Chapter 3: CPU Scheduling

Compiled by: Kassawmar M.

March, 2023

Outline

- Introduction to scheduling
- When to schedule
- Scheduling Criteria
- Types of scheduling
- Schedulers and its type
- Categories of scheduling algorithms
- Scheduling algorithm Goals
- Scheduling policies in different environments
- Thread Scheduling

Introduction to CPU scheduling

- To make your day more logical and efficient, you work on a schedule.
- An OS operates in a similar manner by scheduling tasks, improving efficiency, reducing delays and wait times, and managing CPU resources better.
- This activity is called **process scheduling**.
- CPU scheduling is the basis of multi-programmed OS.
- Multi-programmed computer
  - Multiple processes running concurrently.
  - Processes compete for the CPU.
  - If there is a single CPU, a choice has to be made which process to run next.

Cont..

- The aim of process scheduling is to assign processes to be executed by the processor or processors over time in a way that meets the system objectives.
- Fundamentally, scheduling is a matter of managing queues to minimize delays and to optimize performance.
- Scheduling in batch systems was simple.
- But in time sharing- algorithms are becoming more and more complex.

When to schedule

When to Schedule?

- A new process is created
  - Since the parent and child processes are in ready state, decision needs to be made whether to run the parent process or the child process.
- A process exits
  - That process can no longer run (since it no longer exists), so some other process must be chosen from the set of ready processes.
- A process blocks
  - when a process blocks on I/O, on a semaphore, or for some other reason, another process has to be selected to run.
  - The reason for blocking may play a role in the selection of the next process, but the scheduler doesn't have enough information.
- I/O interrupt
  - Scheduler decides to run the newly ready process, continue the interrupted process or run another process in the ready queue.

Scheduling Criteria

- CPU utilization – keep the CPU as busy as possible.
- Throughput – the number of processes that complete their execution per time unit
- Turnaround time – amount of time to execute a particular process (total time spent on the system)
- Waiting time – amount of time a process has been waiting in the ready queue
- Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment).

Types of scheduling

1. Non-preemptive Scheduling

- A scheduling algorithm picks a process to run and then just lets it run until it blocks, terminates or voluntary releases the CPU.
- Even if it runs for hours, it will not be forcibly suspended.
- Once the processor starts its execution, it must finish it before executing the other. It can’t be paused in the middle.
- CPU utilization is less efficient compared to preemptive Scheduling.
- Waiting and response time of the non-preemptive Scheduling method is higher.

Examples: First Come First Serve, Shortest Job First, Priority Scheduling.

Cont…

2. Preemptive Scheduling

- A CPU allocated to the process for limited time.
- Picks a process and lets it run for a maximum of some fixed time.
- A processor can be preempted to execute the different processes in the middle of any current process execution.
- CPU utilization is more efficient compared to Non-Preemptive Scheduling.
- Waiting and response time of preemptive Scheduling is less.

Examples: – Shortest Remaining Time First, Round Robin.

Cont..

Preemptive Scheduling	VS	Non-Preemptive Scheduling
CPU is allocated to a process for limited time.	Resource Allocation	CPU is allocated to a process till it complete burst time or switches to waiting state.
Process can be interrupted in between	Interrupt	Process can not be interrupted in between
CPU utilization is more efficient	CPU utilizatio	CPU utilization is less efficient
Preemptive scheduling is flexible	Flexibility	Non-Preemptive scheduling is rigid

Schedulers

- **Scheduler**
  - Part of operating system which selects the process
  - Uses scheduling algorithm.
- Their main task is to select the jobs to be submitted into the system and to decide which process to run.
- Schedulers are three types
  1. Long Term Scheduler
  2. Short Term Scheduler
  3. Medium Term Scheduler

Long Term Scheduler

- It is also called **job scheduler**.
- Long term scheduler determines which programs are admitted to the system for processing.
- Job scheduler selects processes from the pool of jobs and loads them into main memory (ready queue) for execution.
- Process loads into the main memory for CPU scheduling.

Short Term Scheduler

- It is also called CPU scheduler.
- CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them.

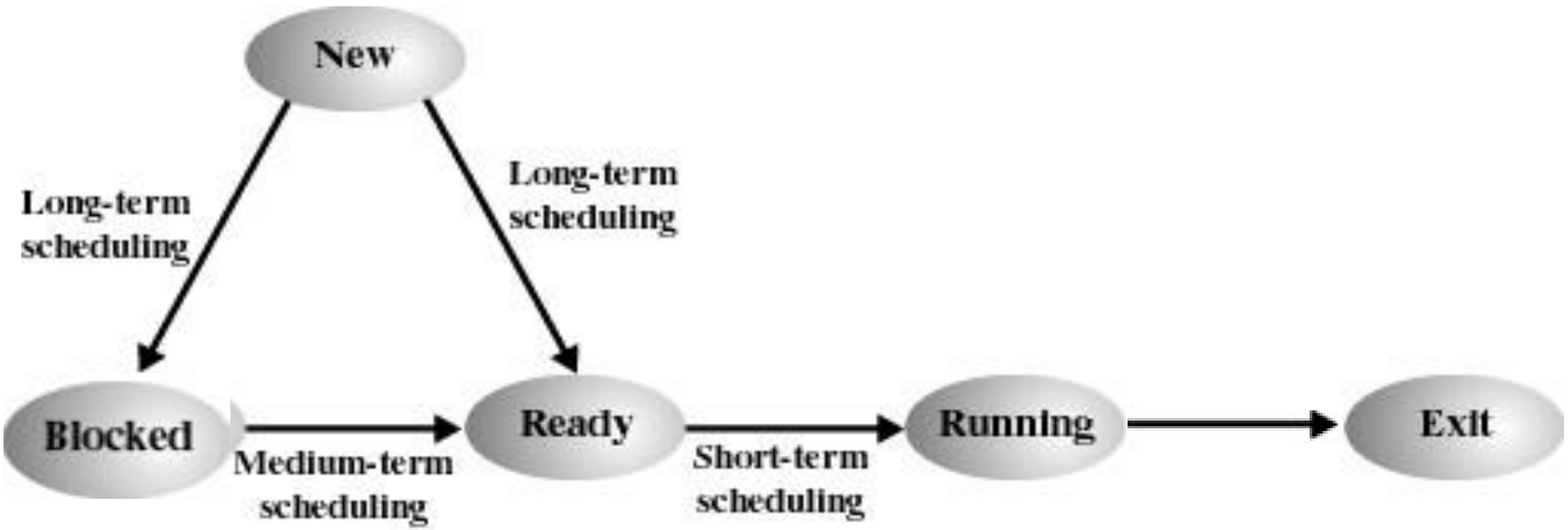
- The method of selecting a process from ready queue is depending on the CPU scheduling algorithm.

- Main objective is increasing system performance in accordance with the chosen set of criteria.
- A **dispatche**r, is module, it connect the CPU to the process selected by the short term scheduler.
- The main function of the **dispatche**r is switching the CPU from one process to another process.

Medium Term Scheduler

- If a process request an I/O in the middle of the execution, then the process removed from the main memory and loaded into waiting queue.
- When the I/O operation completed the job moved from waiting queue to ready queue.
- This two operation performed by the medium term scheduler.
- Medium term scheduling is part of the swapping.

Cont…..



Scheduling and Process State Transitions

Categories Of Scheduling Algorithm

- **Batch-** there is no user waiting for output , it can be preempted or non-preemptive.
- **Interactive** – preemption is needed
- **Real time** – may or may not need preemptive

Scheduling Algorithm Goals

All systems

- Fairness - giving each process a fair share of the CPU
- Policy enforcement - seeing that stated policy is carried out
- Balance - keeping all parts of the system busy. (avoid overload in 1 part)

Batch systems

- Throughput - maximize jobs per hour
- Turnaround time - minimize time between submission and termination
- CPU utilization - keep the CPU busy all the time

Cont….

Interactive systems

- Response time - respond to requests quickly
- Waiting Time (wt): for each process time spent in ready queue.
- Proportionality - meet users' expectations.

Real-time systems

- Meeting deadlines - avoid losing data
- Predictability - avoid quality degradation in multimedia systems

Cont…

- Scheduling of processes is done to finish the work on time. Below are different time with respect to a process.
  - Arrival Time: Time at which the process arrives in the ready queue.
  - Completion Time: Time at which process completes its execution.
  - Burst Time: Time required by a process for CPU execution.
  - TurnAround Time: Time Difference between completion time and arrival time.

    Turn Around Time = Completion Time - ArrivalTime
  - Waiting Time(W.T): Time Difference between turn around time and burst time.

    Waiting Time = Turn Around Time - Burst Time

Scheduling policies in different environments

- Scheduling in Batch Systems:
- Scheduling in Interactive System
- Scheduling in Real Time System

Scheduling in Batch Systems

- There is no users impatiently waiting at their terminals for a quick response,
- Non-preemptive algorithms, or preemptive algorithms with long time periods for each process are often acceptable
- This approach reduces process switches and thus improves performance
  - First Come First Served
  - Shorted Job First
  - Shortest Remaining Time Next

Scheduling algorithms in batch sys

Cont…

1. First Come First Served

- It’s the simplest of all algorithms.
- It is non preemptive and use linked list to keep track the jobs
- Works in FIFO
- It is not optimal.
- When the running process blocks or exits, the first process on the queue is run next.
- When a blocked process becomes ready, like a newly arrived job, it is put on the end of the queue.

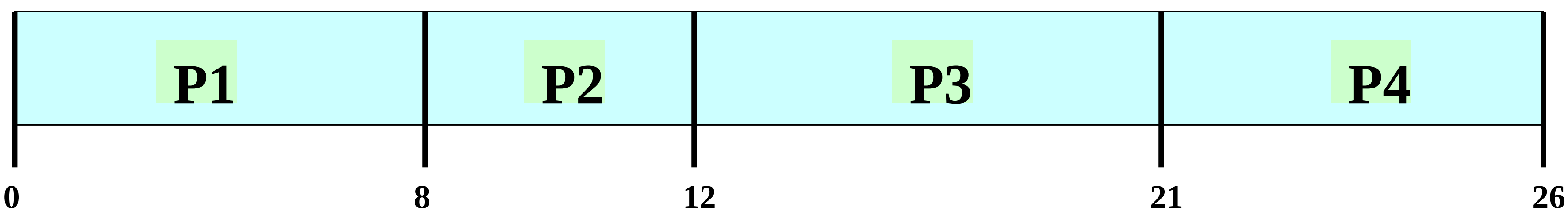
Cont….

Example

EXAMPLE DATA:

Process	Arrival Time	Service Time
1	0	8
2	1	4
3	2	9
4	3	5

The Gantt chart for the schedule is:



Average wait time = ((8-0) + (12-1) + (21-2) + (26-3))/4 = 61/4 = 15.25

Waiting time for $P_1 = 8$; $P_2 = 11$; $P_3 = 19$, $P4=23$

Cont…

2. Short Job First

- This another non-preemptive batch algorithm.
- The scheduler selects the shortest job first
- Shortest job first is only optimal when all the jobs are available simultaneously.
- Shortest job first is provably optimal.

- When multiple batch jobs are sitting in a queue with the same priority, the scheduler runs the shortest job first.

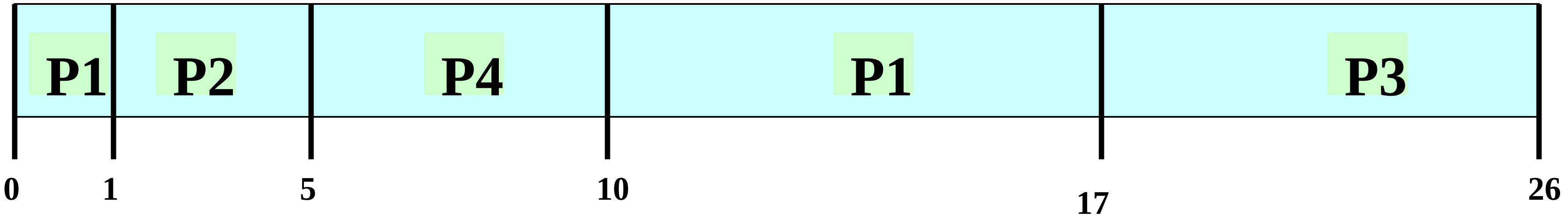
Cont….

Example

EXAMPLE DATA:

Process	Arrival Time	Service Time
1	0	8
2	1	4
3	2	9
4	3	5

The Gantt chart for the schedule is:



Average wait = ((17-0) + (5-1) + (26-2) + (10-3))/4 = 52/4 = 13.0

Cont….

3. Shortest Remaining Time Next

- It is preemptive (it is a preemptive version of the SJF).
- Choose whose remaining run-time is the shortest
- New short jobs get good service

Scheduling in Interactive System

- Preemption is essential to keep one process from hogging the CPU and denying service to the others

  - Priority Scheduling
  - Round-robin scheduling
  - Multiple Queues
  - Guaranteed Scheduling
  - Lottery Scheduling
  - Fair-share Scheduling

Scheduling algorithms in interactive sys

Cont..

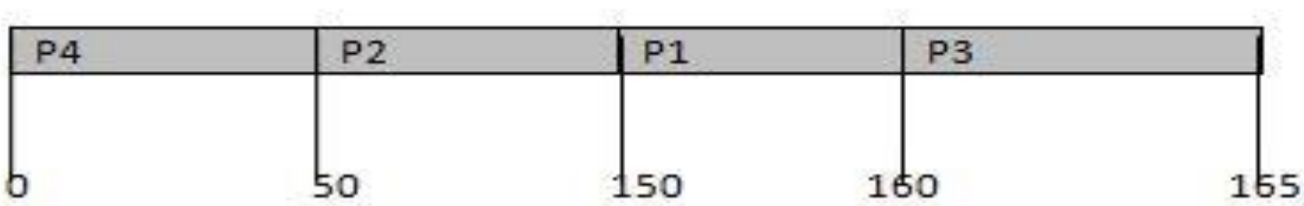
1. Priority scheduling

- Each process is assigned in a priority.
- Schedule highest priority first. All processes within same priority are in FCFS.
- Priority may be determined by user or by some default mechanism.
- The system may determine the priority based on memory requirements, time limits, or other resource usage.
- This idea is critical in a multi-user environment and also in a single user with several applications running at a time.
- **Starvation** occurs if a low priority process never runs. Solution: build aging into a variable priority.

Cont..

Process	Arrival Time	Service Time	Priority	Start Time	Finish Time
p1	0	10	1	150	160
p2	0	100	2	50	150
P3	0	5	1	160	165
P4	0	50	3	0	50

Based on the priority , the order of process show in ff Gantt chart



Average wait = ((160-0) + (150-0) + (165-0) + (50-0))/4 = 131.25

Cont….

2. Round- Robin scheduling

- Processes will be given equal priority
- Each process is assigned a time interval called **quantum**
- Therefore each process will be given an interval of time (quanta of time).
  - ✓1 quantum, 2 quanta…..
- Setting the quantum should have to be done carefully
- **The two main things in setting quantum**

– Setting the quantum too short causes to many process switches that will result in performance degradation.

– But setting it too long may cause poor response to short interactive commands.

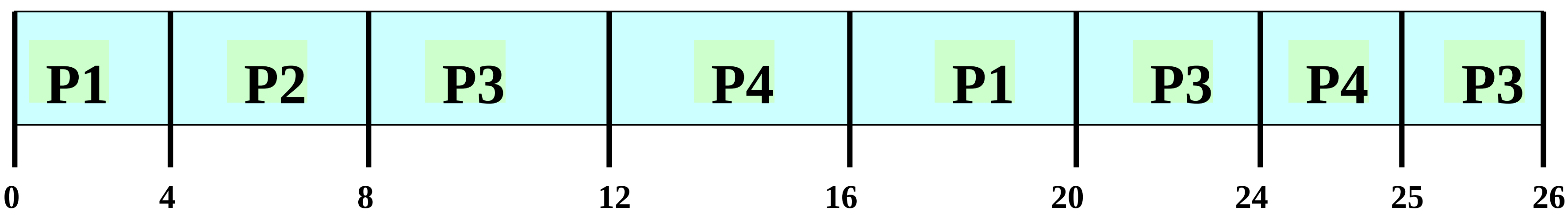
Cont…

Example

EXAMPLE DATA:

Process	Arrival Time	Service Time
1	0	8
2	1	4
3	2	9
4	3	5

Round Robin, quantum = 4, the Gantt chart is



Average wait = ((20-0) + (8-1) + (26-2) + (25-3))/4 = 74/4 = 18.5

Cont…

3. Multiple queues

- It is independent scheduling algorithm
- It groups jobs of similar characteristic together
- Each queue can have its own schedule algorithm
- Priority can be assigned to each queue.

Scheduling in Real Time System

- A **real-time** system is one in which time plays an essential role.
- There are two types of real time system

1. ***Hard real-time systems*** – required to complete a critical task within a guaranteed amount of time.
   - Resource reservation – knows how much time it requires and will be scheduled if it can be guaranteed that amount of time
   - Requires special purpose software running on hardware dedicated to their critical process
2. ***Soft real-time computing*** – requires that critical processes receive priority over less fortunate ones.
   - System must have priority scheduling where real time processes are given the highest priority.

Cont..

Algorithms under real time system

- Rate monotonic scheduling
- Earliest deadline First scheduling

➢ Reading more about the two real time system scheduling algorithm.

Cont…

Rate monotonic Scheduling

- Assumptions:
  - ✓ Each periodic process must complete within its period
  - ✓ No process is dependent on any other process
  - ✓ Each process needs the same amount of CPU time on each burst
  - ✓ Any non-periodic process have no deadline
  - ✓ preemption has no overhead

Cont…

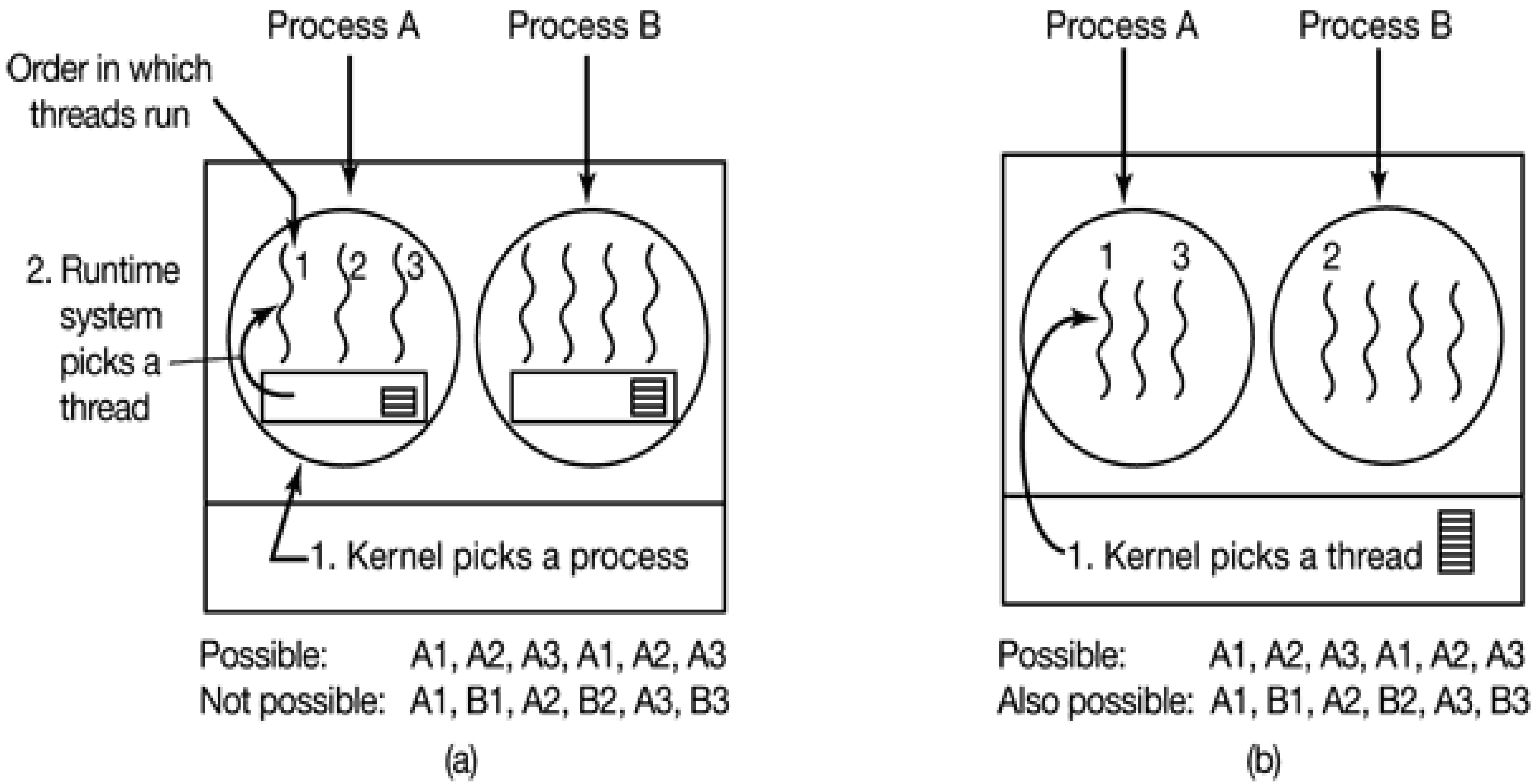
Earliest Job First

- Dynamic scheduling
- Do not need process to be periodic
- Process can use different amount of CPU time
- Scheduler keeps list of runnable process based on their deadline order
- if the process at ready state ,the system checks at deadline and pick the nearest deadline.

Thread Scheduling

1. **User-level threads:**
   - Since the kernel is not aware of the existence of threads, it operates as it always does, picking a process.
   - The runtime system of the process decides which thread to run next.
   - Since there are no clock interrupts to threads, this thread may continue running as long as it wants to.
   - Round-robin scheduling and priority scheduling are most common.
   - The only constraint is the absence of a clock to interrupt a thread that has run too long.
2. **Kernel-level threads**
   - Here the kernel picks a particular thread to run.
   - The executions of thread in the process is scheduled by the kernel rather than runtime system.

Cont…



(a) Possible scheduling of user-level threads with quantum.
(b) Possible scheduling of kernel-level threads

End of Chapter Three

Thanks for you attention!!!