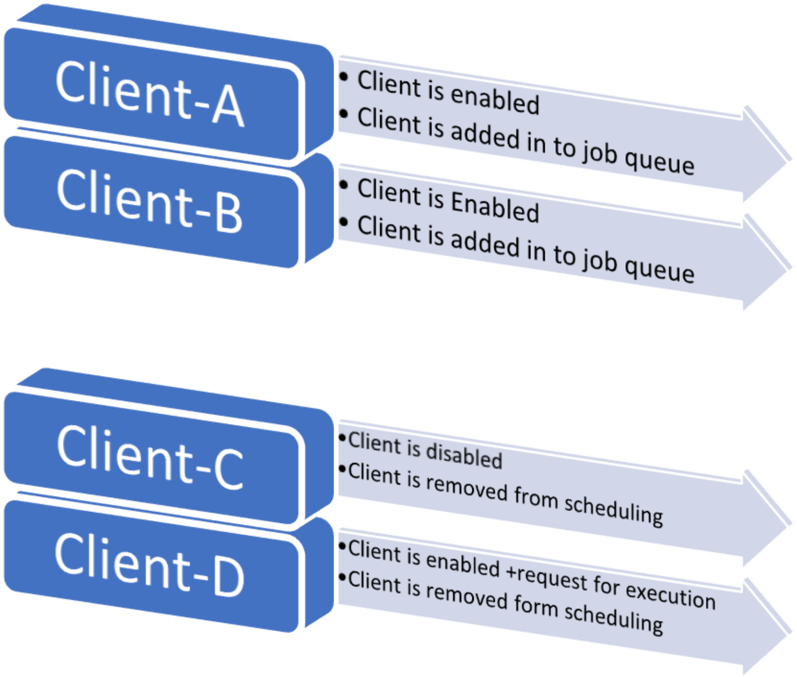
Scheduler-ASWIN

This is a simple program to schedule or share single thread between multiple clients or with multiple functions.
It's a complete generic implementation, with very good OS abstraction. Currently done for Linux and working on windows interface.
You can find great flexibility to add different handlers and scheduling according to your need.

Please feel free to use it and help me to improve the implementation.

Here is a general overview:

To add jobs to the scheduler: client side **aswScheduler_Create()** will return the scheduler context which has to be used across the implementation.
You can register the client handler to schedule using **aswScheduler_RegisterClient()**.
The scheduling of clients can be enabled and disabled using **aswScheduler_EnableClient()/aswScheduler_DisableClient(),** Enabled by default during registration.
You can add the clients in to jobs queue using **aswScheduler_RequestSchedule (),** but it won't trigger the execution.
Where **aswScheduler_RequestRun()**, will add the request to job queue and trigger explicit execution of the scheduler task context.
There is **aswScheduler_LogLevel()** to modify the log level for debug information, default error level.



Client-A

- Client is enabled
- Client is added in to job queue

Client-B

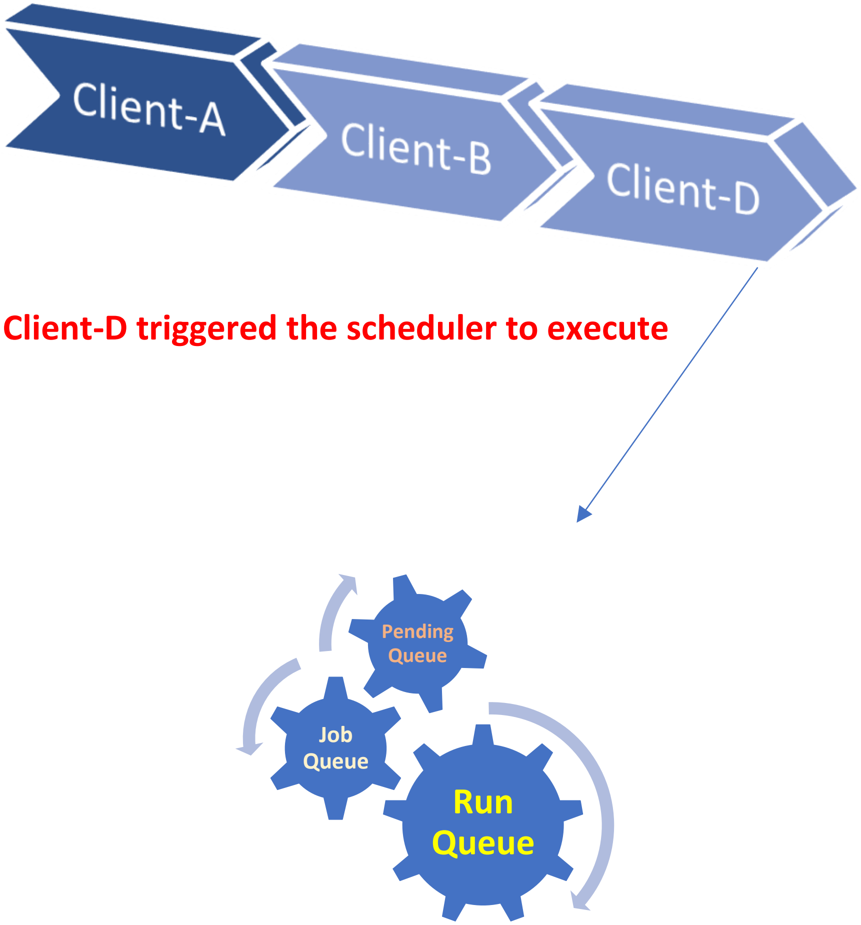
- Client is Enabled
- Client is added in to job queue

Client-C

- Client is disabled
- Client is removed from scheduling

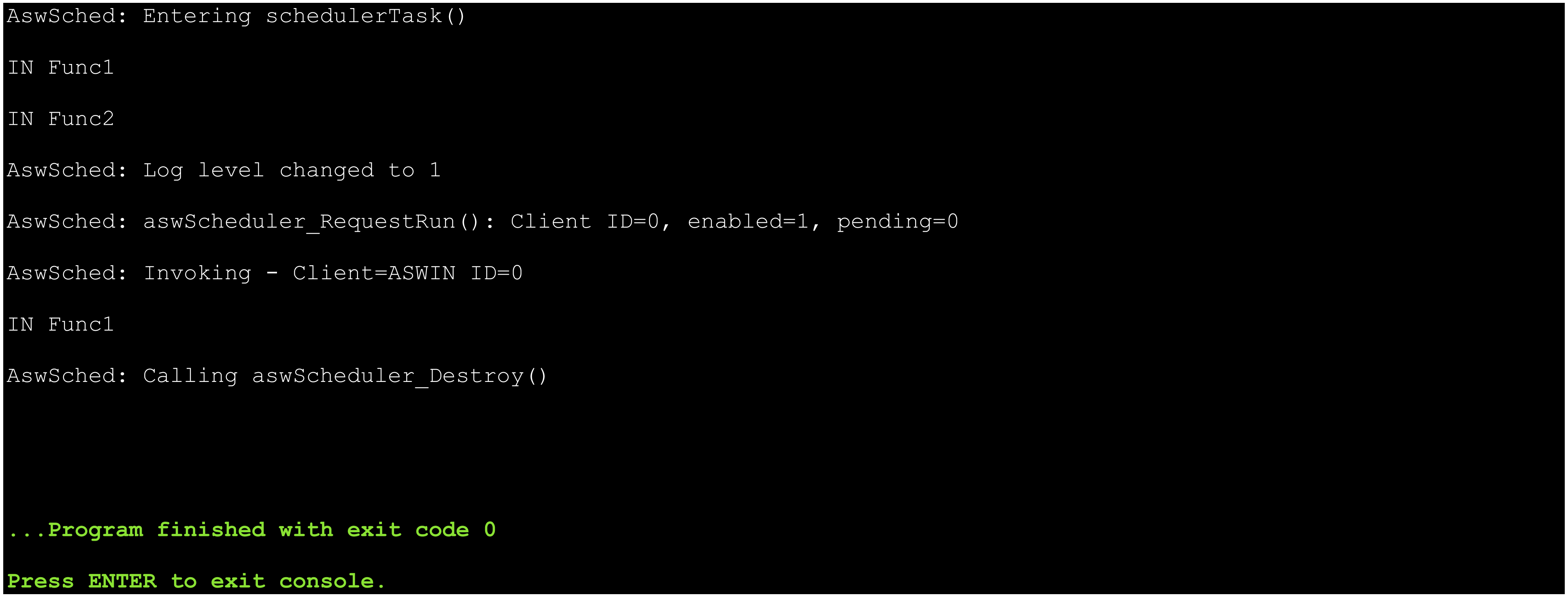
Client-D

- Client is enabled +request for execution
- Client is removed form scheduling



Client-D triggered the scheduler to execute

Expected result using Test code [main.c]:



```
AswSched: Entering schedulerTask()

IN Func1

IN Func2

AswSched: Log level changed to 1

AswSched: aswScheduler_RequestRun(): Client ID=0, enabled=1, pending=0

AswSched: Invoking - Client=ASWIN ID=0

IN Func1

AswSched: Calling aswScheduler_Destroy()




...Program finished with exit code 0

Press ENTER to exit console.
```