Operating System

1) Turn around Time (A) = Finish Time (A) − Arrival Time (A) = 3.000 − 0.000 = 3.000

Turn around Time (B) = Finish Time (B) − Arrival Time (B) = 9.000 − 1.001 = 7.999

Turn around Time (C) = Finish Time (C) − Arrival Time (C) = 13.000 − 4.001 = 8.999

Turn around Time (D) = Finish Time (D) − Arrival Time (D) = 15.000 − 6.001 = 8.999

Average Turn around Time (FCFS) = (3.0001 + 7.999 + 8.999 + 8.999)/4 = 7.24905

waiting time (A) = (3.000 − 3.000) = 0.000

waiting time (B) = 7.999 − 6 = 1.999

waiting time (C) = 8.999 − 4 = 4.999

waiting time (D) = 6.999

Average waiting Time = (1.999 + 4.999 + 6.999)/4 = 3.49925

Grant chart for FCFS

|--- A ---|--- B ---|--- C ---|--- D ---|
0.000 3.000 9.000 13.000 15.000

Grant chart for LRTF

|--- B ---|--- C ---|--- D ---|--- A ---|
1.001 7.001 9.001 15.001

Turn around time (B) = 7.001 − 1.001 = 6.000

Turn around time (C) = 9.001 − 4.001 = 5.000

Turn around time (D) = 15.001 − 6.001 = 9.000

Turn around time (A) = 15.001 − 0.000 = 15.001

Turn around time for LRTF = (6.000 + 5.000 + 9.000 + 15.001)/4 = 8.75025

all waiting time for LRTF is 0. so average waiting time is 0.

Grant chart for RR (time quantum = 2)

|---- A ----|---- B ----|---- C ----|---- D ----|---- A ----|-- B --|-- B
0.000 2.000 4.000 6.000 8.000 10.000 12.000

---|--- D ---|---- B ----|---- A ----|
14.000 15.000 16.000

waiting time (A) = 16.000 − 3 = 13.000
waiting time (B) = 10.999 − 6 = 4.999
waiting time (C) = 1.999
waiting time (D) = 5.999

∴ Average waiting time for RR = 25.997/4 = 6.49925

Turn around time (A) = 13.000
Turn around time (B) = 10.999
Turn around time (C) = 5.999
Turnaround time (D) = 7.999

∴ Average Turn around time RR = 40.997/4 = 10.24925

2) a) Difference between short-term & Long-term scheduler

Short-term	Long-term
i) It is cpu schedular.	i) It is a Job scheduler.
ii) It selects processes from ready queue which are ready to execute & allocates cpu to one of them.	ii) It selects processes from Job pool & loads them into memory for execution.
iii) Speed is fast.	iii) Speed is less than short-term schedular.
iv) Access ready queue & cpu.	iv) Access job pool & ready queue.
v) It executes frequently. It executes when cpu is available for allocation.	v) It executes much less frequently. It executes when memory has space to accomodate new progress.

b) Difference between synchronous & Asynchronous Multi-processing.

Synchronous	Asychronous
i) Sequential exn − In synchronous multiprocessing task or processes are executed one after other in pre-determined order.	i) Parallel exn − In Asychronous Multi-processing tasks run independently on concurrently.
ii) Blocking in nature.	ii) non-blocking.
iii) exn order is predictable & easy to understood.	iii) exn order of task if not preditermind & may very each time program runs.
iv) Straight forward implement.	iv) well situated for tasks that involve I/O operation.

c) A counting semaphore was initialized to S=12. Then 10P (wait) and 4V (signal) operations were computed on this semaphore. What will be the final value of S?

=> Initial counting semaphore 12

wait operation = 12P

signal " = 4V

Final value F = 12 + 10×P + 4×V

= 12 + 10×(-1) + 4(+1)

= 12 - 10 + 4 = 6 Ans

d) Match the following.

A. NUMA → Multicore processor (5)

B. Aging → Priority scheduling (3)

C. Context switching overhead → Round Robin (2)

D. Convoy Effect → FCFS (1)

E. Mutual Exclusion → Peterson's solution (4)

e) Suppose a computing unit consists with 50% Parallel and 50% serial components. Also let this contains 8 processing core. What will be the max speed-up?

=> The parallelizable 50 units happen in 50/8 = 6.25 unit of time.

total duration = (50 + 6.25) = 56.25

speed up = (unparallelizable total runtime) / (Partially parallelized total runtime)

$= \frac{100}{56.25} = 1.778$ Ans

3. a. i) True

ii) False

iii) False

iv) True.

b. Fill in the blanks.

b. i) Execution Time

ii) Preemption

iii) NUMA

iv) Thread control Block

c) suppose $n = \lambda * w$ (where, λ is the average no. of processors arrive into the ready queue; and w is the avg waiting time) what is n? In which situation the system remains in stedy state? when the system may become unsteady?

=> Little law states that $L = \lambda * w$

n is the average number of process in ready queue.
A steady state situation occurs when the system is in balance, meaning that the rate of which process arrive into the ready queue is equal to the rate of process are being executed & removed from the ready queue.
In an unsteady state n will be fluctuate & the system may experience backlogs.

d) If func 1 execute first -

A becomes $10-5 = 5$
C " $5*5 = 25$

If func2 execute first -

A becomes $5*10 = 50$
C " $50-5 = 45$

If func2 execute after func1 -

A becomes $5*10 = 50$
B " $50-5 = 45$

If func1 executes after func2 -

A becomes $45-5 = 40$
C " $5*40 = 200$

∴ The sum of these distinct value is $= 25+45+200 = 270$