



Parallel programming

Robotics and Computer Vision
BPC-PRP

Ing. Jakub Minařík
Brno University of Technology
2025



- Concurrence vs parallel
- Better performance, resource utilization, scalability
- Race conditions
- Deadlocks
- Synchronization issues



T1	T2
-----	-----
1. Read Acct (\$100)	
2.	Read Acct (\$100)
3. Write New Amt (\$90)	
4.	Write New Amt (\$50)
5.	End
6. End	



T1	T2
-----	-----
1. Lock(x)	
2.	Lock(y)
3. Write x=1	
4.	Write y=19
5. Lock(y)	
6. Write y=x+1	
7.	Lock(x)
8.	Write x=y+2
9. Unlock(x)	
10.	Unlock(x)



`std::mutex`

- Better for complex data
- Can cost performance
- Deadlocks
- `std::lock_guard`
- `std::unique_lock`

`std::atomic`

- For simple data
- Only for simple use
- atomic operation - in one step
- Limited by hardware



- `std::thread, std::async`
- **Condition variables** – `std::condition_variable`
- **Barriers** – `std::barrier`
- **Futures and promises** – `std::future, std::promise`
- **Shared locks** - `std::shared_mutex, std::shared_lock`



Jakub Minařík

203294@vut.cz

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation



Robotics and AI Research Group