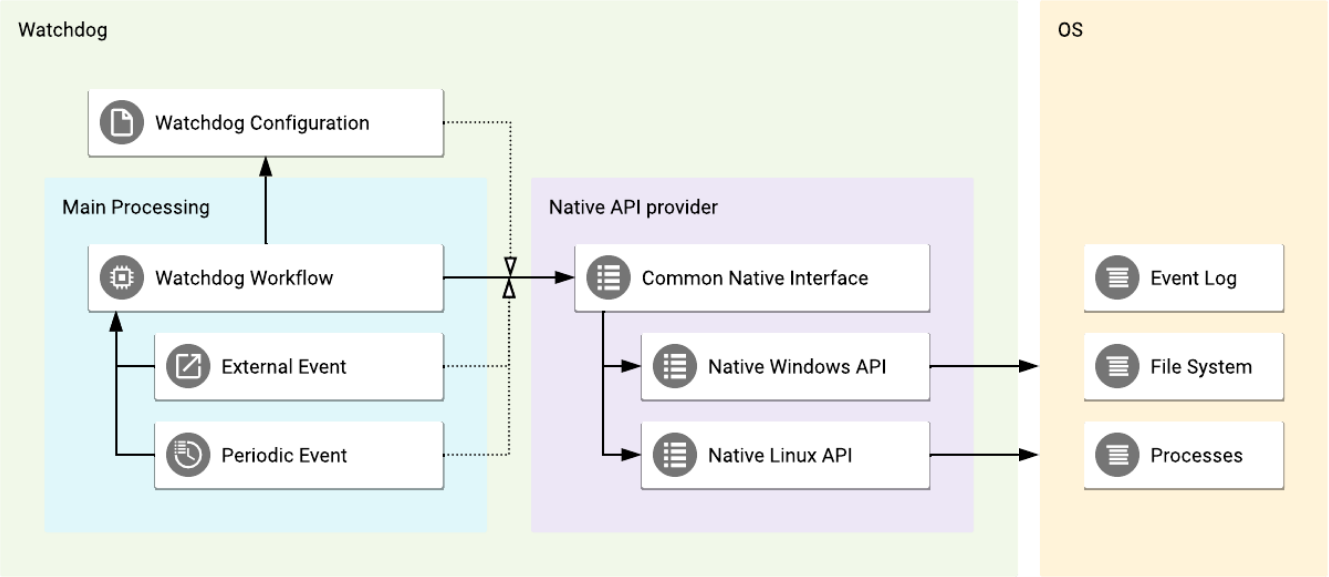
Linux Developer code test

Requirements

- All code should be commented
- All code should be written in C++ (except configuration files)
- The project should be developed using C/C++ for Visual Studio Code and should be fully compatible with it
- All the prerequisites and requirements for the project should be clearly documented

If one or more of the features cannot be implemented - please provide your research results (what have you tried and why do you think it cannot be implemented?)

Diagram



Tasks

Task 1: Monitor applications/processes in Linux

Note: please use the Diagram above for a reference.

Requirements:

1. The configuration requirements:
    a. The watchdog configuration should be read on startup from the plain text file (any common configuration format is acceptable: YAML, JSON, INI, etc.)
    b. The configuration should be reloaded dynamically when the changes are made in the configuration file
    c. The configuration file should describe the list of processes/applications which need to be monitored by watchdog
    d. Each monitored process/application in the configuration file can have startup parameters
2. The processing requirements:
    a. All the configured processes/applications should be checked on startup and started with the configured startup parameters if they are not running already
    b. If a new process/application is added into the configuration file after the watchdog is already started - the watchdog should dynamically check and start such new process/application with the configured startup parameters
    c. If one of the configured processes/applications is stopped or killed - it should be automatically restarted by watchdog with the configured startup parameters
    d. All the processes/applications removed from the configuration file after the watchdog is already started should no longer be monitored and restarted by the watchdog (until such processes/applications are added back to the configuration file)
    e. The process/application monitoring should be implemented using system events or periodic checks or both
3. All the Native OS API should be implemented as a separate component
    a. This component should encapsulate all the Native API details and provide a simple interface for all the actions
    b. There should not be any other way to access the Native OS API, only through this component
    c. It should be possible to implement support of other operating systems (for example support Windows in addition to Linux) by extending the component
4. All the meaningful events and actions done by watchdog should be logged into the system event log and available for the later review and analysis
5. All the essential watchdog workflows should be covered by integration tests
6. All the individual component methods should be covered by unit tests

Task 2 (optional): Monitor foreground application in Linux

Note: please use the Diagram above for a reference.

Requirements:

1. In addition to the standard configuration requirements (see Task 1), the configuration file is allowed to have one application marked as a foreground process.
2. For the application marked as a foreground process, the watchdog should implement a workflow to always keep such application in the foreground and visible on top of all other running applications.
3. The configured foreground application should be checked on startup and should be brought in foreground on top of all other applications
4. The configuration for the foreground application should be dynamically read and applied by watchdog, for example if we change the foreground application in the configuration file after the watchdog is already started - the watchdog should bring the new application in foreground and start monitoring it instead of the old foreground application

Task 3 (optional): Implement both Linux and Windows monitoring

Note: please use the Diagram above for a reference.

Requirements:

1. Implement the full watchdog workflow for both Linux and Windows in the same project by isolating the Native OS API in one component
2. Use separate build configurations to build the watchdog application from the same source code for both Linux and Windows