

# MUTEX

→ MUTual EXclusion

↳ acts as a flag or lock

↳ so only one thread can access a section of a code at a time

↳ a mutex blocks other threads from accessing the code

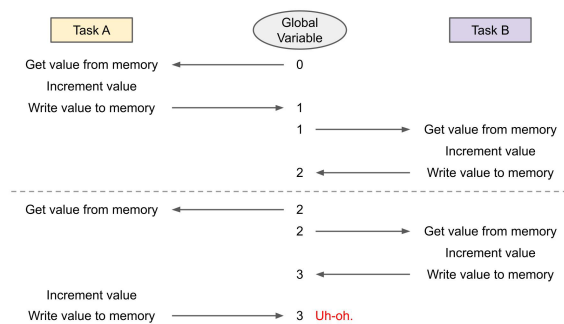
↳ ∴ the executed code is thread-safe & information is not corrupted by other threads

↳ without mutex, a **race condition** might occur

↳ where multiple threads attempt to read & modify a common resource  
eg. global variable & serial port

Applications :

① Task to increment global variable



→ Race condition is created

↳ the timing of sequence of events changes the outcome

use mutex to eliminate the race-condition

↳ allows mutual exclusion of thread execution in a critical section

• In RTOS → mutex is a global binary value that can be accessed atomically

→ if a mutex is not available, the thread is forced to wait

## ② Task with mutex

Task A	Mutex	Global Variable	Task B
Check for and take mutex	1	0	
Get value from memory	0	0	
	0	0	Check for and take mutex
	0	0	Wait/yield
Increment value	0	0	
Write value to memory	0	0	
Give mutex	0	1	
	1	1	
	0	1	Check for and take mutex
	0	1	Get value from memory
	0	1	Increment value
	0	1	Write value to memory
	0	2	Give mutex
	1	2	