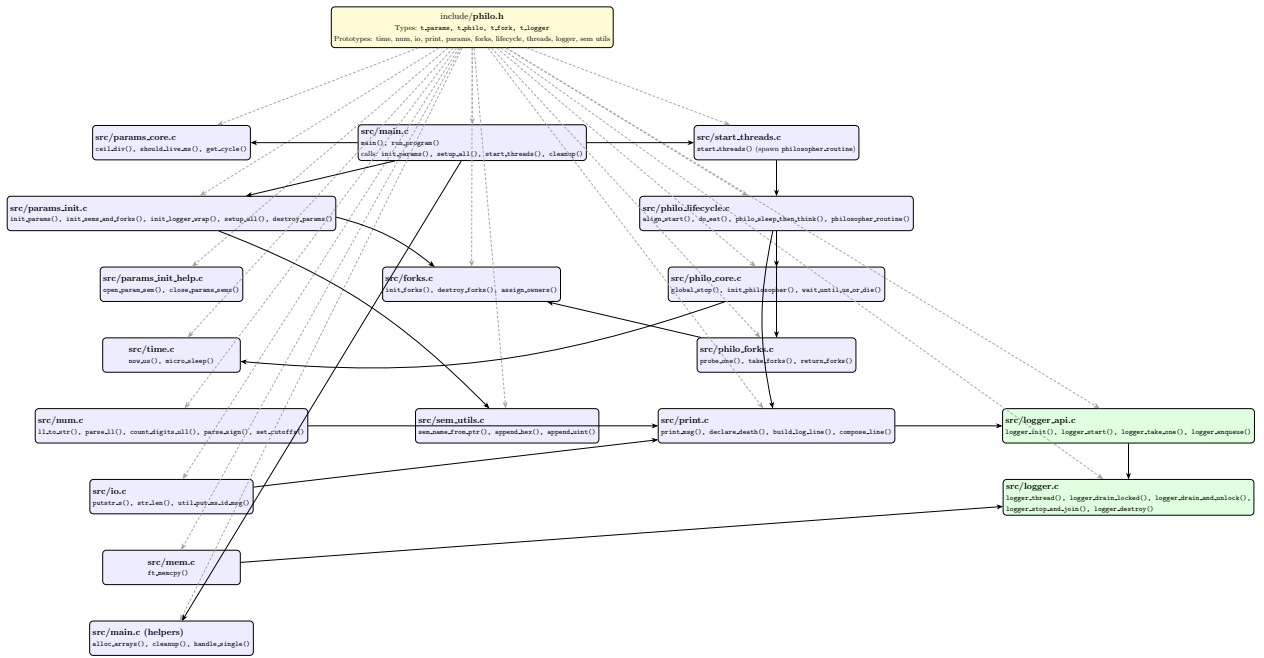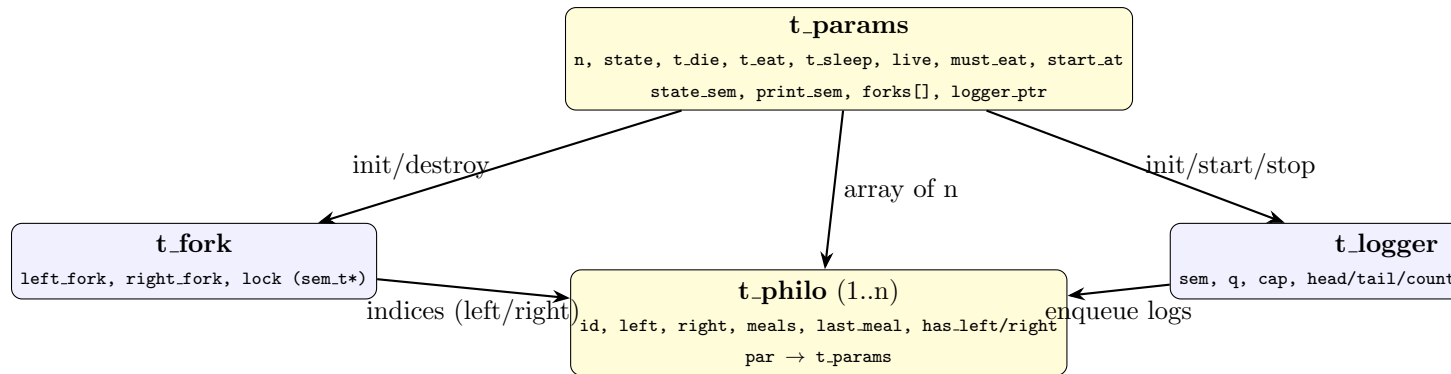# Philosophers (Semaphore Version) — File Structure Flow

Data Structures (from `philo.h`)

1 Project Structure (Updated)

1.1 main.c

Entrypoint: parses args, initializes params/semaphores/forks/logger, starts threads, handles single-philosopher edge case, and runs `cleanup()` on exit.

1.2 params_core.c & params_init.c & params_init_help.c

Timing helpers (`ceil_div()`, `get_cycle()`, `should_live_ms()`), parameter parsing, and all semaphore/fork allocation: `init_params()`, `init_sems_and_forks()`, `init_logger_wrap()`, `setup_all()`, `open_param_sem()`, `close_params_sems()`.

1.3   start_threads.c

Creates philosopher threads and starts the background logger thread.

1.4   philo_lifecycle.c, philo_core.c, philo_forks.c

Implements the life cycle (`philosopher_routine()`), precise waiting with death checks, and fork acquisition/release using semaphores.

1.5   print.c, logger_api.c, logger.c

Builds formatted log lines and enqueues them to a dedicated logger thread that drains the queue and serializes output (`print_sem` respected).

1.6   forks.c

Low-level fork ownership table and (de)initialization for `t_fork` entries.

1.7   time.c, num.c, io.c, mem.c, sem_utils.c

Time utilities in microseconds, number/string helpers (e.g., `ll_to_str()`), minimal I/O functions, memory helpers, and semaphore name/hex utilities.

1.8   include/philo.h

Single header with all types and prototypes (semaphore-based API, logger primitives, timing utilities).