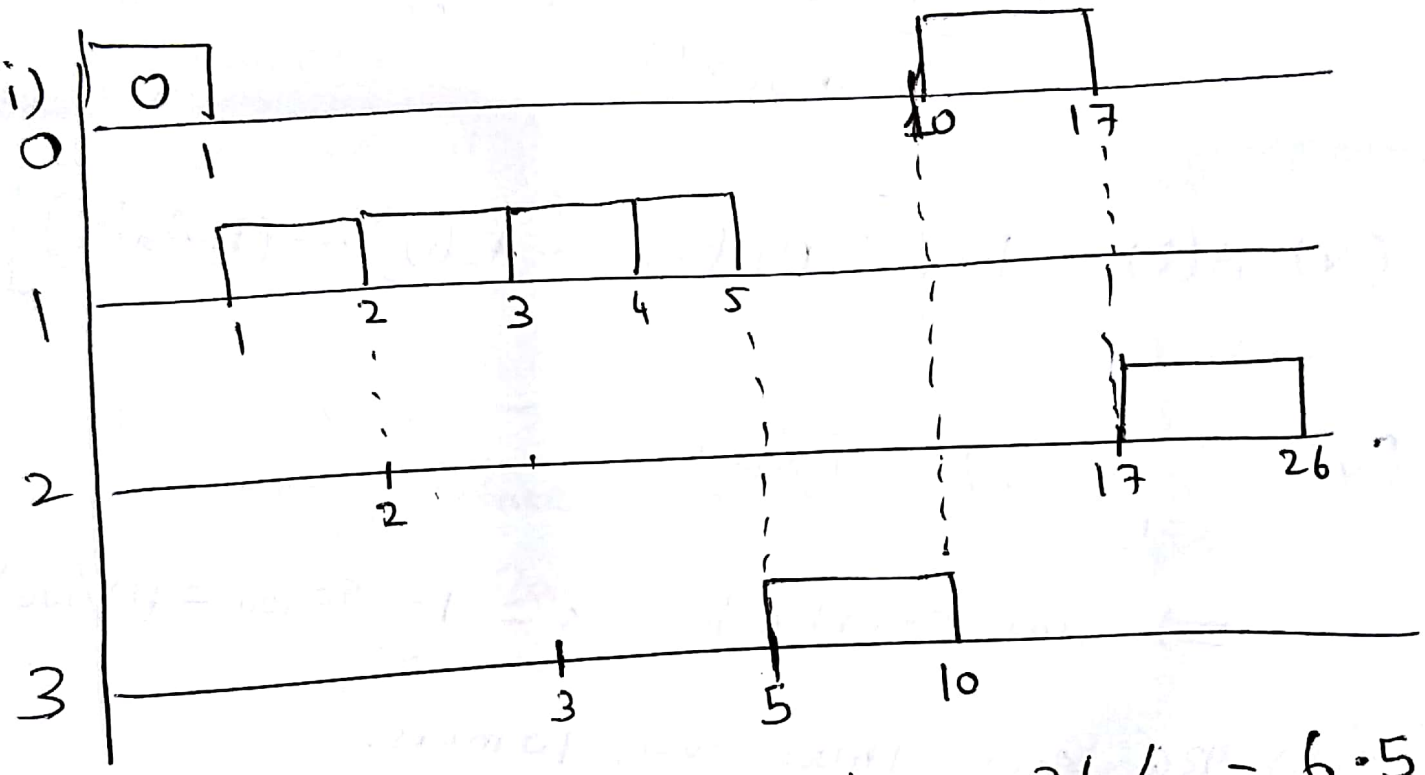
(i) For $n=2$, $\sum_{i=1}^{2} U_i \leq n(2^{1/n}-1)$

$\Rightarrow \sum_{i=1}^{2} U_i \leq 0.824$

(ii)



average waiting time of tasks $= 26/4 = 6.5$

(alternatively, $9+0+15+2 = 26/4 = 6.5$)

(iii) min. separation : $\boxed{GCD(F, p_i)}$ where F is the frame size and p_i is the periodicity of task i.

(iv) λ: failure rate of a core

$F(t) = 1 - e^{-\lambda t}$ (assuming CFR)

$G(t) = F(t) \cdot F(t) = (1 - e^{-\lambda t})^2$

$\Rightarrow R(t) = 1 - G(t) = 1 - (1 - e^{-\lambda t})^2$

$= (2e^{-\lambda t} - e^{-2\lambda t})$

$\because \lambda(t) = f(t)/R(t)$ $\left(f(t) = -\frac{dR(t)}{dt} \right)$

$= \frac{\lambda(1 - e^{-\lambda t})}{(1 - 0.5e^{-\lambda t})}$ $\Rightarrow$ not a CFR model

(v) $A(t) = 1 - \left[(1 - A_1(t))(1 - A_2(t)) \cdots (1 - A_n(t)) \right]$

(vi) $\sum_{i=1}^{2} (e_i/p_i) = (90/100)$

$\Rightarrow$ Time available for BG $= 1 - 90/100 = (10/100)$

$\Rightarrow$ BG gets 1msec every 10 msecs.

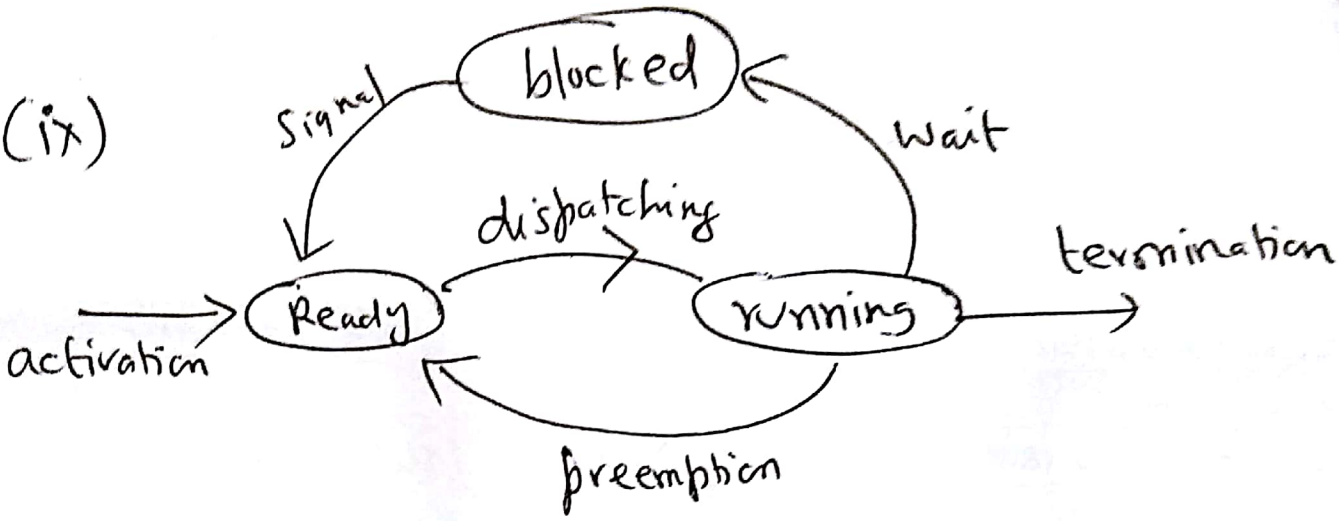
$\Rightarrow$ BG would take $100 * (10/1) = \underline{1000 \text{ msecs}}$ to compute.

(vii) If a user level program is given the ability to disable interrupts, then it can disable the timer interrupt and prevent context switching from taking place, thereby allowing it to use the CPU without letting other processes to execute.

(viii) (a) FCFS - discriminates against short jobs since any short job arriving after long jobs will have longer waiting times

(b) RR - treats all jobs equally so short jobs will be able to leave the system faster since they will finish first.

(ix)



(x)

- State space needed is smaller
- Lower implementation costs
- Less overhead at run-time
- Less cache-pollution, small mem needed