

# CSE3241: Operating System and System Programming

Class-Thread

Sangeeta Biswas, Ph.D.

Assistant Professor

Dept. of Computer Science and Engineering (CSE)

Faculty of Engineering

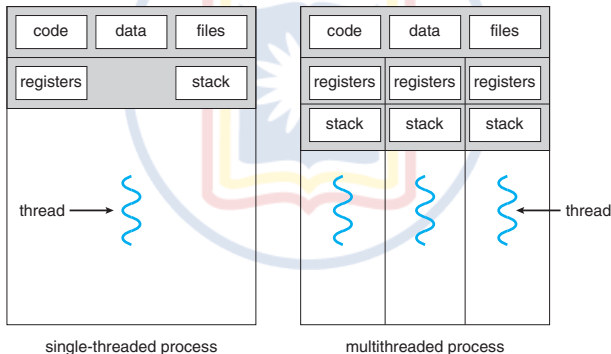
University of Rajshahi (RU)

Rajshahi-6205, Bangladesh

E-mail: [sangeeta.cse@ru.ac.bd](mailto:sangeeta.cse@ru.ac.bd)

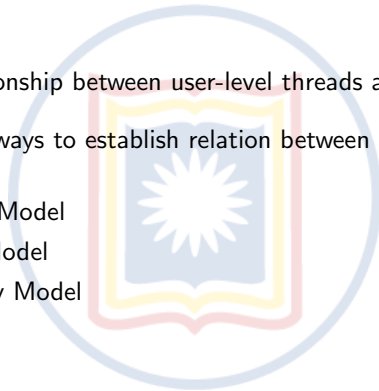
# Single-threaded vs. Multi-threaded Processes

- A thread is a basic unit of programmed instructions that can be managed independently by a CPU scheduler.
- We generally write programs which run as single-threaded processes.
- Threads belong to the same process run concurrently.



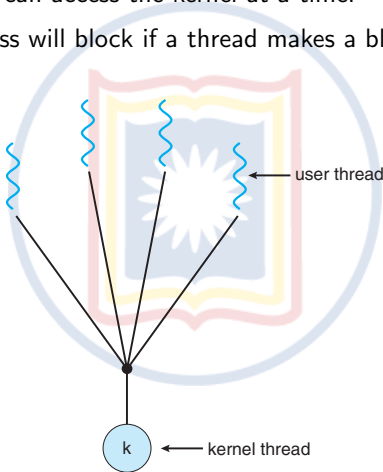
# Multithreading Models

- There is a relationship between user-level threads and kernel threads.
- Three common ways to establish relation between user threads and kernel threads:
  1. Many-to-One Model
  2. One-to-One Model
  3. Many-to-Many Model



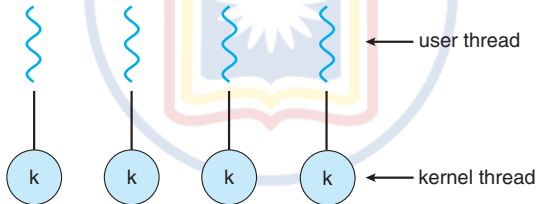
## Many-to-One Model

- Many user-level threads are mapped to one kernel thread.
- Only one thread can access the kernel at a time.
- The entire process will block if a thread makes a blocking system call.



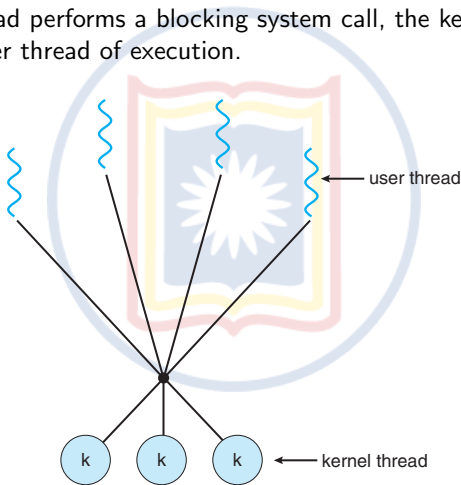
# One-to-One Model

- Each user thread is mapped to a kernel thread.
- Another thread is allowed to run when a thread makes a blocking system call.
- Creating a user thread requires creating the corresponding kernel thread.



## Many-to-Many Model

- Many user-level threads are mapped to a smaller or equal number of kernel threads.
- When a thread performs a blocking system call, the kernel can schedule another thread of execution.



## Two-level Model

- Many user-level threads are mapped to a smaller or equal number of kernel threads.
- A user-level is **also** allowed to be bound to a kernel thread.

