New in 6.d: Thread-Safe, Atomic Operations

We now have a new type—atomicint—and a new set of atomic routines for it. Atomic operations can be safely performed from multiple threads at the same time, without any locking. The new functions are atomic-assign, atomic-fetch, atomic-fetch-inc, atomic-fetch-dec, atomic-fetch-add, atomic-fetch-sub, atomic-inc-fetch, atomic-dec-fetch, and cas.

Unicode lovers rejoice! These are also available as fancy-Unicode operators that use ☢ U+269B ATOM SYMBOL: infix ☢=, prefix ☢+, prefix ++☢, postfix ☢++, prefix --☢, postfix ☢--, infix ☢+=, infix ☢-=, and infix ☢−=

# Regular ops: not thread safe! Wrong result!
my int $total = 0;
await start { for ^20000 { $total++ } } xx 10;
say $total; # OUTPUT: «188758»

# Using atomic operations: thread-safe and correct
my atomicint $atotal = 0;
await start { for ^20000 { $atotal++ } } xx 10;
say $atotal; # OUTPUT: «200000»