E7

7.3, 7.4
7.9, 7.10 Q34

* key points from lecture 7
  - schedule periodic and aperiodic tasks
    * earlier deadline first scheduling
    * rate monotonic scheduling
    * deadline monotonic scheduling
    * polling server

* extra notes

  lcm: least common multiple

E7.3

* definition of schedule:
  - given a set of tasks $J = \{J_1, J_2 \cdots J_n\}$, a schedule is a map $\sigma: R^+ \rightarrow \{0, 1, \cdots n\}$, assign a task at each time t.

$$\sigma(t) = \begin{cases} k \neq 0, & \text{CPU execute } J_k \\ 0, & \text{CPU is idle} \end{cases}$$

P3 of L7

- timing constraints: (Independent Periodic Tasks)
  - worst-case Computation time C_k: time need to execute J_k
  - period T_k
  - relative deadline D_k — P8 of L7
  - worst-case response time R_k,
- for independent periodic tasks, the schedule length

$$\mathrm{lcm}(T_1, \cdots, T_n)$$

P10 of L7

- utilization factor

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i}$$

$U > 1$, not schedulable (independent of scheduling algorithm)

$U \approx 1, \leq 1$, hard to schedule

- Rate monotonic scheduling (RM)
  - fixed priorities to tasks, s.t.

$$T_i < T_j \implies J_i \text{ Priority} > J_j \text{ priority}$$

  - definitely schedulable if — P16-19 of L7

$$U < n(2^{1/n} - 1) < 0.69 \quad \text{(only sufficient)}$$

→ always scheduable if $U < 0.69$ by RM

→ might exist an RM schedule if it does not hold.

  - optimal for fixed-priority scheduling

- earliest deadline first scheduling (EDF)
  - dynamic priorities to tasks based on absolute deadlines
    execute task with shortest time to deadline d_k
  - for independent periodic tasks $D_k = T_k$, schedulable if and only if

    $$u \leq 1$$

  - cpu can be fully utilized by EDF.

---

extra notes:

- you will be faster when you are familiar with the topic
- be careful, very easy points in the exam

(a) schedulable by RM?

utilization factor $u = \frac{1}{3} + \frac{2}{4} + \frac{1}{7} = 0.97 > 3(2^{\frac{1}{3}} - 1) = 0.78$ (not satisfied)

since only sufficient, might exist one.

thus we need to draw the schedule to verify

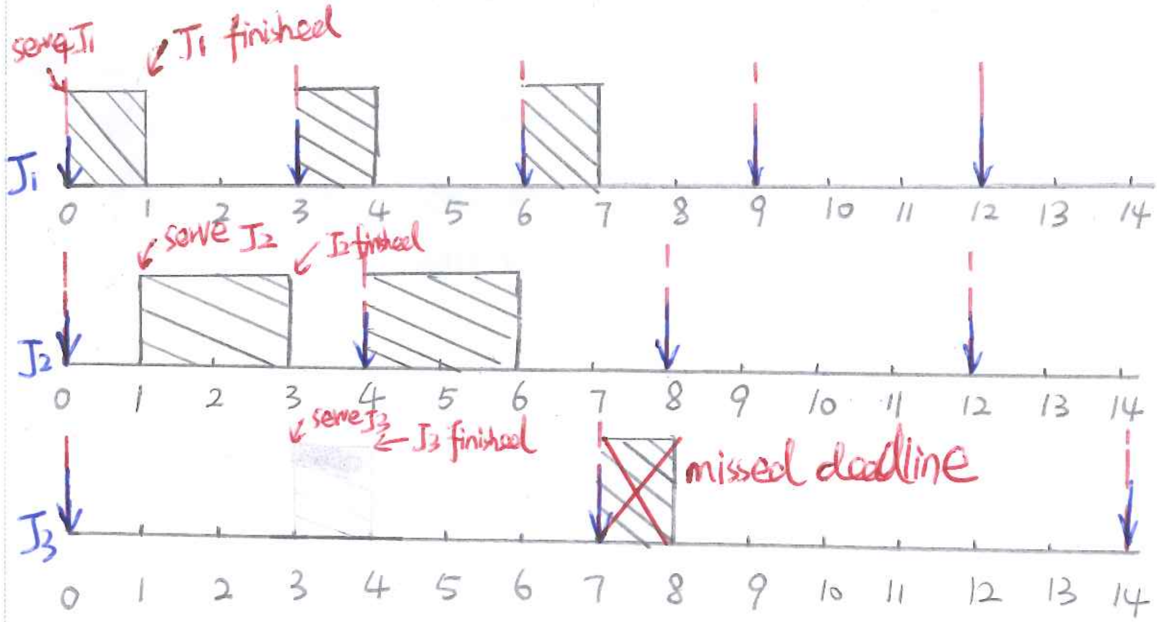
step 1: shedule length

$$\mathrm{lcm}(3, 4, 7) = 84$$

step2: Priority analysis

shorter periodic, higher priority ✓ $\Rightarrow$ priority $J_1 > J_2 > J_3$

step3: draw the schedule



RM can suspend tasks

finished served.

Priority

$J_1 > J_2 > J_3$

1° highligh the realease time by P_K

2° analyze along with the time steps, by follwing the priority

3° check missed deadlines

(b) scheduable by EDF?

Utilization factor $U = 0.97 < 1$, scheduable for sure

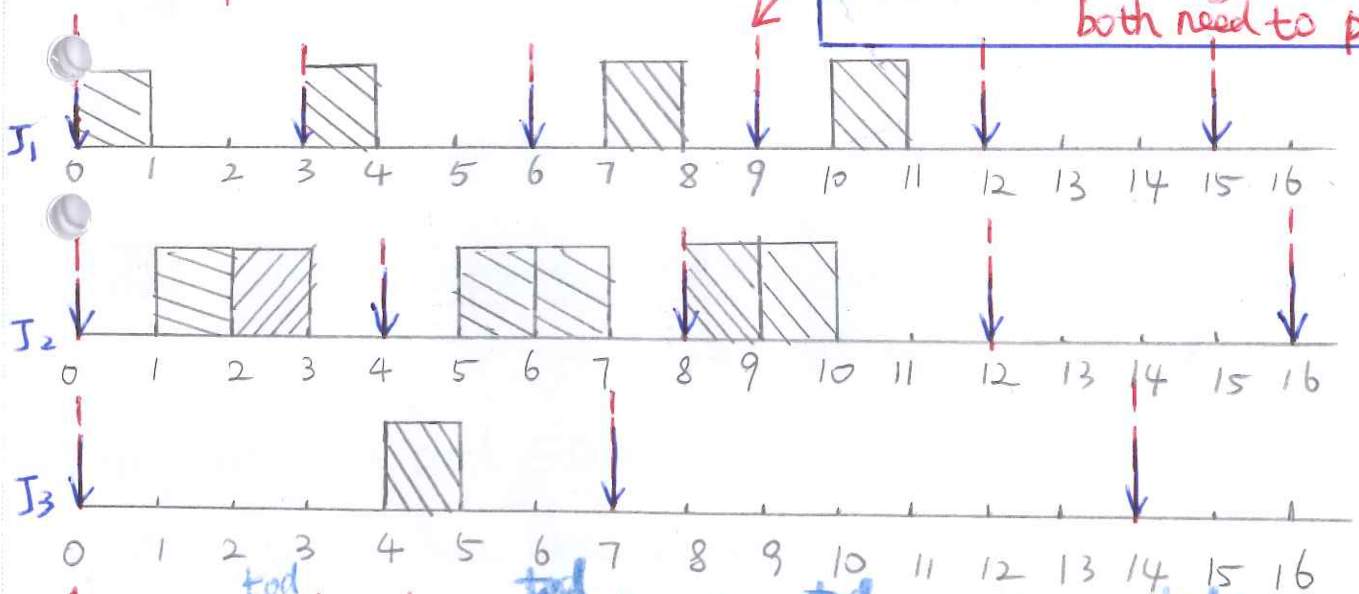
draw the schedule

step 1: schedule length 84

step 2: dynamic priority, execute tasks with shortest time to deadline)

step 3: draw

same deadline, choose anyone you prefer
both need to point out



tod, t=0: J_1, 3 ✓ tod; J_2 4; J_3, 7	tod, t=1: $J_1 = \infty$; $J_2 = 3$ ✓ tod; $J_3 = 6$	tod, t=2: $J_1 = \infty$ tod; $J_2 = 2$ ✓; $J_3 = 5$	tod, t=3: $J_1 = 3$ ✓ tod; $J_2 = \infty$; $J_3 = 4$	tod, t=4: $J_1 = \infty$ tel; $J_2 = 4$; $J_3 = 3$ ✓

E7.4

* choose among different scheduling policy
* calculate worst-case response time without drawing the schedule.

extra notes

(a) minimize the response time of J_i:

↑ time between realease and termination

compare RM and EDF

RM: worst-Case response time 1

EDF: - - - - 2

Conclusion: with EDF, it's possible to schedule tasks are not scheduable by RM, but we loose control on the response time of some tasks.

lcm(4,5,10) = 20
Short, take-home exercise
Same as E7.3

(b) Worst-case response time calculation

1° independent periodic tasks
2° fixed priority

P22 of 47

Given tasks $J_1, \cdots J_n$ with decreasing fixed priority, the worst-case response time R_i are the smallest positive solution to

$$R_i = C_i + \underbrace{\sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j}_{\text{pre-emption by higher-priority tasks}}$$

ceiling: smallest following integer

floor: largest previous integer

In our case,

$$\boxed{J_2 > J_1 > J_3}$$

For task J_2,

$R_2 = C_2$ (highest priority) $< D_2$

For task J_1,

$$R_1 = C_1 + \left\lceil \frac{R_1}{T_2} \right\rceil C_2 = 1 + \left\lceil \frac{R_1}{5} \right\rceil \cdot 2 \Rightarrow R_1 = 3 < D_1$$

For task J_3,

$$R_3 = C_3 + \left\lceil \frac{R_3}{T_2} \right\rceil C_2 + \left\lceil \frac{R_3}{T_1} \right\rceil C_1$$

$$= 3 + \left\lceil \frac{R_3}{5} \right\rceil \cdot 2 + \left\lceil \frac{R_3}{4} \right\rceil$$

$R_3 > 0 \Rightarrow R_3 > 6$

$5 < R_3 \leq 8 \Rightarrow R_3 = 3 + 2 \cdot 2 + 2 = 9$

$8 < R_3 \leq 10 \Rightarrow R_3 = 3 + 2 \cdot 2 + 3 = 10$

$\Rightarrow R_3 = 10 \leq D_3$

Thus schedulable by priority fixed $J_2 > J_1 > J_3$, the worst-case response time for J_2 is 2.

E 7.9

P24–27, L7

* polling server
  - a periodic task for serving aperiodic tasks
  - guarantee CPU utilization for aperiodic tasks

* periodic T_s, server capacity C_s

↓

scheduled by algorithms for periodic tasks

↓

once activated, serve pending aperiodic requests within its capacity

↓

several scheduling strategies for aperiodic requests.

* ~~arrival time~~ a_i ⟶ release time r_i, instead

extra notes

step 1: schedule periodic tasks with the polling server

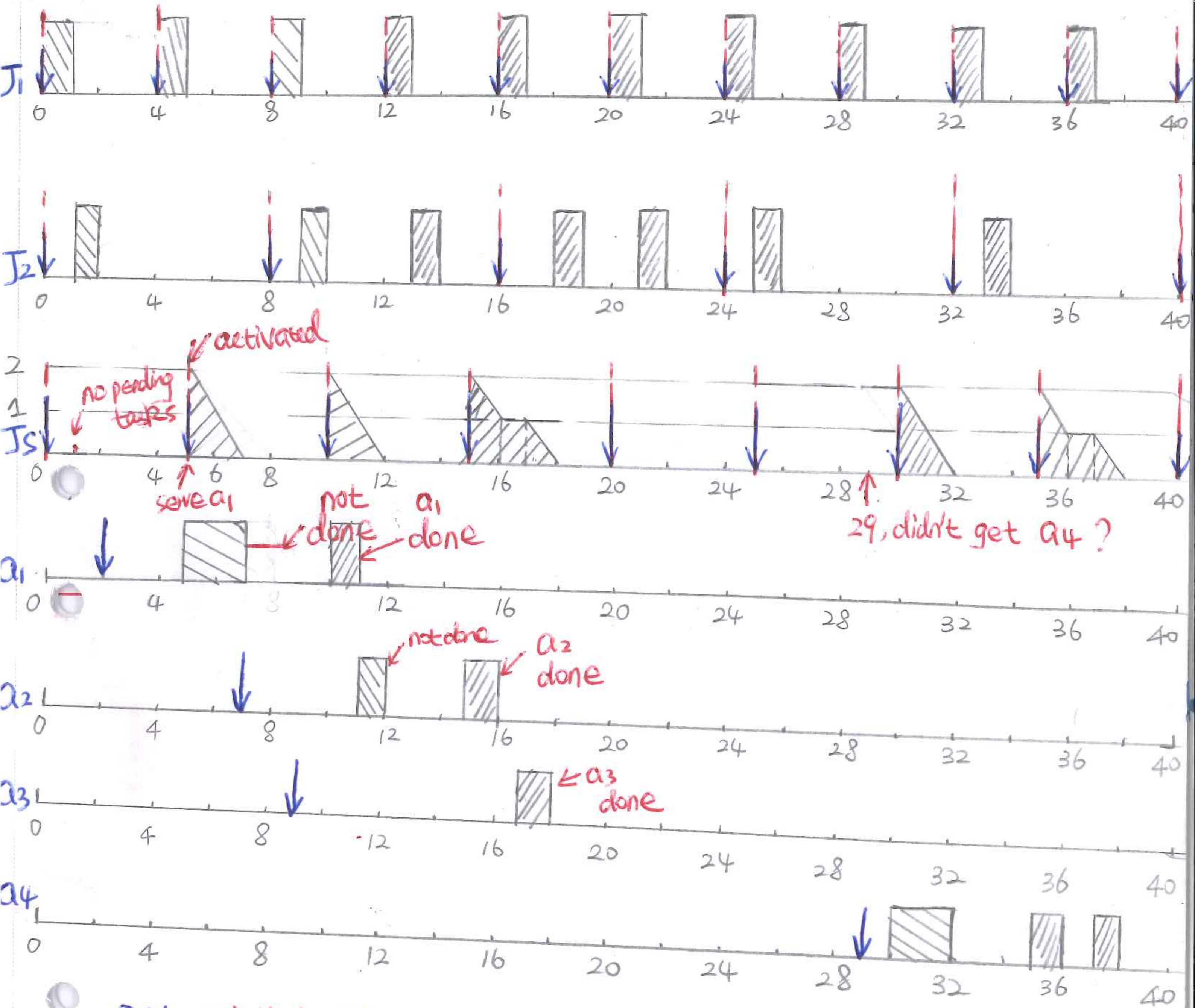
utilization factor $U = \frac{1}{4} + \frac{1}{8} + \frac{2}{5} = 0.775$

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

⟹ scheduable by RM

step 2: schedule length lcm (4, 8, 5) = 40

step 3: draw



RM: priority $J_1 > J_S > J_2$

- at t=7. a_2 is also pending. we follow "first come first serve"
- at t=1, no aperiodic task waiting. so pass to the next periodic task. suspended
- at t=2, even when aperiodic task come afterwards, can not be re-activatived! in the next period

Difference between polling server and ordinary periodic task

capacity ?

wait until the polling server is re-activated!

E 7.10

extra notes

similar to E 7.9

		C_i	T_i	D_i
• two periodic tasks :	J_1 :	1	3	3
	J_2 :	1	4	4
• polling server :	J_s :	3	6	6

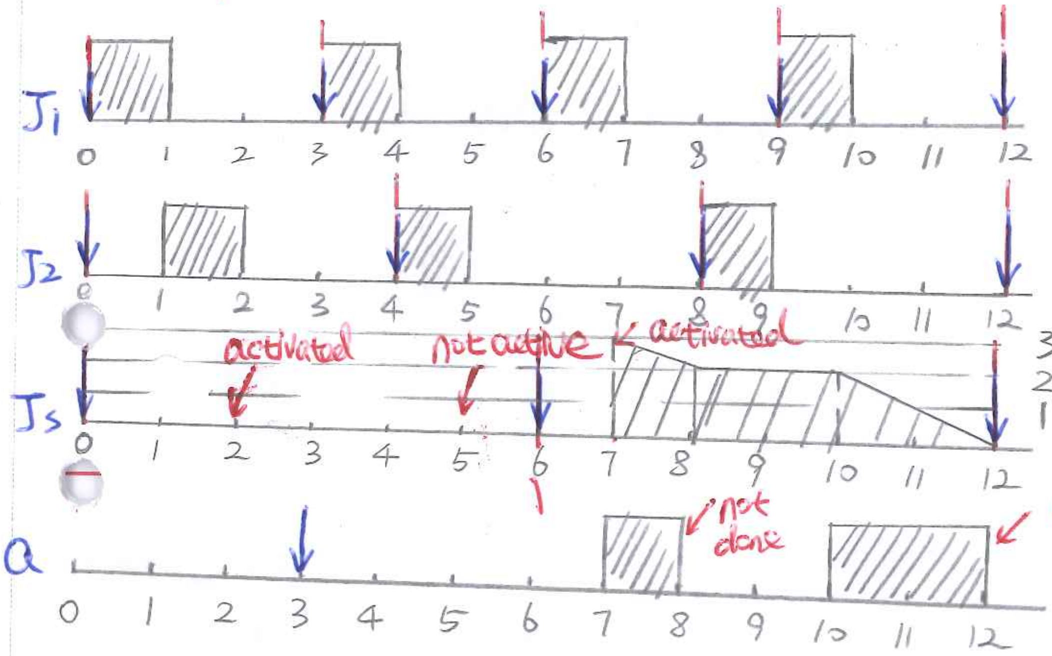
• aperiodic task : arrival time / release time $t = 3$
computation time $C_a = 3$

step 1 : schedule length lcm(3, 4, 6) = 12

step2 : rate monotonic algorithm

fixed priority : $J_1 > J_2 > J_s$

step 3 : draw



been preempted by higher order tasks

t=5, even though there are aperiodic pending, it has to wait until the polling server is re-activatied!
reactivated

t=2, activated by RM scheduler, but no pending tasks, the polling server is suspended

check P_{27} of L7