Lab Exercise: Jenkins – The Scheduler revisited

In this exercise you will get acquainted with Jenkins as a continuous integration (CI) tool. Through this exercise you will gain proficiency in the setting up and using of Jenkins projects to build your C# projects and execute the associated unit tests.

It is a prerequisite to this exercise that you have watch the video Jenkins.mp4 on the course Blackboard site and that you have completed the previous `Scheduler` exercise (Git Basics).

Exercise 1:
Clone an available repository and copy your solution from the previous `Scheduler` exercise to this repository. Do this in the same manner as you have done before (1 team member initially clones, adds, commits and pushes the code, the rest of the team then clones the repo and ensures the code builds out-of-the-box).

Exercise 2:
Create a Jenkins project which will build your project. Ensure that the Jenkins project executes every time you do a "git push" to your repository.

NOTE: You are not going to get a build running off Jenkins in the first attempt – probably, some setting is wrong, some path name is off or some URL is misspelled. Everything you need to know to correct the error and get the Jenkins project building is in the console output of the project. Consult this, find out what is wrong and correct the mistakes. Eventually, you will have a Jenkins build that is kicked off seemingly by magic every time you do a Git push – it's worth waiting for!

Exercise 3:
Add the following operations to class `Scheduler`. Make decisions on details of operation and exceptions as necessary. Ensure that each time some team member commits code, the Jenkins build is kicked off.

`Kill(string name)`	Kill a given thread. Consider if the thread to kill is active or does not exist
`SetPriority(string name, Priority pri)`	Set the priority of the given thread. Consider if the thread is active or does not exist

Exercise 4:
Once you have the understanding of what happens from you do a git push to your repository until Jenkins has finished executing a build, set the Jenkins build up to execute the unit tests in your solution also, as described in the video. Either start making a new Jenkins project from the DemoProject, or add the necessary step to the Jenkins build you already have.