

## Звіт

Виконавець: Репетій Юлія Олександрівна, група К-24

Умова задачі:

кількість полів  $m=2$

	Поле 0		Поле 1		
варіант	read	write	read	write	string
№19	5%	40%	5%	5%	45%

У структурі використано два м'ютекси — `mtx0` і `mtx1`, кожен з яких захищає окреме поле. Читання та запис кожного поля виконуються під відповідним м'ютексом, що дозволяє потокам незалежно працювати з різними елементами без взаємного блокування. Така схема забезпечує високу паралельність операцій та відсутність гонок даних. Для формування рядкового представлення обидва м'ютекси блокуються одночасно у фіксованому порядку. Це гарантує консистентність знімка структури і виключає можливість дедлоку. Обрана схема є простою, ефективною та забезпечує безпечний доступ до даних у багатопотоковому середовищі.

Таблиця з усередненими результатами.

Програма запускалася у режимі Release, одиниці виміру часу – секунди. Для трьох потоків було усереднено результати коли частоти відповідають умові, частоти рівні та коли протилежні до умови.

1.48487	1.40661	1.49863
2.68609	2.4435	2.04295
3.11286	3.01985	2.95667

Висновок: як і очікувалося, програма працює швидше тоді, коли багато читань і мало записів, тому що кожне читання блокує лише свій власний м'ютекс і не заважає іншим читанням. Конфліктів майже немає, блокування короткі, тому система працює швидко. Коли ж записів багато і часто викликається `operator string()`, ситуація погіршується. Запис блокує один м'ютекс, а `operator string()` блокує обидва м'ютекси підряд, тому їм доводиться чекати один одного. Через це виникає значне змагання за блокування. Чим більше записів та перетворень у строку — тим повільніше стає робота.

Що реально було зроблено самостійно: розроблено структуру даних, створенно код для генерації файлів та функцію для заміру часу. З використанням `srpreference.com` було побудовано схему мютексів.