

Студент групи К-25 Дубас Дмитро Олександрович

Умова задачі (Варіант 3, m=3):

- Поле 0: read – 10%, write – 10%
- Поле 1: read – 50%, write – 10%
- Поле 2: read – 5%, write – 5%
- Операція string: 10%

Схема захисту даних та її обґрунтування: Використовується 3 `std::shared_mutex` (по одному на кожне поле). `read()` (`get()`) та `string()` використовують `shared_lock`, `write()` (`set()`) – `unique_lock`. Ця схема оптимальна, оскільки сумарні операції читання (75%) значно переважають запис (25%). Це максимізує паралелізм, дозволяючи одночасні читання, тоді як запис блокує лише одне конкретне поле.

Табличка з результатами виконання (мс):

	1 Thread	2 Thread	3 Thread
Variant	115.835	318.557	169.837
Uniform	168.708	415.595	233.272
Skewed	207.719	355.733	231.856

Висновки: Результати в 1 потоці: найшвидший час у Variant 3 (115.835 мс), де схема блокувань є оптимальною, та найгірший у Skewed (207.719 мс). При переході до 3 потоків (169.8 мс для 150 тис. операцій) спостерігається гарне масштабування: час на одну операцію значно знижується (з ~116 мс до ~57 мс на 50 тис. операцій), що підтверджує ефективність `shared_mutex`.

Що реально було зроблено самостійно: Самостійно реалізовано структуру `ProtectedData` з `shared_mutex` на кожне поле, генератор тестових файлів `generateFile` та функцію запуску потоків з вимірюванням часу `runTest`