Operating system

Date: 25/08/24

→ system software

Functionality of OS

Concurrency:

→ multiple process active at once.

defination of OS ? + functionality

→ Processes can communicate with

Assignment example → Process may need resources mutually exclusive, access to some resources.

→ CPU scheduling, resource management.

Memory management:

→ Allocates the memory to processes, move processes between disk and memory.

File system:

→ Allocates space for storage of program and data on disk.

Networks and distributed computing:

→ Allow computers to work together.

Security and protection.

Why to study operating system?

(*) Abstraction operating : How do you give the users the illusion of infinite resources (CPU time, memory etc)

* System design → Performance and simplicity of OS.
→ Functionality of hardware and software.

* Primary intersection Point : OS is the Point where hardware, software, Programming languages, data structures and algorithms are come together.

responsibility of OS → file, process, memory mana- and security.

#Process management:

→ OS manages user programs and system programs.

→ Encapsulated in a process (A Process

includes the complete execution context. (code, PC, registers, resources).

→ A Process is one instance of a program in execution. many process might be running at the same time.

→ OS must have : create, delete, suspend, resume and schedule processes

→ support inter - process communication and handle deadlock.

Not Process & Program are different

deadlock → process তার resource আর প্রয়োজন resource

OS

Date: 28/08/2

Memory Management:

Primary memory → Provides direct access CPU

→ Processes must be in main memory to execute.

OS Must → Keep track of memory in use.

→ Keep track of unused memory

→ Protect memory space.

→ Allocate, deallocate spaces for processor.

→ Swap process ↕ memory ⟷ disk.

Policies:

* Decide when to load each process in memory.
* Decide how much memory space to allocate each process.
* Decide when a process should be removed from the memory.

short term → main memory
long " → secondary "

File system :

→ A file is a long term strong entity, a named collection of presistent information that can be read or written.

→ A file system support directions, which contains files and other directiones (name, size, date created, dede modified) etc.

OS must → Create and delete directionies
→ Manipulate files and directiones.
read, write, catched, remove, copy etc
→ Provide general higher level services
→ Backups.

Disk management :

Disk (HDD) :

→ The action actual hardware that sits underneath the file system.
→ large enough to storce all user

Programs and entire OS.

OS must → Keep track of used and unused memory

→ Keep track of 'bad blocks'

→ Disk Scheduling

The process sheduling is the activity of th

Procese manager that handles the remove

of the running process from the CPU

and the selection of another process

on the ~~baris~~ basis of a particular

strategy

there are two types of process schedu

① Non-Pre-emptive ; Here the resources can't be taken from a process ~~not~~ until the

(মাঝে কোনো Break করানো যাবে না) যখন Process Active হবে

queues → main memory
Process → secondary
Process completes execution. The switching of resources occurs when the running Process terminates.

③ Preemptive: Here the OS allocates the resources to a Process for a fixed amount of time. During resource allocation, the Process switches from running state to ready state or waiting state to ready state

The OS maintains the following important Process scheduling queues.

* Job queues: This queue (main → secondary) keeps all the processes in the system.

* Ready queues: This queue keeps a set of all processes residing in main memory ready and waiting to execute. A new Process always Put in this queue

Ready for Process

Assignment:

what is the long term scheduler and short term scheduler? write difference between them.

1. what is Process control block (PCB)
2. what kind of infor save in PCB when context sw

Context switching: A context switching is the mechanism to store and restore the state or context of a cpu in Process control block so that a process execution can be resumed from the same point at a later time.

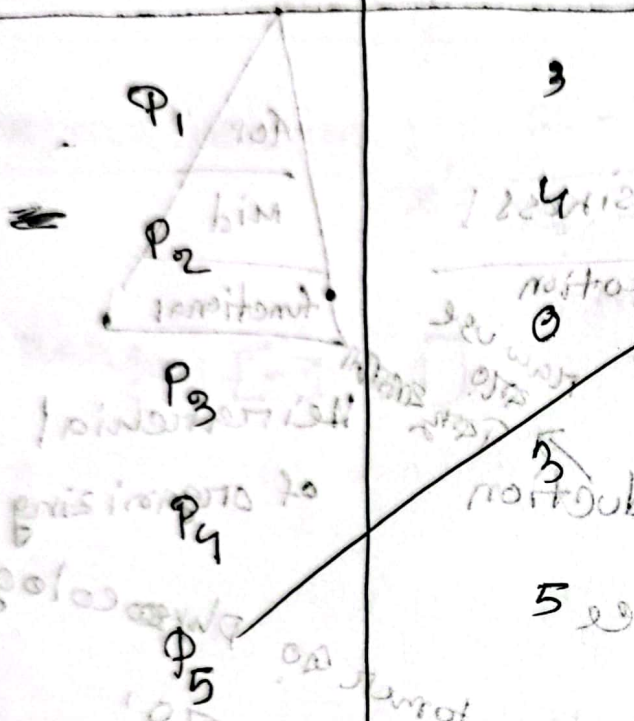
PCB → প্রত্যেকটা process এর info, রাখে, কতটুকু execute হয়েছে, কখন Break, নিলো,

3. Context switching diagram.

first count first serve

Process scheduling Algorithms: FCFS (Non-Prec

Process	Arrival time	burst time
P_1	3	1
P_2	4	5
P_3	0	2
P_4	3	7
P_5	5	5



last second

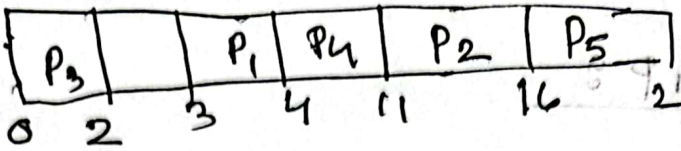
(CT-A.T) (TT-B.T) fir

Process	Arrival time	burst time	Completion time	Turn Around time	waiting time	Respo time
P_1	3	1	4	1	0	0
P_2	4	5	16	12	7	7
P_3	0	2	2	2	0	0
P_4	3	7	11	8	1	1
P_5	5	5	21	16	11	11

2TM

6a

GANTT chart



↳ burst time

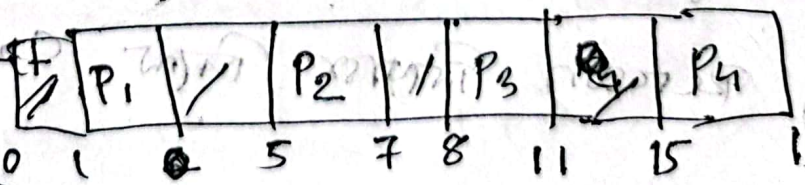
average turn around, throughput = n / total ti

Response, waiting time

= 5 / 21

Process	Arrival time	burst time	Completing time	Turn around time	waiting time	Response time
P_1	1	2	3	2	0	0
P_2	5	2	7	2	0	0
P_3	8	3	11	3	0	0
P_4	15	4	19	4	0	0

GANTT Chart



Non-Premeptive এর waiting time & Response time সমান হয়।

Premptive এ same হতে পারে।

WT ও RT same কেন হয়?

কারণ.

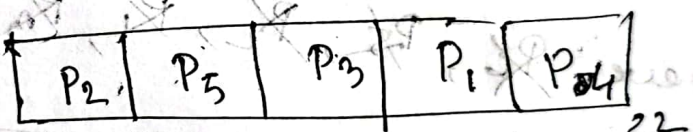
OS

Date: 02/09/24

SJTF (Shortest Job First) non preemptive :

Process	Arrival time	Burst time	C.T	T.T	W.T	R.T
P_1	4	5	15	11	6	6
P_2	0	2	2	2	0	0
P_3	1	5	10	9	4	4
P_4	6	7	22	16	9	9
P_5	2	3	5	3	0	0

GANTT chart



Ready Queue : ~~P2~~ ~~P5~~ ~~P3~~ ~~P1~~ P1 P4

P_5 এর 2nd assign করা Job time কম

Bubble sort এ Pass $(n-1)$

step $(n-\text{Pass})$

Priority Scheduling non Preemptive :

Process	Arrival time	Burst	Priority	C.T	T.T	W.T	R.T
P_1	3	5	2	8			
P_2	2	4	1	4			
P_3	5	1	3	13			
P_4	4	7	4	21			
P_5	1	6	3				
✓ P_6	0	2	5				

Ready Queue: ~~P_6~~, ~~P_5~~, ~~P_2~~, ~~P_1~~, ~~P_3~~, P_4

GANTT

P_6	P_2	P_1	P_5	P_3	P_4

0, 2, 6, 14, 17, 18, 25

Priority same হলে আগে যেটির Arrival time কম সেটা আগে,

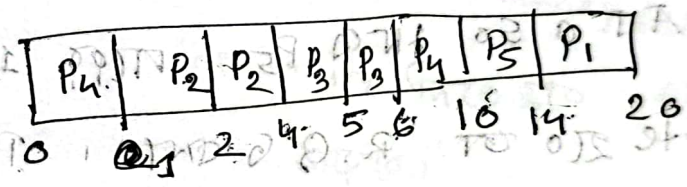
Operating System. Date: 08/09/24

SJF (Preemptive) / SRTF (shortest Remaining time first)

Process	A.T	B.T	CT	TT	WT	R.T
P_1	2	6,1	20	18	12	12
P_2	1	3,2,0	4	3	0	0
P_3	4	2	6	2	0	0
P_4	0	5,4	10	10	5	0
P_5	6	4	14	8	4	4

R.Queue: ~~P_4~~, ~~P_2~~, P_1, ~~P_3~~, P_5

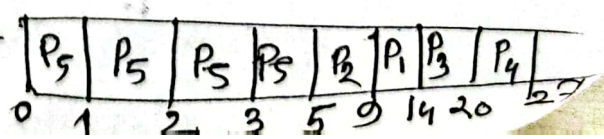
GANTT chart



Process	Priority	A.T	B.T
P_1	2	3	5,0
P_2	1	2	4,0
P_3	3	5	6,0
P_4	4	1	7,0
P_5	5	0	5,4,3,2,0

R.Q: P_5, P_4, P_2, P_1, P_3

GANTT



Priority scheduling

P	priority	A.T	B.T	CT	T.T	W.T	R.T
P_1	2	3	5,1,6	11	8	3	3
P_2	1	2	4,3,1,6	6	4	0	0
P_3	3	5	6,7	20	15	9	9
P_4	4	1	7	27	26	9	19
P_5	3	0	5,4,3	14	14	9	0

not same

R.Q ; P_5, P_4, P_2, P_1, P_3

GANTT :

P_5	P_5	P_2	P_2	P_2	P_1	P_3	P_3	P_4

0 1 2 3 5 6 11 14 20 27

P_5 Q. AT 0 So 1st Q P_5 পাবে 1 sec Q P_4 Accecute হবে এবং R.Q Q যাবে। Priority 2 sec Q P_2 কাজ করে P_2 যাবে। এবং Accecute হবে 4 sec.

OS lab

Date: 11/9/24

```
n = int(input("Num of process: "))
arr = [[0 for i in range(7)] for j in range(n)]
Print(arr)
```

↳ 2D array বানানোর জন্য

output এমন,

Num of Pro : 5

[[0,0,0,0,0,0,0],[0,0,0,0,0,0,0]]

↳ 5 টা

```
for i in range(n):
    Pid, at, bt = input().split()
    arr[i][0] = int(Pid)
    arr[i][1] = int(at)
    arr[i][2] = int bt
Print(arr)
```

another way

```
n = int(input()).
arr =
for i in range(n):
    for j in range(3):
        arr[i][j] = int
            (input())
Print(arr)
```

OS

Date: 15/09/24

~~Round Robin~~ SRTF

Process	AT	BT	CT	TT	WT	RT
P_1	0	6,5,				
P_2	1	3,2,0				
P_3	2	5,4,0				
P_4	3	1,0				
P_5	4	4,0				

1 2 3 1 4

R: ~~P_1~~, ~~P_2~~, ~~P_3~~, ~~P_4~~, ~~P_5~~

Gantt chart: | P_1 | P_2 | P_3 |

0 1 2

2 sec এর পর আরেক process আসবে

Gantt chart: | P_1 | P_2 | P_3 | P_4 | P_2 | P_3 | P_5 | P_1 |

0 1 2 3 4 6 10 14 19

Round Robin: Time Quanta = 2 → CPU burst

Process	AT	BT
P_1	0	6,4,2
P_2	1	3,1
P_3	2	5,3,1
P_4	3	1,0
P_5	4	4,2,0

1 2 3 4 5 6 7 8 9 10 11

Ready : P_1, P_2, P_3, P_1, P_4, P_5, P_2, P_3, P_1, P_5, P_3

Chart :

P_1	P_2	P_3	P_1	P_4	P_5	P_2	P_3	P_1	P_5	P_3

0 2 4 6 8 9 11 12 14 16 18 19

Quanta = 2

P	AT	BT
P_1	0	8, 6, 4, 2, 0 — 8, 4, 0
P_2	1	6, 4, 2, 0 — 6, 2, 0
P_3	3	3, 1, 0 — 3, 0 — 2, 0
P_4	5	2, 0 — 5, 1 — 4
P_5	6	6, 2

R : P_1, P_2, P_3, P_2, P_4, P_5, P_1, P_3, P_2, P_5, P_1

Chart :

P_1	P_2	P_3	P_2	P_4	P_5	P_1	P_3	P_2	P_5	P_1

0 2 4 6 8 10 12 14 16 17 19 21 23

R : P_1, P_2, P_3, P_1, P_2

P_1	P_2	P_3		

0 4 8 11

R : P_1, P_2, P_3, P_1, P_4, P_5, P_2, P_4, P_5

P_1	P_2	P_3	P_1	P_4	P_5	P_2	P_4

0 4 8 11 15 19 23 25 26

1 2 3 4 5 6 7 8 9 10 11

Ready : P1, P2, P3, P1, P4, P5, P2, P3, P1, P5, P3

Chart :

P1	P2	P3	P1	P4	P5	P2	P3	P1	P5	P3
0	2	4	6	8	9	11	12	14	16	18	19

Quanta = 4

P	AT	BT
P1	0	8, 4, 0
P2	1	6, 2, 0
P3	3	3, 0
P4	5	2, 0
P5	6	4

R : P1, P2, P3, P1, P4, P5, P2, P1, P3, P2, P5, P1

Chart :

P1	P2	P1	P3	P2	P4	P5	P1	P3	P2	P5	P1
0	2	4	6	8	10	12	14	16	17	19	21	23

R : P1, P2, P3, P1, P2

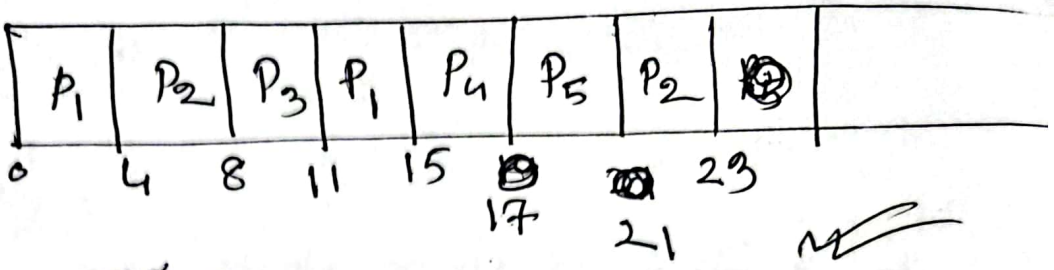
P1	P2	P3		
0	4	8	11	

R : P1, P2, P3, P1, P4, P5, P2, P4, P5

P1	P2	P3	P1	P4	P5	P2	P4	
0	4	8	11	15	19	23	25	26

R: P1, P2, P3, P1, P4, P5, P2, P3.

chart:



64, 11, 12, 22, 25, 34, 90

64, 34, 25, 12, 22, 11, 90

Bubble sort:

for (3, 7) → 2 to 7 ... loop

i=2 for (start, n-i+start-1) → 2-6 ...

↳ 7 - 2 + 2 - 1 = 6

a[2] > a[3]:

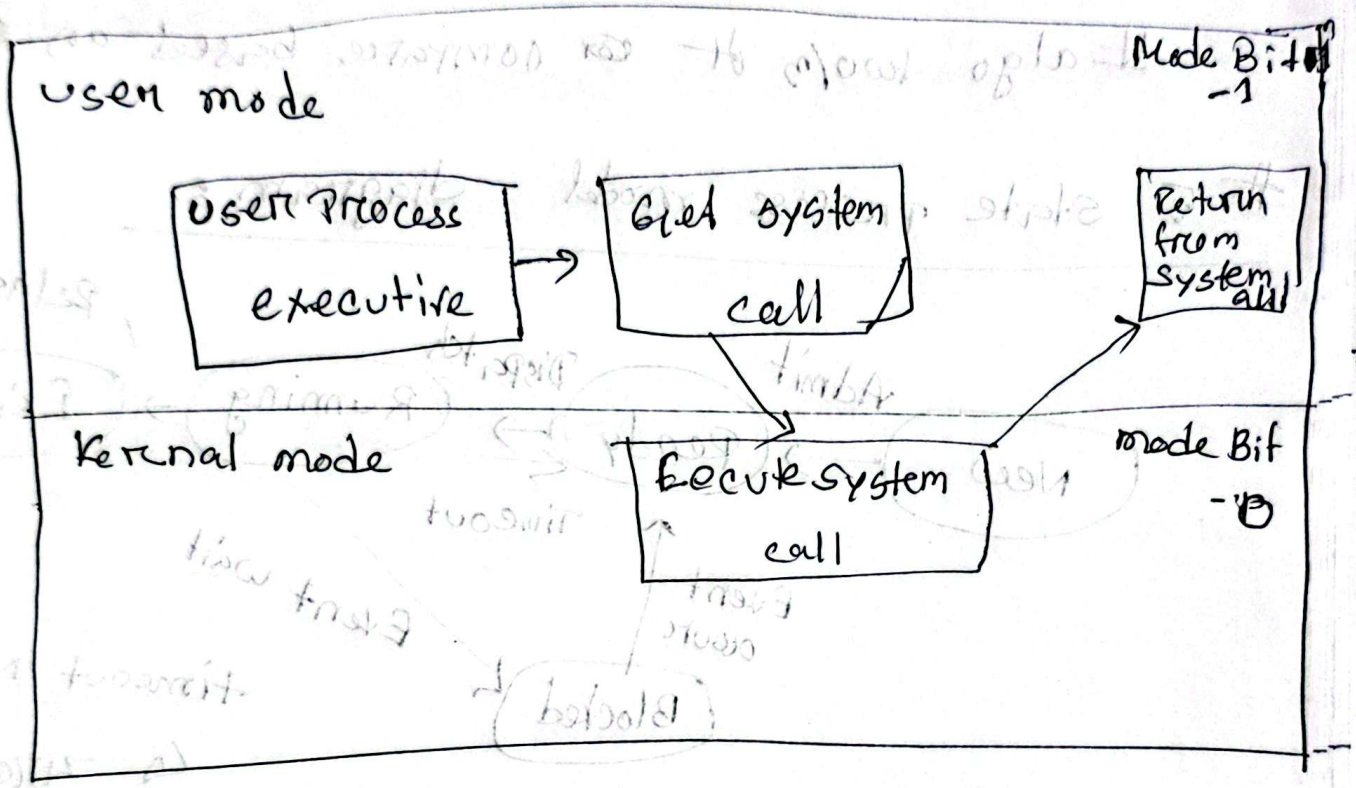
a[2], a[3] = a[3], a[2]

Swap True:

OS

Date: 22/09/24

User mode Vs Kernal mode: → OS (Hardware control করে)



PC on করলে যে UI আসে সেটা User mode এ.

Printer এ কাজ করে OS Goes OS এর কাজ হচ্ছে I/O system.

System call এ হচ্ছে User hardware এর call করছে

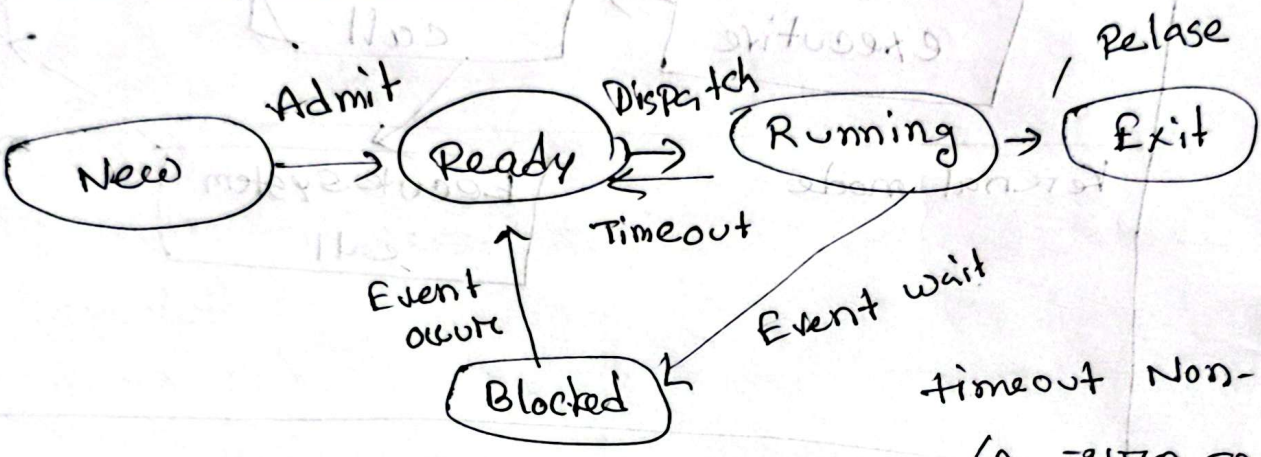
Sub: 23/09/24

what is system call? ~~Submission~~

Basic some system call name and their work.

algo two/three ~~of~~ compare based on Performance.

5 state process model diagram :



timeout Non-Preemptive ↳ নয়,

arrow দিয়ে কোথা থেকে কোথায় যাচ্ছে।

Process Arrive হলে তা কোন Job Queue তে যায়, execute হলে Ready Queue তে আসে;

→ Ready Queue ← Running
Ready Queue (wait করা)
Exit (execute করা)

→ Running ⇒ Blocked (execution চলাকালীন Resource চাহিলে Block হয়ে)
Ready (time out হলে) নতুন Process

Blocked → Resource কম হলে delay না করে Blocked থাকে।

6th algo → 1. FCFS (Non)

2. SJF (no)

3. Priority (non)

4. SRTF/SJF (Pre)

5. Round around ?

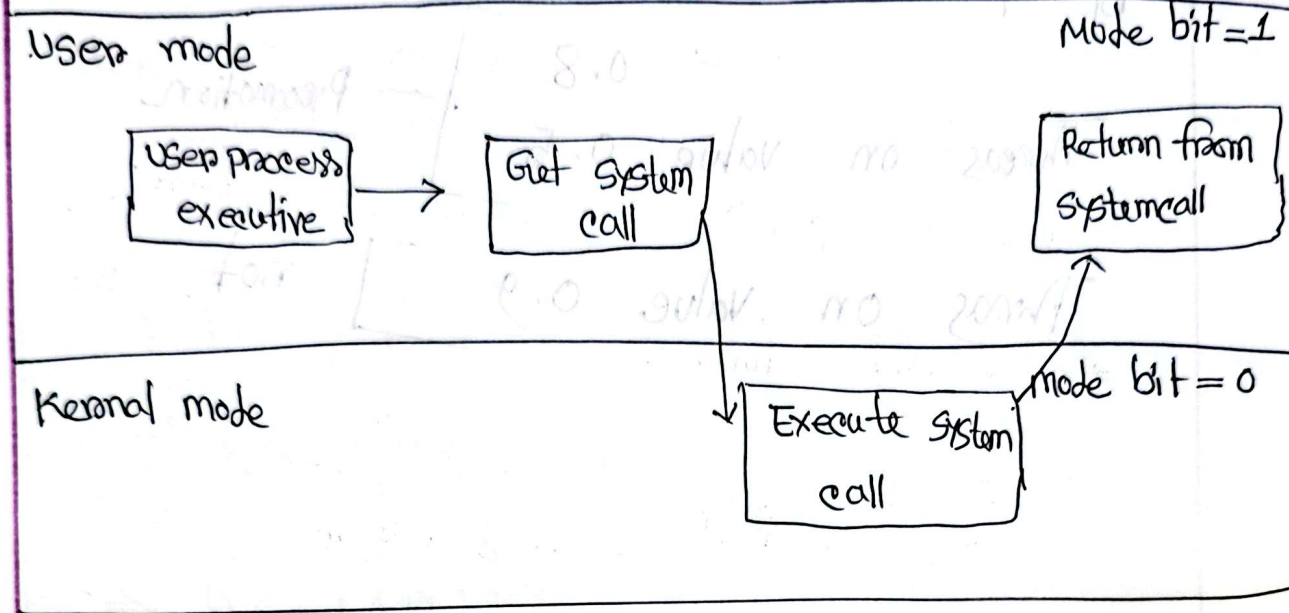
6.

compare two algorithms.

OS

22.09.24

User mode vs Kernal mode



Kernal হচ্ছে operating system এর hardware part control করে।

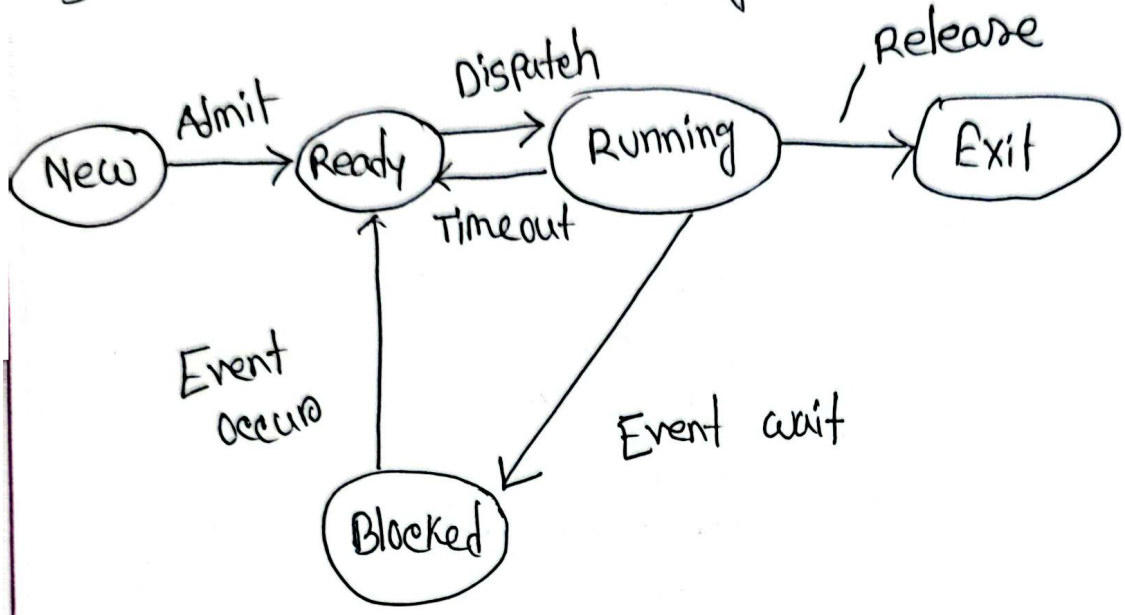
Mode থাকে 0. PC এ কাজ করার সময় সবকিছু user এ করা হয়। printer এ print করতে চাইলে PC এর নিজে PC নিয়ে যাচ্ছে। PC কাজ করে operating system এর I/S

or System call → hardware related support নিতে পারে।

Assignment

1. What is system call?
2. Basic কয়েকটা system call func নাম লেখ নিয়ে যাবে

5 State process model diagram :



* → সব execution হয়ে গেলে release
* time এর ক্ষেত্রে ।। না হলে time out হয়ে ready queue,
* non preemitive এ নাই
* Blocked - সমাধান অন্যের resource না পেলে process blocked হয়,

৫টা algorithm Scheduling

3. ২টা algorithm compare করা হচ্ছে ৬টা criteria দিয়ে।

algorithm এর performance কিভাবে measure করব।