

# **Grand Quiz Spring 2021**

**Subject Code CS604 lecture 1 to 22** 

Solved by Riz Mughal







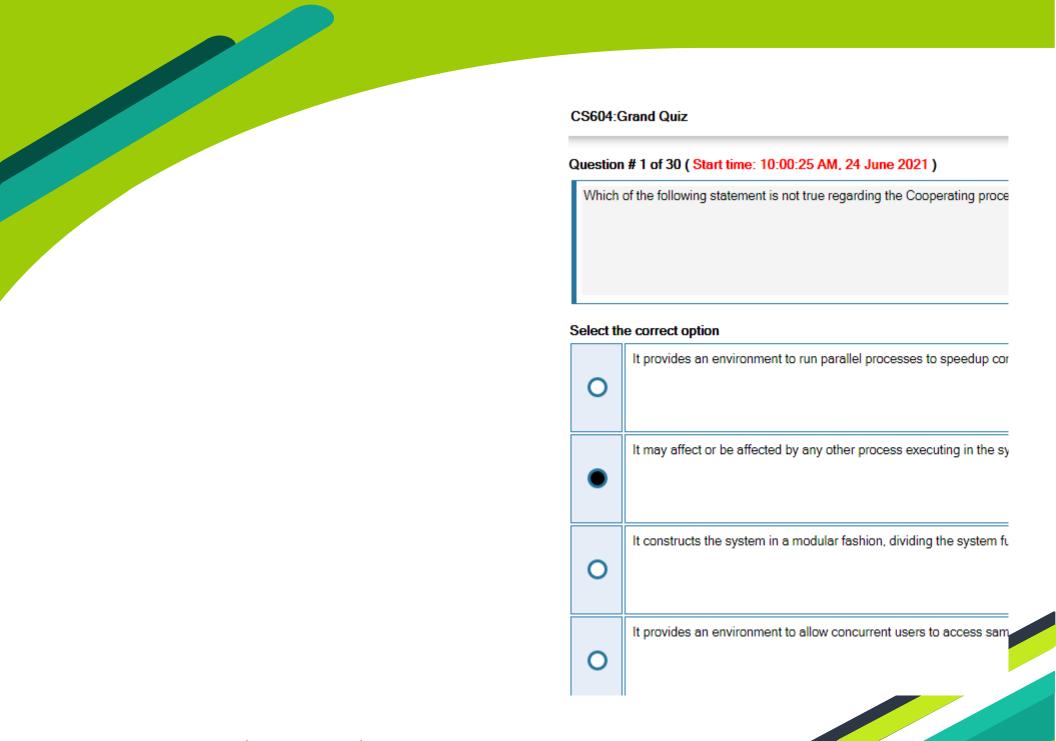
https://www.youtube.com/channel/UCINsFwDiB62SValCcPDZbRQ/playlists

# **Dear Viewers:**

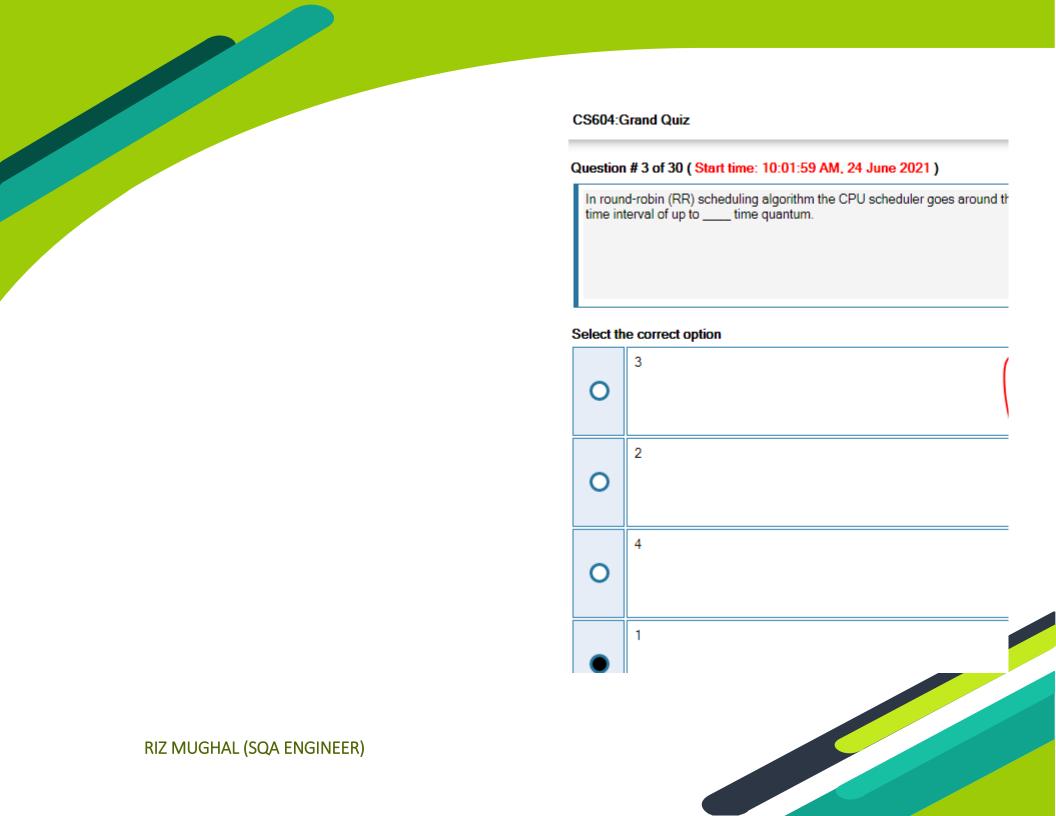
I'm providing 100% correct quiz solution. You can visit my YouTube channel and get more information about all other subjects' quizzes and final year project (CS619).

RIZ MUGHAL (SQA ENGINEER)

Riv Murghall





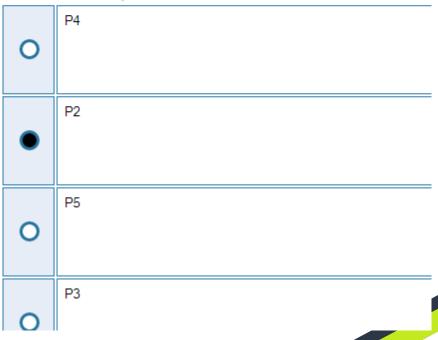






#### Question # 5 of 30 (Start time: 10:02:28 AM, 24 June 2021)

P1, P2, P3, P4, P5 are five processes with ticket numbers P1=1, P2=2, P3 process enters critical section after P1.

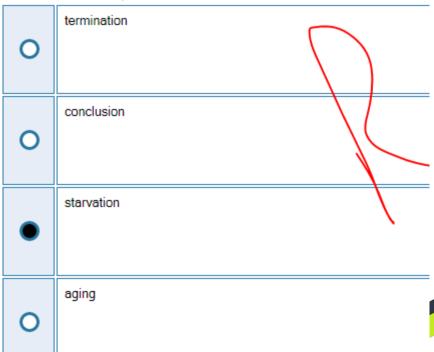


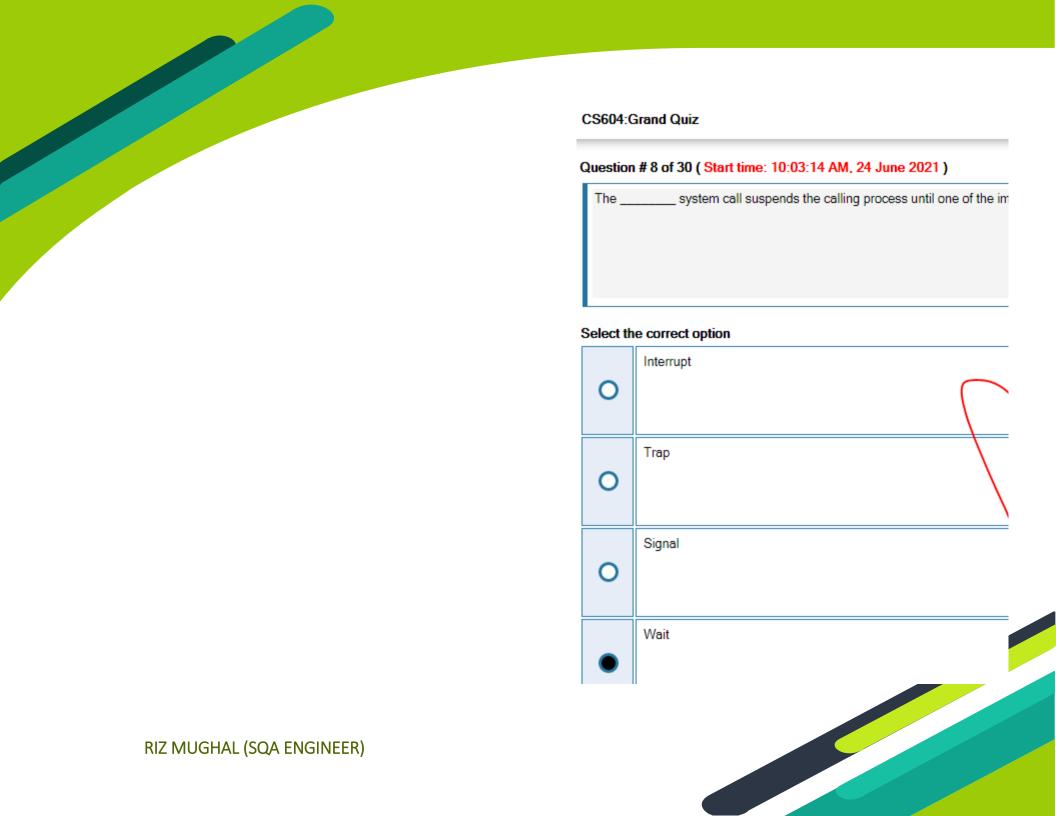




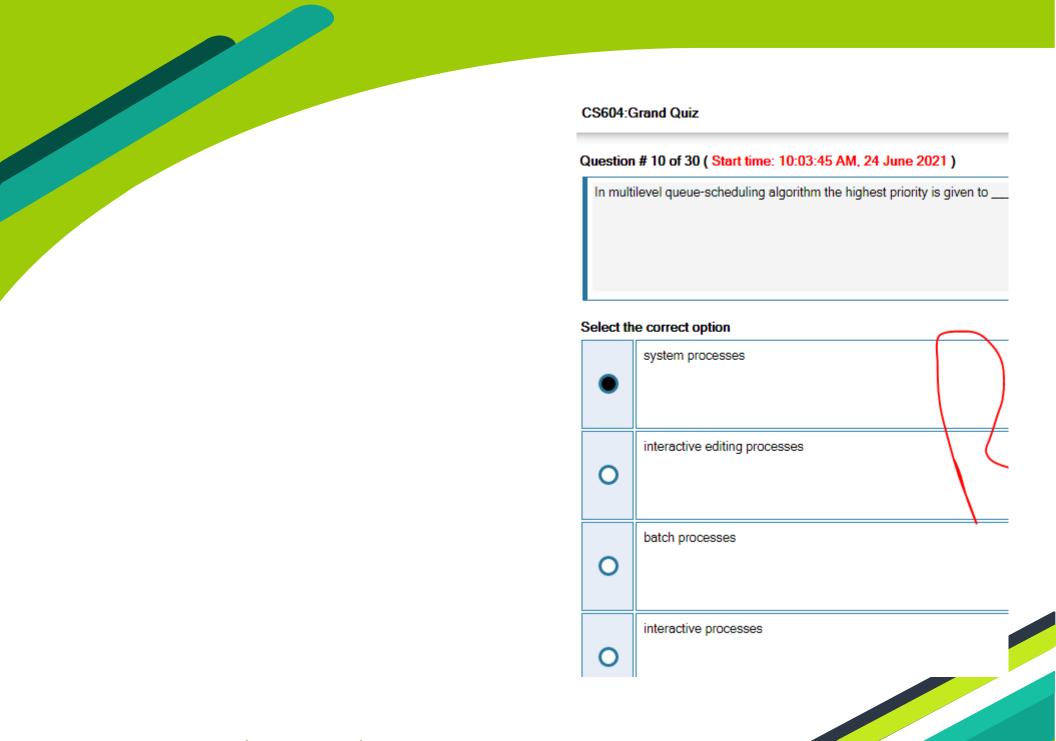
Question # 7 of 30 (Start time: 10:02:58 AM, 24 June 2021)

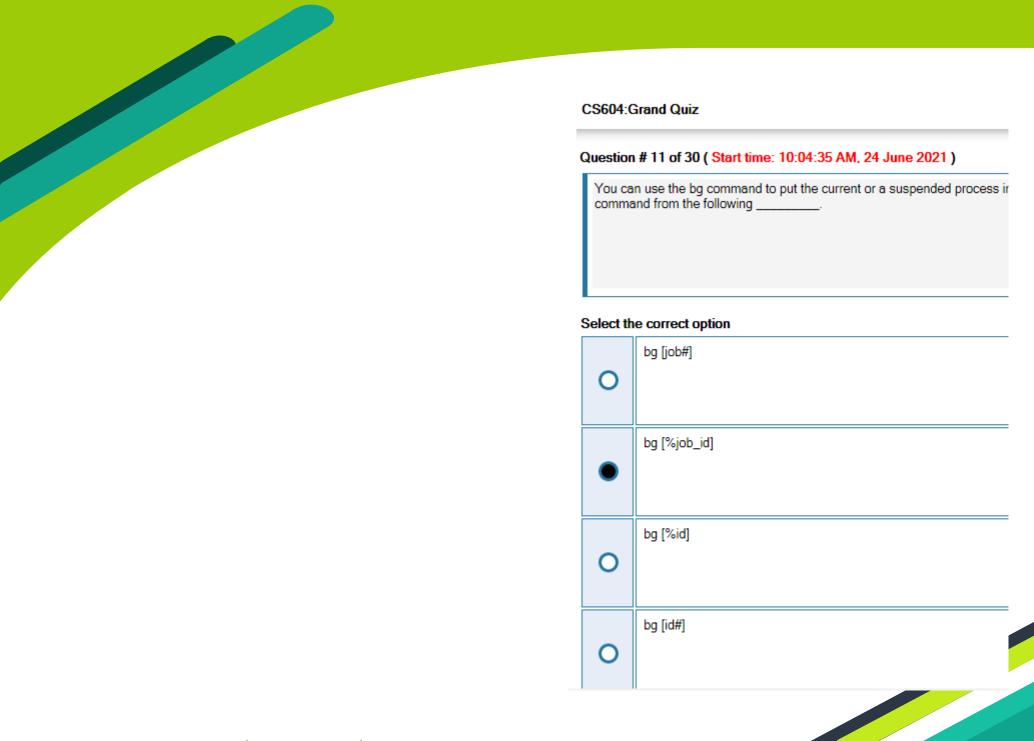
A major problem with priority- scheduling algorithms is \_\_\_\_\_\_.











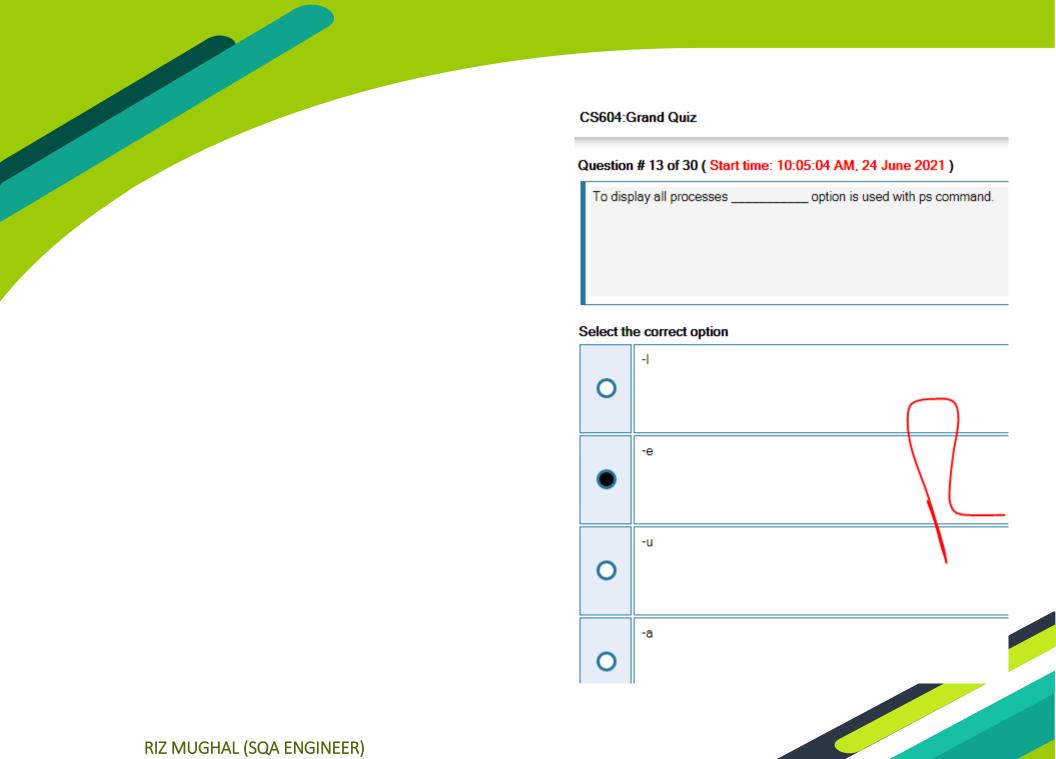


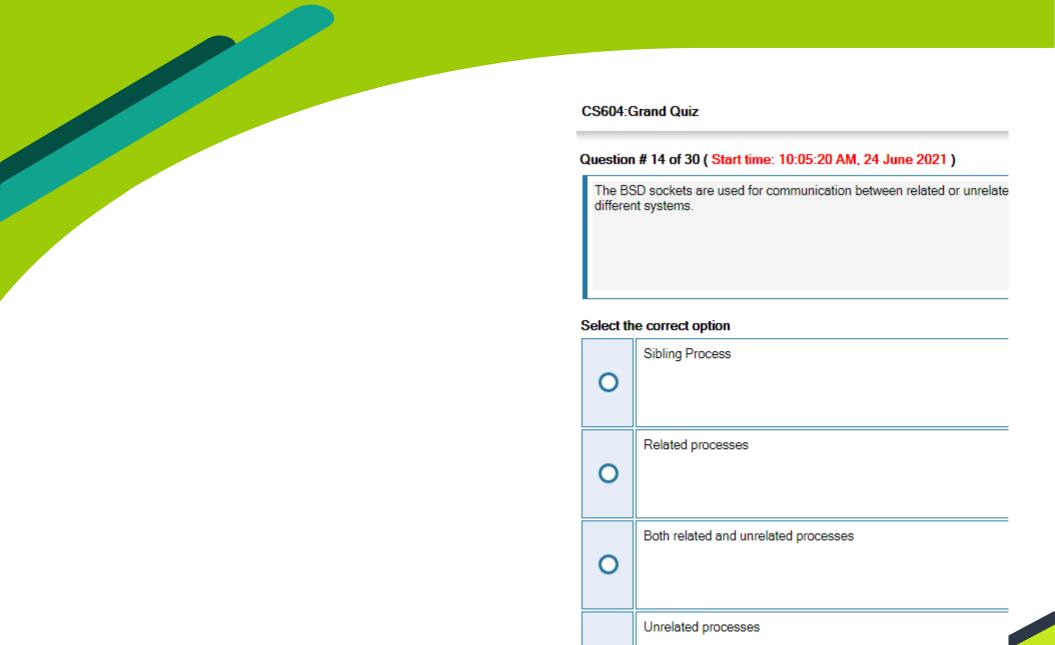
# Question # 12 of 30 ( Start time: 10:04:48 AM, 24 June 2021 )

Given below two statements can be categorized in some sort of message pa

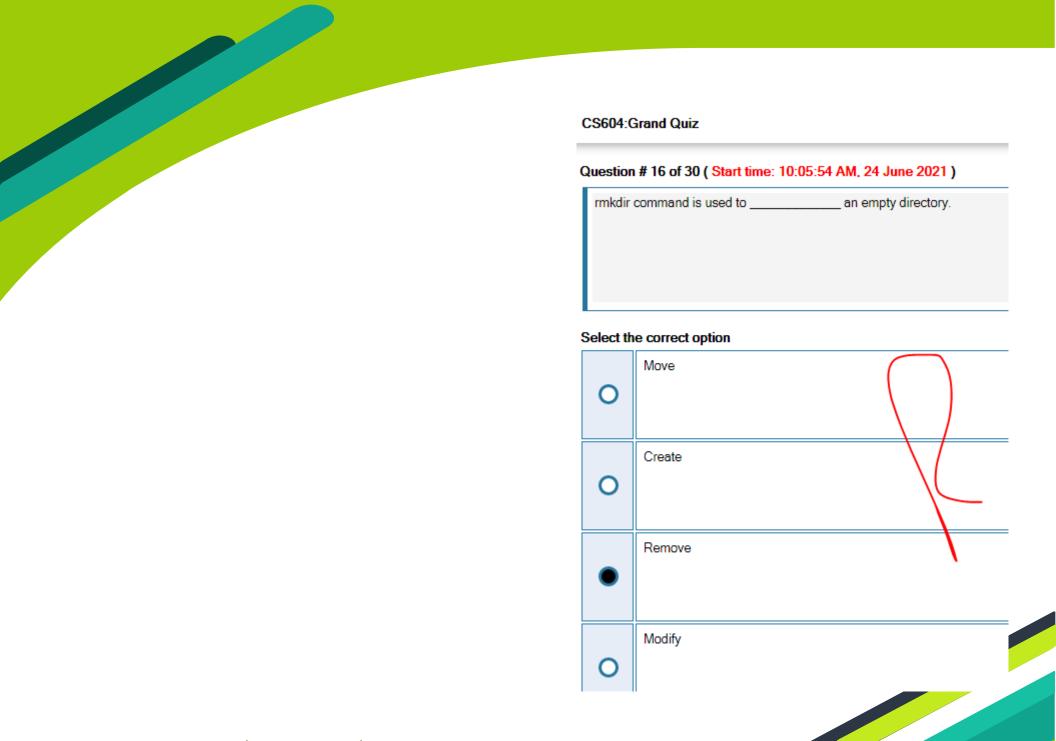
- Send (A, message) Receive (B, message)

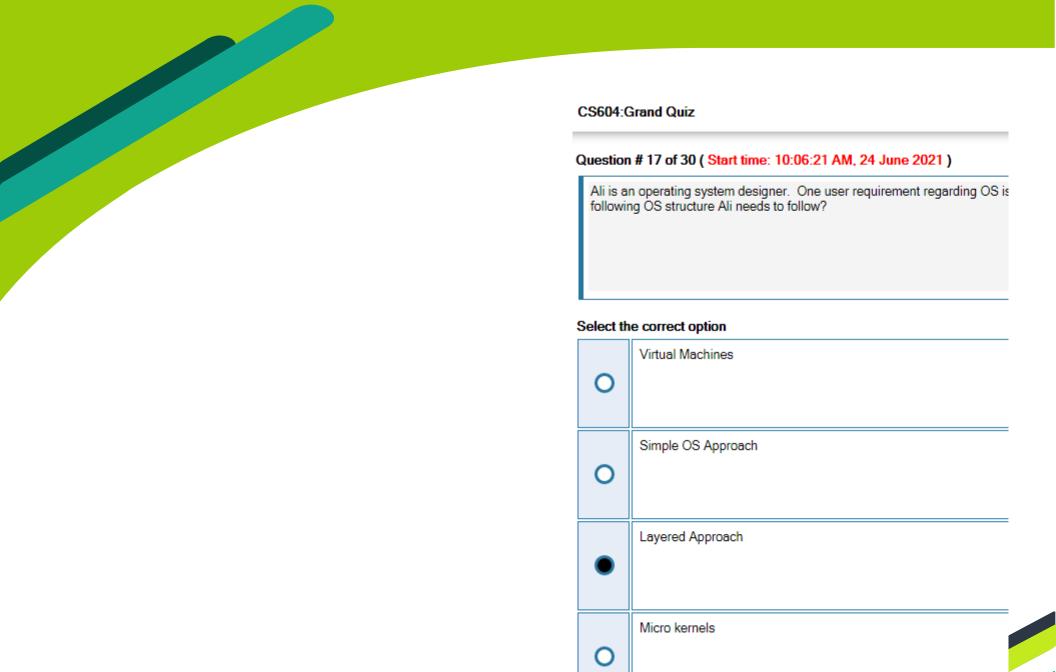
•	Direct communication
0	Send by copy communication
0	Explicit buffering
0	Synchronization

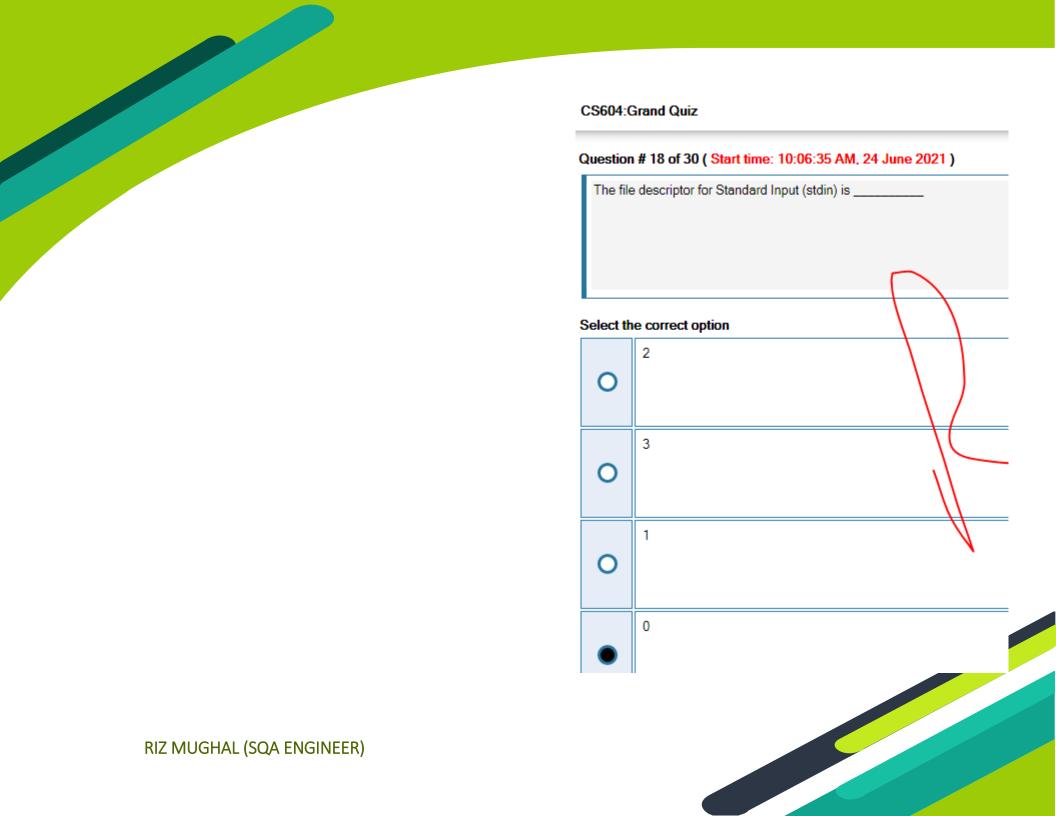














# Question # 19 of 30 ( Start time: 10:06:50 AM, 24 June 2021 )

UNIX System V Scheduling Algorithm in every second, the priority number ready to run is updated by using the following formula:

#### Select the correct option

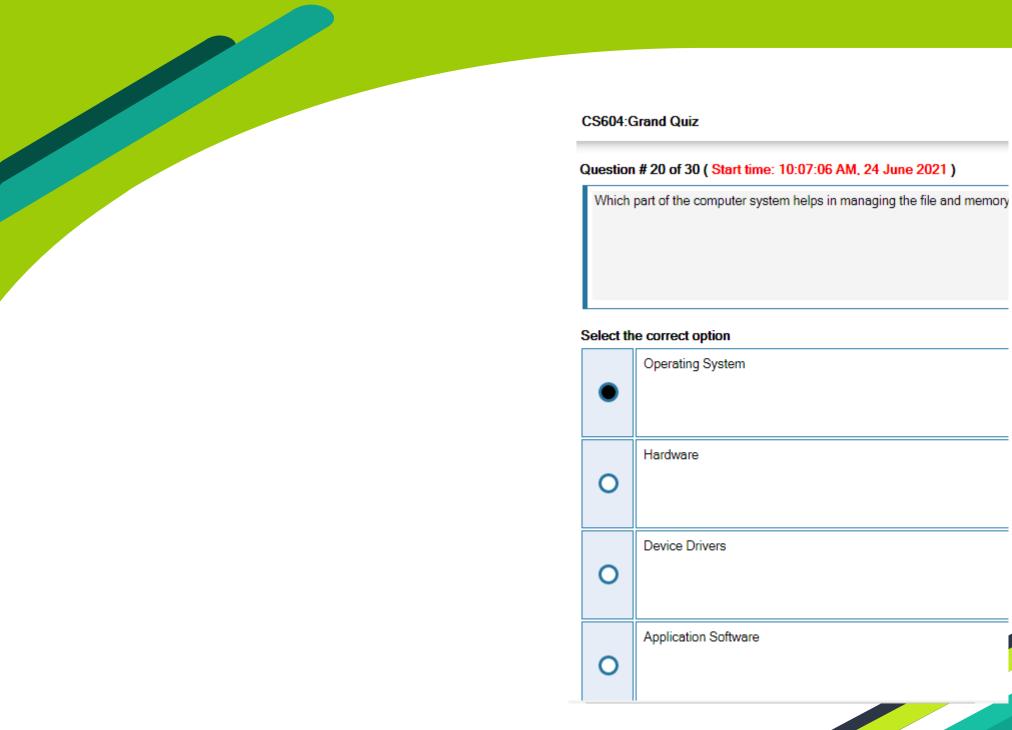
Priority # = (Recent CPU Usage)/2 + Threshold Priority + nice

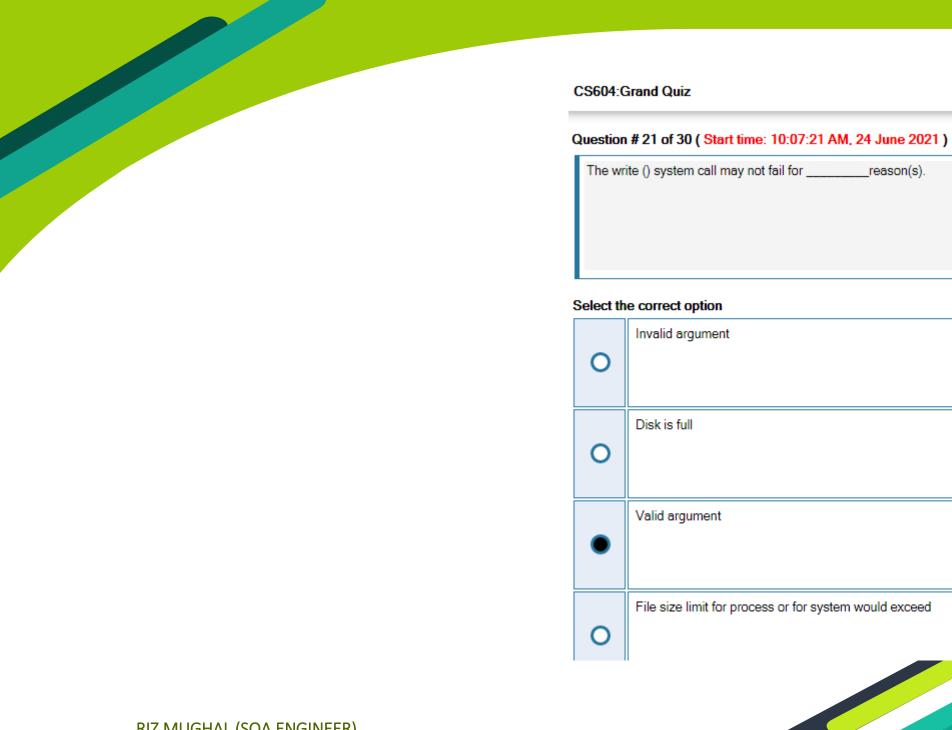
Priority # = (Recent CPU Usage)/2 + nice

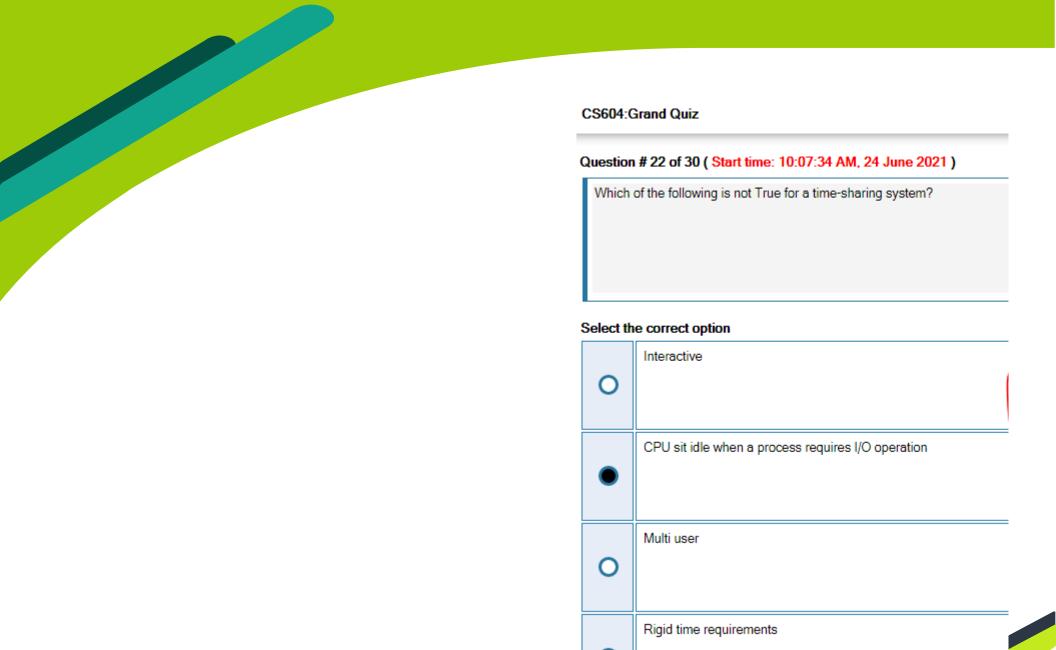
Priority # = Threshold Priority + nice

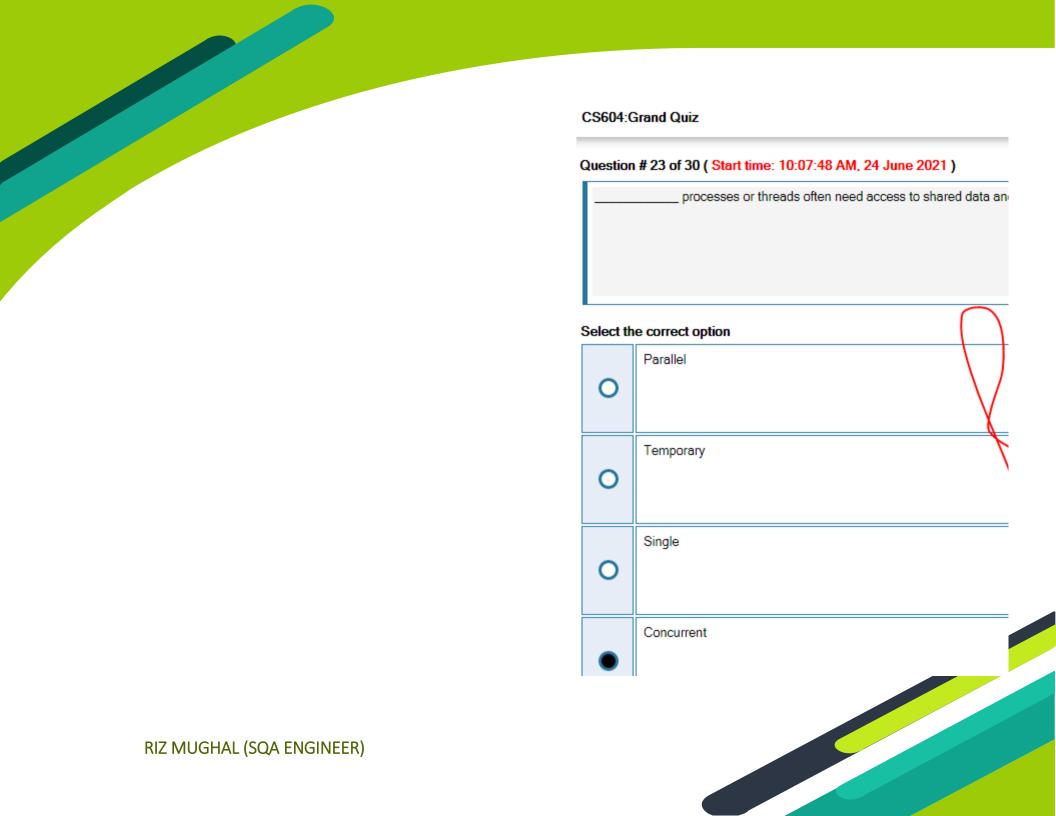
Priority # = (Recent CPU Usage)/2 + Threshold Priority

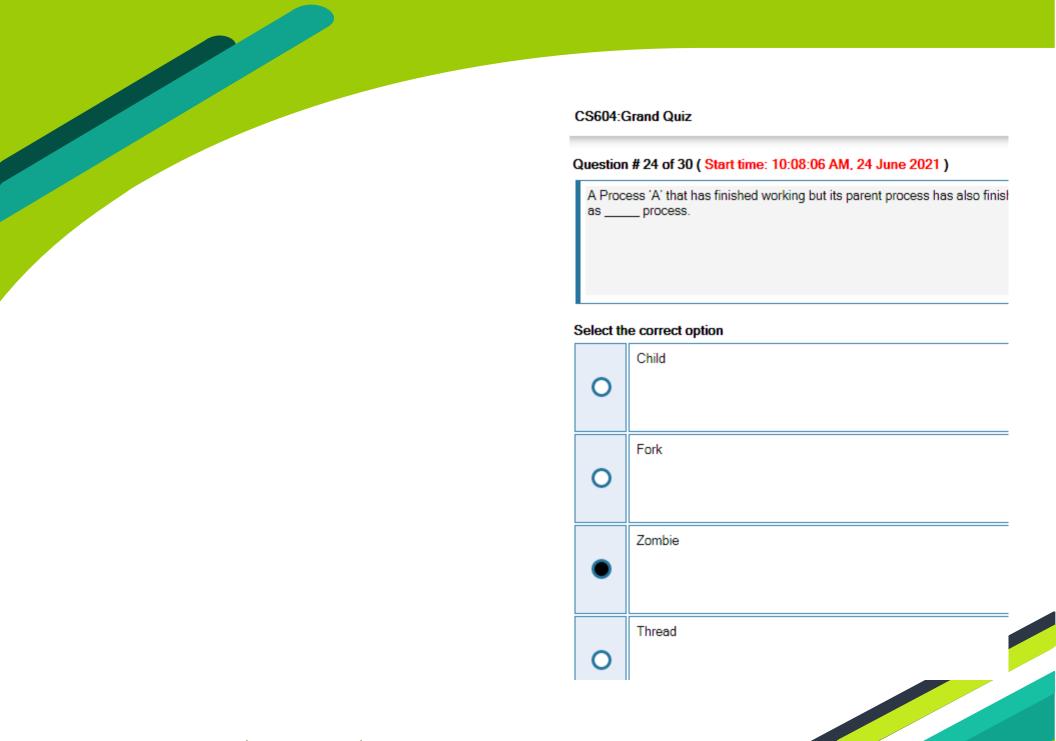
Priority # = (Recent CPU Usage)/2 + Threshold Priority













# Question # 25 of 30 ( Start time: 10:08:24 AM, 24 June 2021 )

Four processes P1, P2, P3, P4 enter the ready queue at time 0 with burst ti











Question # 29 of 30 ( Start time: 10:09:26 AM, 24 June 2021 )

When a process P1 switches from the running state to the waiting state be called------

0	Termination
0	Switching
•	Non preemptive
0	Preemptive

