

Przy pomocy języka C, napisz prostą bazę danych. Baza danych ma spełniać następujące warunki:

- ilość rekordów/pól bazy jest stała, znana podczas procesu kompilacji
- w pamięci stałej jest przechowywana w formie binarnej w pliku
- w pamięci RAM jest przechowywana w strukturze (local\_db)
- jest obiektem w formie biblioteki dynamicznej
- musi wspierać wielowątkowość
- musi być kompilowalna pod Linuxem za pomocą Makefile.

#### local\_db:

Każdy program korzystający z biblioteki posiada własną kopię lokalną bazy, nazywaną się local\_db. Kopia lokalna jest wypełniana zawartością z pliku podczas uruchamiania aplikacji.

#### Operacje na bazie:

odczyt: odczytanie zawartości bazy z pliku do local\_db

zapis: zapisanie zawartości local\_db do pliku

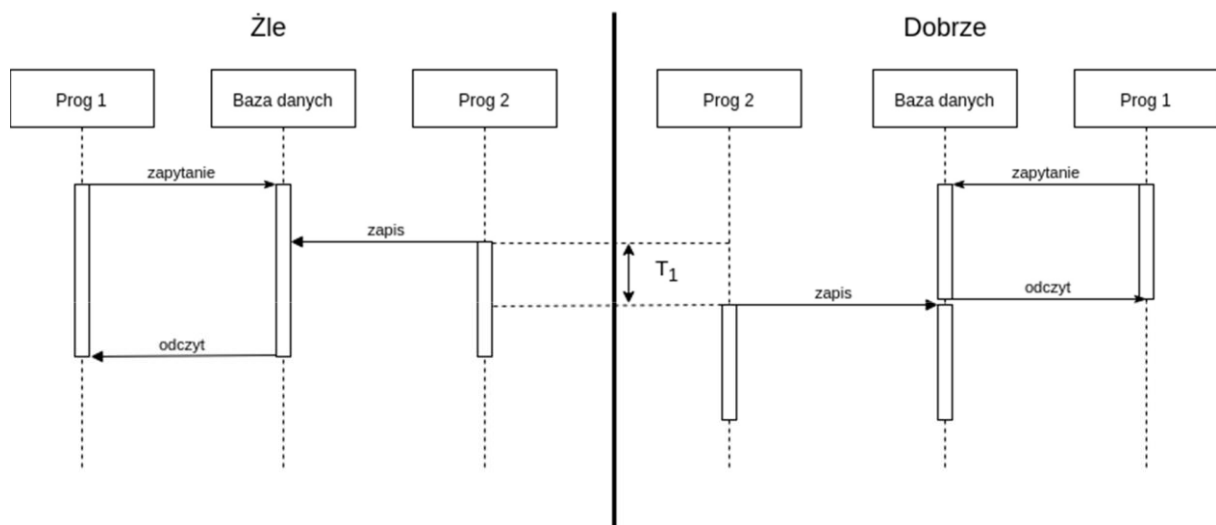
zmiana: zmiana wartości pojedynczego rekordu, zmiany dokonuje się w local\_db

#### Problem:

1. Problem wyścigu - wielowątkowość.

Gdy aplikacja 1. odczytuje zawartość bazy z pliku do local\_db, a w tym samym czasie aplikacja 2. zapisuje bazę do pliku, wówczas zawartość bazy aplikacji czytającej bazę 1. będzie 'mieszanką' wartości sprzed zapisu i po zapisie bazy przez aplikację zapisującą 2.

Zaproponuj mechanizm w bibliotece bazy danych zapobiegający ww. sytuacji.



$T_1$  - czas jaki Prog2 czeka aż operacja odczytu Prog1 się zakończy

## 2. Spójność bazy danych.

Wyobraź sobie następującą sytuację: aplikacja 1. ma zainicjalizowaną swoją bazę local\_db. Następnie aplikacja 2. zapisuje swoją bazę lokalną local\_db, w której jeden z rekordów został zmieniony, do pliku. Powstaje sytuacja, gdzie aplikacja 1. ma nieaktualną wartość rekordu, gdyż rekord ten został zmieniony i zapisany przez aplikację 2.

Zaproponuj rozwiązanie, które będzie informowało wszystkie aplikacje korzystające z bazy danych, o tym, że zaszły zmiany w bazie danych. Dodatkowo, rozwiązanie to powinno informować jakie rekordy zostały zmienione oraz odświeżyć zawartość tych rekordów w lokalnej bazie danych wszystkich aplikacji.

