



- provides : thread init, spawn, join, destroy
- signal handler for all threads.
- main(), entry point

signal handler() [requires knowledge of every thread locks (specified in module.h)]

module_init() for each module (specified in module.h)

module_exec() spawns thread for each module

module_cleanup() cleans up after exited modules

exit()

err_printer()

main.h

- To be included by every module [source code]
- extern definition of done → controls loops for all threads
- extern definition of thread local errno custom
- prototype for err_printer()

modules

Wednesday, July 1, 2020 18:46

modules.h

- a) Includes `module_main.h` for each module that defines:
- `int module_init()` function. [optional]
 - `void * module_exec(void *)` [mandatory]
 - `void module_exit()` function [depends on `module_init()`]

b)

- ① Array of `module_init` functions, static declaration
- ② Array of `module_exit` functions, static declaration
- ③ Array of `module_exec` functions, static declaration
- ④ Array of `pthread_cond_t` on which modules might sleep for the signal handlers.

- ① Allocation of all memory for the cross-module variables in this file [static]
- ② Declaration of all routines.

module.h

Wednesday, July 1, 2020 19:18

Main module header, contains all function prototypes.
Not to be included in anything but module.c.

Extern prototypes for

- ① module_init
- ② module_exit
- ③ module_exec
- ④ pthread_cond_t *.

Only to be included by main, in (modules.h).

module - extern.h

Wednesday, July 1, 2020 19:24

Extern definitions for any synchronizers.

Extern definitions for any variables shared between modules.