

Starts a transaction T in P by creating a Journal object j for the running thread

Checks the borrowing rules and returns a mut ref to data

$\left. \begin{array}{l} T \\ \lambda \end{array} \right\} P::\text{transaction}(\bullet | j | \{$
 $\quad \text{let mut data} = \text{data.borrow_mut}(j);$
 $\quad \bullet *data = \text{new_value};$
 $\quad \bullet \}$
 $\bullet)$

Takes an undo log before mutably dereferencing data and persists it using `clflush`

Drops the logs

Commits changes if the transaction is successful and persists them; otherwise, rolls back.

Drops the local mut ref

● Hidden low-level operations