SET A
International Institute of Information Technology, Hyderabad
(Deemed to be University)
CS3.301 Operating Systems and Networks – Monsoon 2025
Quiz 1
Answer key

Ans 1
[1.5 points for correct answer]
(d) AI Tutor: 150, LangOS Main: 75 (each process maintains independent values)

[1.5 points for explanation]
Reason: After fork(), both processes have independent copies of variables. The child updates its own copy to 150, while the parent updates its own copy to 75. No sharing occurs.

Ans 2
[1.5 points for correct answer]
When a process running in kernel mode is preempted, the current “kernel stack pointer value” is saved into the process’s “PCB (process control block)”

[1.5 points for explanation]
Why: When preempted in kernel mode, the process is using its kernel stack. The OS must capture the CPU context—crucially the kernel SP (stack pointer) value—and store it in the PCB, so that on rescheduling the kernel can restore the registers and resume exactly where it left off. (The user stack isn’t what’s active here; the kernel stack is.)

Ans 3

Part A *[3 points]*

[1 points for shortest job first]

SJF (Gantt: P1[0–3] → P3[3–9] → P4[9–11] → P2[11–23])

- P1: Turnaround = 3−0 = 3; Response = 0−0 = 0; Waiting = 3−3 = 0
- P2: Turnaround = 23−1 = 22; Response = 11−1 = 10; Waiting = 22−12 = 10

- P3: Turnaround = 9−3 = 6; Response = 3−3 = 0; Waiting = 6−6 = 0
- P4: Turnaround = 11−4 = 7; Response = 9−4 = 5; Waiting = 7−2 = 5

Averages (SJF):
- Avg Turnaround = (3+22+6+7)/4 = 9.5
- Avg Response = (0+10+0+5)/4 = 3.75
- Avg Waiting = (0+10+0+5)/4 = 3.75

[1 point for short time completion first]

STCF (Gantt: P1[0–3] → P3[3–4] → P4[4–6] → P3[6–11] → P2[11–23])

- P1: Turnaround = 3−0 = 3; Response = 0−0 = 0; Waiting = 3−3 = 0
- P2: Turnaround = 23−1 = 22; Response = 11−1 = 10; Waiting = 22−12 = 10
- P3: Turnaround = 11−3 = 8; Response = 3−3 = 0; Waiting = 8−6 = 2
- P4: Turnaround = 6−4 = 2; Response = 4−4 = 0; Waiting = 2−2 = 0

Averages (STCF):
- Avg Turnaround = (3+22+8+2)/4 = 8.75
- Avg Response = (0+10+0+0)/4 = 2.5
- Avg Waiting = (0+10+2+0)/4 = 3.0

[1 point for round robin]

RR (q = 4, Gantt: P1[0–3] → P2[3–7] → P3[7–11] → P4[11–13] → P2[13–17] → P3[17–19] → P2[19–23])

- P1: Turnaround = 3−0 = 3; Response = 0−0 = 0; Waiting = 3−3 = 0
- P2: Turnaround = 23−1 = 22; Response = 3−1 = 2; Waiting = 22−12 = 10
- P3: Turnaround = 19−3 = 16; Response = 7−3 = 4; Waiting = 16−6 = 10
- P4: Turnaround = 13−4 = 9; Response = 11−4 = 7; Waiting = 9−2 = 7

Averages (RR):
- Avg Turnaround = (3+22+16+9)/4 = 12.5

- Avg Response = (0+2+4+7)/4 = 3.25
- Avg Waiting = (0+10+10+7)/4 = 6.75

Part b *[2 point]*

- Turnaround time (lower is better): STCF (avg 8.75) — smallest average.
- Response time (lower is better): STCF (avg 2.5) — best responsiveness for short/interactive jobs.
- Waiting time (lower is better): STCF (avg 3.0) — lowest average waiting.

Summary recommendation: STCF (preemptive SJF) performs best across all three metrics for this arrival pattern — it finishes short interactive jobs fastest and minimizes avg turnaround/response/waiting.

Ans 4
[1.5 points for correct answer]
Option (b) — 5ms at top, then 10ms/20ms below, with a moderate boost every 50ms — matches these goals.

[1.5 points for explanation]
- Give very small time slices at the top so interactive jobs get quick turnarounds and can finish before being demoted (reduces latency).
- Increase time slices at lower levels so CPU-bound jobs run longer once demoted (reduces context-switch overhead and improves throughput).
- Use a moderate priority-boost interval (not too frequent to waste CPU on boosting, not too infrequent to starve interactive tasks) so long jobs don't permanently starve short interactive tasks and responsiveness is preserved.

Ans 5
[1.5 points for correct answer]
b. UDP - includes checksum for error detection but has no flow control

[1.5 points for explanation]
Reason:
a. TCP (error detection, flow control, guaranteed delivery): Correct but adds latency via ACKs/retransmissions → not ideal for real-time.
b. UDP (checksum, no flow control): Correct and best for minimal delay, though reliability must be handled at app level.
c. TCP (connection-oriented, retransmission): True but retransmissions cause delay → poor for interactive typing.
d. UDP (auto
correction, flow control): Incorrect — UDP only detects errors, no correction/flow control.
e. TCP (faster due to lightweight nature): Incorrect — TCP is heavier than UDP, so slower for low-latency.

Ans 6

[1.5 points for correct answer]
Answer: (f) TCP

[1.5 points for explanation]
a. FTP: Incorrect — used for file transfer, not email.
b. SMTP: Incorrect here — SMTP is an application-layer protocol for sending mail, not a transport protocol.
c. HTTP: Incorrect at transport layer — its application-layer (though modern email clients may use HTTPS for APIs).
d. RTP: Incorrect — used for real-time audio/video streaming, not email.
e. UDP: Not suitable — no reliability, ordering, or guaranteed delivery → bad for email.
f. TCP: Correct — provides reliable, ordered, connection-oriented delivery, essential for email protocols (SMTP, IMAP, POP3) to function correctly.

BONUS

Question 7 ***(3 points)***

• Q1. Virtual runtime update equation *(1 point)*

– Correct formula: $\Delta vruntime = \Delta t_{exec} \times 1024 / w_{task}$
(1.0 point)

– Partial credit: correct proportional reasoning without full formula
(0.5 points)

• Q2. Effect of nice values *(2 points total)*

– (a) Rate of increase of vruntime *(1 point)*

∗ Low nice (high priority): Higher weight ⇒ slower vruntime increase *(0.5 points)*
∗ High nice (low priority): Lower weight ⇒ faster vruntime increase *(0.5 points)*

– (b) Amount of CPU time received *(1 point)*

∗ Low nice (high priority): Receives more CPU time (*0.5 points)*
∗ High nice (low priority): Receives less CPU time *(0.5 points)*

Total: 3 points

Question 8 ***(2 points)***

• Part (a): Terminal Output (1 point)

– Award *1 point* if the terminal output is exactly:

BEFORE

– Award *0 points* otherwise.

• Part (b): File Content *(1 point)*

– Correct final contents of output.txt:

CHILD
EXEC

PARENT

(1.0 point)

– Partial credit: Identifies redirection into output.txt correctly,
but with missing/misordered lines *(0.25 points)*

Total: 2 points