي.

د user interface له مخې سافټوېر په دوه ډوله دی.

1 - GUI software : graphical user interface يې لکه ډيندوز

2 - CLI software : command line interface

د (based) او (copy right) له مخې سافټوېر په څلور ډوله ده.

1 - Commercial software

2 - Shareware software

3 - Freeware software

4 - Public domain software

### System software

دا هغه سافټوېر دی چې په مستقيم ډول د کمپيوټر د هارډوېر سره کار لري او هارډوېر منيچ کوي. نور سافټوېرونو ته د انسټال کېدو زمينه برابروي، او په کمپيوټر کې ورته د استفاده کولو زمينه ورته برابروي. او د کمپيوټر او يوژر تر منځ ارتباط برابروي. د کمپيوټر داخلي دندې او د هارډوېر برخې لکه مانيټور پرينټر اوداسې نورې برخې کنټرولوي.

### Various types of system software

په درې ډوله دی: 1- operating system 2- device driver 3- language processor

### :Operating System

کمپيوټر User ته په فعاليت راولي. يا هم د کمپيوټر او User تر منځ د ارتباط پول رامنځ ته کوي. يا هم د کمپيوټر او يوژر تر منځ د انټرفيس خاصيت لري.

### Need for Operating System

O/s د نورو سافټوېرونو لپاره د يو platform په شکل دنده ترسره کوي. او دغه د نور پروگرامونه يا سافټوېرونو لپاره زمينه مساعده کوي. تر څو په کمپيوټر کې استال او د استفاده وړ وگرځي. د مختلفو تخنيکونو څخه په استازي سره امنيت (security) رامنځته کول. بغير له OS څخه کمپيوټر د استعمال وړ نه ده.

### History of Operating System

Operating system څلور نسلونه لري.

#### First Generation

دغه جنريشن له ۱۹۵۵-۱۹۴۵ کال پورې رامنځ لومړی ځل لپاره ميخانيکي کمپيوټرونه رامنځته شول. په دغه جنريشن کې O/s سيستم نه وو په دغه کې له Plug Board څخه استفاده کېدله. له ۱۹۴۵-۱۹۵۵ کال پورې په کمپيوټر کې کوم OS نه وو په

مستقیم ډول کمپیوټر ته هدايات کيدل. کله چې په ۱۹۴۰ کال کې ميخانيکي کمپیوټر رامنځته شوی وو په هغه کې هم OS نه وو په Machine language کې به ورته code کيدو مثال يی plug board ده.

### Second Generation Operating System

دغه په ۱۹۶۵-۱۹۵۵ کال کې رامنځ ته شو پدغه o/s کې يو O/s رامنځ ته شوچې ددغه O/S نوم GMOS(General Motor operating system) او دغه لومړنی O/s وو چې رامنځ ته شو. پدغه o/s باندی په ۱۹۵۰ کال کې کار پيل شوی وو. دغه O/s د IBM کمپیوټرونو لپاره رامنځ ته شوی وو او دغه o/s د RAP په نوم کمپنی لخوا رامنځ ته شوی وو او په دغه جنيريشن کې يو بل o/s هم رامنځ ته شوی دی چې نوم يی UNIX وو او په ۱۹۶۰ کال کې رامنځ ته شو.دا OS د (North American Aviation) له خوا معرفی شو. مثال يی batch OS ده.

### Third Generation Operating System

دغه Generation O/S له ۱۹۸۰-۱۹۶۵ کال پوری رامنځ ته شو او په دغه Generation کې د Multi Programing کانسيپټ رامنځ ته شو چې په يو کمپیوټر کې زياتی دندی ترسره کړو. او هم Multi O/S رامنځ ته شو،يعنی پدغه کې CPU بيکاره نه پاتی کيدی. پدغه کې کله چې يوه وظيفه ترسره نشی بله وظيفه راشی دی ته Multi Programing وايی. DEC-PDP په ۱۹۶۱ کال کې د Main Computer پدغه جنيريشن کې رامنځ ته شوی دی.

### Fourth Generation Operating System

د شخصی Personal کمپیوټرونو هر کارونکي دغه نسل استعمالولی شی د شخصی کمپیوټرونو کانسيپټ د mini computer سره يو شی دی. مايکرو سافټ ويندوز په ۱۹۷۵ کال کې رامنځته شو. دغه O/s له ۱۹۸۰ څخه پيل او تر ننه پوری دوام لری يا هم تر ننه پوری استفاده کيدی. پدغه Generation O/s کې د Multi-tasking کانسيپټ رامنځ ته شو.چې دغه به په يو وخت کې زياتو کسانو ته خدمات وړاندی کول.

### کمپیوټر څلور اساسی برخي لری.

Input      Output      Processing      storage

او O/s د کمپیوټر د برخو مدیریت په غاړه لری. او ټول فایلونه منظم کوی په کمپیوټر کې o/s ډیټا خوندي ساتي تر څو Unauthorized (غیري مسوول) کسان ورته لاس رسی پیدا نکړی.پرته له O/s څخه کمپیوټر د استفادی وړ نه ده.

### Important function of an Operating System

- Memory Management
- Process Management
- Device Management
- File Management
- Security
- Control Over System performance
- Job Accounting
- Error deleting aids
- Coordination between other software and User

### Component of Operating System

#### o/s دري اساسی برخي لری.

- Kernel: دا د o/s هسته ده او د کمپیوټر هارډوېر اداره کوی او دغه د کمپیوټر هارډوېر ته تر ټولو نژدی برخه ده.

د یوزر، منابعو او هارډویر تر منځ د پل رول لوبوی، او کومی غلطی چې ترسره کیږی د هغه د سمبالښت دنده هم پر غاړه لری.

• **Shell:** دا د کمپیوټر او یوزر تر منځ د interface رول لوبوی، او دغه د یو ژباړن رول لوبوی او Shell یوزر ته ډېر نږدی دی. کومی پروسی چې د o/s پواسطه سرته رسیری دا د Shell پواسطه یوزر ته وړاندی کیږی، دغه Shell په دوه ډوله دی **CLI** او دوهم **GUI**. د کمپیوټر kernel په **high level** ژبه نه پوهیږی همدا shell دی چې داسی ژبی ته ډیټا ژباړی ترڅو OS کرنل درباندي پوه شی.

• **File System:** File : فایل د ارتباط لرونکو معلوماتو مجموعه ده اودا فایلونه په HDD یا secondary میموری کی ساتی خامو موادو ته وایی لکه ویډیو، اډیو، انځور، او ډاکومنټ ته فایل ویل کیږی او ددغه ټول چې مدیریت کوی دغه دنده د File System ده. د File System دنده دغه ده چې په کمپیوټر کی ډیټا په منظم ډول ذخیره کړی. د فولډر د تنظیم او سمبالښت دنده پر غاړه لری، دنده یی دا ده چې فولډر کی ډیټا په منظم شکل ذخیره کړی او دا په Logical شکل باندي ده. زمونږ Space ضایع نشی او لاس رسی ورته اسانه وی او وخت ضایع نشی. بعضی مهم File System عبارت دی له :

- FAT (File Allocation Table): دا پخوانی ډول د فایلونو دی چې پخوانیو ویندوزگانو کی استعمالیده.
- Ext (Extended File System): په Linux او Unix کی کارول کیږی.
- NTFS (New Technology File System): منځینی فایل سیستم ډول دی. چی ویندوز کی استعمالیږی.
- HFS (Hierarchical File System): په mac os په کارول کیږی.
- APFS (Apple File System): نوی فایل سیستم چی په apple او mac په I/O الو کی کارول کیږی.

### Types of Operating System

**Batch Operating System:** دغه هغه O/s ده چې په دوهم جنیریشن کی رامنځ ته شو په دغه O/s کی ټولی وظیفی په گروپونو ویشل کیږی او کومی وظیفی چې یو بل ته ورته والی ولری هغه په هماغه گروپ کی ځای پر ځای کیږی. او کومی دندی چې لومړی راغلی وی هغه به لومړی ترسره کیږی او په یوه وخت کی به یواځی یوه وظیفه تر سره کیږی. په ۱۹۷۰ کال کی ډېر مشهور شو، په دغه O/s کی کمپیوټر مستقیمه اړیکه له یوډر سره نه درلوده.

### Advantage of Batch O/S

- دغه O/s په دی ښه پوهیږی چې یوه دنده به څومره وخت ونیسی.
- ډېرو خلکو ته په یو وخت کی خدمات وړاندی کوی.
- ډېر وزگار نه پاتی کیږی.
- په دغه ډېر غټ کار Manage کولی شو.

### Disadvantage of Batch O/S

- لومړی باید یوډر د Batch O/S په اړه سم معلومات ولری کله چې تیروتنه رامنځ ته شی دغه تیروتنه پیدا کول ډېر گران کار دی.
- دغه O/S قیمت وی.
- کله چې په دغه O/S کی لومړی دنده ناکامه شی نو هغه وخت پوری نه پریږی کله چې یی بشپړه کړی نه وی.

### Multi programming o/s

دغه ډېرو وظیفو ته اجازه ورکوی ترڅو په Main memory کی واوسیږی. پدغه کی کله چې یو دنده تر سره نشی بل دنده باندي کار کوی او بله دنده تر سره کوی، او دغه ډېر کار په کم وخت کی ترسره کوی.

### Advantage of Multi programming O/S

- په دغه کی لوړه او موثره CPU کارول کیږی.
- CPU ځینی پرگرامونه ته خپل وخت ټاکي او کار ورباندي ترسره کوی.

### Multi programming O/S disadvantage of

- په دی کی CPU ته مهالویش ته ضرورت لری.
- په main memory کی ډبرې دندی نه مدیریت کوی.

### Multi-Processing Operating System

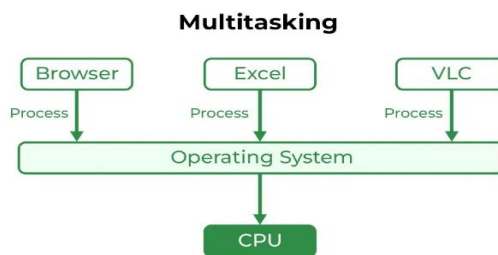
ډبر کارونه په یو وخت کی ترسره کوی ډبر کارونه په کم وخت کی ترسره کوی په دغه کی میموری مشترکه ده. او په دغه کی د CPU شمیر زیات دی.

### Advantage of Multi-Process Operating System

- اعتبار یی زیات ده
- د ډبر وخت د ضایع کېدو مخنتوی کوی
- تاوان یی دا ده : پیچلی دی خکه دا په ورته وخت کی ډبرې CPU کاروی.

### Multi-tasking Operating system

دغه O/S ډبر کارونه په یو وخت کی ترسره کوی ډبرې وظیفی په عینی وخت کی تر سره کوی او په دغه کی مونږه یه یوه CPU او میموری باندی زیات کارونه کولی شو



### Advantage of Multi-Tasking Operating System

- په ورته وخت کی د ډبرو کارونو ملاتړ کوی زیات کارونه په یو وخت کی کوی.
- Memory management ښه تر سره کوی.
- تاوان یی دا ده: ډبر پروسیسرونه په یو وخت کی بوخت وی تر څو کارونه بشپړ کړی نو CPU ډبره تودوخه تولیدوی.

### Network Operating system

کله چې موږ له یوه کمپیوټر څخه بل کمپیوټر ته معلومات شپړ کوو د هغه معلوماتو اداره کول او د هغه mange کول د معلوماتو امنیت دنده هم په غاړه لری د File system دنده هم په غاړه لری کمپیوټر ته څوک لاس رسی ولری او څوک یی ونلری دغه دنده هم پر غاړه لری. په دی زیات کمپیوټرونه وصل کیږی او دا ټول د یو سرور څخه کنټرول کیږی او ددوی سکوروټی هم پر غاړه لری

### Advantage and Disadvantage of Network o/s

- مرکزی سرور لری، امنیتي گواښونه یی د سرور له لاری اداره کیږی.
- دغه نوی تکنالوژی ده
- سرور ته لاس رسی د مختلو خابونو څخه کیږی.
- تاوان یی دا ده : سرور یی گران ده، ساتنه او تازه کول یی په منظم ډول اړین دی.

### Real time Operating system

دغه o/s د وخت په اساس کار کوی هر ی دندی ته وخت ورکوی چې دنده تر سره کیږی او هر دنده په خپل وخت تر سره کوی کوم چې ورته ټاکل شوی وی. لکه ترافیکی اشاری. چې د مشخص وخت لپاره عیار شوی ده او په هماغه اساس کار کوی.

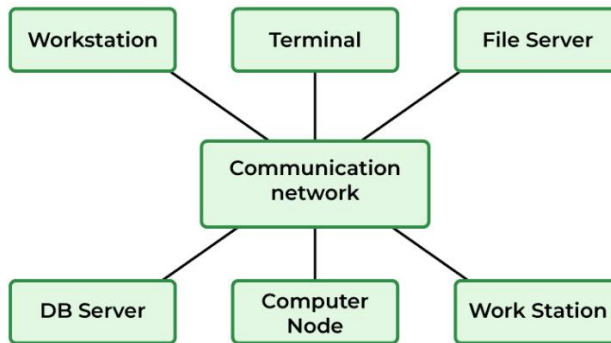


### Time sharing Operating system

زیاتو User ته خدمات وړاندی کوی خو د وخت په اساس وخت دا توپه کوی د هر شی لپاره وخت ویشی.

### Distributed Operating system

پدغه o/s کی مختلف کمپیوتر سره وصل شوی وی ټولی وظیفی په یو وخت کی تر سره کوی او په گډه سره یی ترسره کوی په دغه O/S کی میموری بیل وی او پروسیس بیل وی. کمپیوترونه په خپل منځ کی سره وصل کیږی خو د یوه سرور څخه کنټرول کیږی په هر یو کی مختلف پرگرامونه انسټال کیږی او مختلفې دندی په یو ځای تر سره کوی.



### Structures of OS

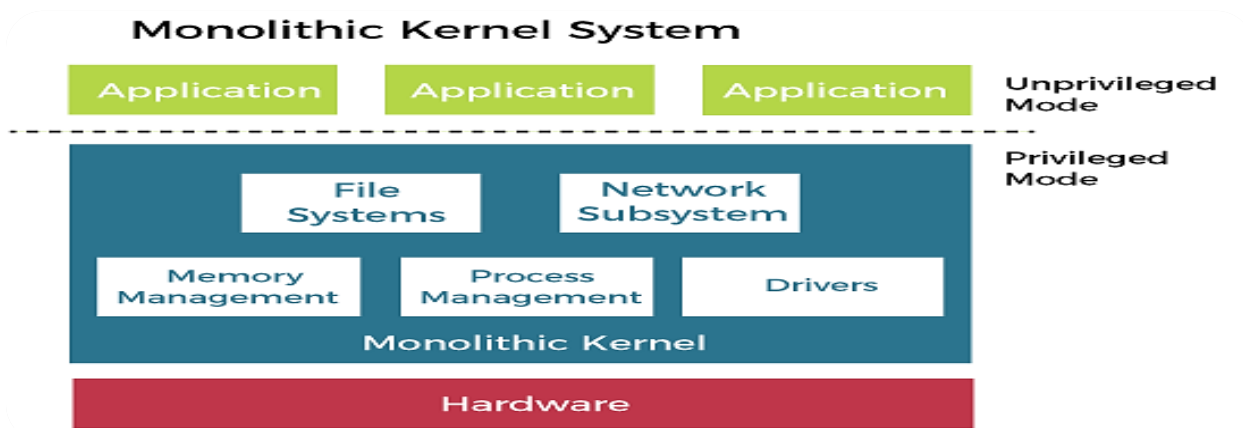
#### Simple Stature

په دغه ساختمان کی موږ څلور Layer لرو په دغه کی لومړی Layer له څلورم Layer سره یا آخری Layer ته لاس رسی لری. په دغه کی کله چې Application Layer خراب شی نو مکمل سیستم ورسره Crash کیږی. دغه ساختمان د لومړی ځل لپاره په Dos کی رامنځ ته شوی وو، دغه Stature خوندي ندی او دغه تر ټولو ساده سټرکچر ده.



### Monolithic structure

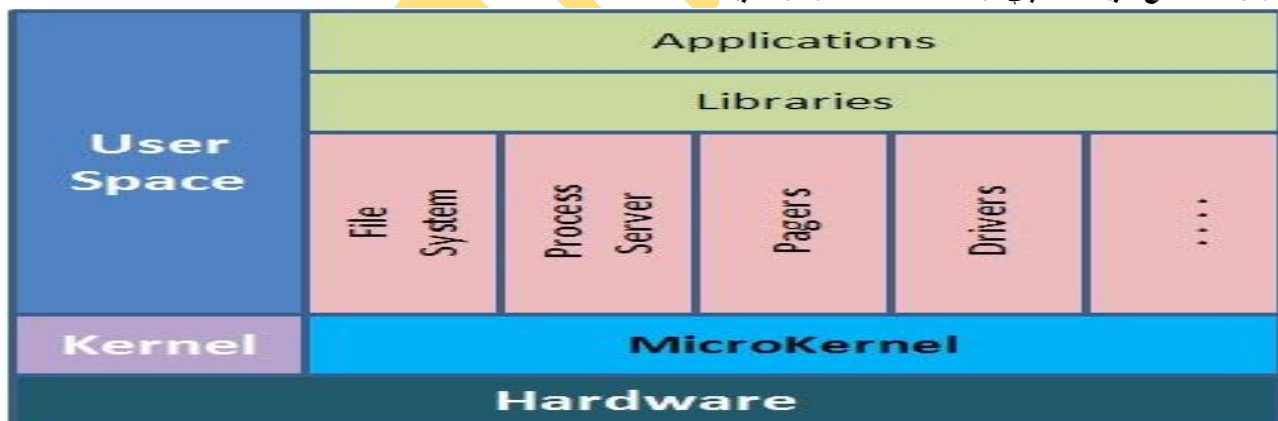
په دغه Layer کې ټولې Function پکې په یوه Layer کې یوځای کېږي. دغه structure په Unix کې رامنځ ته شو په دغه structure کې Debugging پکې سخت ده. په دغه structure کې Kernel وجود لري. او ددغه structure مراقبت سخته ده. ددی ته څه ور اضافه کول او کمول ترې سخت دی. د user space څخه یوازې call کیږي او kernel space ورته جواب وایی. اپریټینګ سیستم دوه ډوله Space لري: 1- User space 2- kernel space



### Micro-kernel structure

په دغه ساختمان کې موږ ده Space لرو. لومړی User Mode دوهم Kernel Mode په دغه structure کې غیر ضروری دندی User Mode ته ورکول کېږي.

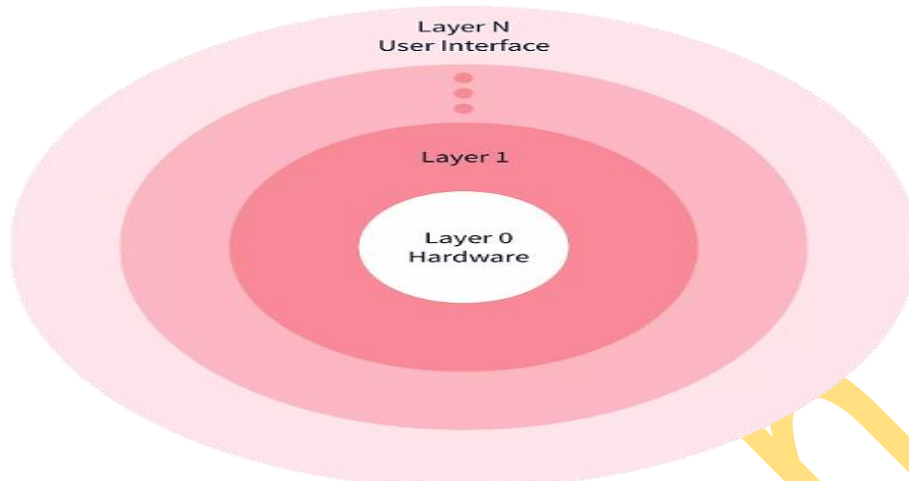
Kernel Mode: په دغه کې System Program پراته وی. د message Passing پواسطه د Client پرگرامونه او سرور ترمنځ ارتباط رامنځ ته کوي، دغه structure موثره نده دغه structure چې ده time west ده. تاوان یې دا ده په دغه کې ډیر وخت ضایع کېږي کله چې یوه Function ترسره کېږي.



### Layered approach structure

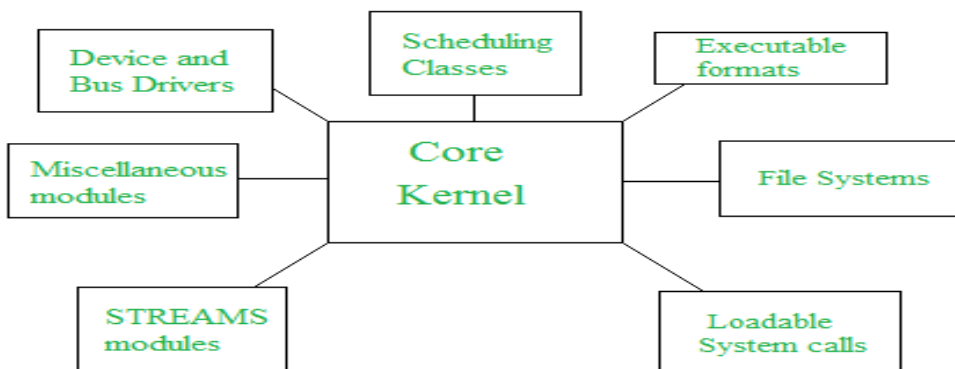
دغه Layer ډیزاین کولو موقیعت سخت ده. دغه Slow structure دی، په دغه کې Debugging اسانه ده، دغه structure خوندي ده، او time west ده، موثره نده، هر Layer کې چې ستونزه پیدا شي یواځې هماغه debugging کوي. که په یو layer کې ستونزه پیدا شي مستقیم لاسرسی ورته کرلی شو.

User mode: دوه برخې لري: 1- client 2- server



### Modular approach Structure

دغه سټرکچر د Layered او د Micro-kernel سره لږ ورته والی لری. هر ماډل پکې Core kernel ته لاس رسی لری، خوندي او موثره ده د Object oriented Programming څخه پکې کار اخیستل کیږی. دغه سټرکچر مهمی دندی راټولوی په دغه می سټرکچر کې Core kernel په منځ کې واقع ده هر برخه یی مستقیم له Core kernel سره اړیکه لری ددغه امنیت خوندي ده ځمکه هر ماډل یو دانه جدا Interface لری. دغه سټرکچر fast ده، تغیرپذیره ده.



### Service and function of OS

- د کمپیوټر او یوذر تر منځ د انټرفیس رول لوبوی.
- د o/s نورو پراگرامونو لپاره د platform رول لوبوی.
- o/s همیشه فعال وی چې کله چې کمپیوټر On شی نو o/s هم کار شروع کوی.
- o/s همغږی کوونکی ده د هارډوېر، یوذر او اپلکشن تر منځ.
- o/s کنټرول کوی او اداره کوی او File system manage کوی.
- کومی عملی چې په o/s کې ترسره کیږی هغه هم کنټرول کوی.
- کله چې دوه پروسی په یو وخت کې د پروسیس حالت کې شی OS د (dead lock) حالت ته ځی. مطلب هیڅ عملیه ته اجرا کوی.

### Services of o/s

Program Execution

Input Output Operations

File Management

Error Handling

Resource Management

Communication between Processes

Networking

System Utilities

User Interface

### Program Execution

دا د OS دنده ده چې يو پروگرام به څنگه Execute کيږي پروگرام مخکي له دینه چې execute شي له HDD نه باید RAM ته لاړ شي او د کومې عملیې په مرسته چې راغلي د OS په مرسته execute کيږي. Os د cpu scheduling algorithm په مرسته دا مشخصوي چې کومه پروسه باید لومړی execute شي او بیا کوم. مهم الگوریتمونه FCFS, SJF څخه عبارت دی. د پروگرام اجراء کیدل کله چې o/s انستال شي هغه بیا نورو پروگرامونو ته د اجراء کیدو زمینه برابروي، د o/s دنده دا ده چې نورو پروگرامونو څنگه اجراء کيږي. پروگرام په رم کې اجراء کيږي ځکه CPU ته نژدی وي وخت نه ضایع کيږي او پروسه به تیزوي. کله چې دوه پروسې په عیني وخت کې Exaction ته ولاړی شي نو په دا وخت deadlock رامنځ ته کيږي. o/s د یوډر پروگرام اپلکشن پروگرام او یوتیلتي پروگرام دنده پر غاړه لری تر څو چې په ښه ډول اجراء شي.

OS د Program execution لپاره لاندې خدمات وړاندې کوي.

Load a program into memory - 1

Execute the program - 2

Handle program execution - 3

Provides a mechanism for process synchronization - 4

Provides a mechanism for process communication - 5

### Input and output Operation

- د I/O عملی د I/O د ډیوایسونو پواسطه ترسره کيږي او دا ډیوایسونه O/S کنټرول کوي
- I/O ډیوایسونه په ډرایورنو کنټرول کيږي او دا ډرایورونه په O/S باندې کنټرول کيږي
- د o/s ډیوایسونو او ډرایورنو او یوزر ترمنځ ارتباط ممکن کوي.
- کله چې I/O ته لاس رسي ولرو دا د لاس رسي زمینه O/s راته برابروي.
- دغه عملیې د ډیوایس ډرایور په مرسته کنټروليزي او ډیوایس ډرایور بیا د OS په مرسته کنټروليزي.
- I/O operation means read or write operation
- O/S provide access to the required I/O devices

### File manipulation & management

فایل ته ویندوز کې فولدر او په Linux کې directories وایي. موږ ډیټا په دوه شکلونو ساتو ۱- فایل سیستم او ۲- ډیټابیس فایلونه د فایل سیستم په واسطه او فایل سیستم د o/s پواسطه کنټروليزي. O/s نورو پروگرامونه ته دا اجازه ورکوي ترڅو یو فایل read یا write کړي. فایل په خپل منځ کې د دقت لرونکي ډیټا مجموعه ده ده ته فایل ویل کيږي، کمپیوټرونه فایلونه په دایمي حافظه کې ذخیره کوي، په فایل سیستم کې ډیټا په ډایرک ډول ذخیره کوي، په فایل سیستم کې د فایل جوړول د فولدر ختم کول، فولدر ته لاس

رسی پیدا کولو ته فایل سیستم وایی او دغه فایل سیستم o/s کنترول کوی، د فایل سیستم د عملیو لپاره زمینه o/s برابروی یعنی دا چی فایل جوړ کړو او یا یې دیلیت کړو. د ډیټا بیکف واخلو دغه بیکف اخیستل د o/s کار ده.

### Communication

د سیستم په مختلفو برخو کې پروسیس ترسره کیری یا پروگرام د اجراء کیدو په حال کې وی نو ددوی تر منځ ارتباط رامنځ ته کول د o/s کار ده، مختلف سیستمونه د Communication Lines پواسطه وصل شوی دی، او دا د سیستم د پروسو تر منځ همغږی رامنځ ته کوی. کله چې د مختلف پروسو تر منځ ارتباط رامنځ ته کیری o/s په دی وخت کې لاندی پروسی ترسره کوی تر څو ارتباط په کامیابی تر سره شی،

✓ د ډیټا لیر رالیر صورت نیسی. هم په عینی سیستم کې او هم په مختلفو سیستمونو کې ارتباط رامنځ ته کیری. دا د o/s پواسطه ترسره کیری او په مختلفو سیستمونو کې ارتباط د Communication Lines پواسطه او په عینی وخت کې د Message prosing پواسطه تر سره کیری.

**Error Handling:** بعضی وخت په بعضی ځای کې واقع کیری، Error په CPU کې یا I/o لو کې یا هم د میموری هادویر کې را پیښیږی.

- او یو Error په یوې اله کې کیدای شی ډیر بد عمل وکړی په هماغه Device کې .
  - دا Error په هادویر او سافټویر کې چې راځی کله چې د لو ناکامه کیدل میموری Error په صفر تقسیمیدل .
  - ددی لپاره چې د error څخه مخنیوی وکړو نو باید د سیستم د افریشن مانیټور د یو سیستم د error د پیدا کولو لپاره او د یوه وړوکی تماس لرونکی اپلیکشن څخه استفاده وشی .
- لاندى مهمی دندی چی o/s یی په error handling کی ترسره کوی.

- O/S باید متواتر شکل پټ یا پنهانی errorسکن (scan) کړی.
- O/S یو ښه عمل صورت نیسی او گرنټی یا ضمانت ورکوی د دقیق او د اعتماد وړ محاسبه کوی.

### Resource management

- ❖ سیستم منابعی د مختلفی منابعو تر منځ شیر وی خو o/s resource sharing منیج کوی .
- ❖ او دا همچنان د cpu وخت منیج کوی ر پروسو تر منځ د cpu scheduling algorithm په استفادی سره .
- ❖ دا په یو سیستم کې د میموری منیجمنټ سره کمک کوی او د i/o devices کنترول کوی .
- ❖ د multi-tasking او multi user په محیط کې منابع لکه : main memory , cpu cycle and file storage مشخص شوی ده د هر یوزر او وظیفه لپاره .

لاندى غټ فعالیتونه د یو o/s ده چی په resource management کی ترسره کړی .

- O/S د Scheduler resources په استفاده هر قسم فعالیتونه منیج کوی.
- Cpu scheduling algorithm د ښه استفادی لپاره استعمالوی.

### Security and security

په o/s کې protection (دفاع، ساتنه) یو میکانیزم ده چی د پروسس، پروگرام یا یوزر د څو منابعو د کمپیوټر په سیستم کې چی کنترول کړی او اکسس کړی. o/s مطمین کوی چی ټولی منابعی د کمپیوټر کنترول او اکسس کړی. او دا مطمین ده چی ټول خارجی منابعی به د غلطی لاسرسی نه به خوندي وی. دا مونږ ته دا اعتبار جوړوی یا تهیه کوی د user name او password په استفادی سره. Protection یو میکانیزم ده یا هم یوه لاره ده چی مونږ ددی په واسطه کنترول کوو د یو پروگرام لاسرسی، پروسس او یا هم د یوزر لاسرسی منابعو ته چی د کمپیوټر سیستم په واسطه تشریح شوی.

لاندى ډیر فعالیتونه چی د o/s په واسطه په protection کی ترسره کیری .

- o/s مطمین کوی چی ټولی د سیستم منابع ته لاسرسی کنترول دی.
- o/s مطمین ده چی ټولی خارجی i/oالی خوندي دی له بی اعتباره لاسرسی څخه.
- o/s تهیه کوی معتبره یا صحیح اینده د هر یوزر لپاره د پاسورډ په معنی.

## Networking

دا خدمات رابطته جوړوی په نیټورک کی د ډیوایسونو ترمنځ لکه : د انټرنیټ وصل کول ، لیږل او را لیږل د data پاکټونو او نیټورک ارتباطاتو منیج کول . د o/s مثالونه چی په نیټورک کی تهیه شوی .

- د ویندوز سرور استفاده
- Linux سرور
- IOS چی د CISCO الو لخوا استفاده کیږی لکه ... switch, router and etc.

## User interface

User interface ډیر مهم ده او ټول o/s دغه تهیه کړی .  
یوزر کولای شی چی دواړه انټرفیس استفاده کړی په o/s کی CLI AND GUI

## What is process in Operating System

کله چی پرگرام د Execution په حالت کی شی په دغه وخت کی ورته process ویلی شو، پرگرام ته پروسیس نشو ویلی او پروسیس ته هم پرگرام نشو ویلی، کله چی پرگرام خلاص کړو او د اجراء کیدو حالت ته شی بیا ورته پروسیس ویلی شو.

Process یوه active Entity ده او program یو passive Entity ده . process په RAM کی وی او پرگرام په Hard disk کی وی .

## Difference between process and program

Process	Program
پروسیس پرگرام نه دی	پرگرام پروسیس نه دی
دغه process چی ده Active Entity ده	دغه passive Entity ده
دغه process په RAM کی دی	دا په Hard disk کی وی
دغه Memory ته CPU ته او I/O ته ضرورت لری .	دا یوازی Secondary memory یا Hard disk ته ضرورت لری
پروسیس د پرگرام د اجراء کیدو یوه بیلگه ده	یو پرگرام څو پروسیس لری
پروسیس په یو پرگرام پوری تړاو لری	

مثال : کله چی په یوه ژبه کی یو پرگرام ولیکو بیا هغه Run کړو دغه Run کولو ته Process and program ویل کیږی . پرگرام لومړی په hard disk کی وی کله چی موږ پری کلیک وکړه بیا Ram ته راځی .

## Attributes or Characteristics of a Process

دغه خاصیتونه چی پواسطه یو پروسیس یو له بل څخه بیل کولی شی،

**Process ID** : کله چی یو پروسیس رامنځ ته کیږی دا داسی پروسیس وی چی په دغه نمبر یا ID باندی بله پروسیس شتون نلری، دغه نمبر څکه نمبر څکه ورکول کیږی چی له نورو پرسو څخه بیل شی.

**Process state** : کله چی یوه پروسه اجراء کیږی ددغو لاندی پړاونو څخه نیږیږی یعنی له Creation څخه نیولی تر completion پوری یا به پروسیس د Read په حالت کی وی یا به د Running په حالت کی وی او یا به د Waiting په حالت کی وی .

**Program counter** : تر ټولو تیز ترینه میموری چی ده هغه رجسټر ده چی په CPU کی موقعیت لری دغف

ه د تیری پروسی ادرس ساتی کله چی نوی پروسیس تر سره شی نو بیا زاړه پروسیس داخل وی .

**Priority** : کله چی یو پروسیس تر سره کیږی نو هری پروسی ته یو Priority Num ورکول کیږی دی کومی پروسی چی Priority په High کی وی نو هغه به لومړی تر سره کیږی .

**General Purpose Register** : کله چی یو پروسیس تر سره کیږی نو ددغه پروسیس اړوند ټول معلوماتو ته ضرورت ده هغه معلومات په Register Memory کی ساتل کیږی د ټولو پروسیس په اړه معلومات ساتی په Running حالت کی یی .

**List of open files** : کله چی یوه پروسه اجراء کیږی یو لړ فایلونه ته ضرورت دی چی د پروسیس سره مرسته کوی .

یا هم ڪلهه جي پروسه ترسره ڪيڙي نو فابلونو ته ضرورت لري.

مثال: ڪلهه جي يو ڏوڪ پروگرام انسٽال ڪوو زيات فابلونو ورسره وي جي دا فابلونو پروگرام سره مرسته ڪوي د اجراء ڪيدو په وخت ڪي

**List of open devices:** پروسيس د اجراء ڪيدو په وخت ڪي مختلف Device ته ضرورت لري لکه: **i/o device, keyboard, mouse** او داسي نور.

**Accounts Information:** هري پروسه ته محدود وخت ورڪول ڪيڙي او د CPU اندازو به نظر ڪي نيول ڪيڙي جي ڏومره Space استعمال ڪيڙي. ڪلهه جي يو پروسيس تر سره ڪيڙي په اڙه معلومات ساتل ڪيڙي.

**CPU Scheduling information:** يعني په دي ڪي زياتي پروسه راڃي بايد ڪومه لومري ترسره شي يو تقسيم اوقات ورته جوڙوي جي ڏهه وخت بايد ڪومه پروسه ترسره ڪري.

### Process Control Block (PCB)

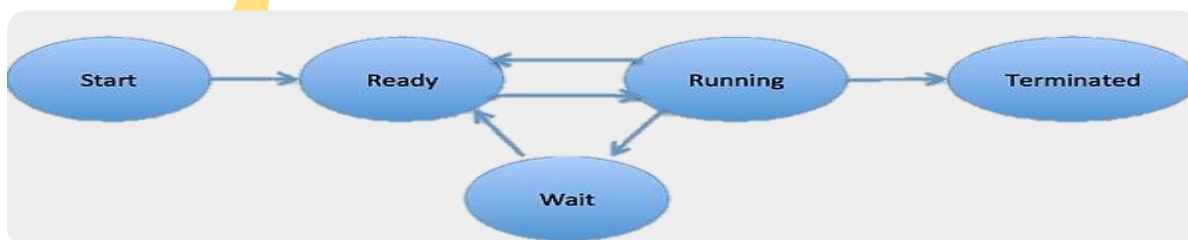
هغه ڄاي جي د يوي پروسه ٽول معلومات پڪي پراڻه وي دي ته Process Control Block ويڻ ڪيڙي.

Process ID
State
Pointer
Priority
Program Counter
CPU Register
I/O information
Account information

### Process Life Cycle states of a process

ڪلهه جي پروسه پيل ڪيڙي له پيل ڪيدو ڏهه تر پايه پوري ڪومو حالتونو جي تيريري دي مڪمل وائڻ ته Life Cycle وايي.

با هم ڪلهه جي د لومري ڄل لپاره يو پروسيس رامنڃنه شي دغه حالت ته



### Process life cycle or states of a process

**Start:** ڪلهه جي لومري ڄل لپاره يو پروسيس جوڙه يا شروع شي هغه ته وايي.

**Ready:** کله چې یوه پروسه ریم ته لاره شی او د اجراء کولو لپاره آماده شی دغه ته **Ready** وایي.

**Running:**

**Waiting** کله چې یو پروسیس تکمیل نشی او په حالت د انتظار کې پاتې شی یعنې **I/O** ته ضرورت وی یا د یوې پروسې خای بله پروسه ونسي پروسیس بیرته **Ready** حالت ته خي او بیا **Running** حالت ته خي.

**Terminated** کله چې یو پروسیس په بشپړ ډول سره تکمیل شی هغه ته **Terminated** وایي

### Process scheduling

دا د **Multi Programming** لپاره ډېره مهمه ده او دا **o/s** ته ددی اجازه ورکوي تر څو پوري ډېری پروسې په یو وخت کې په میموري کې اوسي، کله چې د یوې پروسې په ختمیدو سره بله پروسه انتخاب کوي دی ته **Process scheduling** وایي. کله چې یوه پروسه **Remove** کړي او پر خای یې بله انتخاب کړي.

### Categories of Scheduling

موږ دوه ډوله **Scheduling** لرو

**Non- preemptive:** کله چې یوه پروسه ورکول شی چې هغه خلاصه (تکمیل) شوی نه وی بیا بله نه اجراء کوي

کله چې د یوه پروسې لپاره یو منبع په لاس ورکول شی تر هغه یې نه پریري تر هغه یې نه پریري چې دا پروسه یې خلاصه کړي نه وی.

**Preemptive:** کله چې یوه پروسه ورکول شی هغه چې مستقیم منابعو ته ضرورت ولري تو هغه پریري بیا بله پروسه باندی کار کوي، با په دغه کې هرې پروسې ته وخت ورکول کیږي یعنی هرې پروسې ته خپله وخت ویشي.

**Program:** د لارښونو مجموعي ته پرگرام وایي مثال: **Home, insert**

**Software:** د پرگرامونو مجموعي ته سافتویر ویل کیږي. مثال: **Word power point**

### Process Scheduling

**Process manger:** دغه هغه پروس لري کوي کوم چې په حالت د **Process** کې وی او پر خای یې بل پروسیس مخکي کوي کیدای شی چې هغه پروسیس **High priority** ولري د پروسیس انتخاب کولو دنده هم پر غاړه لري.

دا د کارونه لپاره یو **table** جوړوي چې تر څو ټول کارونه په خپله وخت کې تر سره کړي د یوې پروسې په نه اجراء کېدو په صورت کې بله پروسې باندی کار کوي او وظیفی ته وخت ویشي چې کومه وظیفه په کوم وخت کې تر سره شی.

دغه د **Multi-programing** له مهمو **Function** څخه ده دغه **o/s** ته اجازه ورکوي چې په یوه وخت کې ډېری وظیفی په میموري کې ووسیږي. هرې وظیفی ته خپل وخت ورکوي او دغه پروسه لومړی وظیفه قطع کوي او پر خای یې بله پروسیس شروع کوي.

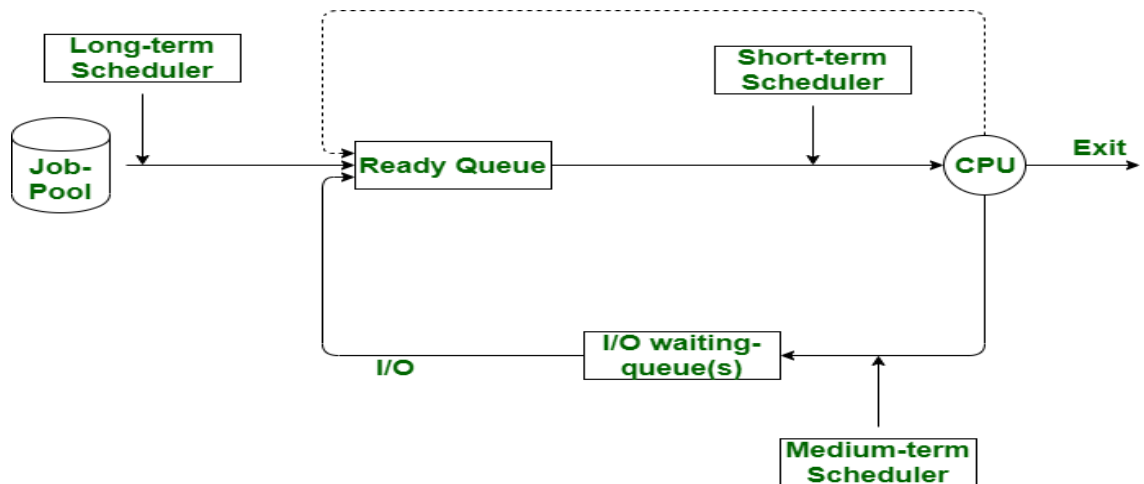
### Types of Scheduler

**1: Long time Scheduler:** دغه ته موږ **Job Scheduler** هم ویلي شو، دنده یې دا ده د **Secondary memory** څخه دندی راوړي **primary memory** ته هلته یې په **ready queue** کې اچوي.

**Short term scheduler:** دغه ته **CPU scheduler** هم وایي چې دندی یې **ready queue** نه انتخاب کړي د ترسره کیدو حالت لپاره دا پریکړه د مختلفو الگوریتمونو په اساس کوي.

**3: Medium term scheduler:** دا دنده داده چې **swapping** (تبدیل) تر سره کړي. یا هم دا دی یوه وظیفه یا پروسیس پر خای بل انتخاب کوي یعنې کله چې یو پروسیس **i/s** ته ضرورت ولري اجراء نشی د هغه پر خای بله وظیفه داخله کړي او په **CPU** کې د پخواني پروسې یو ادرس ساتي چې خومره کار مخکي پری شوی وی.





### What is scheduling Algorithm?

**algorithm**: د یو کار دی ترسره کولو لپاره یو تگلاره دی چی کوم وخت کی باید وشي یا هم چی دندو ترسره کولو یا د یوه کار ترسره کولو لپاره کومه تگلاره چی کار کوو ورته algorithm وایی. دغه معلوم وی چی کومه پروسه به اجرا کیږی خو په multi programming کی . کومه لومړی اجرا خومره وخت کی او بیا کومه اجرا شی دا scheduling algorithm کوی . هدف یی چی cpu بیکاره پاتی نه شی او وخت ضایع نه شی . په multi programming کی پروسو ته وخت ورکوی په منظم ډول یی ورکوی

او scheduling algorithm دا چی کله زیاتی پروسی په یو وخت کی cpu ته راخی د هغه منظم کول اداره کول چی کومه وظیفه په کوم وخت کی باید ترسره شی او لومړی کومه وظیفه ترسره شی هغه باندی کار کوی یوی پروسی لپاره منظم انتخاب او یو وخت انتخاب کوی کله چی یو پروسس ترسره شی په ډاډه سره بله پروسه ورته رسوی او وخت د ټولو پروسو تر منځ ویشی

### Purpose of Scheduling Algorithm

د سیستم منابع «ram, i/o, cpu» خاص utilise کیږی ترڅو سمه استفاده تری وکړی .

وخت مونږ ته بچت کوی یعنی د وخت ضایع کیدو څخه مخنیوی کوی  
cpu اداره او منظم کوی

**Through put** : د cpu په capability پوری مربوطه دی چی سومره توانایی لری . یا time په unit کی چی کوم کار ترسره کیږی ورته وایی .

**Turn round time**: Cpu پروسی په انتظار کی نه پاتی کیږی د پروسس وخت ته وایی .

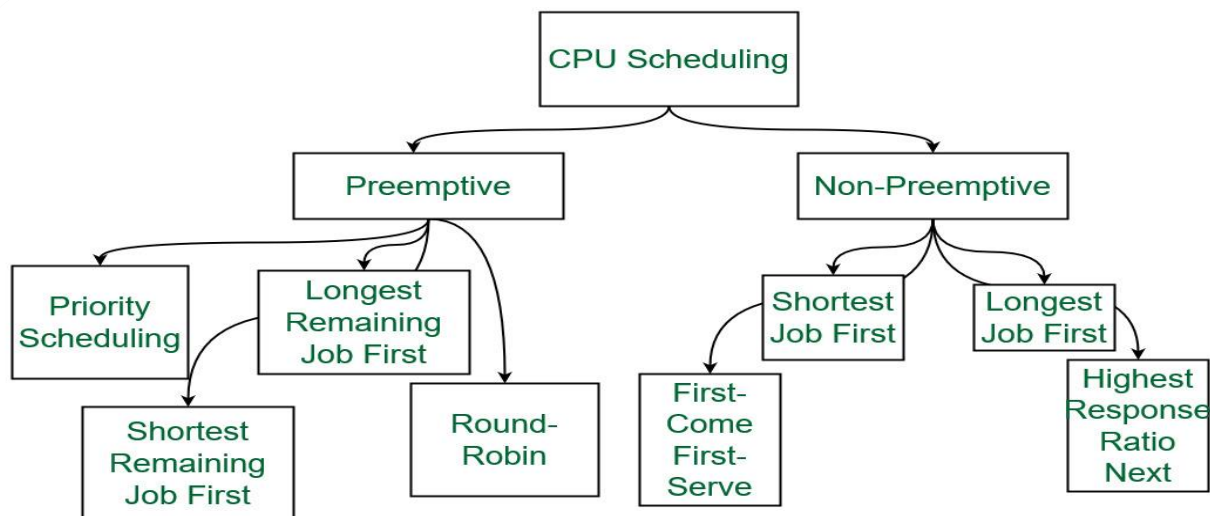
**Respond time**: cpu څخه د بیرته راتلو وخت ته وایی .

### Scheduling strategies

**Pre Emotive**: د یوی پروسی پر خای بله پروسه انتخاب کوی. په هغه حالت کی چی یوه پروسه I/O حالت ته ضرورت ولری او یا له high protiz سره راورسیږی یا کله چی یوه پروسه انتخاب کوی که هغه I/O ته ضرورت ولری هغه پریردی بیا بله انتخاب کوی هغه ترسره کوی دا تغیر منونکی ده .

**Non pre Emptive**: تر هغه وخت پوری به یو پروسی باندی کار کوی ترڅو یی خلاصه کوی نه وی هغه مشخص وخت به ترسره کوی . یا چی ددی په اساس یوه پروسه انتخاب شوه تر هغه وخت یی نه پریردی چی ترڅو یی نه ترسره کوی .

### CPU Scheduling Stricture



**Process** : د یو پروگرام د اجرا کیدو حالت ته پروسس ویل کیږی .

**FCFS**: فول شکل یی first come first serve لومړی وظیفه چی کوم راخی هغه به اجرا کیږی .

Process	Arrival time	Burst time
P1	0	4
P2	1	3
P3	2	1
P4	3	2
P5	4	5

**SJF(shortest job first)**: کوم کارونه چی کوچینی وی هغه به ترسره کیږی لومړی او دغه non preemptive ستراتیژی ده .

خانگرتیاوی : د هغه تیم اوسط یی کم وی ، او د هری دندی سره خپل وخت پکی مشخص وی .

**LGF( Longest Job first )**: دادی SJF مخالفه دی په دغه چی د کومی پروسس burst time زیات وی هغه لومړی ترسره کیږی دغه هم non preemptive ستراتیژی دی . یعنی هغه پروسه به لومړی اجرا کیږی چی زیات وخت نیسی .

### Priority scheduling

د اولیت پر اساس پروسه اجرا کیږی د کومی چی high priority وی لومړی به هغه اجرا کیږی .

هر پروسس ته د اولیت نمبر ورکول کیږی چی له دغه څخه وروسته short time server کولای شی چی یو پروسس انتخاب کړی څومره چی د یو پروسس نمبر لوړ وی هغومره به ژر ترسره کیږی بعضی وخت دوه پروسس یو خای cpu ته راشی چی یو رقم priority لری نو لومړی به هغه ترسره کیږی چی لومړی راغلی وی .

Burst time

Waiting time

Process	Arrival time	Executive time	Priority	Service time
P1	0	5	1	9
P2	1	3	2	6
P3	2	8	1	14
P4	3	6	3	0

### Priority algorithm scheduling

P3	P1	P0	P2
0	6	9	14
			22

کوم الگوریتم کی چی یو پروسس زیات وخت باسی یو ښه الگوریتم نه دی څومره چی کم انتظار باسی هغه ښه الگوریتم دی .

**Shortest remaining time first** : کوم دنده چی کم وخت لپاره تکمیل ته پاتی وی هغه ترسره کیږی او دغه یو preemptive الگوریتم دی په دغه که چیری بله وظیفه راغله او کوم وخت یی اړتیا وه هغه باندی کار کوی ځکه چی یو preemptive ستراتیژی ده .

**Round robin scheduling algorithm**: دا یو preemptive الگوریتم دی (پریښودونکی) هری پروسسی ته خاص یا fix وخت ورکوی د cpu وخت د ټولو پروسو ترمنځ ویشی که چیری هغه وظیفه په خپل وخت کی پوره شی ښو ښه به وی کنی بیا تری راگرخی او دغه fcfs ته ورته ده فرق یی داده چی ددی quantum time په اساس وظیفه اجرا کوی.

هغه وخت چی یو وظیفی ته ورکول کیږی هغه ته quantum time وایی او دغه وخت 100 meli sec , 10 meli sec پوری وی .

Process	Execution time or burst time
P1	24
P2	3
P3	3

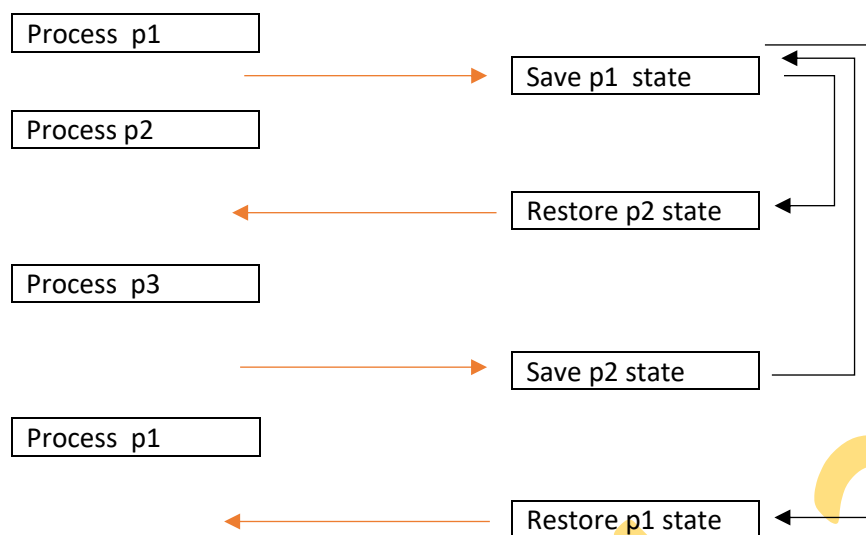
P1	P2	P3	P1	P1	P1	P1	P1
0	4	7	10	14	18	22	26
							30

**Multi-level scheduling algorithm** (سپیډی زیات دی ددی الگوریتم) : څو الگوریتمونه سره یو ځای کوی او په هغه باندی یوه وظیفه ترسره کوی . دغه الگوریتم هغه وظیفی سره یو ځای کوی چی مشترکی ځانگړنی سره ولری لکه : I/O ته ضرورت لری یو طرفه یی کوی ځینی یی بیا cpu وخت ته ضرورت لری هغه بل طرفه کوی دایی دنده ده . هرو queue ته یو priority ورکول کیږی او د queue گانو په داخل کی له Round Robin څخه استفاده کیږی .

**Context switch** (پریښودل ، شروع کول د یوی وظیفی)

کله چی یو پروسس پریښودل شی او بیرته چی له کوم ځای څخه شروع شی هغه د context switch دی .

په دغه کی دوه مهم کارونه Store & restore د یو وظیفی په اړه ټول معلومات په PCB کی ساتی او بیرته یو پروسس له هغه هلقه کی چی پاتی شوی له هغه ځای څخه یی بیرته restore کوی .



### Threads in operating system

یو پروسس چی په څو برخو وویشل شی هغی برخی ته یی threads وایی او دغه په logical سره په برخو ویشی او دغه threads له پروسس څخه رامنځته کیږی او دغه threads مونږ light weight process هم ورته وایی یعنی کم وزنه پروسه په یوه پروسه کی یو execution په path باندی ویشل کیږی. او د threads سره مونږ د cpu سپید (speed) زیاتیری او دی دوی اداره کول هم د O/S سیستم په غاړه دی هر یو threads له یوی پروسس سره اړیکه لری. او څو پروسس له یو threads سره اړیکه نلری او Threads هغه وخت ښه او تیز کار کوی چی cpu گانی یی زیاتی وی .

دی ته کم وزنه پروسه هم ویلای شو (light weight process)

په یو پروسس کی زیات functionalities رامنځته کیږی او هر function ته threads ویلای شو .

په یوه پروسه کی په ډیر کم توپیر راتلای شی .

کله چی څو threads په یوه وخت کی ترسره کیږی نو بیا process زر ترسره کیږی .

ددوی management هم د اپریتینگ سیستم په غاړه دی .

کله چی پروسس زیات وی نو multi threads ښه کار کوی .

د مسلسلو وظایفو جریان ته چی په یوه process کی وی ورته threads وایی .

### Why do we need threads in operating system?

ځکه چی د یو سیستم سرعت ورسره زیاتیری د threads رامنځته کول او جوړولو دا تیز ده نسبت process ته په threads کی context switch دا تیز ده نسبت process ته .

د رامنځته کولو او له تکمیل کوی د threads پکی تیز ده.

Context switching پکی تیز دی.

د ټولو threads په خپل منځ کی data & code سره شیر کوی او performance ورسره لوړیری .

**Context switch:** د یوی پروسس وړل او پر ځای یی بلی پروسس راوستلو ته وایی.

### Threads VS Process

**process:** زیاتو منابع ته ضرورت لری.

**Threads:** کمی منابع ته ضرورت لری.

**process:** creation & termination دغه د پروسس slow دی. د پروسسی اجرا کیدل پکی سلو دی

**Threads:** په دغه کی تیز دی .

**process:** هر پروسس خپل بیل کود او دیتا لری او راجیستر بی هم بیل دی .

**Threads:** پدی کی بیا code او Data شیر کوی خپل منخ کی یواخی راجیستر بیل وی پکی .

**process:** د پروسس په منخ کی communication ډیر slow دی .

**Threads:** خو په threads کی بیا تیز دی .

**process:** context switch په پروسس کی slow دی .

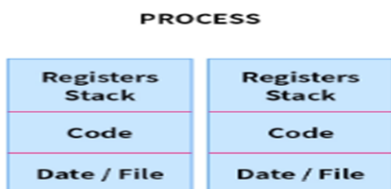
**Threads:** او پدی کی بیا تیز دی.

**process:** ټولی پروسسی یو له بل څخه خپلواکی دی .

**Threads:** په دی کی پروسسی یو له بل سره ارتباط لری .

Two different process

two threads of single process



Two Different Process

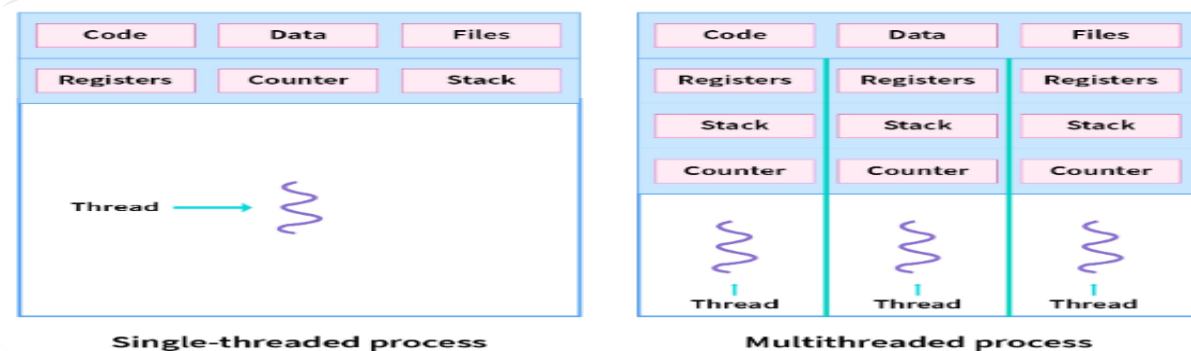


Two Threads of a single process

### Components of threads

Threads components ۳ دری برخی لری.

1. Program counter
2. Register set
3. Stack space



**Threads** په عموم کی په دوه ډوله دی.

1. User level thread (ULT)

2. Kernel level threads (KLT)

1. په یوزر space رامنځته کېږی او kernel تری خبر نه وی

- که چیری پکی یو threads بلاک شو نو process هم بلاک کېږی
- Context switching پکی تیز ده او management پکی تیز ده .
- هغه لیول چی په kernel کی رامنځته کېږی او manage کېږی ورته ULT وایی .

2. او هغه space چی په kernel کی جوړېږی او هلته منیج کېږی ورته KLT وایی

- که چیری پکی یو threads بلاک شی نو process نه بلاک کېږی
- Context switching پکی تیز دی .
- All management پکی سلو دی .

1. هغه threads چی هغه په kernel level کی رامنځته کېږی هغه ته ( user level thread ) وایی user level

چی کوم thread رامنځته کېږی دغه kernel دی د یو پروسس په نظر کوری .

2. او هغه threads چی په user level کی رامنځته هغه ته kernel level threads وایی .

1. دغه user library په واسطه manage کېږی . او د دغو creation & termination ډیر تیز دی . که چیری په یو

threads په user space کی black شو نو نور ټول threads سره black کېږی ځکه چی ټولو threads ته user level space د یو پروسس په سترگه کوی .

2. **Kernel level threads** : دغه کی creation & termination ډیر slow دی په دغه ټولو threads ته د

threads په سترگه کوری د هغه بل خلاف دی .

Threads : یو پروسس چی په برخو برخو ویشو هری برخه یی threads وایی .

**Multi threads** : کله چی څو threads په یو وخت کی execution ترسره کړی یا هم پروسس سره تړاو ولری دغه ته

multi threads ویل کېږی .

### Advantages of threads

- Responsiveness
- Resource sharing
- Economy
- Better communication
- Microprocessor architecture utilization

**نوټ :** چیری چی څو پروسس د شیر او recourse ته هڅه کوی نو دلته race condition رامنځته کېږی ددغه د مخنیوی لپاره process synchronization لرو .

**Parallelism** : د څو پروسس په موازی شکل یو وخت کی اجرا کېدنی ته parallelism یا concurrency ورته وایی.

- پروسی یا threads په موازی شکل پکی ترسره کیږی .
- په multi-processor کی ترسره کیږی .
- همزمانه د پروسو execution (اجرا کیدل) ترسره کیږی .
- په multi core کی ترسره کیږی .
- دلته سرعت په خاطر ښه ده .

**Concurrency:** دغه کی هم څو پروسی په یو وخت کی ترسره کیږی په دغه context switch صورت نیسی چی دا context switch مونږ ته په سترگو نه معلومیږی. قیمت دغه څه ده .

- دلته هم په یوه وخت کی څو پروسی ترسره کیږی. خو توپیر یی په دی کی ده چی دلته context switching صورت نیسی خو دوی ترمنځ کومه وقفه او سرعت چی وی هغه مونږ نشو لیدلی .
- په single processor کی ترسره کیږی یا single core
- او دا بیا قیمت په خاطر ښه دی ځکه چی ارزانه دی .

### Process in synchronization (د پروسو همغږی کول)

کله چی څو پروسو د پروگرام یو برخی ته کوشش کوی چی ترڅو د پروگرام په همدا برخه باندی execution ترسره کړی په یوه وخت کی . نو د حالت د همغږی کولو لپاره process synchronization ترسره کوو .

که چیری همغږی نه شی نو race condition رامنځ ته کیږی دا د race condition د همغږی کولو لپاره process synchronization ترسره کوو . که ونه شی دیتا با ثباتی له منځه ځی او دیتا result هم بدلیری .

#### Process synchronization په اساس پروسی په دوه ډوله دی.

- 1 Independent process: کله چی یوه پروسه اجرا کیږی (execution) نو هغه د بلی پروسی په execution اثر نلری.
  - 2 Cooperative process: کله چی یوه پروسه (execution) کیږی نو هغه د بلی پروسی په execution اثر نلری. او process synchronization په همدی cooperative process کی رامنځته کیږی .
- کله چی دوه پروسی هم زمان واقع شی د دغه د مخنیوی لپاره synchronization څخه کار اخلو .

**Conversance:** کله چی دوه پروسی p1, p2 دواړه داسی یو منفی کوی اول یی جمع کوی دواړه ته out put راځی خو کوم چی لومړی راغلی وی هغه ته لومړی output ځی. کله چی زیاتی وظیفی ترسره کیږی نو په دغه پروسس synchronization ډیر په کاریری .

#### دی synchronization په اساس پروسی په دوه ډوله دی .

- 1 Depended: معنا یی مستقل پروسس په دغه کی یو پروسس په بل باندی تاثیر نه کوی یعنی د یوی وظیفی پروسس د بلی وظیفی په پروسی باندی تاثیر نلری.
- 2 Comparative: په دغه کی د یو وظیفی پروسس د بلی وظیفی په پروسس باندی تاثیر لری او synchronization ته په comparative کی ضرورت دی.

### Race condition

هغه د سیالی حالت ده چی په یوه پروگرام کی څو پروسی د پروگرام یوه برخی ته چی هغه شپږده د ټولو ترمنځ او پرلپسی ورته لاسرسی هڅه کوی چی مختلف افریشن ترسره کوی ورباندی نو بیا race condition رامنځته کیږی .

- دا کومه برخه چی د پروسو ترمنځ شپږ دی ورته لاسرسی هڅه کوی دی ته critical section وایی .
- دا همزمانه چی لاسرسی چی ورته کیږی چی باثباتی رامنځته کیږی او د دیتا result بدلیری دی ته critical section problem وایی .

کله چی دوه پروسی په یو وخت کی ترسره کیږی هره پروسه به کوشش کوی زما **trough put** به صورت ونیسی او بله به کوشش کوی چی زما **trough put** به صورت ونیسی او دغه وخت کی **race condition** رامنخته کیږی .

تولی پروسی چی دغه ته هڅه کوی هغه ته **thread section** وایی .

**Critical solution**: کله چی دوه پروسی هم زمان اجرا کیږی نو ددی لپاره باید دوه پروسی **Critical solution** ته داخل شی.

### Critical solution problem

د **Critical solution** ته دوه پروسی ننوتلو مخنیوی

### Solution to critical section problem

حل لپاره دری شیان باید وکړی ترڅو **critical** حل شی

**Mutual exclusion**: - باید یوه وخت کی یو پروسس ته **critical section** ته داخل شی کله چی دا پروسس **complete** شی نو بیا دی بل **process** داخل شی . یا باید یو پروسه په یو وخت کی **Critical section** ته داخل شی کله چی هغه پروسه **complete** شوه بیا بله داخلیدلی شی .

**Progress**: - باید دا **critical section** بی وزگار نه شی پاتی او پروسی باید د معلوم وخت لپاره انتظار وباسی ترڅو بی هغه بله پروسه سرته نه رسیږی او دا انتظار ورته باید معلوم وی . یا یو پروسه باید د معلوم وخت لپاره انتظار وباسی ترڅو مخکی وظیفی یا پروسه پای ته ورسپیږی ترڅو **critical section** بی وزگار پاتی نشی انتظار باید د پروسی معلوم وی کله چی یو پروسه **Critical section** ته ریکویست کوی یو دا باید محدود وی .

**Solution** په دوه ډوله دی.

**Bounded waiting**: کله چی پروسه د **critical solution** لپاره ریکویست کوی دا باید محدود وی .

دوه دانی **solution** لری :



**Software based solution** : **Peterson's solution** : دا یو سافتویر بیسډ سلوشن دی . ترتولو لومړنی سلوشن او د نورو لپاره اساس ده . او دا **solution** د دوو **process** هماهنگ کولای شی. ترڅو **critical section** ته داخل شی . او دا **solution** دوه ډوله **item** لری ترڅو دوه پروسی ترمنځ شیر وکړی . **int turn & Boolean Plag** . یا هم دا یو بیسډ سلوشن سافتویر دی دا ترتولو لومړی سلوشن او د نورو **solution** لپاره اساس ده او دا **solution** د دوه همایت کولای شی ترڅو **critical solution** ته داخل شی او دا ډول ډیټا **item** لری چی ترڅو دوه پروسی ترمنځ شیر وکړی . **int turn** او بل بی **Boolean plug** .

**Int turn**: د کومی پروسی نمیر ده چی **critical section** ته داخل شی دا بی ساتی که مونږ **set turn** کی **set I** کړو نو ا ته به ځی او که **set** کړو نو **z** ته به ځی .

**Boolean plug**: که چیری یو پروسس **true** وی نو پروسس تیاره ده چی داخل شی او که چیری **false** وی بیا نو تیار نده که چیری پروسه **ready** وی نو پروسه داخل کیدلی شی او که نه وی **ready** نو نه شی داخل کیدلی دا هم بنیایی .

دغه الگوریتم ته **humble** الگوریتم هم وایی په دغه کی یوه پروسه بلی ته خپل نوبت ورکوی ترڅو داخل شی دغه ته **class selection** هم وایی .



**Hardware based selection:** hardware-based selection test and lock ده. دا يو synchronization problem دی يو variable لری چی شیر دی د ټولو ترمنځ چی lock variable دی او همدغه variable د پروسو ترمنځ synchronization رامنځ ته کوی او همدا lock variable دوه variable اخیستلی شی (0/1) دی کله چی صفر وی پروسس داخل کیدای شی او که ۱ یو وی بیا نشی داخل کیدای .

او د ټولو لومړی lock variable صفر ته set وی که چیری پروسه په critical section کی وی بیا true وی او کله چی پروسه نه وی پکی بیا false سیټ کوی او دا د ډیرو پروسو ترمنځ هماهنگی رامنځ ته کوی .

### Semaphore

دا یو software based solution ده اودا سلویشن Edsger dijstra وړاندی کړی او یو non-negative variable solution ده او یو value اخلی او دا به منفی نه اوسپړی کوم چی کومی پرسی چی critical section ته داخل کیری باید دا access variable کوی او کله چی پروسه ورپیل access کړه یوه پروسه به دوه operation څخه استفاده کوی .

**Wait:** دا په p سره بنودل کیری او له Dutch څخه اخیستل شوی عبارت له proberen هدف یی کتل دی د یوی پروسی .

**Signal:** په v سره بنودل کیری او له Dutch څخه اخیستل شوی عبارت دی verhogen او هدف یی اضافه کول دی «increment» او دا ترټولو ښه الگوریتم دی.

- په دغه کی پروسس مستقیم لاسرسی نه پیدا کوی باید هغه دوه operation ترسره کیری .
- کله چی د s value صفر وی یا د صفر څخه کوچینی وی نو condition یی true ده نو باید انتظار وکړی تر څو هغه بله complete شی .
- او که s value یی صفر څخه لوی وی نو بیا false دی نو پروسه داخل کیدای شی کله چی داخل شو پروسه باید decrement وکړی «که ۱ وه باید ۰ یی کړی» معنا چی بله پروسه شته دا ټول د wait برخه ده .

**Signal:** دا برخه دا دنده پرغاړه لری هغه decrement چی شوی هغه پروسه چی complete شوی نو بیا increment کوی تر څو بله پروسه داخل شی.

### Introduction to Dead lock

په operating system کله چی یو پروسه exited کیری هغه system security ته ضرورت لری او دغه resource software هم کیدای شی او hardware هم کیدای شی لکه system library چی software ده او memory, I/O, cpu چی hardware ده. هغه وخت کی دغه resource نشی کولای چی ټولو requirement یا غوښتنی پوره کړی نو پدی دغه وخت کی dead lock رامنځته کیری .

یو پروسس یوی منابع ته غواری چی لاسرسی ولری نو له دری مرحلو څخه تیریری

- 1 . Request resource
- 2 . Use resource
- 3 . Leave resource

او بیا اخیر کی دا منابع پریردی .

### Dead Lock

☺ کله چې په operating system کې یوه پروسه ترسره کېږي نو resource ته ضرورت لري چې له hardware او software منابعو څخه عبارت ده .

☺ څو پروسې چې غواړي اجرا شي نو هر پروسس منابعو ته ضرورت لري او هري پروسه یو resource لري او هره پروسه یوه منابع لري او غواړي چې نوري منابع هم لاسته راوړي چې ټولې پروسې په حالت کې د circular wait کې شي نو dead lock رامنځته کېږي .

☺ **Dead lock:** هغه حالت ده چې په هغه کې د پروسو یو سیټ بلاک شي او هره پروسه یوه منبع لري خو د بلي پروسې له امله ورته لاسرسی نشي کولای .

کله چې پروسه یوې منبع ته لاس رسي کوي نو له دغو مرحلو څخه تیرېږي .

1. Requests a Resources
2. Use the Resources
3. Releases the Resources

### Dead lock condition

کله چې دا لاندې ۴ condition په یوه وخت کې صورت ونیسي نو dead lock واقع کېږي له دوی څخه چې ۳ واقع هم شي بیا هم مونږ dead lock ته نشو تللي .

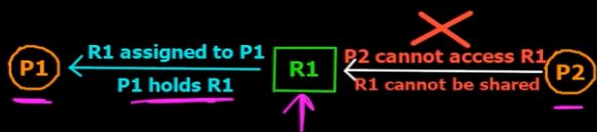
1. **Mutual exclusion:** په یوه وخت کې دوه پروسو ته access نشي کیدلی ځکه shareable نه ده .  
یا په دغه سیستم کې چې مونږ سره کوم resource دی هغه shareable نه دی یعنی په یو وخت کې د دوه پروسو په واسطه باندې نشي access کیدلی په یو وخت کې یواځې د یوې پروسې په واسطې باندې access کیدلی شي .
2. **Hold and wait:** یو resource له خانه سره ساتي بلي ته انتظار کوي . یا یوه پروسه ساتل او بلي ته انتظار کول .  
یا په دغه یو کې یو resource له خانه سره ساتل کېږي او بلي ته انتظار کوي .
3. **Non preemption:** کله چې یوه پروسس یو resources لري خو بله process منابع ته ضرورت لري خو لومړی resource نه پرېژدی . یا تر هغه وخت پوري یوې پروسې څخه یو resource نشي اخیستلای تر څو چې هغه یې په خپله complete کړي نه وي .
4. **Circular wait:** ټولې پروسې د یو بل منابع ته لاسرسی کوي خو دوی resource نه پرېژدی . یا کله چې دغه دري کاندیشن صورت ونیسي بیا circular condition ته خو معنا یې دایروي انتظار را منځته کېږي دی پروسو تر منځ .

### Conditions for Deadlock to occur -

Deadlock can arise if following four conditions hold simultaneously (Necessary Conditions)

#### 1) Mutual Exclusion - ✓

One or more than one resource are non-sharable (Only one process can use at a time)



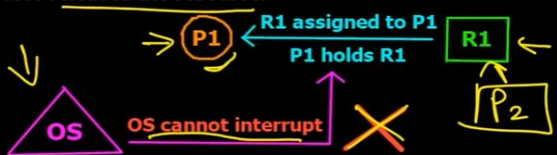
#### 2) Hold and Wait -

A process is holding at least one resource and waiting for other resources.



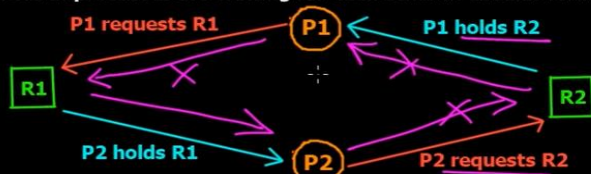
#### 3) No Preemption - ✓

A resource cannot be taken from a process unless the process releases the resource.



#### 4) Circular Wait -

A set of processes are waiting for each other in circular form.



### Handling dead lock

1. **Dead lock prevention or avoidance:** سیستم تر هغه اجازه مه ورکوه تر څو چی **dead lock** ته لاره شی.
2. **Dead lock detection and recovery:** لومړی **dead lock** پیدایځه او بیا یی جوړ یا هم حل یا یی ریکوری کړه. او یا ددغه دنده د ستونزو کشف او د جوړیدل دی.
3. **Ignore the problem all together:** نو دلته سیستم **reboot or restore** کول دی کومی کارونه مو چی ورسره کړی وی هغه ټول له سره کبیری خو دا باید ونه شی خو بیا هم **dead lock** رامنځته کبیری سیستم **restore** کړه چی ټولی پروسی بیرته له سره شی.

### Dead lock handling strategies

1. **Preemption:** پروسی تری ختم کړه له **dead lock** څخه یعنی **end** کړه پروسی. د یوی پروسی منبع پلې ته ورکوی.
2. **Roll back:** په **o/s** کی چی پروسی ترسره شوی د هغه ټولو لیست ساتی نو په **roll back** کورو کوم خای چی غلطی نه هغه خای ته خو کومی پروسی چی وروسته وی هغه ټولی له منځه ځی. له پروسی څخه بیرته شاته لاره شته سیستم **restore** خو کوم پروگرامونه چی په سیستم کی وی هغه له منځه وړی .
3. **Kill one or more process:** یا یو پروسه ختمه کړی تر څو دا **cycle** ماتنه شی او پروسی روانی شی او یا ټولی پروسی ختم کړه.

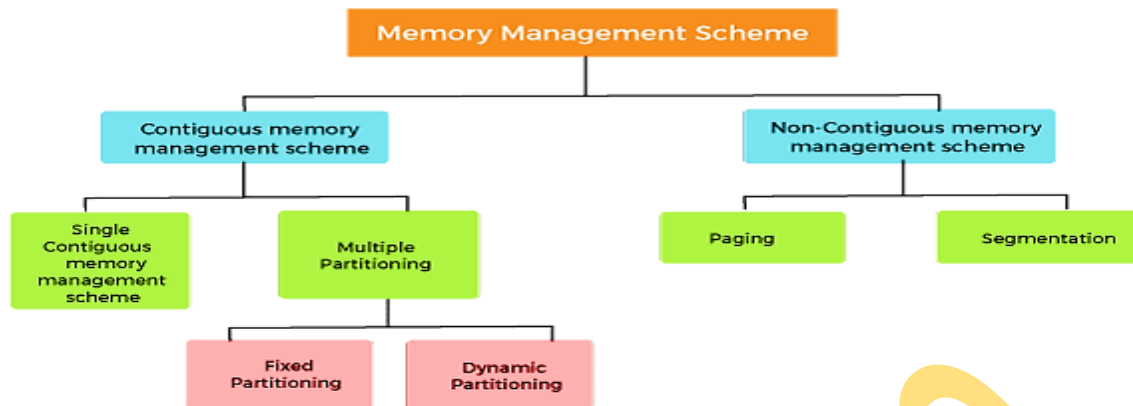
### Memory management

د کمپیوټر تر ټولو مهمو برخو څخه ده ترڅو دیتا ذخیره کړی. د **o/s** له مهمو دندو څخه د میموری اداره کول دی. میموری منجمنت د **Primary Memory** سمبالښت دنده په غاړه لری تر څو په بڼه ډول استفاده تری وشي. **Concurrency** ته لاره هواره شی او هغه وخت صورت نیسی چی ډیری پروسی په **RAM** کی وی.

- ☺ **Primary memory:** ډیره کوچینی ده او موقت وخت لپاره دیتا ذخیره کوی.
- ☺ ډیرو پروسی غواری **ram** ته راځی نو **memory management** دا کار کوی چی نوبت د چا ده او څنگه به ترسره شی. او دا د **ram** کوچینی سایز ده دا ټولی دندی پکی خای پر خای کوی.
- ☺ د اپریټینگ سیستم به مهمی دندو څخه یوه **memory management** ده.
- ☺ په بڼه شکل سره استفاده ځینی وشي.
- ☺ **Concurrency** ته لاره هواره شی (هغه وخت صورت نیسی چی ډیری پروسی په ریم کی وی).
- ☺ **Swapping** د **memory management** یو تخنیک ده.
- ☺ د هری دندی کی څومره میموری ته ضرورت وی همغه اندازه میموری ورکوی.

### Memory management technique

دغه په دوه ډوله دی ۱. **Contiguous memory management schemes** ۲. **Noncontiguous memory management schemes**



Classification of memory management schemes

1. **Contiguous:** په دغه پروسه په پرلپسې توگه خای پر خای کیری او په منځ کی کوم space پریښودل کیری.

⊗ **Single contiguous:** ساده ترین Memory management scheme ده. اوس نه استعمالیری. په دا کی primary memory په دوه برخو ویشل کیری. یو lower او دویم یی higher دی. په lower part کی o/s وی او په higher part کی د user program لپاره وی او یواځی یوه پروسه په یوه وخت کی ترسره کیری. ⊗ **Multiple partitioning:** پروسه په ډیرو برخو باندی ویشل کیری ډیری پروسه پکی خای کیدلی شی. زیاتی پروسه په یوه وخت کی ترسره کیژی. د CPU وخت هم پکی نه ضایع کیری. یا پروسه په ډیرو برخو ویشل کیری زیاتی پروسه په یوه وخت کی پکی تر سره کیری د CPU وخت هم نه ضایع کیری.

⊗ **Fixed partitioning:** میموری په ډیرو blocks ویشل کیری خو blocks یی فکس وی کوم سایز چی ورکړل شی هغه یی سایز ده او دا په خکه د میموری space ضایع کیری. او سایز هغه وخت ورکول کیری کله چی دا سیستم جوړوی دا سایز یی same او different هم کیدای شی.

⊗ **Dynamic partitioning:** په کومه اندازه چی space ضرورت وی هغه رقم block ورکوی خو block په مسلسل ډول وی دا چی تغیراوستلای شی.

2. **Noncontiguous memory:** په پرله پسې ډول نه دی چی هر خای سنور کیری ترمنځ space هم واقع کیدای شی.

کوم black چی ورکول کیری د هغه track په اپریټینگ سیستم له خان سره ساتی.

**Track:** کله black ورکول شو څومره بلاک .... څومره بلاکونه ډک وی هغه وخت کی ضرور ده کله موژ سره میموری زیاته وی.

⊗ **Paging:** په دا کی process په صفحو باندی ویشل کیری. خو صفحی fixed size وی یعنی بیا تغیر پکی نه شو راوستلای. په دا کی په پرلپسې ډول نه ذخیره کیری. دلته میموری هم په برخو ویشل کیری. او د پروسه هر یی برخه ته page وایی او memory برخه ته page frame وایی. د پروسه ادرس هم په برخو پکی ویشل کیری چی logical address او physical address سره یوځای شی ورته maps وایی. او دا mapping د memory management unit لخوا ترسره کیری.

⊗ **Segmentation:** میموری او پروسه دواړی په برخو ویشل کیری خو که fixed size وی تغیرکی راوستلای شو. د میموری هر segment پروسه ته ورکول کیری. په دا کی track ساتل لږ سخت ده. یو segment مساوی دی له میموری بلاک سره او segment table هم لری او دوه ډوله معلومات پکی ذخیره کیری.

1. **Noncontiguous:** دغه په هر خای کی store کیری پروسس او ترمنځ یی space هم رامنځته کیری.

### Non-contiguous allocation

### Management technique

دغه هغه تخنیک ده چی O/S ورته میموری به non contiguous ډول ورکوی تر څو په دغه کی پروسی ذخیره شی د پروسس لپاره به یو د ram په اخر کی او بل به د Ram په سر کی وی په غیر منظم ډول به ذخیره شوی وی .

هغه block چی o/s بی دی میموری ته ورکوی هغه به non contiguous وی او هم میموری ټول track له خانه سره ساتی . په دغه کی مونږ non contiguous میموری څخه ضایع کیدو مخنیوی کوی .

### دغه تخنیک په دوه لاری کیدلی شی چی مدیریت کری O/S .

#### 1. Paging

دغه د صفحی په معنا ده یو پروسس را اخیستل کیری په صفحو یا page باندی ویشل کیری دغه صفحی به fixe سایز ( size ) لری تغیر هیڅکله نه کوی کله چی پروسه په برخو ویشل شو نو non contiguous ته زمینه مساعده شوه چی پروسه په هر خای به ram ذخیره کری په دغه کی هم پروسس او هم میموری په برخو ویشل کیری پروسس چی په برخو ویشل شی هغه ته page وایی او میموری چی په برخو ویشل شی هغه ته page frame وایی physical address دا په میموری کی وی . logical address دغه د cpu له خوا ورکول کیری .

نوبت : page size د frame size سره مساوی ده .

کله چی physical page او logical page سره یو خای شی نو دغه ته mapping وایی او دغه کار د memory mapping دی . کله چی یو پروسس په پیچونو باندی ویشل شی نو د page numbers او د page length دا ټول په یو table کی ساتل کیری چی هغه ته page table وایی .

#### 2. Segmentation

دغه کی یو پروسس یا میموری په برخو برخو ویشل کیری دغه چی ده دی تغیر ور ده نسبت paging ته . په segment table کی د segment number , segment length دا په کی ذخیره کیری .

دوه ډوله معلومات د یو segment په باره کی ساتل کیری .

1 . Base : هغه دی segment number ته وایی

2 . Limit : segment full size ته وایی.

#### Fragmentation (برخی ته ویل کیری)

یوه مشکل ده چی د میموری د ضایع کیدو سبب ګرخی .

مثال : یوه میموری په مختلفو برخو ویشلو ته وایی . کله چی د یو frame size بی 2mb وی او د page size بی 1mb وی کله چی په دغه بی ذخیره کری نو 1mb سایز پاتی کیری نو دغه ته fragmentation وایی .

Fragmentation په دوه برخو ویشل شوی ده:

1 . Internal fragmentation: کله چی د یوی پروسی سایز کوچینی وی د میموری د سایز څخه یعنی کوچینی پروسس په لوی سایز میموری کی مو ذخیره کری وی نو نیم ونیول شو او نیم پاتی شو نو دغه ته وایی.

#### څنگه دا internal fragmentation له منځه وړلی شو ؟

که چیری مونږ dynamic partition څخه استفاده وکری نو دغه ستونزه له وړلی شو.

2. External fragmentation: که چیری دغه دوام وکری نو زمونږ د میموری سایز کمولی شی.

مثال : مونږ د یو پروسی یو برخه لرو غواړو چی په میموری کی بی ذخیره کړو د میموری سایز دغه پروسی ته کافی ده نو دغه سایز پرلپسی نه ده نو ځکه دا مونږ نشو ذخیره کولی نو دغه space ضایع شو دغه ته external fragmentation وایی .

### How to remove external fragmentation?



Process of need 45kb memory space

### Virtual memory

د hard disk د یوې برخې څخه د ram په توګه استفاده کیږي همدې برخې ته virtual memory وایي .

- ✓ زموږ سره مرسته کوي چې bigger program زموږ سره execution کړي .
- ✓ کله چې پروسه په page وویشل هغه هم page ته وړل کیږي میموري ته .

### How virtual memory works?

پدې کې swapping ترسره کیږي مطلب دا چې د یوې پروسې پر ځای بله انتخابیږي. یوه پروسه ایسته کوي او پر ځای بله پروسه انتخاب کوي . کومه پروسه کی زیاتې وخت ورته کال نه وی شوی هغه ایسته کوي یو ځای بله پروسه انتخاب کوي . خو دا نظر ریم ته سلو ده .

### What limits the size of virtual memory?

دوه شیان دی چې د virtual memory سایز limits کولای شی .

- 1 . Hard disk size
- 2 . Address bus تعداد بی زیات وی.

### Advantages of virtual memory

- \* پروسې په تیزی سره اجراء کیږي .
- \* د پروسې اجرا کیدل به همزمانه دېرشې .
- \* Ram به کوچینی وی کوم پروگرام چې رن کوو هغه سایز به لوی وی.
- \* ضرورت بیا نشته چې موږ زیات ram واخلو .

### Dis advantages of virtual memory

- \* وخت پکې ضایع کیږي ځکه swapping او swap out پکې کیږي
- \* سیستم سلو وی پکې
- \* Switching د application پکې وخت نیسی.
- \* هارډیسک space پکې مصروف وی .

### Demand paging

- ⊙ Page غوښتل چې کله ورته ضرورت وي. ورل کيږي که ضرورت نه وي نه وري. ځکه د ميموري space ضايع کيږي .
  - ⊙ کله چې پروسه رم ته راوړل شي او هغه په page وويشل شي. چې په دا کی داسی page وی چې استعمال شوی نه وی هغه خای خالی کوی او په خای بل page راوړی نو دی ته demand paging وایی. که ضرورت شي page بیا بیرته راوړی .
- دا کار د یو ډول algorithm په اساس کيږی .

### Page replacement algorithm

یو بدلول په بل page ورته page replacement وایی. دا algorithm هغه وخت راخی چې کله virtual memory څخه استفاده کيږی .

MRU – 4    LRU -3    OPR -2    FIFO -1

1. **FIFO (First in first out)**: لومړی کی کوم page راغلی وی هغه out کوی په خای یی بل راوړی.
2. **OPR (optimal page replacement)**: کوم په خای بل page کی زیات استعمال نه شي هغه out کوی په خای یی بل page راوړی.
3. **LRU (Least recently used)**: کوم page چې په نژدی وخت کی زیات استعمال شوی نه وی هغه out کوی په خای یی بل راوړی.
4. **MRU (Most recently used)**: هغه page چې په نژدی وخت کی زیات استعمال شوی وی هغه out کوی په خای بل راوړی.

### Trashing

کله چې کوم page replace, شی نو سمدستی هغه ته ضرورت شي نو ورته trashing وایی او دا مشکل د algorithm له وجی راخی نو زموږ time ضایع کيژی دلته .

### I/O device management

ډیر output او input الی لرو هر هغه څه چې کمپیوتر ته وصل کيږی I/O دی . دا څه پرزی چې کمپیوتر لری دا ټول manage ته اړتیا لری . management دا په غاړه لری چې دوی manage کوی او دوی ترمنځ هماهنگی رامنځ ته کړی

### Device drivers

د هر device لپاره خپل driver وی ترڅو device فعال کړی او استفاده ځینی وکړی او دا system software ده. د کمپیوتر هره پروژه له میخانیکي او الکترونیکي برخو جوړه شوی. د پروزی او ډرایور تر منځ ارتباط ساتی او هارډویر مهم دنده bit په block بدلیژی.

- ⊛ O/S په مستقیم ډول هم له یوه ډیواس سره communication نه کوی د driver نه.
- ⊛ O/S له drivers نه په استفادی سره I/O ټول ډیوایس Manage کوی.
- ⊛ Drivers د O/S لپاره انترفیس رول لوبوی.
- ⊛ Drivers دا زمینه برابروي ترڅو o/s له devices سره communication ترسره کيږی.

### Device controller

دا هارډویر دی د device drivers او device ترمنځ انترفیس رول لوبوی دا کوچینی کوچینی چپونه دی . Electronic component ته device controller وایی. هره الة device controller ته ضرورت لری دا هم کیدای شي چې د یو controller په واسطه زیات devices کنترول کړی .

**مهمه دنده** : د cpu او device ترمنځ دا communication د bits جریان ده دا جریان په blocks بدلوی او ډیتا یی بدلوی.

### Synchronous VS asynchronous

**Synchronous**: کله چی i/o افریشن راوان وی cpu انتظار حالت کی وی. بله پروسه کار نه کوی. وخت پکی ضایع کیږی .

**Asynchronous**: د i/o افریشن راوان وی د cpu هم. نو پدی کی وخت نه ضایع کیږی .

### Communication to I/O devices

Cpu باید یوه لاره ولری چی د I/O سره Communication وکړی. درې دانې طریقې دی چی communication وکړی. په دريو طریقو CPU له I/O سره ارتباط ټینګیدلی شی.

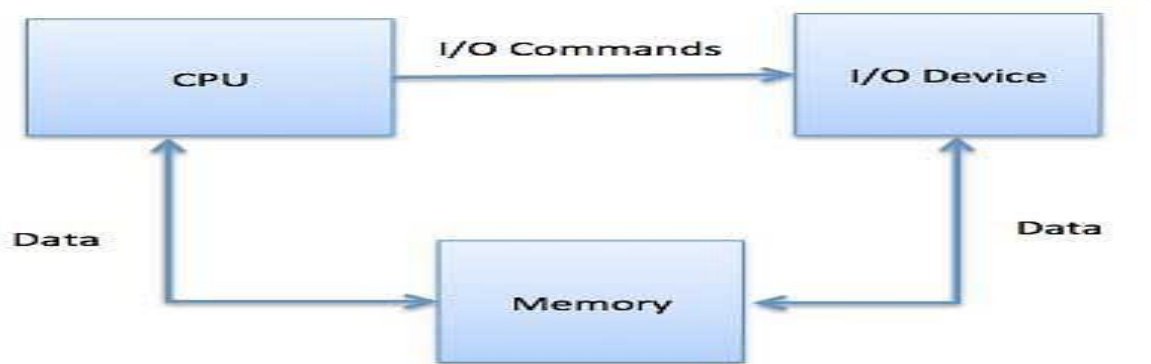
1 - Special instruction      2 - memory mapped I/O      3 - Direct memory access (DMA)  
**نوټ**: interrupt : د execution نارمل حالت بدلولو ته وایی .

### Special instruction I/O

د cpu د instruction څخه استفاده کوی او د هغه instruction دی چی د I/O د control لپاره جوړ شوی وی .

### Memory-mapped I/O

د CPU په مرسته چی کومي عملیې اجراء کیږی په CONSOL پدی کی ادرس د میموری او I/O تر منځ شیر کیږی او I/O مستقیم ډول له میموری څخه ډیټا راخیستی شی.



### Memory instruction I/O

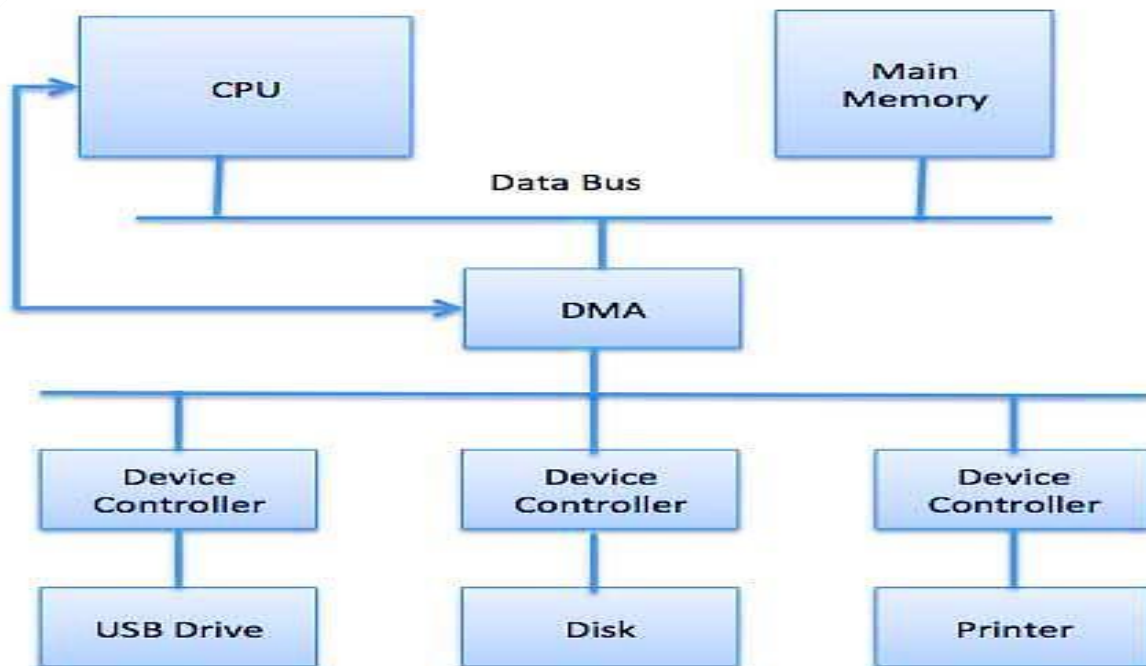
د I/O devices او processor د میموری یوه برخه شیر کیږی او دا ډواړه د شیر برخی ته مراجعه کوی ترڅو ډواړه communication ترسره کړی .

**Buffer**: کله چی data د دوو devices ترمنځ تبادلې کیږی او د میموری په یوه برخه کی د موقت لپاره تم کیږی ورته وایی. د cpu او I/O د buffer نه په استفادی سره communication ترسره کوی .

### Direct memory access (DMA)

دلته په جریان کی ته اچول کیږی ځکه په i/o په direct ډول memory ته لاسرسی کوی او communication ترسره کیږی . دا پروسه د direct memory access (DMA) پواسطه کنټرول کیږی . دلته cpu اجازه ورکوی چی په مستقیم ډول سره لاسرسی وکړی . سپید ( speed ) یی تیز وی . یا په مستقیم ډول میموری ته لاس رسی کیږی. CPU په جریان کی نه وی بهترین میکانیزم دی ځکه Interrupt نه واقع کیږی دا هادوپر دی او CPU وزگار پاتی کیږی.





### Polling I/O

دا ساده لاره چې I/O دیوایس له پروسیسر سره ارتباط ونیسی پروسه پکې د CPU لخوا تر سره کیږی د پروسیسر لخوا په نوبت سره I/O چک کیږی چې دیته Polling وایی.

### Interrupts I/O

په دی کې په مسنقیم ډول CPU ته Call کیږی تر څو I/O ته توجه وکړی.

### Polling VS Interrupts

- \* کله چې د devices نه cpu ته سیگنال راشی او دا signal ددی لپاره راغلی وی چې توجو بل څه ته اړول غواړی چې اوس واقع شو ورته interrupts وایی .
- \* کله چې cpu د هر device چیک کوی چې ما توجو ته به اړتیا نلری دغه ته polling وایی او دا پروسه د cpu لخوا ترسره کیږی. او interrupts د cpu لخوا ورکول کیږی

### Storage management

دری ډوله ستوریج لرو on premise – local ده او cloud ده .

**Local:** کوم چې زموږ سره کمپیوټر کې موجود وی.

**On premise:** سرور ورته اخلو د LAN نیټورک په اساس ټول devices هغه ته لاسرسی لری او storage تری اخلی.

**Cloud:** چې دا د انټرنیټ څخه اخیستل کیږی.

x یوه پروسه ده چې د device – منیج کوی کوم چې دایمی وخت لپاره ډیټا ذخیره کوی.

x د سیستم performance ورسره ښه کیږی او د data integrity ورسره ښه وی .

x که ښه مدیریت نه شی نو قیمت به یی زیات شی یعنی cost .

x Performance ورسره ښه کیږی د storage management په اساس.

x په اسانی سره ډیټا ته لاسرسی کیږی .

x Storage capacity ورسره سمیږی.

### Storage management key attributes (ولی مهم ده)

- \* Performance

- ✿ Reliability
- ✿ Recovery
- ✿ Capacity

### Disk management

یو له مهمو storage device څخه ده . که چیرې دیتا په منظم ډول ډیسک کی ځای پر ځای وی سیستم performance ورسره بڼه کیږی .  
 O/S دا هم ده چی د disk په ځانگړو طریقو سره manage شی ترڅو دیتا په منظم ډول ځای پر ځای شی. اکسس ورته اسانه شی او space ضایع نه شی .

### (اهداف)Goals of disk management

- ✿ دیتا په ترتیب ډول سره کول دی .
- ✿ دیتا چی گډوډه وی نو سیستم به سلو وی نو دا گډوډی له منځه وری .
- ✿ د دیتا د ضایع کیدو مخنیوی کوی .
- ✿ دیتا منظم شکل سره organization کوی .

### Disk management tasks

- Partitioning**: په منظم ډول دیتا ساتی.
- Formatting partitions to different file system**: د فایل حالت بدلول بل حالت ته.
- Defragmentation**: د ستوریچ د ضایع کیدو مخنیوی کیدو کوی.
- Back up**: د دیتا کاپی اخیستل که چیرې دیتا خراب وی یا له منځه ولاړه شی.

### Disk scheduling

کله چی څو پروسی غواړی اجرا شی نو سیالی کوی هر ه یوه غواړی چی اجرا شی نو دا کار پکی disk scheduling کوی ترڅو چی کومه اجرا شی . او هری پروسی ته خپل نوبت ورکوی چی ترسره شی .  
 Disk ته په دوه ډوله د I/O راځی یو بی read او بل بی write ددغو دواړو مدیریت کول د disk scheduling دی . په خپل نوبت سره ریکویست جواب ورکوی . هغه نور په انتظار کی پاتی کیږی . کله چی ریکویستونه سره لری وی نو د disk dram زیات حرکت کوی او که نژدی وی نو بیا (dynamic random access memory) dram کم حرکت کوی .

### Disk scheduling algorithm

#### FCFS (FIRST COME FIRST SERVE)

لومړی چی کوم ریکوست راغلی وی لومړی به هغه اجرا کیږی که د Read لپاره وی که د write لپاره وی دلته dram زیات حرکت کوی وخت ضایع کیږی پکی.

#### SSTF (Shortest seek time first)

: کوم چی نژدی وی او په لږ وخت کی ورته رسیدلی شی هغه لومړی اجرا کوی. کوم چی لږ وخت نیسی هغه اجرا کړه . یا dram ته چی کومه پروسه نژدی وی هغه ترسره کړه .

#### Secom

: لومړی disk dram یو طرفه ته ځی هغه ترسره کوی بیا بل طرف ته ځی هغه ترسره کوی. او شروع د منځینی برخی نه کوی .

### File system

- ✿ هر هغه شی چی کمپیوتر بی د ذخیره کولو وړتیا لری ورته file وایی .
  - ✿ هغه دیتا چی خپل منځ کی دقت لری ورته file وایی . چی کمپیوتر بی په secondary میموری کی ذخیره کوی
- it is collection of interrelated data.
- ✿ مجموعه د bit او ریکارډونو ده .

### File Attributes

**Name:** It is the only information stored in a human-readable form.

**Identifier:** Every file is identified by a unique tag number within a file system known as an identifier.

**Location:** Points to file location on device.

**Type:** This attribute is required for systems that support various types of files.

**Size:** Attribute used to display the current file size.

**Protection:** This attribute assigns and controls the access rights of reading, writing, and executing the file.

**Time, date and security:** It is used for protection, security, and also used for monitoring

### Introduction to file system

د o/s یوه برخه ده چی په کمپیوټر کی د فایلونو ذخیره کولو او اداره کولو دنده په غاړه لری او په منظم ډول یی ذخیره کوی .

### Advantages of file system

**Organization:** ترتیب کول په منظم ډول سره.

**Data protection:** د فایلونو ساتل د ویروس څخه اکسس یی او کنترول. degree ورته برابره کوی چی څوک read کوی او څوک یی write کوی.

**Improved performance:** چی څومره فایلونه په منظم ډول ځای پر ځای شی نو د سیستم performance ورسره high کیږی وخت به زیات نیسی. او تیز وی .

### Dis advantages

**Compatibility issue:** په مختلف کمپیوټرونو کی مختلف سیستمونه کارول چی د دوی ترمنځ ستونزه راځی چی data یو بل ته نه استول کیږی.

**Disk space overhead:** د هارډیسک space ضایع کول.

**Vulnerability:** د سکورتی خیال نه ساتل.

### Various file system

- ☺ **Fat (file allocation table):** An older file system used by older versions of Windows and other operating systems.
- ☺ **NTFS (new technology file system) :** A modern file system used by Windows. It supports features such as file and folder permissions, compression, and encryption.

- ☺ **Ext (extended file system):** A file system commonly used on Linux and Unix-based operating systems.
- ☺ **HFS (hierarchal file system):** A file system used by macOS.
- ☺ **APFS (apple file system):** A new file system introduced by Apple for their Macs and iOS devices.

### Various operation performed by file system

**Create---read---write----open-----delete-----close**

**Append:** د یوی ډیټا په سر په بله ډیټا وراچول کوی دواړه ساتی .  
**Truncate:** فایل کی چی کومه ډیټا ده هغه ډیټا له منځه وری فایل نه ختم کوی.

### File access mechanisms

مختلف طریقوی موجود دی چی موږ ته فایل را پیدا کوی .  
**Sequential access:** په پرلپسې ډول لاسرسی کوی دی یو په بل پسې فایلونه کتل کیږی.  
**Direct random access:** په مستقیم ډول لاسرسی کول دی دا چی څنگه راخی مختلف algorithm کار دی. د index په مرسته لاسرسی ورکوی .  
**Indexed sequential access:** دواړه direct and sequential څخه استفاده کوی دا یو hybrid میکانیزم دی لومړی په index پیدا کوی بیا وروسته د ډیټا په داخل کی چی سرچ کوی د sequential څخه استفاده کوی.

### Security

یو میکانیزم ده چی د هغه په مرسته زموږ سیستم او ډیټا خوندي شی سیستم کی Disk, RAM او CPU راخی. غیر قانونی لاس رسی سیستم او ډیټا ته له دی څخه امنیت ته Security وایی.

### Vulnerabilities

په غیرقانونی ډول زموږ ډیټا ته له یوی لاری ننوتلو ته وایی.

### What is CIA triad

**Confidentiality:** زموږ ډیټا او سیستم ته په هغه کسان لاس رسی پیدا کوی چی اجازه ورکړل شوی وی.  
**Integrity:** څنگه چی ډیټا لیږل شوی ده هماغه ډول باید خپل خای ته ورسی لار کی تغیر رانشی.  
**Availability:** سیستم او نیټورک باید هفته کی اوو ورخی او ۲۴ ساعت فعال وی.



### threats

هر هغه څه چې د سيسنم او ډيټا د خرابوالی توان ولری دپته Thread وایی.

### Physical Threats

It is a potential cause of an occurrence/event that could result in data loss or physical damage.

### Non-physical threats

A non-physical threat is a potential source of an incident that could result in

- Hampering/restricting of the business operations that depend on computer systems.
- Sensitive data or information loss
- Keeping track of other's computer system activities illegally.
- Hacking id & passwords of the users, etc.

Shafiqkhan

**Get more e-books from [www.ketabton.com](http://www.ketabton.com)  
Ketabton.com: The Digital Library**