Day 2

Task-1

1)

import time

print("Hello World")

time.sleep(300)

2)

run main.py file

run below comands

ps -a

3)

run a python file

run below comands to see process states

ps -a | grep "R" (in Running state)

ps -a | grep "S" (in sleep)

ps -aus (all process with state info)

4)

create process with different sleep timers to see this

Ques-> How does the operating system decide which process gets CPU time? What

role does the process scheduler play in this?

A scheduling method is used to decide which job will be dispatched for execution.

Process schedulers are fundamental components of operating systems responsible for deciding the order in which processes are executed by the CPU.

In simpler terms, they manage how the CPU allocates its time among multiple tasks or processes that are competing for its attention.

Process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process based on a particular strategy.

can use Round Robin, SJF(preemptive, non-preemptive) etc.

Task-2

1)

import threading

import time

def print\_num():

for i in range(10):

print(i)

time.sleep(1)

def print\_alpha():

for i in 'abcdefgh'

print(i)

time.sleep(1)

thread1 = threading.Thread(target=print\_num)

thread2 = threading.Thread(target=print\_alpha)

thread1.start()

thread2.start()

thread1.join()

thread2.join()

2)

use ps to get PID

open terminal and run->

top -d -p $PID

htop -d 1 -p $PID

Ans->   
Multithreading improves the performance of a system, especially on multi-core processors, by allowing tasks to run in parallel.  
Concurrency -> Concurrency refers to the ability of a system to handle multiple tasks at once, though not necessarily simultaneously  
Synchronization -> When multiple threads access shared resources concurrently, synchronization ensures that data consistency is maintained.  
  
Task-3  
Solution- >  
4 conditions for Dead lock->

Mutual Exclusion -> Resources held 1 thread at a time

Hold and Wait -> 1 resource is blocked and waiting for another

No Pre-emption -> No pre-emption allowed in running task.

Circular wait -> A loop must be present in the resource allocation graph

Prevention can happen to avoid at least one of the above conditions.

Task-4->

Solution->  
To improve the efficiency of I/O operations, data is temporarily stored in buffers, which are small regions of memory. Buffering allows data to be read or written in chunks rather than one byte at a time.

Increasing the buffer size allows more I/O interrupts to be handled while other tasks can continue to execute.

Task-5->

Algos->  
1) FCFS -> Simple to execute, inefficient movement of disc

2) SSTF -> efficient, May cause Starvation of faraway locations

3) SCAN -> Moves in one direction the reverses while fulfilling in-between tasks  
 can cause wait to certain requests.

4) LOOK -> just like SCAN but does not move to the end instead moves till last request in the direction thus overhead of finding end request is there

5) C-LOOK -> Does not reach end of disk, overhead of finding end request is there.

Task-6->  
A computer screen shot of a black screen

Description automatically generated  
1) Physical Layer -> Transmits raw bit stream over the physical medium

2) Data Link Layer -> Defines the format of data on the network.

3) Network Layer –> Decides which physical path the data will take

4) Transport Layer -> Transmits data using Transmission protocols including TCP and UDP

5) Session Layer -> Maintains connections and is responsible for controlling ports and sessions

6) Presentation Layer ->Ensures that data is in a usable format and is where data encryption occurs.

7) Application Layer -> Human Computer interaction layer, where applications can access the network services.

Task-7->

TCP-> TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.

UDP-> UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.  
  
Task-8->  
Process creation -> Fork() or exec()

Manged by Kernel.

Ps command to show details like PID.  
to terminate use kill command.

Task-9->  
Linux use tree like structure to organize files  
/ is the root directory  
r (Read)  
w (Write)  
x (Execute)  
can give different permissions to owner, group and others using commands->  
chmod  
chown  
chgrp  
  
Task-10->  
Shell scripts can be used to automate tasks like backup, log rotation etc  
we can write a scheduler or cron job to execute the tasks.  
can be executed for efficiency and without manual intervention.