

Systems Software

Assignment 3

Filippo Ghirardini

Freie Universität Berlin
ghira@zedat.fu-berlin.de

Threads
December 1, 2024

1 Individual overview

Mutual Exclusion

Signaling

Synchronization

Coordination

Communication

Cooperation

2 Comparison

Mutual Exclusion

Definition: Ensures that multiple participants do not access critical resources simultaneously to prevent conflicts.

- **Participants:** Processes or threads competing for shared resources.
- **Relationship:** Contention for exclusive access.
- **Example:** Two threads locking a shared variable for modification.

Definition: The mechanism by which one participant notifies another about an event or a change in state.

- **Participants:** A sender (initiator) and a receiver (listener).
- **Relationship:** One-way interaction; the sender triggers the receiver.
- **Example:** A producer signals a consumer when new data is ready.

Definition: Coordinates actions or events between participants to ensure they occur in a predefined order or timing.

- **Participants:** Processes or threads that depend on each other's progress.
- **Relationship:** Interdependent operations, often with timing constraints.
- **Example:** A barrier where all threads must arrive before proceeding.

Definition: Organizing tasks or resources to achieve a common goal without strict timing or sequencing constraints.

- **Participants:** Multiple entities working together.
- **Relationship:** Shared understanding and planned actions.
- **Example:** Team members dividing tasks in a software project.

Definition: Exchange of data or information between participants to share state or facilitate interaction.

- **Participants:** Sender and receiver; can be symmetric or asymmetric.
- **Relationship:** Information transfer, which may involve protocols or direct messaging.
- **Example:** Two processes exchanging messages via a message queue.

Definition: Collaborative effort by multiple participants to achieve a shared objective, typically requiring communication, coordination, or both.

- **Participants:** Independent entities with aligned goals.
- **Relationship:** Collaborative and interdependent.
- **Example:** Two microservices working together in a distributed system.

Comparison Table

Concept	Focus	Participants	Relationship
Mutual Exclusion	Resource control	Competing processes	Contention
Signaling	Event notification	Sender, receiver	One-way trigger
Synchronization	Timing/order	Dependent entities	Sequential actions
Coordination	Task planning	Collaborative entities	Shared understanding
Communication	Data exchange	Sender, receiver	Information transfer
Cooperation	Goal achievement	Independent participants	Collaboration

Relationships in Context

- Mutual exclusion ensures safe access (e.g., between threads).
- Synchronization ensures order (e.g., producer-consumer).
- Coordination focuses on task allocation (e.g., distributed tasks).
- Signaling provides event-driven mechanisms (e.g., interrupts).
- Communication enables interaction (e.g., network messages).
- Cooperation integrates efforts toward a common goal (e.g., teams).