

Scenario:

Module 3 is preloaded from file.

Module 1 is the "target" module, launched by the main program.

Module 1 requires i/p from Modules 2 and 3.

Since Module 2 is not started, Module 1 launches it
by trying to access its memory.

Key:

module

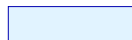
①

thread

module ①

change module state -> **Finished**

state lock



module-specific code



inter-thread dependency



Main program

construct modules

-> **Uninitialised ***

if required, load stored modules

-> **Finished ③**

launch target module(s) ①

if uninitialised ①

launch ①

-> **Running ①**

start thread ①

(else do nowt)

wait for all modules to finish

destroy modules

Module ①

generic thread stuff

access modules ② ③:

if uninitialised ②

launch ②

-> **Running ②**

start thread ②

(else do nowt)

(if uninitialised ③)

else do nowt

loop:

wait for i/p from ②

process i/ps from ② ③

write to memory

-> **Finished ①**

Module ②

generic thread stuff

loop:

get i/p from file

process i/p

write to memory

-> **Finished ②**