Date ___/___/___

Embedded Systems

realtime - part of embedded systems
- Millions of computing systems are built
- different system with different purposes
- Embedded within large electronic devices.

Computing system embedded within large electronic devices are embedded systems.

- Billions of units produced yearly Vs millions of desktops.

Consumer Electronics : Cellphone, digital camera, video games.
- Home Appliances : Microwave
Office Automation : Fax, Copier
Business Equipments : Cash reader, alarm System

Characteristics of Embedded Systems
Single functioned - executes repeatedly
Tightly Constrained - low cost, low power, small size, fast computing.
our course ← Reactive and real time - continually reacts to changes in the system's environment.
- Must compute certain results in real time without delay.
Individual units are computing devices (Microprocessor)

GPP - General purpose processor
SPP - Single " "
ASP - Appl'n specific " " - (micro controller)
designed for specific function.

→ Design Challenge
→ Designer goal
- Construct an implementation that fulfil desired functionality, optimality.
- Design metrics
A measurable feature of a system's implementation
- Optimizing designing metrics is key challenge.

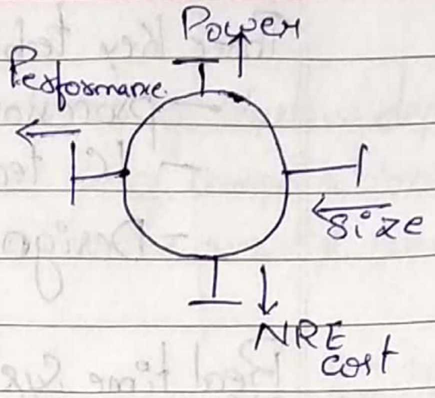
Common Metrics

Unit cost	time to prototype
NRE cost (one-time)	time to market
Size	Maintainability
performance	Correctness
power	Safety.
flexibility	

To best meet the optimization challenges.
The designer must have expertise with both software / hardware.

Date __/__/____

Costs :



NRE Cost
Unit Cost

Total Cost = NRE Cost + Unit cost × # of units

$$\text{per product Cost} = \frac{\text{NRE Cost}}{\text{\# of Units}} + \text{Unit Cost}$$

The larger the volume, lower the per product Cost.

The performance design
Clock frequency,
instructions / sec.

Eg: Dig. Camera - User care about how fast it processes image, but not ~~latency (response time)~~. Clock speed

Throughput
No. of tasks per second.
CameraA processes 4 images/second.

latency (response time)
CameraA process an image in 0.25 second.

Technology :-
A manner of accomplishing a task, Using technical processes methods or knowledge

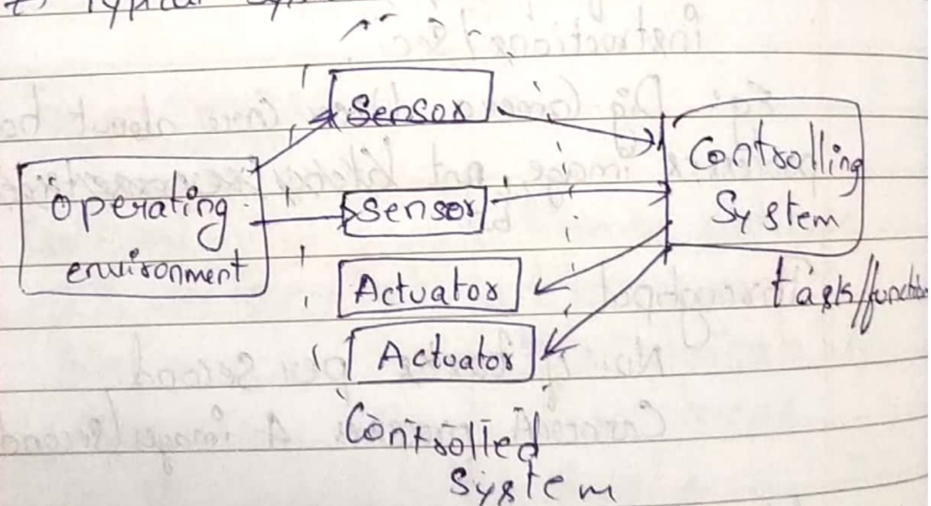
Page No.

Date __/__/__

Three key technologies for Embedded Systems.
- processor technology
- IC technology
- Design "

Real time Systems :-
A system is called real time.
Whenever we need to quantitatively express time to describe its behavior.
How to describe the behavior of a system?
List inputs to the system and the corresponding system response.

A typical system



A Sensor converts some physical environments to electronic signals.

An actuator converts electric signals from computer into physical actions

Saathi

Date __/__/____

In RT system, the correctness of system depends not only on the logical result of computation but also on the time at which the results are produced.

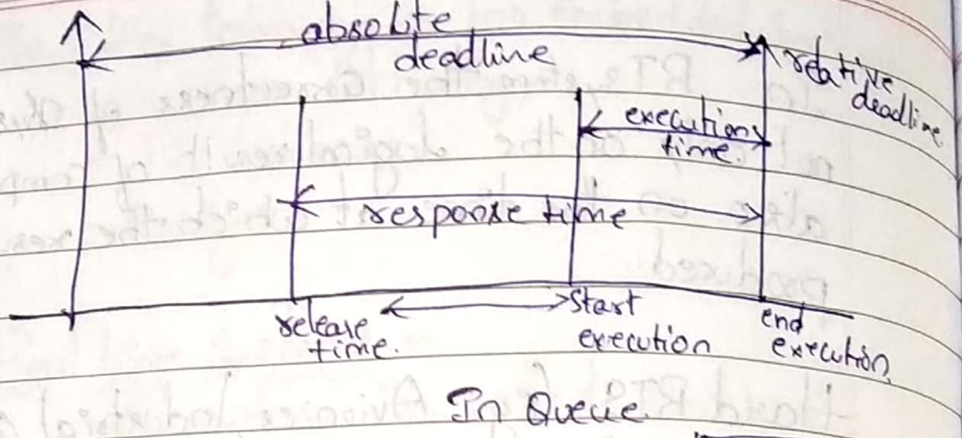
Hard RTS (eg: Avionics, Industrial control appl'n, on-board computers)
Firm RTS (Banking, Online transaction processing, Satellite based)
Soft RTS (Video Streaming, railway reservation system, web browsing)

Hard deadline	Firm deadline	Soft deadline
Penalty due to missing deadline is a higher order of magnitude	Penalty and reward are in same order	

Embedded System = Hardware + RTOS + Applic'n Pgm.

RTOS:-
An Embedded System responds to external inputs. If response time to an event is too long, the system fails.

General purpose OS:-
Not designed for real time use.
Hence, Real time OS help tasks meet their deadline.



Period = deadline

How to specify deadline and how to meet task deadline?

→ By proper task scheduling, task precedence constraints, various tasks resource requirements etc.

Characteristics of ES

Real time – Every RT task is associated with some time constraints.

Correctness Criteria

Safety and Task Criticality – A critical system is one whose failure causes whole system damage.

– A safety system does not cause any damage.

– A " critical real time system is one where any failure causes severe damage.

→ Safety and Reliability.

Independent concept in traditional systems

– A safe system does not cause damage even when it fails.

Date ___/___/___

A system can be safe and unreliable, system and vice-versa.

A safe unreliable system (word processor)
A unsafe reliable " (gun).

An unreliable system can be made safe upon failure by reverting to <u>fail-safe state</u> (traffic lights)

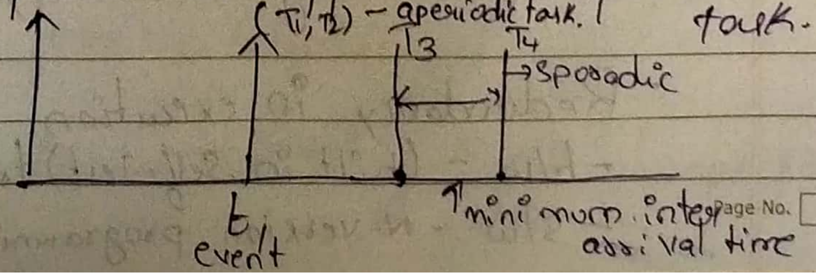
for a Safety-Critical system (RTS)
No fail safe state exists.

for a Safety-Critical System the only way to achieve safety is by making it reliable.

Realtime tasks.

Periodic tasks	Aperiodic tasks	Sporadic tasks.
after every time-period execution time (e_i) Periodic " (P_i)	(a_i, r_i, e_i, d_i) a-arrival time r-ready time d-deadline "	inter arrival time among successive instances of a periodic task rather strictly being periodic

acceptance test applicable to sporadic task.



Date __/__/____

Saathi

Hard RTS → Periodic Task $P(\phi, r, e, d)$
Firm RTS (Time driven)
Soft RTS

Phase ($\phi = 0$), release time, execution time, Period (or) deadline

→ Aperiodic task - event driven
(a, r, e, d)
arrival time, release time, execution time, deadline

→ Sporadic task
Known min. inter-arrival time
have deadline.
all the sporadic tasks need to pass the acceptance task

$$P(\underbrace{T_1, T_2, T_3, \ldots \boxed{3} \ldots T_n}_{\text{Scheduled task.}})$$

Task Constraints :-

1) Deadline Constraint
2) Resource " - Shared access (read-read)
Exclusive " (write - x) (mutual exclusive)
3) Precedence " -
$T_1 \rightarrow T_2$: Task T2 can start executing only after T1 finishes its execution
4) Fault-Tolerant Requirements :-
- To achieve higher reliability for task execution
- Redundancy in execution
- h/w - (built in self-test) triple modular redundancy
- s/w - N-version programming, recovery block

Page No.

Scanned by CamScanner

Saathi

Date ___/___/______

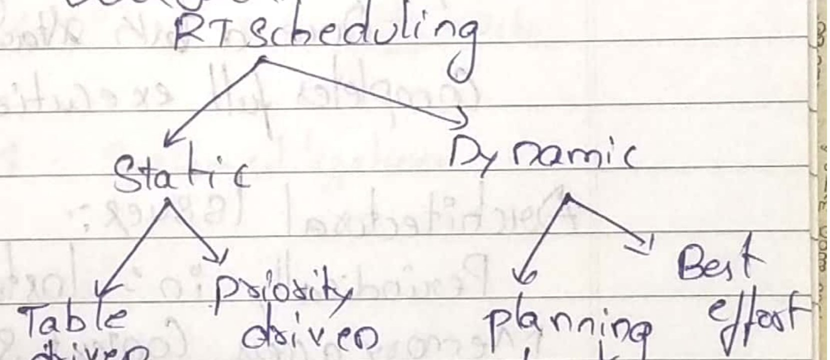
Common Misconceptions:-

Real time Computing is equal to fast Computing.
Real " pgm is assembly coding, priority, interrupt pgm, device drivers.
RTS operate in State environment.

RTS Issues:-

Resource Mangagment Issues
Architectural Issues
Software Issues

Scheduling Paradigms:-



→ Notion of Predictability.

→ Optimal Scheduling
A Static scheduling is said to be optimal if for any set of tasks, it always produces a feasible Schedule.

Dynamic Scheduling: If it always produces a

Page No.

Date __/__/__

feasible scheduler whenever a static algo with complete prior knowledge of all the possible tasks can do so.

Static - periodic tasks
Dynamic - both periodic & Aperiodic task

Preemptive Scheduling :-
Task executed is preempted and resumed later. preemption occurs to execute higher priority task.

Non-preemptive Scheduling:
- Once a task starts executing, it completes full execution.

Architectural Issues :-
Periodically in :- Instruction execution time, Memory access, Context Switching, Interrupt handling

RT systems usually avoid Caches and Super scalar features. Support for error handling (self-checking circuitary, Voters, system monitors)
Support for fast and reliable Communication (routing, priority handling buffer and timer - management).

Scheduling RT system :-

Periodic task

A ($P = 6, e = 4$) Periodic task

B ($P = 4, e = 1$) Period $\propto \frac{1}{\text{priority}}$

Period = deadline.

$\phi = 0$

LCM (6, 4) = 12 (hyperperiod)

Rate Monotonic algorithm is static algo, optimal.
if periods are prime, Hyperperiod is very long.

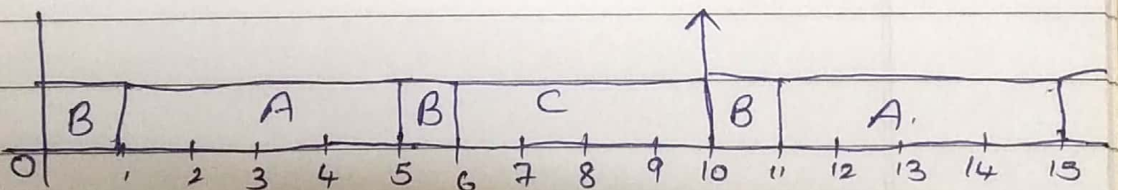
T_A ($P_A = 10, e_A = 4$) u_A (utilization) $= \frac{e_A}{P_A} = 0.4$

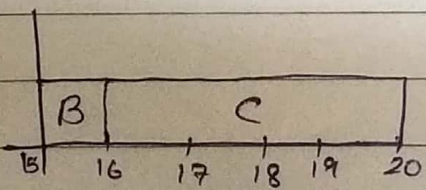
T_B ($P_B = 5, e_B = 1$) $u_B = 1/5 = 0.2$

T_C ($P_C = 20, e_C = 8$) $u_C = \frac{8}{20} = 0.4$.

$\sum_{i=A}^{C} u_i = 1$ - System utilization.

Hyperperiod = 20.





Demand for e.g 1.2

$8 = 2 * 4$

$4.8 = 4 * 1.2$ De > Supply

$8 = 1 * 8$

$\overline{20.8}$

$\boxed{u_i > 1}$ Not schedulable.

if $u_i = 1$ sometimes schedulable.

Date __/__/__

$T_A (P=10, e=4)$ $u_A = e/P$

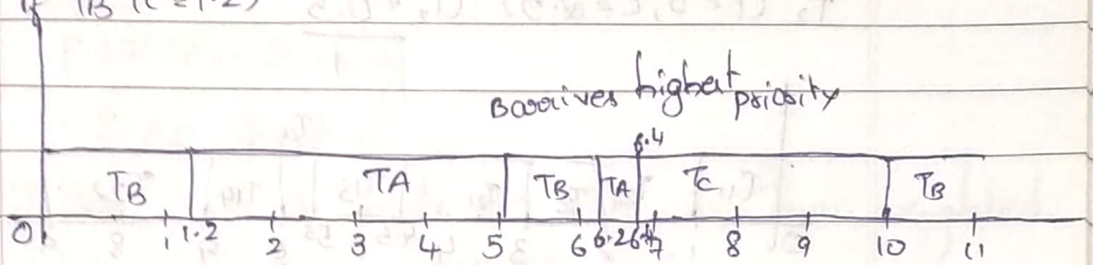
$T_B (P=5, e=1)$ $u_B =$

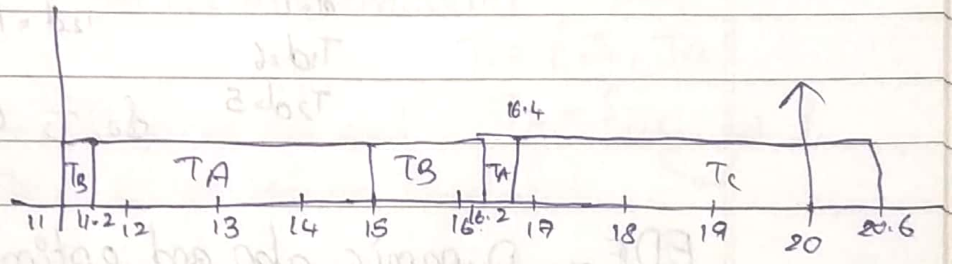
$T_C (P=20, e=8)$ $u_C =$

1

if $T_B (e=1.2)$ $\sum_{i=A} u_i = 10.04 > 1$ not Schedulable.

if $T_B (e=1.2)$



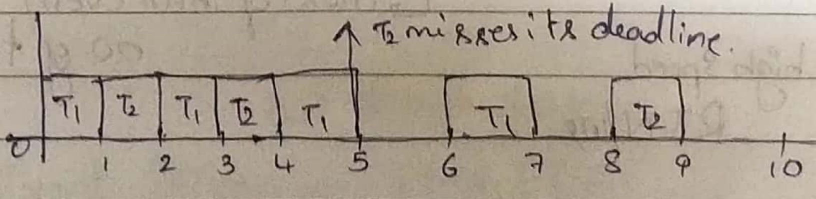


$T_1 (P=2, e=1)$ $u_1 = 0.5$

$T_2 (P=5, e=2.5)$ $u_2 = 0.5$

1

Hyperperiod (10)



RMA <1 (Schedulable)

>1 (not ")

$=1$ (sometimes Schedulable)

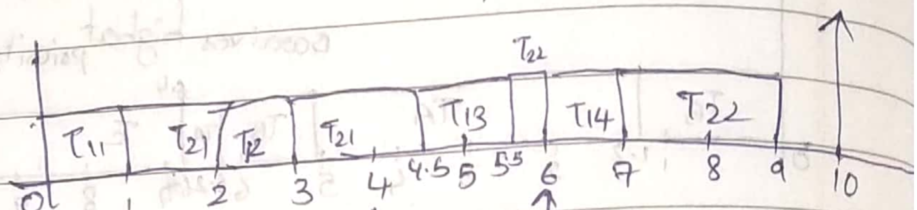
Page No.

Date __/__/__

EDF (Earliest deadline first):-

Instead of tasks, jobs are considered
→ Individual instances of tasks are job.

T_1 ($P = 2$, $e = 1$) $u_1 = 0.5$ Hyperperiod (10)
T_2 ($P = 5$, $e = 2.5$) $u_2 = 0.5$
$\overline{\quad 1 \quad}$



at = 6
$T_1d = 6$
$T_2d = 5$

$T_1d = 8$
$T_2d = 10$

So, T_2 Continues at 4 upto 4.5

EDF - Dynamic algo and optimal.
RMA - Static algo.

1) Clock driven (less tasks) Scheduling algo (or) Table driven Schedule
2) weighted round robin
3) Event driven algo

} Scheduling algos.

↳ most of them Event Drive
no. of tasks more

↳ for high speed RTN/ws.

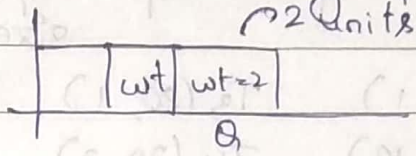
Date __/__/__

weighted round robin algo
time slice

By adjusting weights, we can speedup the task.
Therefore, it is used in highspeed RT n/ws.

2 units of execution

$\boxed{\frac{Q}{n}}$



processor sharing

i) Size $Q = \sum wt$

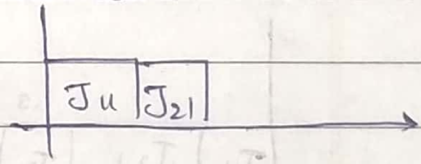
if one job depends on other job, then weighted robin increases response time.

Precedence Constraint.

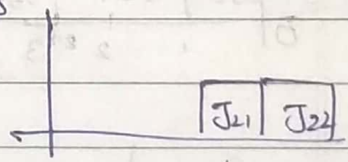
$T_1 = \{J_{11}, J_{12}\}$

$T_2 = \{J_{21}, J_{22}\}$

weighted robin is not used in case of precedence constraint.

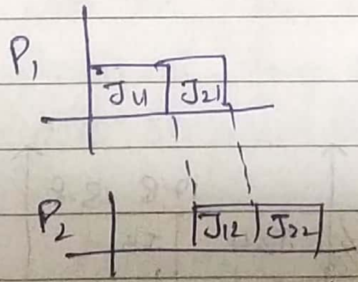


One processor



store & forward switch

Unix pipes

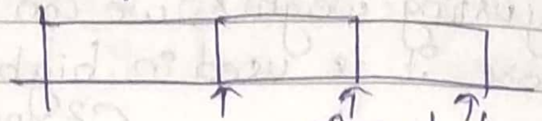


cut through switch.

Event Driven ← RMA, EDF
Scheduler schedules the jobs.
events (release time, end time).

Cyclic Scheduling algo.

frame



size of frame - design parameter

T_1 (4, 1) T_3 (20, 1)

T_2 (5, 1.8) T_4 (20, 2)

Hyperperiod P(20)

1 1.8 1

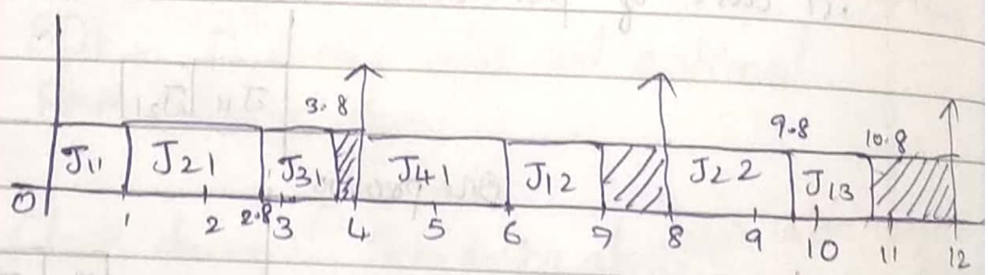
J_{11} (0, 4) J_{21} (0, 5) T_{31} (0-20)

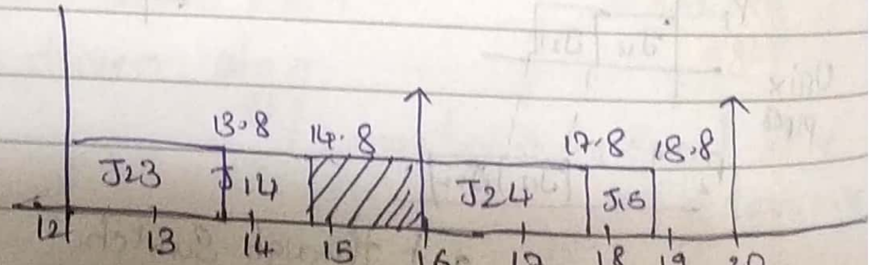
J_{12} (4, 8) J_{22} (6, 10)

J_{13} (8, 12) J_{23} (10, 15) T_{41} (0-20) 2

J_{14} (12, 16) J_{24} (15, 20)

J_{15} (16, 20)

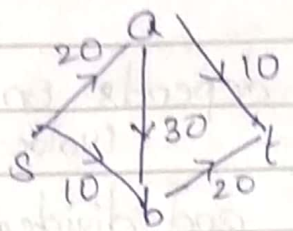




0	J_{11}
1	J_{21}
2.8	J_{31}
3.8	Idle
4	J_{41}
6	J_{12}

16	J_{24}
17.8	J_{15}

Network flow problem - A model for cyclic Scheduler



s → t

s → a → b → t = 20

s → a → t = 10

s → b → t = 10

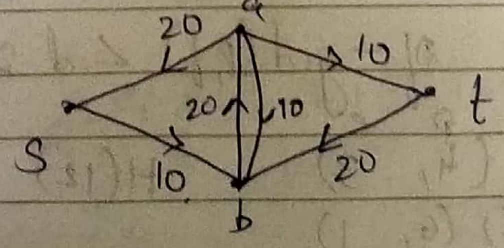
s → b → a → t = 10.

ford fulkerson algo:

1) find any path from Source — destination say 'p'
2) find the edge, lowest weight say c
   s - a / b - t
3) Push the flow on path wtc
4) Augment the graph.
5) goto step ① (till No-path s → t).

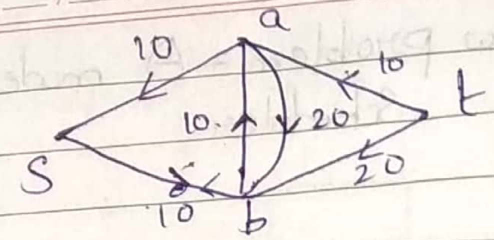
Augment is reversing the lowest wt.

s → a → b → t



Date __/__/__

$S-b-a-t.$

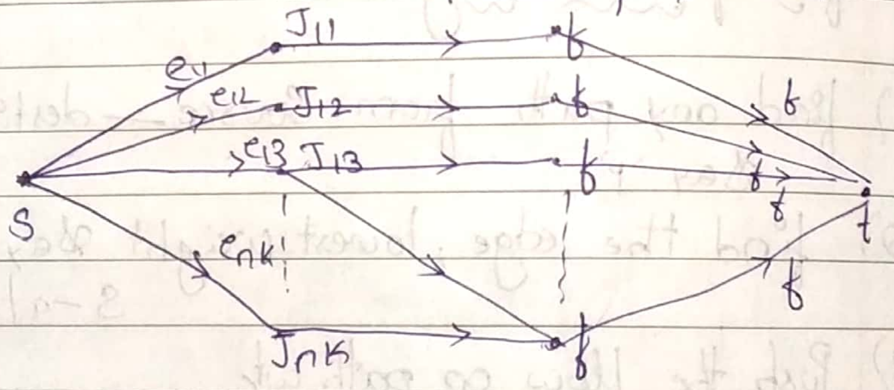


The no. of iterations depends on the weight (value of input)

Take the highest weight and divided it by 2 the complexity will be logarithmic

→ No. of task are less
↳ polynomial complexity.

→ No. of tasks more ↑ Hyperperiod ↑
↳ pseudo polynomial.



if the job gets release before the frame, then it gets scheduled in that frame.

frame should be greater than execution period
frame " " integral multiple of hyper period.

$2f - gcd(P, f) < d$ should satisfy.

	P	e
T_1	(4,	3)
T_2	(6,	1)

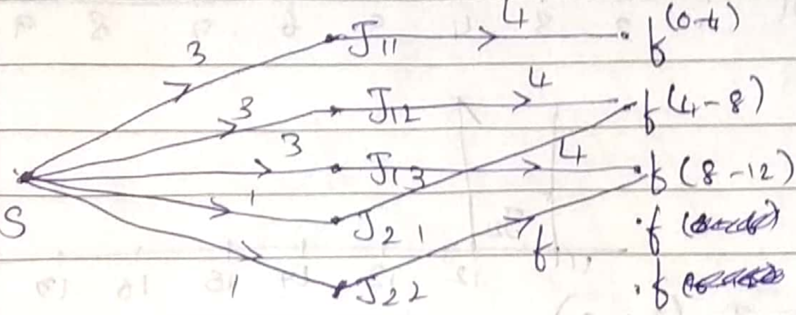
H(12)

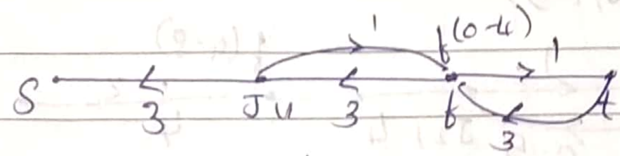
Date ___/___/___

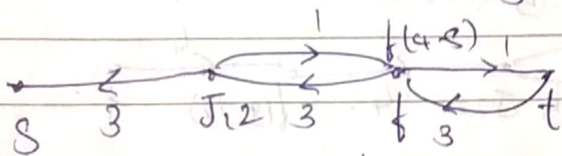
J_{11} (0-4) J_{21} (0,6) $f = 4$.

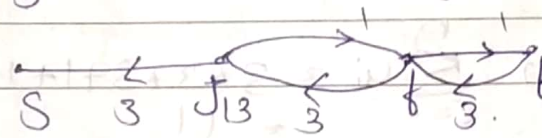
J_{12} (4,8) J_{22} (6,12)

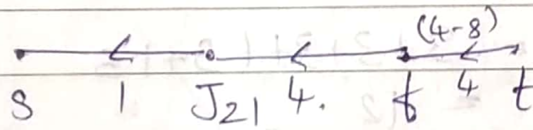
J_{13} (8,12)

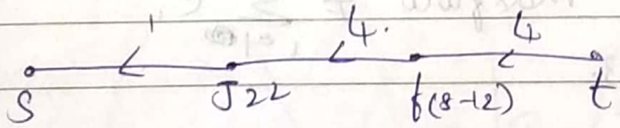


①


②


③


4)


5)


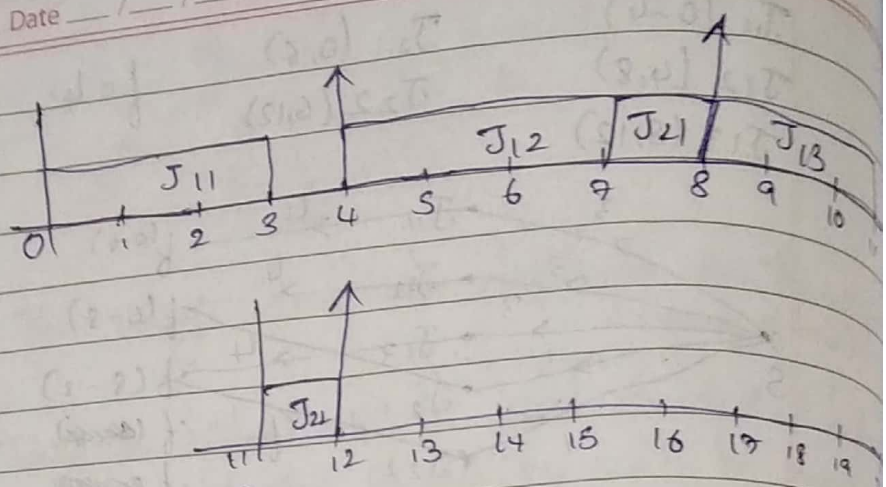
Stopping cond'n No. path from $s \to t$.

T_1 (4, 3) T_2 (6, 1)

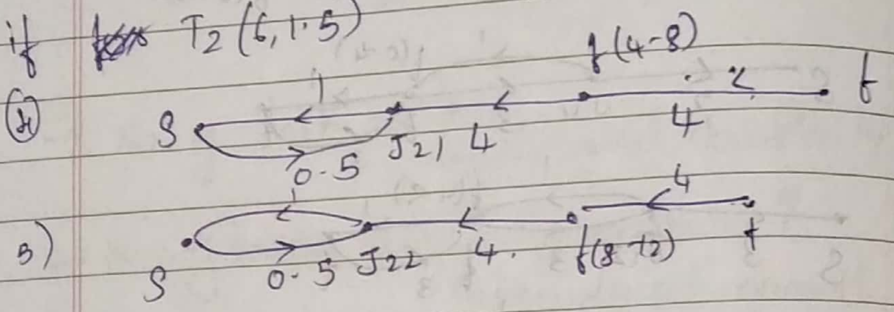
max flow $\equiv \sum_{i=1}^{n} u_i$

$$\sum_{i=1}^{n} e_i = 3+3+3+1+1 = 11$$

Date __/__/____



if ~~f~~ $T_2 (6, 1.5)$



max flow $\sum_{i=1}^{n} u_i = 3+3+3+1+1 = 11$

$\sum_{i=1}^{n} e_i = 3+3+3+1.5+1.5 = 12.$

max flow $\neq \sum_{i=1}^{n} e_i$

if $f=3.$

Slice the job with highest execution.

$T_1 (4, 3) \longrightarrow T_{1a} (4, 1.5)$

$T_{1b} (4, 1.5)$

$T_2 (6, 1.5)$ $f = 1.5.$

max flow $= \sum_{i=1}^{n} e_i = 12$

$T_1 (4, 1)$

$T_2 (5, 1.8)$

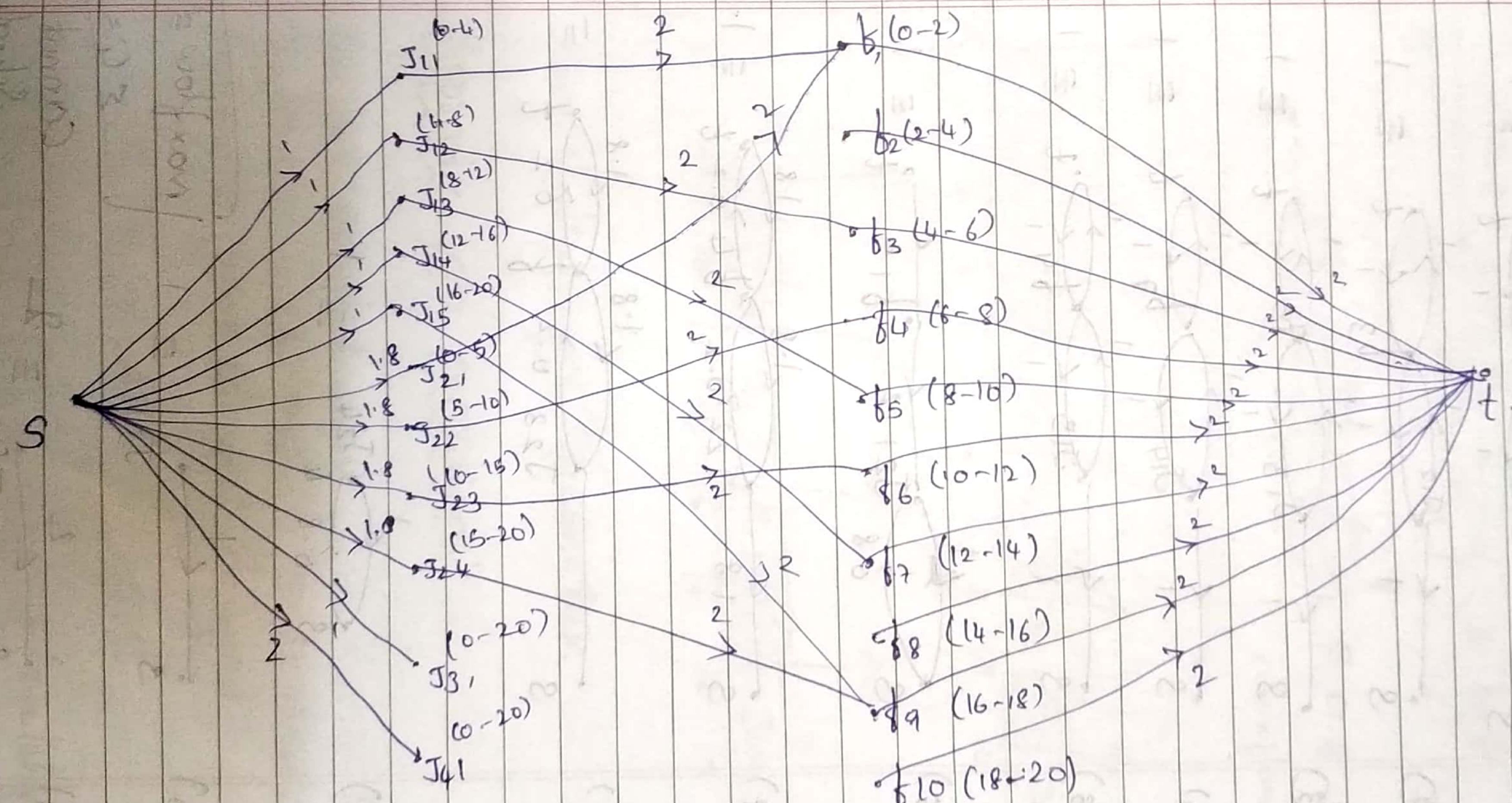
$T_3 (20, 1)$

$T_4 (20, 2)$

$2f - gcd(f, P) < d.$

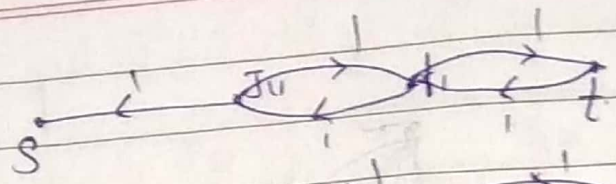
11 job needs

Page No.

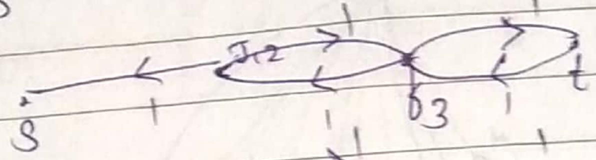


J31, J41 can be mapped to any frame.

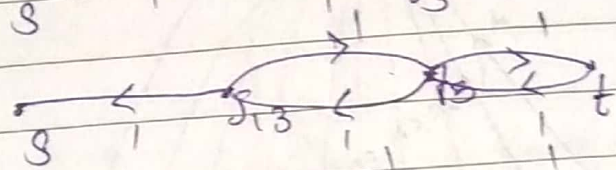
Date __/__/__

(1) 

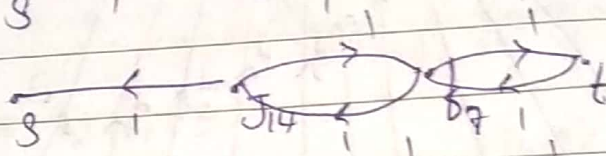
$= 1$

(2) 

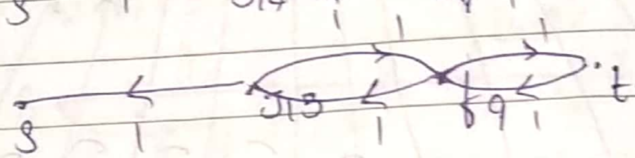
$= 1$

(3) 

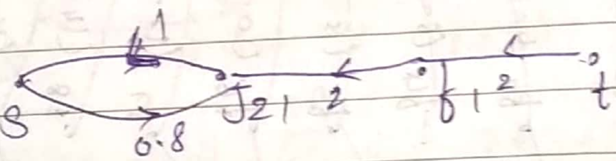
$= 1$

4) 

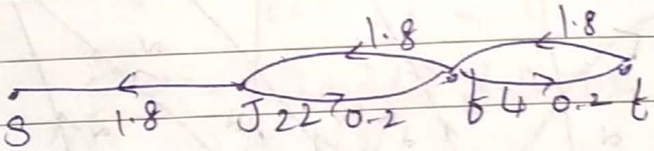
$= 1$

5) 

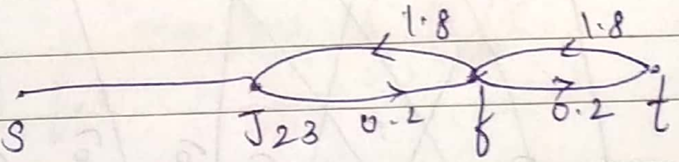
$= 1$

6) 

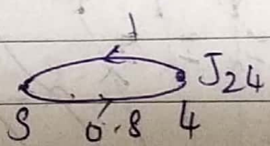
$= 1$

7) 

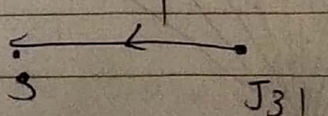
$= 1.8$

8) 

$= 1.8$

9) 

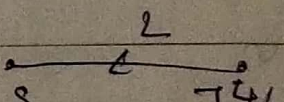
$= 1$

10) 

$= 1$

max flow $= 13.6$

$\Sigma C_i = 15.2$

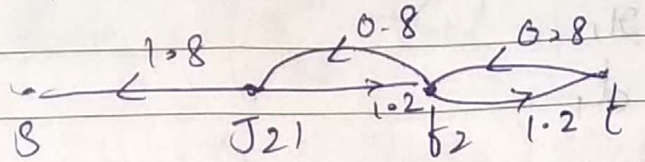
11) 

$= 2$

cannot be Scheduled this way

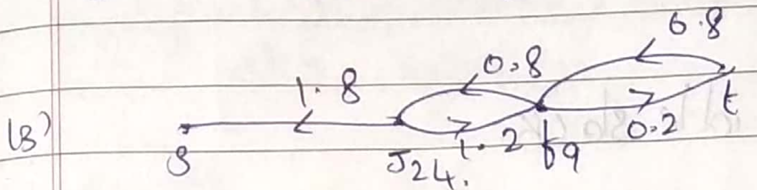
Date ___/___/___

replace ~~B and~~



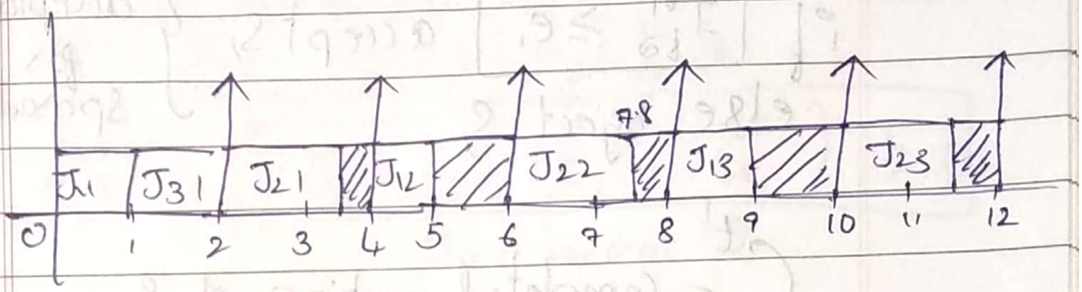
Ξ 0.8.

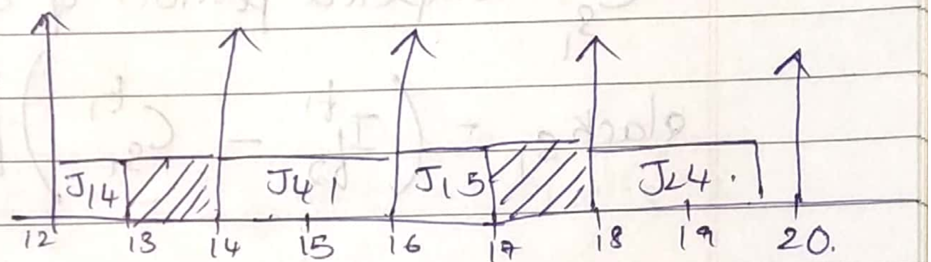
(B)



Ξ 0.8

maxflow = 15.2





Slack stealing for Aperiodic task.

Sporadic job execution :-

- Can get accepted upon acceptance test
- Should not miss periodic tasks deadline.
- Aperiodic jobs can get delayed.

$S_1 (r_1, e_1, d_1)$.

findout $[t_1, t_2]$ a interval (available frame interval).

Date __/__/__

Such that

$S_{f_1} \geq r_1$

$e_{f_2} \leq d_1$

$I_{f_2}^{f_1}$ - idle slack.

if $\boxed{I_{f_2}^{f_1} \geq e_1}$ accept S_1
else reject S_1

} acceptance test for sporadic job.

$C_{S_1}^{t'}$ - completed portion of S_1.

$\text{slack}_{S_1} = \left(I_{f_2}^{f_1} - C_{S_1}^{t_1} \right)$ $\boxed{\text{at } t'}$ e_{f_2}

t' is frame boundary.

→ $S_2(r_2, e_2, d_2)$ arrived.

Case ① $d_2 < d_1$

$S_{f_3} \geq r_2$ $\quad e_{f_4} \leq d_2 \quad [f_3, f_4]$

if $\left(I_{f_4}^{f_3} \geq e_2 \right)$ accept

else reject.

Saathi

Date __ / __ / ____

$$\text{slack}_{S_1} = \left(I_{t_2}^{t_1} - e_1 \right)$$

if ($e_2 \le \text{slack}_{S_1}$) accept S_2.
else reject S_2.

Case ② $d_2 > d_1$

If $\boxed{IS_2 - e_2 < 0}$ reject S_2.

else if $\boxed{IS_2 - e_2 - (e_1 - C_{S_1}^{t'}) \ge 0}$
then accept S_2.

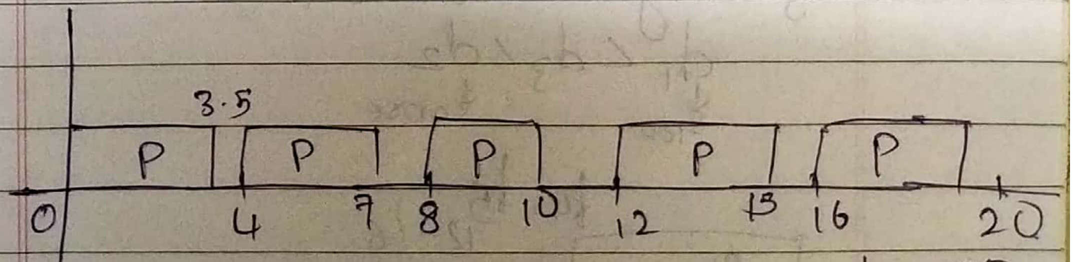
else
reject.

$$\text{Slack}_{S_2} = IS_2 - e_2 - (e_1 - C_{S_1}^{t'})$$

$Q(S_1, S_2 \cdots S_K, S_{K+1}, \cdots S_i)^{t}$

accepted Sporadic jobs

Sless		Smore
$(S_1 \cdots S_K)$	d_{K+1}	$(S_{K+2} \cdots S_i)$
$d_{S_{less}} < d_{K+1}$		$d_{S_{more}} > d_{K+1}$



$S_1 (3, 3.5, 17)$ no. of frames ⑤.

$S_2 (7, 1, 29)$ $S_3 (11, 0.5, 12)$.

Page No.

Date __/__/__

S_1 Recognised at $t=4$

f_2, f_3, f_4 $\quad [\underset{f_1}{4}, \underset{f_2}{16})$

$$\boxed{I_{f_2=16}^{f_1=4} = 4 \geq e_1 = 3.5}$$ accept S_1

S_2 recognised at $t=8$.

8-12 frame (f_3)

12-16 f_4

16-20 f_5

20-24 f_6

24-28 f_7

$[f_3 = 8, f_4 = 28)$

$I_{f_4}^{f_3} = 5.5$ $\quad$ at $t' = 8$.

$5.5 - 1 > 0$.

$\text{slack}_{S_2} = 5.5 - 1 - (3.5 - 1)$

$= 4.5 - (2.5)$

$= 2 > 0$ accepted.

S_3 recognised at $t=12$.

$\underset{S_{1e}}{d_1} < d_3 < \underset{S_{\text{nose}}}{d_2}$

f_4, f_5

12-16

16-20

$$\boxed{I_{e_{S_3}}^{S_{S_3}} = 2}$$

Page No.

Date ___/___/___

$I_{S_{K+1}} - e_{K+1} - \forall_x (e_x - C^{t'}_{S_x}) < 0$ $t' = 12$

$= 2 - 0.5 - (3.5 - 3)$

$= 1 > 0$

Now, $\forall_{x \in S_{more}} (slack_{S_x} - e_{K+1}) \geq 0$

$2 - 0.5 \geq 0$ $\boxed{\text{accept.}}$

General Case.

if $(I_{S_{K+1}} - e_{K+1}) < 0$

then reject

else if $\underbrace{\left(I_{S_{K+1}} - e_{K+1} - \forall_{x,\, S'_x \in S_{less}} (e_x - C^{t'}_{S_x}\right)}_{A} < 0$

the Reject.

else if $(\forall_{x \in S_{more}} (slack_{S_x} - e_{K+1})) \geq 0$

then accept.

else Reject

$\boxed{slack_{S_{K+}} = A}$

Page No.

Practical factors :- for Sporadic jobs & Aperiodic jobs.

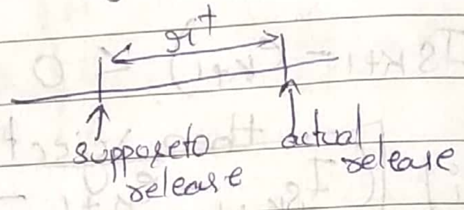
1) Execution time jitter

$e_i < e_i^{+t}$ (+time)

e_i ↑ Th, e_i^{+t} ↑ actual

① Sporadic job with least deadline
② Aperiodic job (Slack stealing)

2) Release time jitter



3) Mode Change → feedback to operator
Mode Change → Sporadic job

$S_1 = \{S_1, S_2 \dots \theta_i \dots S_n\}$

$S_2 = \{ \dots \}$

Priority Based Scheduler :-

no. of tasks are more.

1) fixed priority. - RMA & DMA

2) Dynamic " - EDF & least Slack First

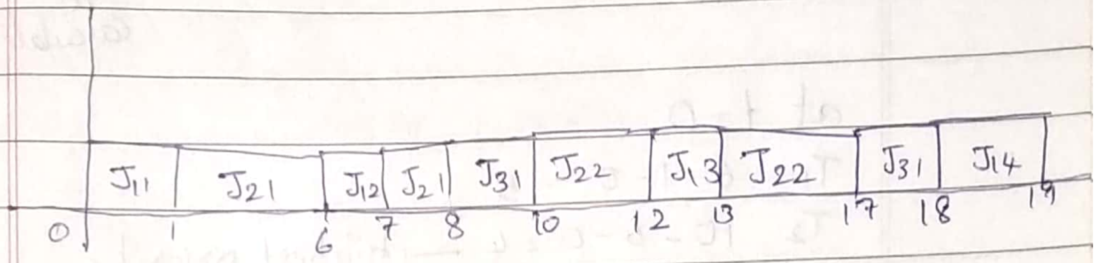
EDF → optimal algo

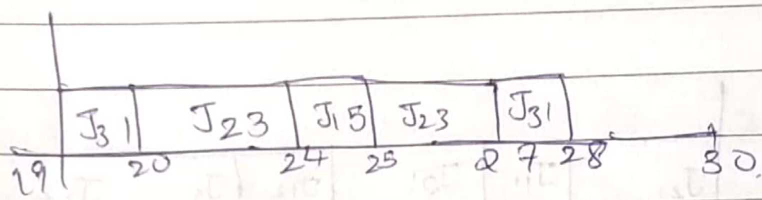
P/d, e

T_1 (6, 1) - 5 Job slices $T_1 > T_2 > T_3$

T_2 (10, 6) - 3 Jobslices

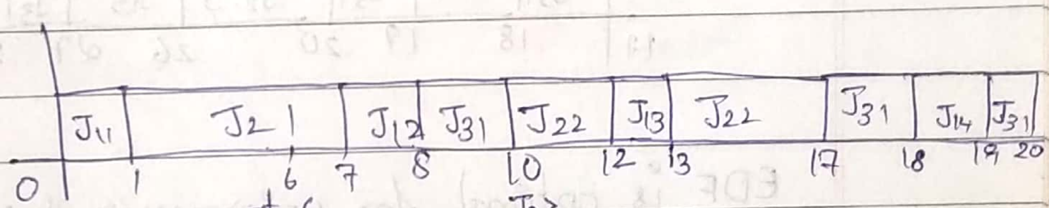
T_3 (30, 5) - J_{31}





$\boxed{RMA = DMA}$

EDF

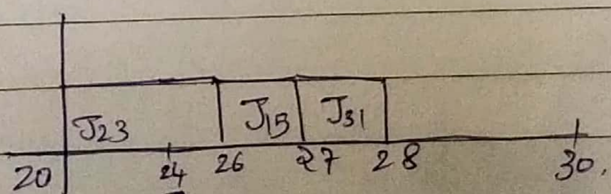


at 6 $T_2 >$

J_{12} Comes but has deadline 12

J_{21} has dedline = 10

So, J_{21} continues



T_2 continues

Least Slack first.

$$\text{Slack} = \underset{\substack{\uparrow \\ \text{absolute deadline}}}{d} - \overset{\substack{\text{remaining execution time} \\ \uparrow}}{e_r} - t$$

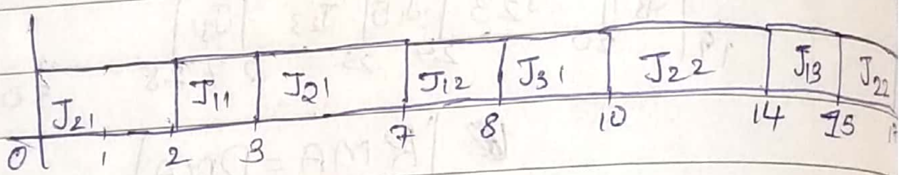
t ← time unit at which this is calculated.

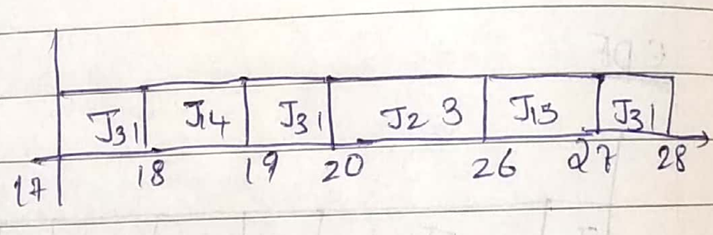
at $t = 0$

T_1 $6 - 1 - 0 = 5$

T_2 $10 - 6 - 0 = 4$ → highest priority

T_3 $30 - 5 - 0 = 25$





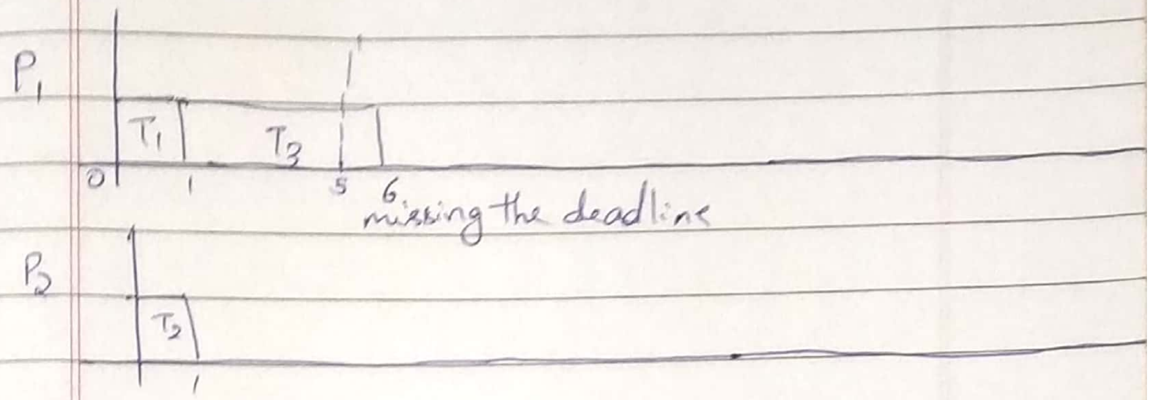
EDF is optimal for uniprocessor system preemption is allowed.

Date ___/___/_____

$T_1 \overset{r}{(}0, \overset{e}{1}, \overset{d}{4})$

$T_2 (0, 1, 4)$

$T_3 (0, 5, 5)$.



Ex:

① $T_1 (\overset{p}{4}, \overset{e}{1})$ $\qquad T_1 > T_2 > T_3$

$T_2 (P=5, e=2, d=7)$

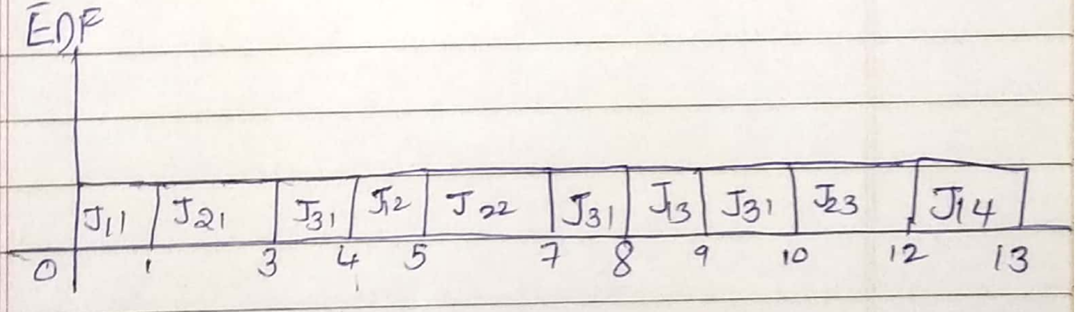
$T_3 (P=20, e=5)$

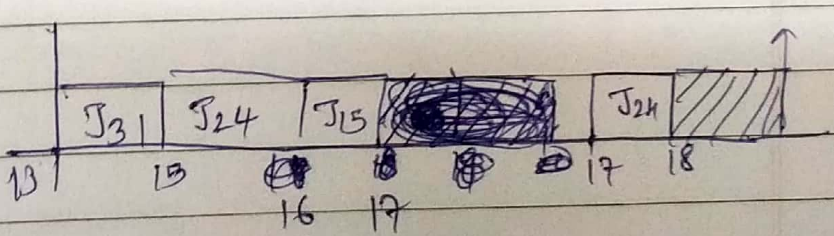
② $T_1 (\overset{p}{4}, \overset{e}{1})$

$T_2 (5, 2, 3=d)$ → RMA / DMA, EDF / LSF

$T_3 (20, 5)$

① EDF





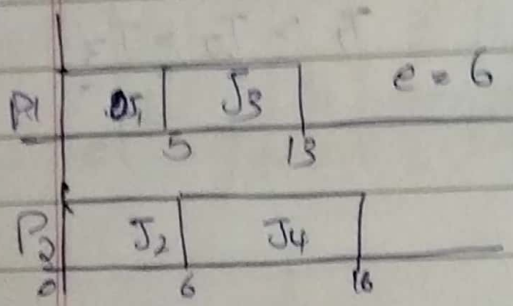
Date

Jitters in e/r / e^+ / r^+ → jitter

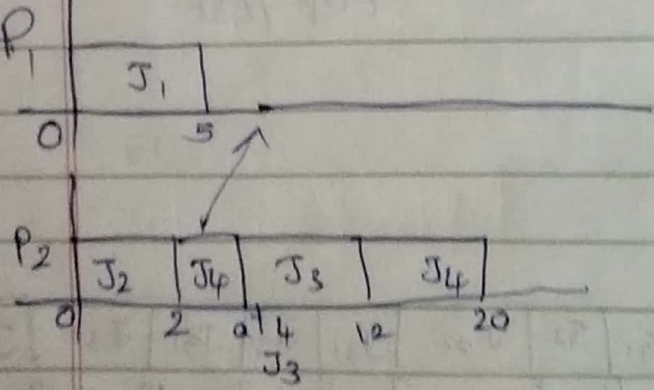
	r	e	d
J_1	0	5	10
J_2	0	[2, 6]	10
J_3	4	8	15
J_4	0	10	20

Priority driven for multiprocessor ✓ Safety Critical
Dynamic
static.

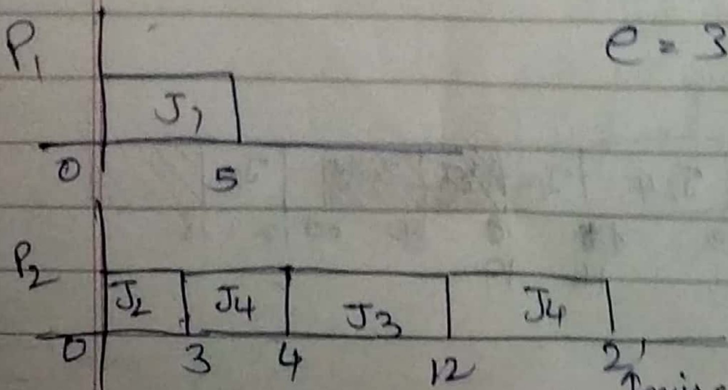
e = 6 for J_2



e = 2 for J_2



e = 3 for J_2



Misses deadline.

Page No.

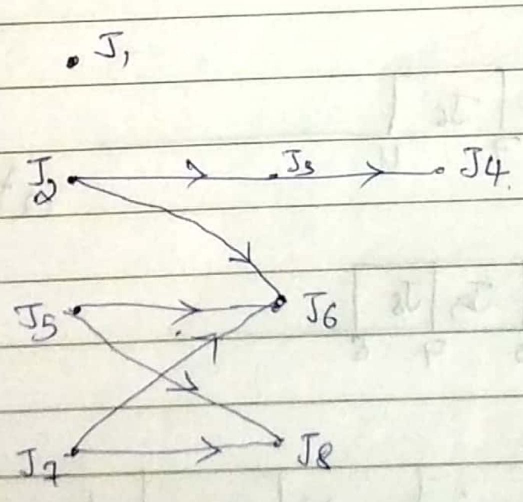
we cannot allow from one processor to other process

Preemptable Non-migratable (Dynamic)

It is dynamic priority driven Scheduler higher priority job preempts lower job priority (preemptable)

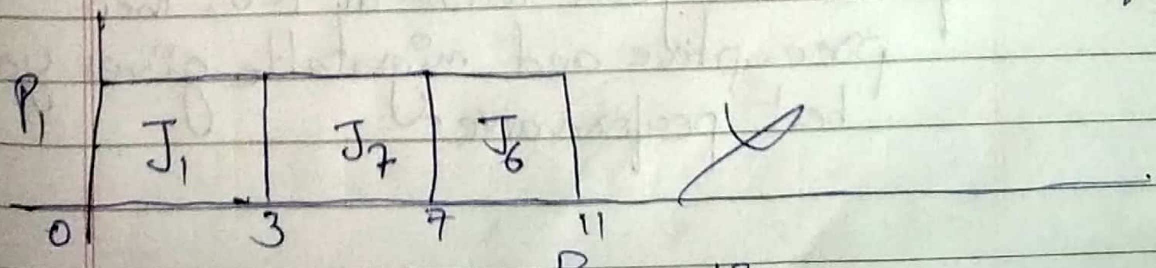
If dynamic priority driven – Pre-emptive migratable. Jitter in (e/r)

J_1 (0, 3)	J_5 (4, 2)
J_2 (0, 1)	J_6 (0, 4)
J_3 (0, 2)	J_7 (0, 4)
J_4 (0, 2)	J_8 (0, 1)
r e	r, e

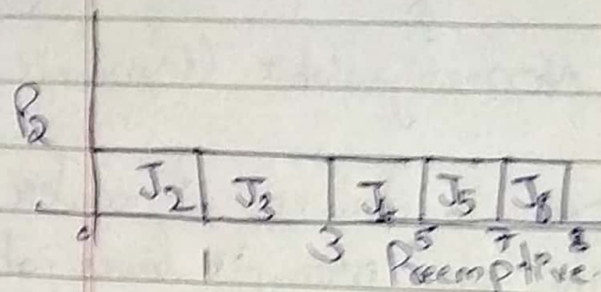


Physical constraint (predecessor)

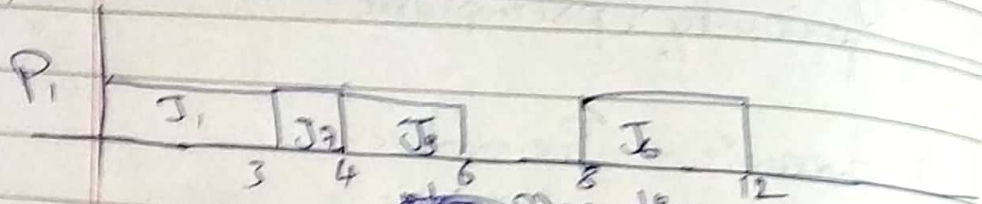
jobs are migratable (Dynamic)



Preemptive.

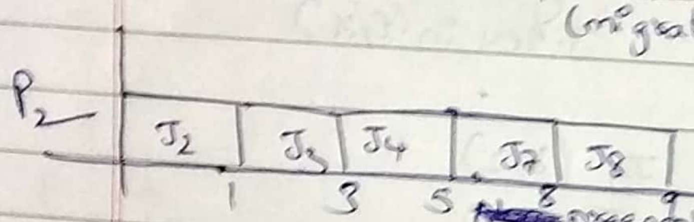


Preemptive



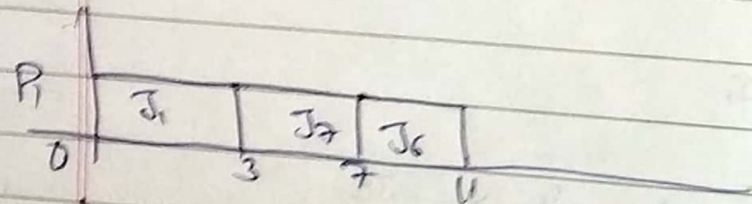
preemptive (migratable)

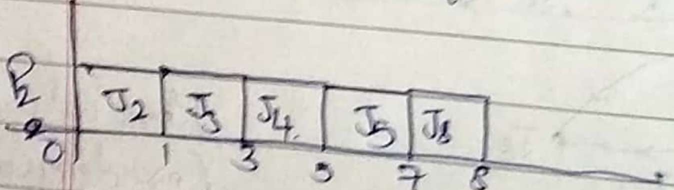
It completes at t=12



preemptive (migratable)

non-preemtable



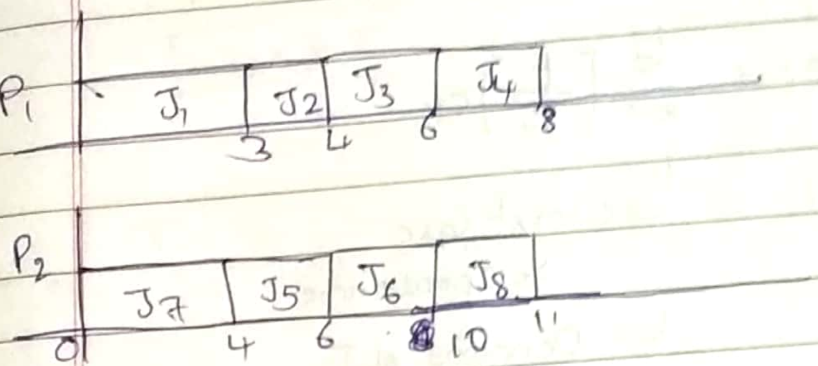


at completes at t=11

If all the jobs release at t=0, then preemptive and migratable gives you best performance.

Date __/__/__

If the Scheduler is static, the we divide the Jobs into two parts like
P_1 (J_1 — J_4)
P_2 (J_5 — J_8)



Emb. Sys 20/08/19

Effective release time & deadline

effective rt — maximum of its own & predecessor (end)

effective deadline — ? — of its own & successor (start)

Uniprocessor system

$U_{EDF} \leq 1$ } schedulable

$U \leq 1$

$U > 1$ — not schedulable (demand more than supply)

supply (100%)

J_{ci} cth instance of ith job of of task T_i

m'ssed its deadline at t

Demand of T_1

$$\sum_{T_i \in T} \frac{e_i}{p_i} t > t$$

Demand of all job $\forall i$

$$\sum_{i=1}^{m} \frac{e_i}{p_i} t \geq t$$

for → $T = \{T_1 \ldots T_m\}$

Maximum utilization of cpu.

supply = 1 (100% utilization)

$$\boxed{\sum_{i=1}^{m} \frac{e_i}{p_i} \geq 1}$$ ← (EDF is not optimal.)

demand supply

cpu is idle $(t-d)$ cpu is never idle

$$\sum_{i=1}^{m} \frac{e_i}{p_i}(t-d) > (t-d)$$

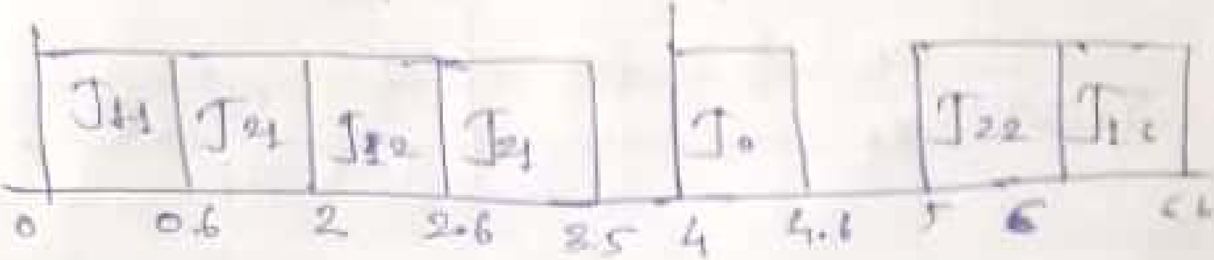
$$\sum_{k \neq i} \frac{e_k(t-d)}{p_k} + \left| \frac{e_i}{p_i}(t-d) \right| > (t-d)$$

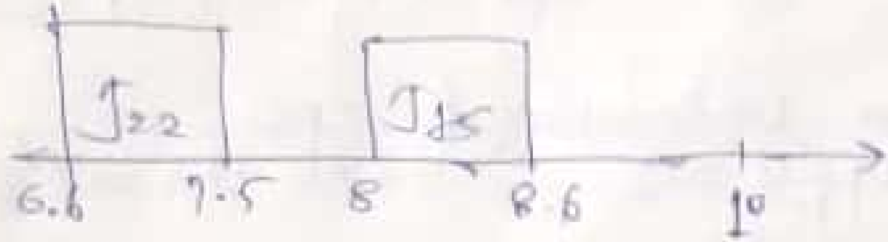
(12)

$T_1 (e = 0.6,\ p = 2)$ $\quad p = d \leftarrow (1)$

$T_2 (e = 2.3,\ p = 5)$ $\quad p \neq d \leftarrow (2)$

EDF





$$\left(\frac{0.6}{2} + \frac{2.3}{5} \right) < 1$$

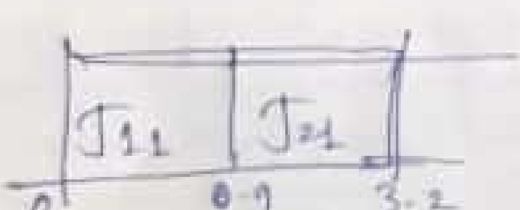
$T_1 (e = 0.9,\ p = 2)$

$T_2 (e = 2.3,\ p = 5)$

$d = 3$

$p = d \leftarrow (1)$

$p \neq d \leftarrow (2)$



$d = 3$

(1) $D > P$

N & S

(2) $P > D$

sufficient but not necessary.

$p \neq d$

$$\boxed{\sum_{i=1}^{m} \frac{e_i}{\min(p_i, d_i)} \leq 1}$$

$T_2(e=2.5,\ p=5)$ } — 1
$T_1(e=1,\ p=2)$

$T_a(0.8,\ 2)$ } — 2
$T_b(3,\ 5)$

$T_a(0.8,\ 2)$ } — 3
$T_b(3.5,\ 5)$

1 $\sum_{i=1}^{m} \frac{e_i}{p_i}$

2 schedule on timeline

J_{11}	J_{21}	J_{12}	J_{21}	J_{13}	J_{22}	J_{14}	J_{22}	J_{15}

0 1 2 3 4.5 5.5 6 7 9

J_{11}	J_{21}	J_{12}	J_{21}	J_{13}	J_{22}	J_{14}	J_{22}	J_{15}

0 0.8 2 2.8 4.6 5.4 6 6.8 9.2 10

J_2

J_{11}	J_{21}	J_{12}	J_{21}	J_{13}	J_{22}	J_{14}	J_{20}	J_{15}

0 0.8 2 2.8 5.1 5.9 6 6.8 10.0 11

deadline of T_1

② Liu - layland Test:-

To check task set is schedulable by RMA or not.

$U_{RM}(n) = n(2^{\frac{1}{n}} - 1)$ n is no. of tasks

$T_1(3,1)$

$T_2(4,1)$

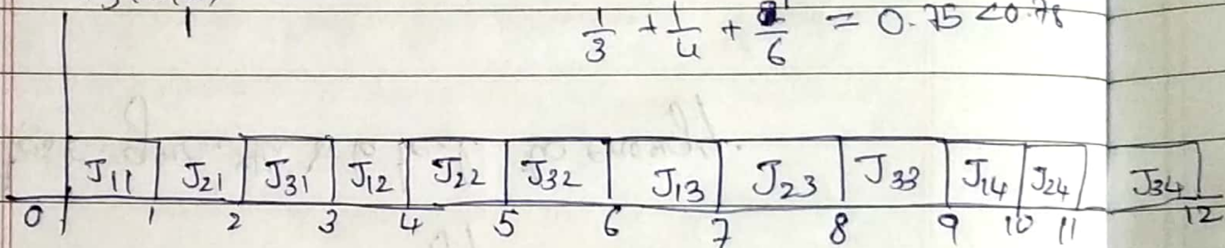
$T_3(6,1)$

$n = 3$

$3(2^{\frac{1}{3}} - 1) = 0.78$

$\sum_{i=1}^{n} u_i \leq 0.78$

$\frac{1}{3} + \frac{1}{4} + \frac{1}{6} = 0.75 < 0.78$



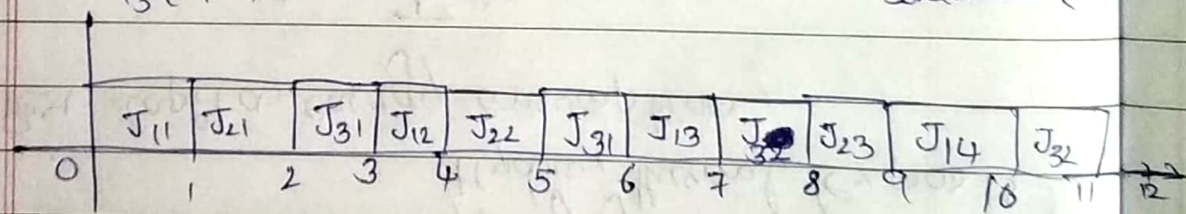
for ex

$T_1(3,1)$

$T_2(4,1)$

$T_3(6,2)$

$\sum_{i=1}^{n} u_i = 0.92 > 0.78$ ✗ not schedulable



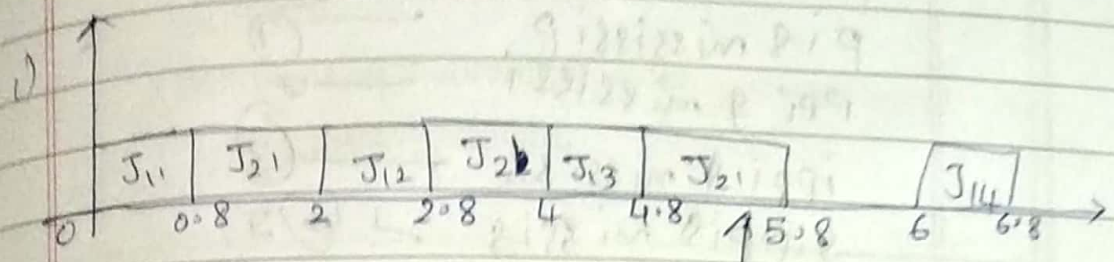
$u \leq 1$ RMA schedule is Necessary but not sufficient.

Date

1) $T_1(0.8, 2)$
$T_2(3, 5)$

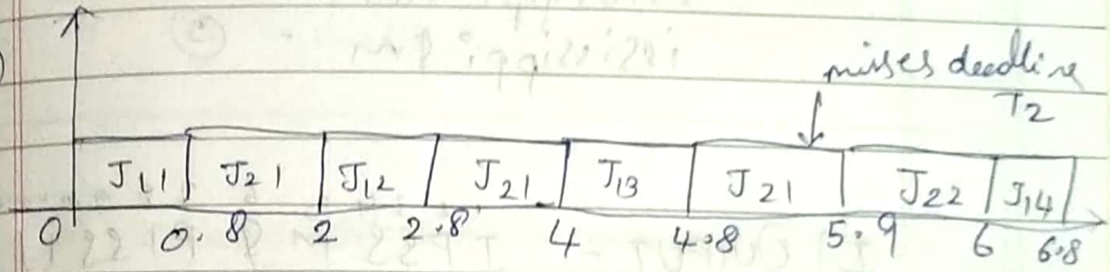
2) $T_1(3.5, 5)$
$T_2(0.8, 2)$

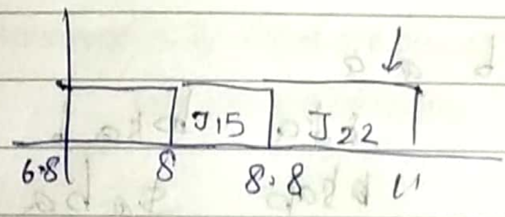
1)



U<1
RMA

2)





$U > 1$

EDF $\Longrightarrow$ RMA

D $\longrightarrow$ S

Test Schedulibity ✓

① $U \leq 1$ RMA Schedule

Periods are harmonically Related

Date ___/___/___

3) Liu - Lehoczky Test

- Completion Time Theorem
- Time Demand Analysis
- Worst case Response time Analysis.

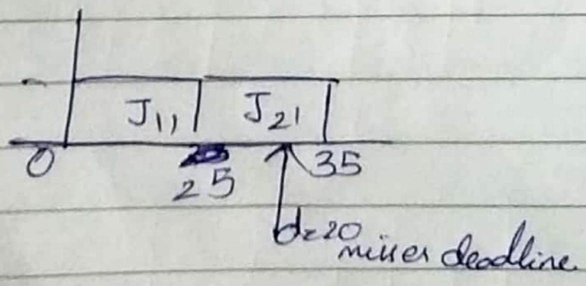
RMA, DMA

P = d.
P ≠ d. DM is optimal

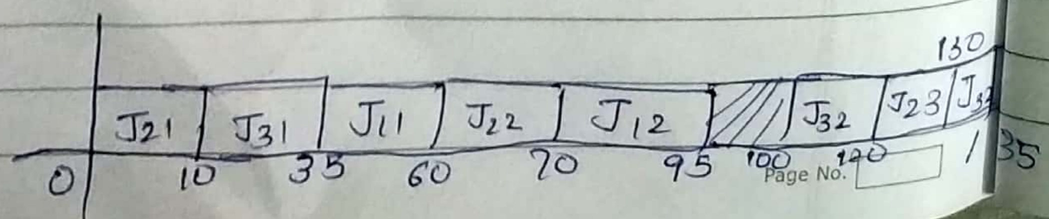
T_1 (P=50, e=25, d=100)
T_2 (P=60, e=10, d=20)
T_3 (P=100, e=25, d=50)

$T_1 > T_2 > T_3$. (RMA)
$T_2 > T_3 > T_1$ (DMA)

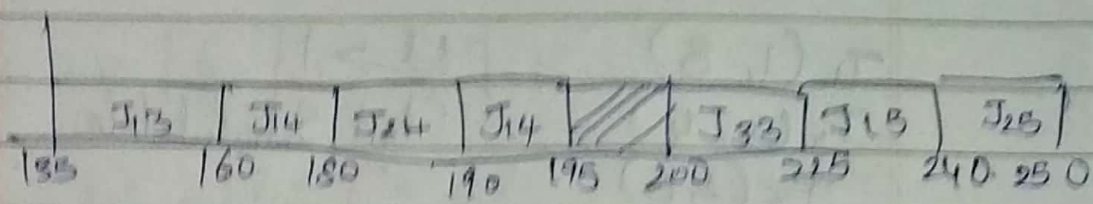
RMA

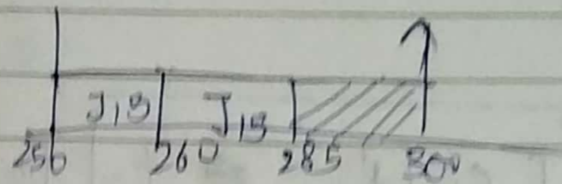


DMA



Page No.





Critical Instance of Task T_i

- time at which it is relased
- time t at which $\forall_{n=i \text{ to } 1-1}$ is released.

$T = \{T_1 \ldots T_{i-1}, T_i \ldots T_n\}$

t

Suppose from 1 to i, release time is '0'

if 1st instance meets its deadline then all there ~~to~~ instances meets deadline.

$J_{11}, J_{21}, J_{31} \ldots J_{i1}$

worst case, response time ~~analysis~~

Date ___/___/___

$T_1(1, 3)$ $U > 1$

$T_2(1, 4)$ but Schedulable

$T_3(2, 6)$

[0-t] supply

Demand

$$\boxed{e_i + \sum_{l=1}^{i-1} \left[\frac{e_i}{p_i}\right] t \quad t} \quad (0-t) \text{ Supply}$$

$$e_i + \sum_{k=1}^{i-1} \left\lceil \frac{t^0}{p_k} \right\rceil e_k = t$$

ceil

T_3: ① $2 + \frac{2}{4} \cdot 1 + \frac{2}{3} \cdot 1 = t$

$= 2 + 1 + 1 = \boxed{t^1 = 4}$

$t^0 = e_2 = 1$

② $2 + \left\lceil \frac{4}{4} \right\rceil \cdot 1 + \left\lceil \frac{4}{3} \right\rceil 1$

$2 + 1 + 2 = t^2 = 5$

③ $2 + \left\lceil \frac{5}{4} \right\rceil 1 + \left\lceil \frac{5}{3} \right\rceil \cdot 1$

$2 + 2 + 2 = t^3 = 6$

2nd ↑ 1.1 6 1 floor

Saathi

Date ___/___/___

(4) $2 + \left\lceil \frac{6}{4} \right\rceil \cdot 1 + \left\lceil \frac{6}{3} \right\rceil \cdot 1$

$2 + 2 + 2 = t^4 = 6$ $\qquad \omega_3 = 6$

(4)

$t^0 = 1 = e_2$

$\boxed{\omega_2 = 2}$

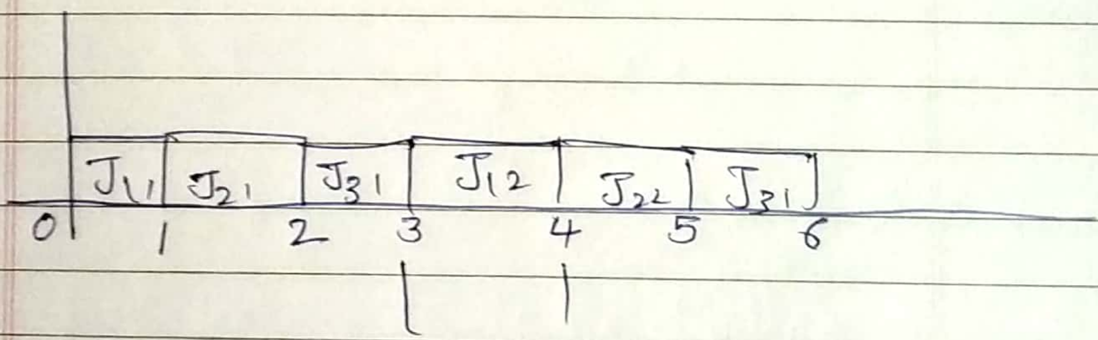
$\boxed{\omega_1 = 1}$

(1) $1 + \left\lceil \frac{1}{3} \right\rceil \cdot 1 = t^1$

$1 + 1 = \boxed{t^1 = 2}$

(2) $1 + \left\lceil \frac{2}{3} \right\rceil \cdot 1 = t^2$

$1 + 1 = \boxed{t^2 = 2}$



Recurrence relation (or) Completion time theorem.

*

23/08/19 Date __/__/__

$T_1 (1, 3)$

$T_2 (1.5, 5)$ $T_1 > T_2 > T_3 > T_4$

$T_3 (1.25, 7)$

$T_4 (0.5, 9)$

RMA.

$$T_i = e_i + \sum_{K=1}^{i-1} \left\lceil \frac{t}{P_K} \right\rceil e_K$$

↳ worst case response time

↳ Demand of T_i

$\boxed{\omega_1 = e_1 = 1}$

$\omega_2 = t^0 = e_2 = 1.5$

$= 1.5 + \left\lceil \frac{1.5}{3} \right\rceil * 1$

$= 1.5 + 1$

$t' = 2.5$

$= 1.5 + \left\lceil \frac{2.5}{3} \right\rceil * 1$

$t^2 = 2.5$

$\boxed{\omega_2 = 2.5}$

$\omega_3 = t^0 = e_3 = 1.25.$

$= 1.25 + \left\lceil \frac{1.25}{3} \right\rceil * 1.5 + \left\lceil \frac{1.25}{3} \right\rceil * 1$

$= 1.25 + 1.5 + 1$

$t' = 3.75$

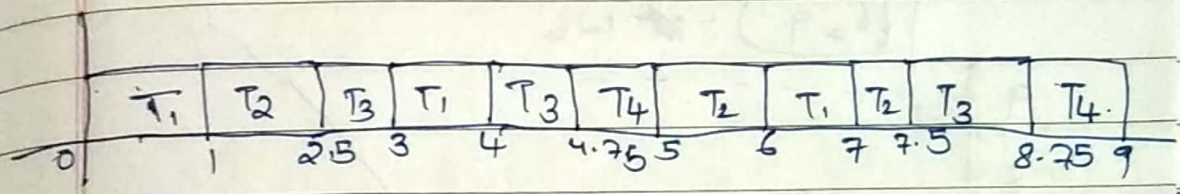
$$= 1.25 + \left\lceil \frac{3.75}{5} \right\rceil * 1.5 + \left\lceil \frac{3.75}{3} \right\rceil * 1$$

$t^2 = 4.75$

$$= 1.25 + \left\lceil \frac{4.75}{5} \right\rceil * 1.5 + \left\lceil \frac{4.75}{5} \right\rceil * 1$$

$= 4.75$

$\boxed{w_3 = 4.75}$



$w_4 = t^0 = e_4 = 0.5$

$$= 0.5 + \left\lceil \frac{0.5}{7} \right\rceil * 1.25 + \left\lceil \frac{0.5}{5} \right\rceil * 1.5 + \left\lceil \frac{0.5}{3} \right\rceil * 1$$

$t^1 = 4.25$

$$= 0.5 + \left\lceil \frac{4.25}{7} \right\rceil * 1.25 + \left\lceil \frac{4.25}{5} \right\rceil * 1.5 + \left\lceil \frac{4.25}{3} \right\rceil * 1$$

$= 0.5 + 1.25 + 1.5 + 2$

$t^2 = 5.25$

$$= 0.5 + \left\lceil \frac{5.25}{7} \right\rceil * 1.25 + \left\lceil \frac{5.25}{5} \right\rceil * 1.5 + \left\lceil \frac{5.25}{3} \right\rceil * 1$$

$= 0.5 + 1.25 + 3 + 2$

$t^3 = 6.75$

$$= 0.5 + \left\lceil \frac{6.75}{7} \right\rceil * 1.25 + \left\lceil \frac{6.75}{5} \right\rceil * 1.5 + \left\lceil \frac{6.75}{3} \right\rceil * 1$$

$= 0.5 + 1.25 + 3 + 3$

$t^4 = 7.75.$

Date ___ /___ /_____

$$= 0.5 + \left\lceil \frac{7.75}{7} \right\rceil * 1.25 + \left\lceil \frac{7.75}{5} \right\rceil * 1.5 + \left\lceil \frac{7.75}{3} \right\rceil * 1$$

6

2.5

$$= 0.5 + 2 * 1.25 + 2 * 1.5 + 3$$

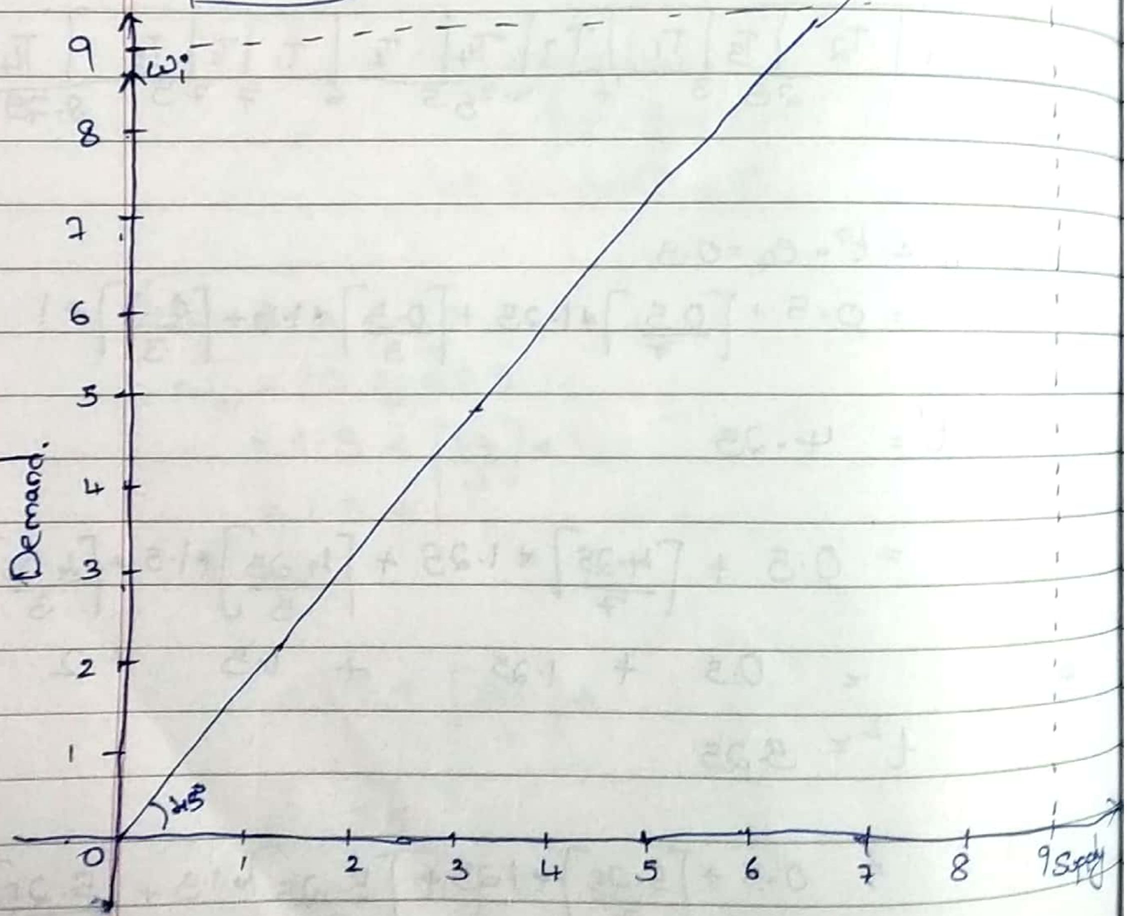
$$= 9$$

$$t^5 = 9$$

$$= 0.5 + \left\lceil \frac{9}{7} \right\rceil * 1.25 + \left\lceil \frac{9}{5} \right\rceil * 1.5 + \left\lceil \frac{9}{3} \right\rceil * 1$$

$$= 0.5 + 2 * 1.25 + 2 * 1.5 + 3$$

$$\boxed{t^6 = 9} = w_4$$



Time-demand analysis.

Scanned by CamScanner

5, 3, 2

Date __/__/__

c p

$T_1 = (1, 2)$

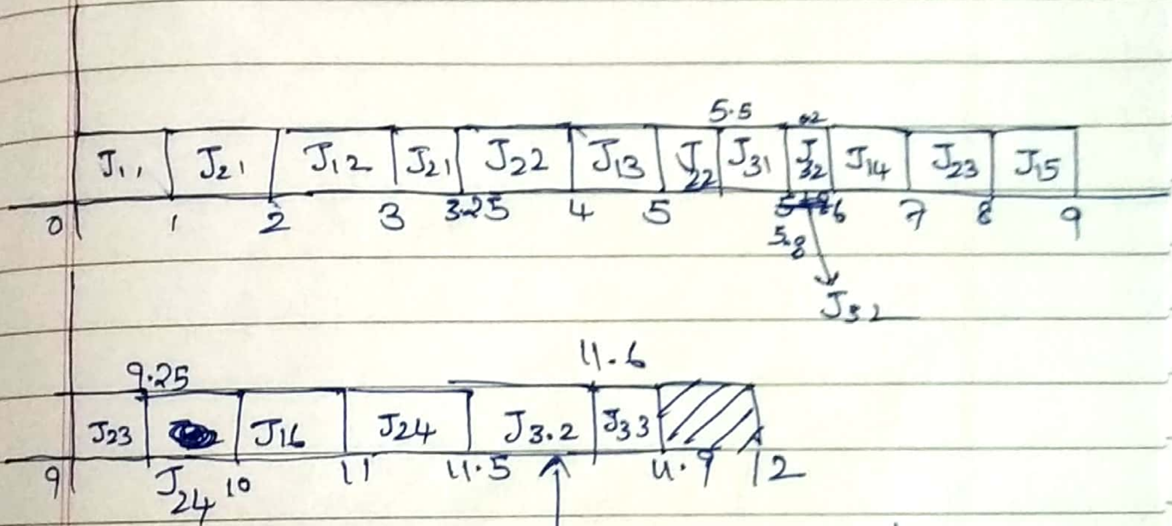
$T_2 = (1.25, P = 3, d = 3.5)$

$T_3 = (0.3, P = 5, d = 6)$

$\omega_1 = 1$

$d. > \omega_2 = 3.25 > P$

$d. > \omega_3 = 5.8 > P$



J_{32} misses the deadline

(0-5) d=6

(5-10) d=6 (11)

(10-15) d=6 (16)

This has happened because $P \neq d$.

All the above algorithms are for $P = d$ (rt=0)

we need some other strategies for $P \neq d$ tasks.

Page No.