OS tut

1) What type of multiplexing → in time, space or both can be used for sharing following resources-

a) CPU
b) memory
c) disk
d) printer
e) key board
f) display

also give real life scenarios for space & time multiplexing.

2) Suppose library procedure called READ & the system call also called READ Is it essential that both should have same name if not which one is more important.

3) If you have to choose btw windows or linux. give 5 reasons.

4) Define essential properties of following OS:

1) batch type OS 2) time sharing OS
3) Real time OS 3) distributed OS.

1 Multiplexing → technique for sharing resources efficiently among multiple users.

→ Time division - resource divided into time slots. each user allocated specific time slot.

→ Freq. division - divides resource's frequency into bands & assigns to each device

→ Code division - assigns unique code to each user trasmit simultaneously.

CPU → TDM, Memory → ~~TDM~~ TDM ~~or~~ or SDM (space),
disk → TDM, or SDM , Printers → TDM
keyboard → not common ∵ generally 1 user at a time.
display → TDM, SDM
all process diff use at diff time , >1 tabs at 1 time

Space multiplexing → given diff memory slots

1) cellular networks → to serve multiple users together. diff users allocated seperate, channel freq within that cell.
2) wifi network 3) Satellite communication

Time multiplexing → given diff time slot

1) Cable television → In CATV systems to deliver multiple television channels by assigning diff time slots over a single coaxial cable.
2) Traffic signal 3) Airport security checkpoint.

2

Library procedure	System call
fxn provided by library to read data from file OR input	request by user to OS for reading data from file OR input

No. it is not essential for both to have same name

system call will be translated by compiler into library procedure
∴ library procedure more imp
system call may / may not.

3) Windows →
1) software availability 3) GUI easier to use
2) user-friendly interface 4) Technical support
5) hardware compatibility.

Linux →
1) Open source & license cost 2) easy ~~co~~ customization
3) stability & reliability 4) security
5) community & support

4) Batch OS →
Similar kind of jobs clubbed together as batches & each batch processed ~~respectively~~ in sequence without user interaction →
1) Non-Interactive 3) efficiency
2) Queue Management

Time sharing OS →
multitasking