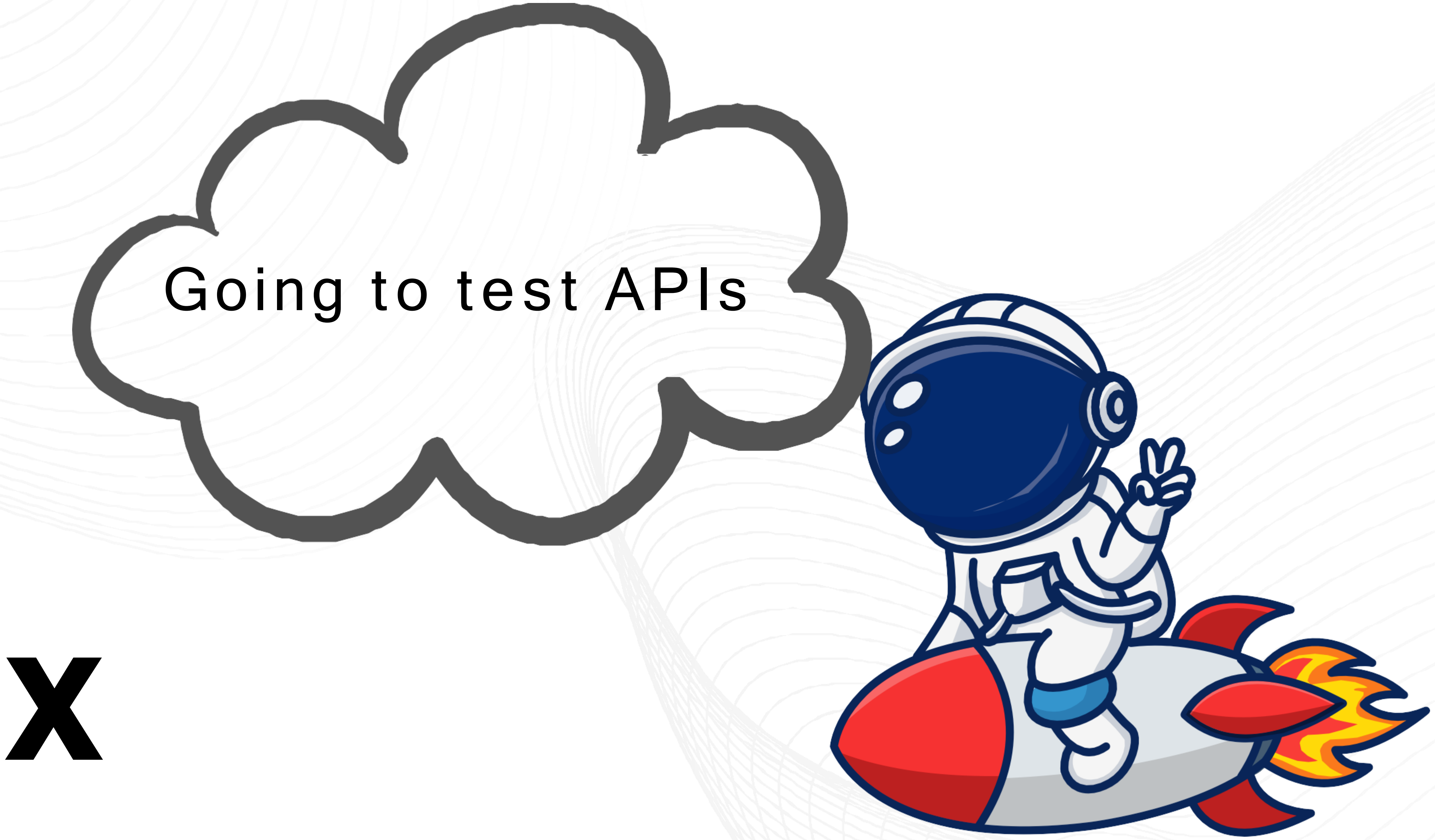
Final Project Proposal



GALAXY X

Operating system

June 13, 2023

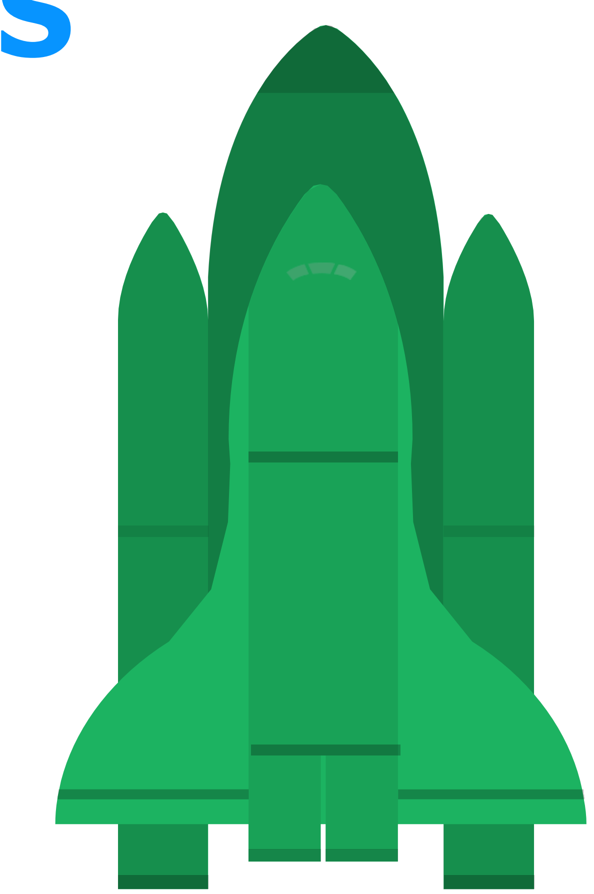
INTRODUCTION

Professor Kim Young Il

TEAM
Galaxy X
Aleksei
Oybek
Joseph
Juyoung
Chul Hyun

AGENDA

1. PROJECT INTRODUCTION
2. SUMMARY OF PROBLEM AND CHALLENGES
3. FINAL DESIGN
4. CONCLUSION
5. RECOMMENDATION
6. DEMONSTRATION



PROJECT INTRO

Key Achievements

- C basics and syntax
- Real-time OS
- API configuration
- GitHub
- 40 tested APIs
- Working environment

Limitations

- Some functions can not be executed
- Laptop limitations (Resources on the web
- Deep understanding basics of computer
- Provided example information
- Programming language

Backround

Spacecraft's Management and Scheduler functions (APIS) expected to be tested

SUMMARY OF PROBLEMS

KNOWLEDGE PROBLEMS

- C programming language
- freeRTOS
- API
- GitHub
- IDE configuration

SOLUTION

- Learnt C basics and syntax
- Followed tutorial, manual and example files
- Asked from seniors
- Followed the leader
- Watched guides

DESIGN METHODS

01.

MSVC port

Only Microsoft Windows was used

Popular compiler for C/C++

Includes IDE

02.

GITHUB

Open source tool to control source file versions

Control team code

Revert back

History of development

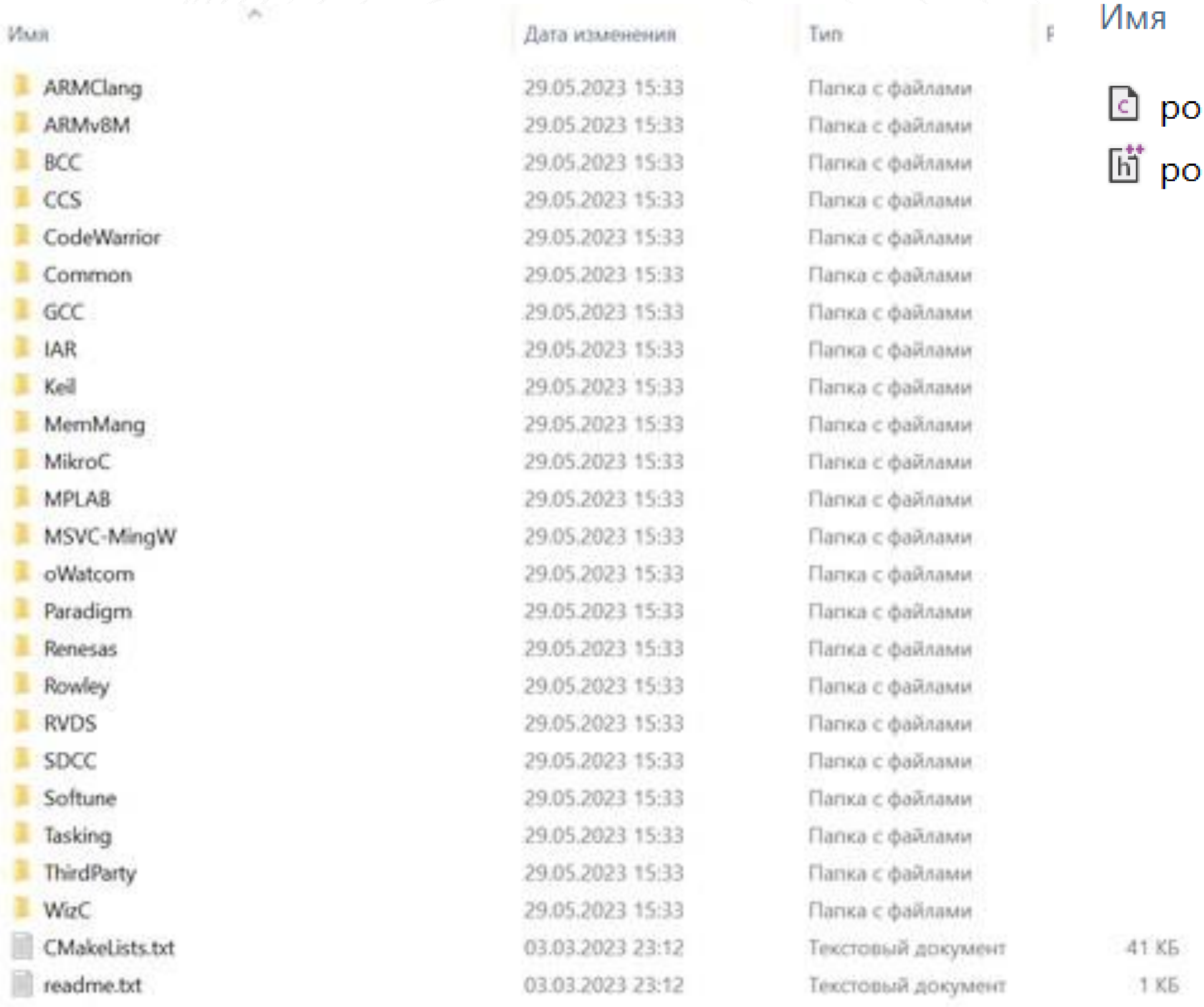
03.

MS VS community

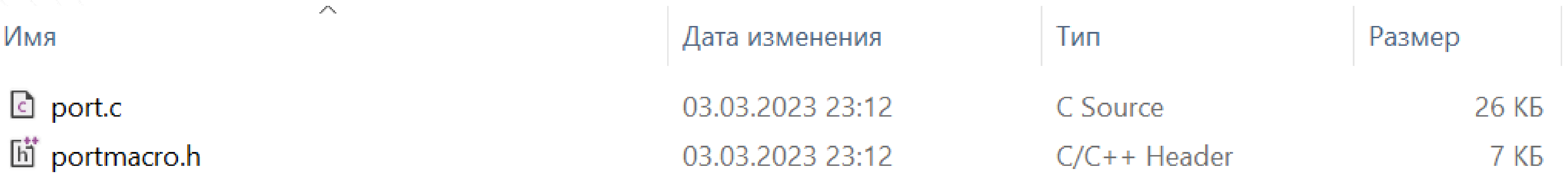
To build, execute and debug C programs

SLN file to set up configuration for compiling and building

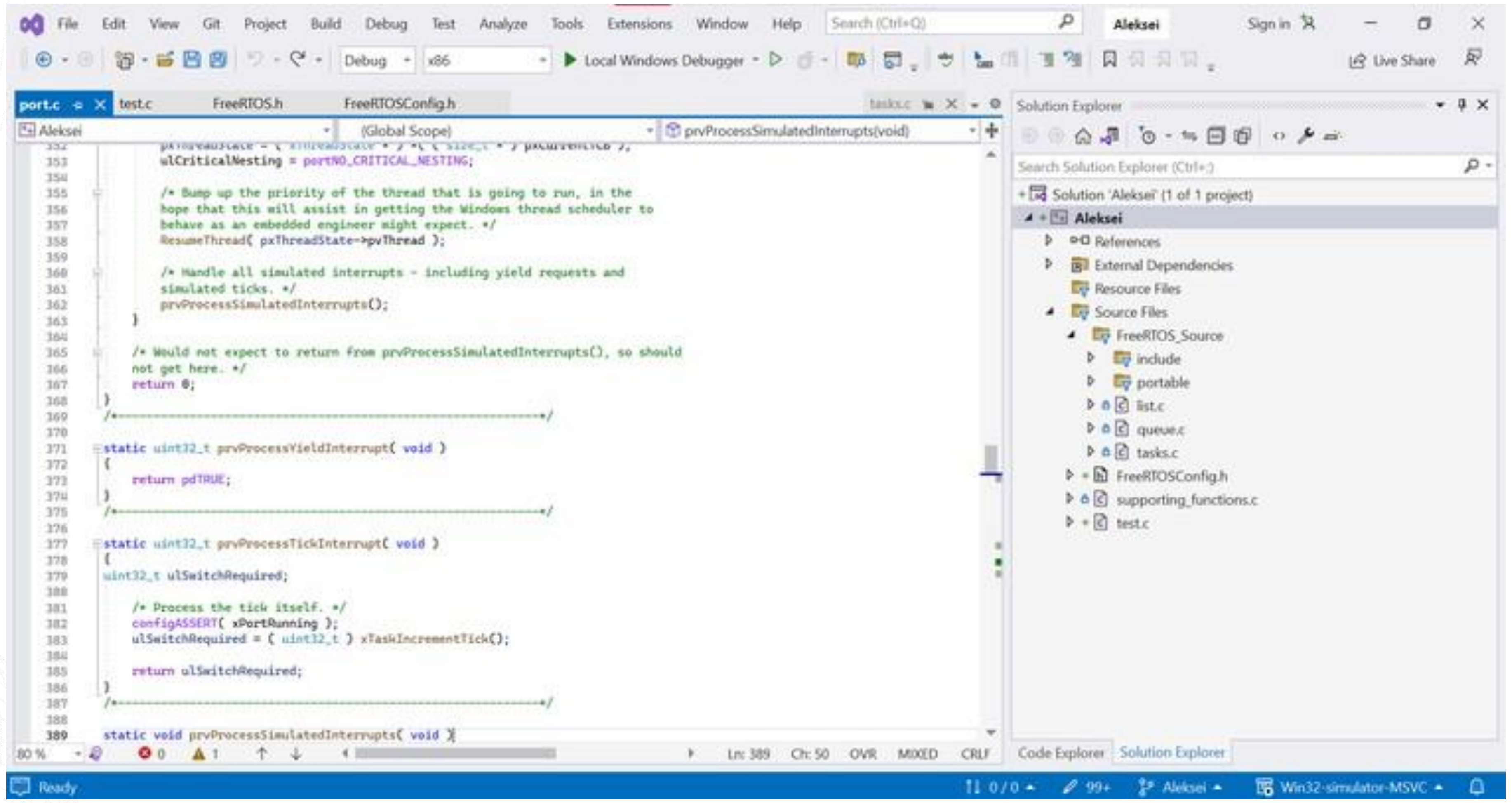
DESCRIPTION OF FINAL DESIGN (MSVC)



Имя	Дата изменения	Тип	Р
ARMClang	29.05.2023 15:33	Папка с файлами	
ARMv8M	29.05.2023 15:33	Папка с файлами	
BCC	29.05.2023 15:33	Папка с файлами	
CCS	29.05.2023 15:33	Папка с файлами	
CodeWarrior	29.05.2023 15:33	Папка с файлами	
Common	29.05.2023 15:33	Папка с файлами	
GCC	29.05.2023 15:33	Папка с файлами	
IAR	29.05.2023 15:33	Папка с файлами	
Keil	29.05.2023 15:33	Папка с файлами	
MemMang	29.05.2023 15:33	Папка с файлами	
MikroC	29.05.2023 15:33	Папка с файлами	
MPLAB	29.05.2023 15:33	Папка с файлами	
MSVC-MingW	29.05.2023 15:33	Папка с файлами	
oWatcom	29.05.2023 15:33	Папка с файлами	
Paradigm	29.05.2023 15:33	Папка с файлами	
Renesas	29.05.2023 15:33	Папка с файлами	
Rowley	29.05.2023 15:33	Папка с файлами	
RVDS	29.05.2023 15:33	Папка с файлами	
SDCC	29.05.2023 15:33	Папка с файлами	
Softune	29.05.2023 15:33	Папка с файлами	
Tasking	29.05.2023 15:33	Папка с файлами	
ThirdParty	29.05.2023 15:33	Папка с файлами	
WizC	29.05.2023 15:33	Папка с файлами	
CMakeLists.txt	03.03.2023 23:12	Текстовый документ	41 КБ
readme.txt	03.03.2023 23:12	Текстовый документ	1 КБ

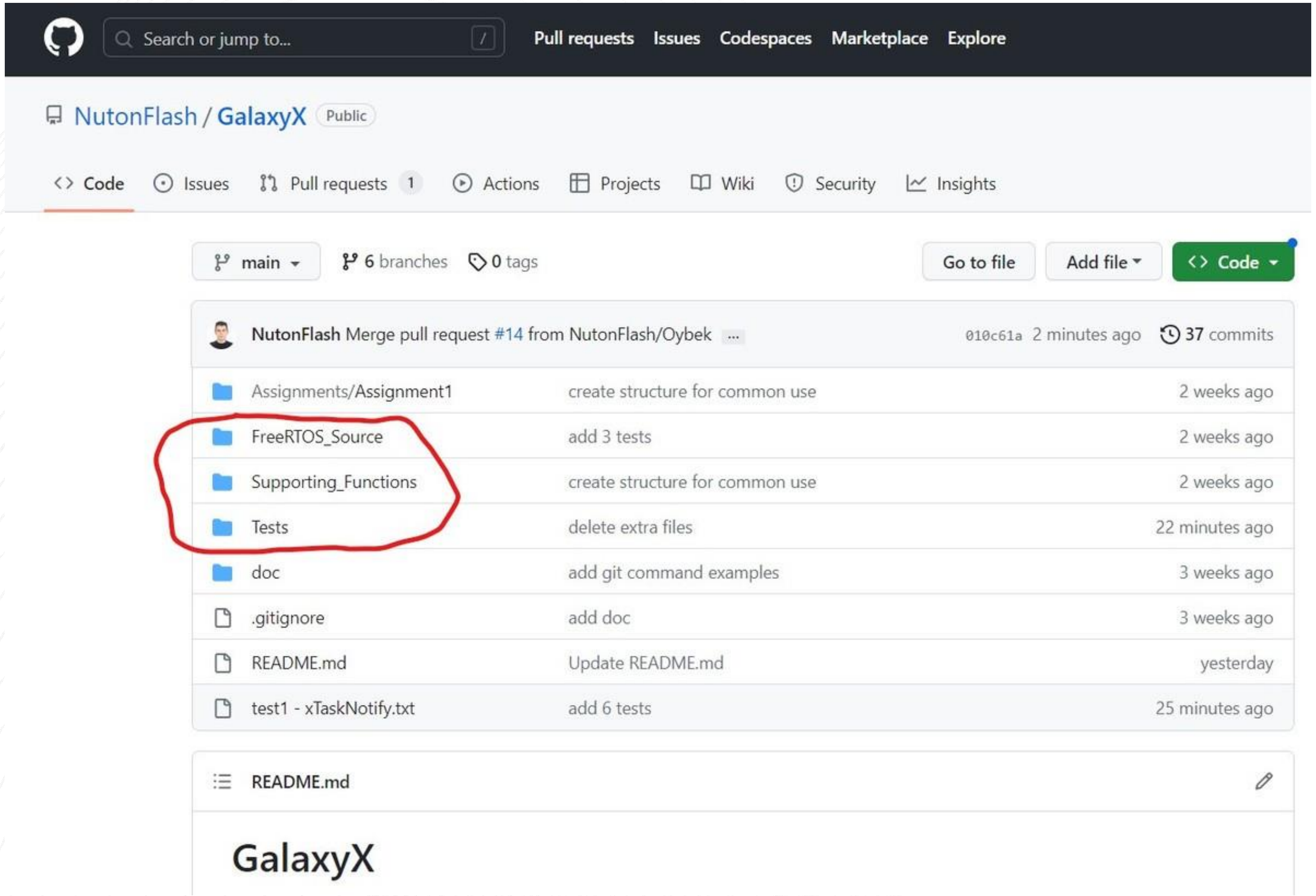


Имя	Дата изменения	Тип	Размер
port.c	03.03.2023 23:12	C Source	26 КБ
portmacro.h	03.03.2023 23:12	C/C++ Header	7 КБ



```c
        pxThreadState = ( xThreadState * ) *( ( size_t * ) pxCurrentTCB );
        ulCriticalNesting = portNO_CRITICAL_NESTING;

        /* Bump up the priority of the thread that is going to run, in the
        hope that this will assist in getting the Windows thread scheduler to
        behave as an embedded engineer might expect. */
        ResumeThread( pxThreadState->pvThread );

        /* Handle all simulated interrupts - including yield requests and
        simulated ticks. */
        prvProcessSimulatedInterrupts();
    }

    /* Would not expect to return from prvProcessSimulatedInterrupts(), so should
    not get here. */
    return 0;
}
/*-----------------------------------------------------------*/

static uint32_t prvProcessYieldInterrupt( void )
{
    return pdTRUE;
}
/*-----------------------------------------------------------*/

static uint32_t prvProcessTickInterrupt( void )
{
uint32_t ulSwitchRequired;

    /* Process the tick itself. */
    configASSERT( xPortRunning );
    ulSwitchRequired = ( uint32_t ) xTaskIncrementTick();

    return ulSwitchRequired;
}
/*-----------------------------------------------------------*/

static void prvProcessSimulatedInterrupts( void )
```

DESCRIPTION OF FINAL DESIGN (GITHUB)

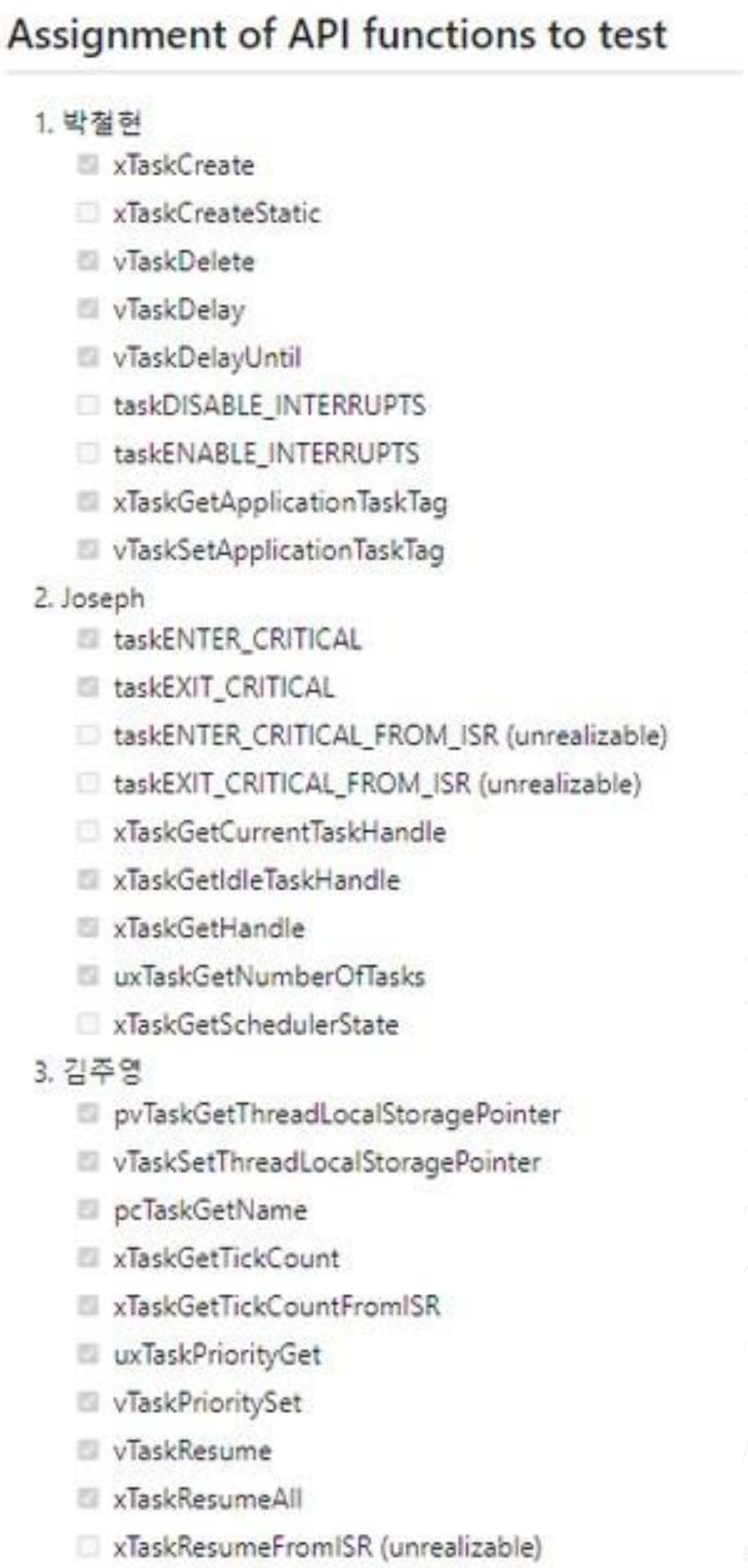


NutonFlash Merge pull request #14 from NutonFlash/Oybek ... 010c61a 2 minutes ago 37 commits

Assignments/Assignment1	create structure for common use	2 weeks ago
FreeRTOS_Source	add 3 tests	2 weeks ago
Supporting_Functions	create structure for common use	2 weeks ago
Tests	delete extra files	22 minutes ago
doc	add git command examples	3 weeks ago
.gitignore	add doc	3 weeks ago
README.md	Update README.md	yesterday
test1 - xTaskNotify.txt	add 6 tests	25 minutes ago

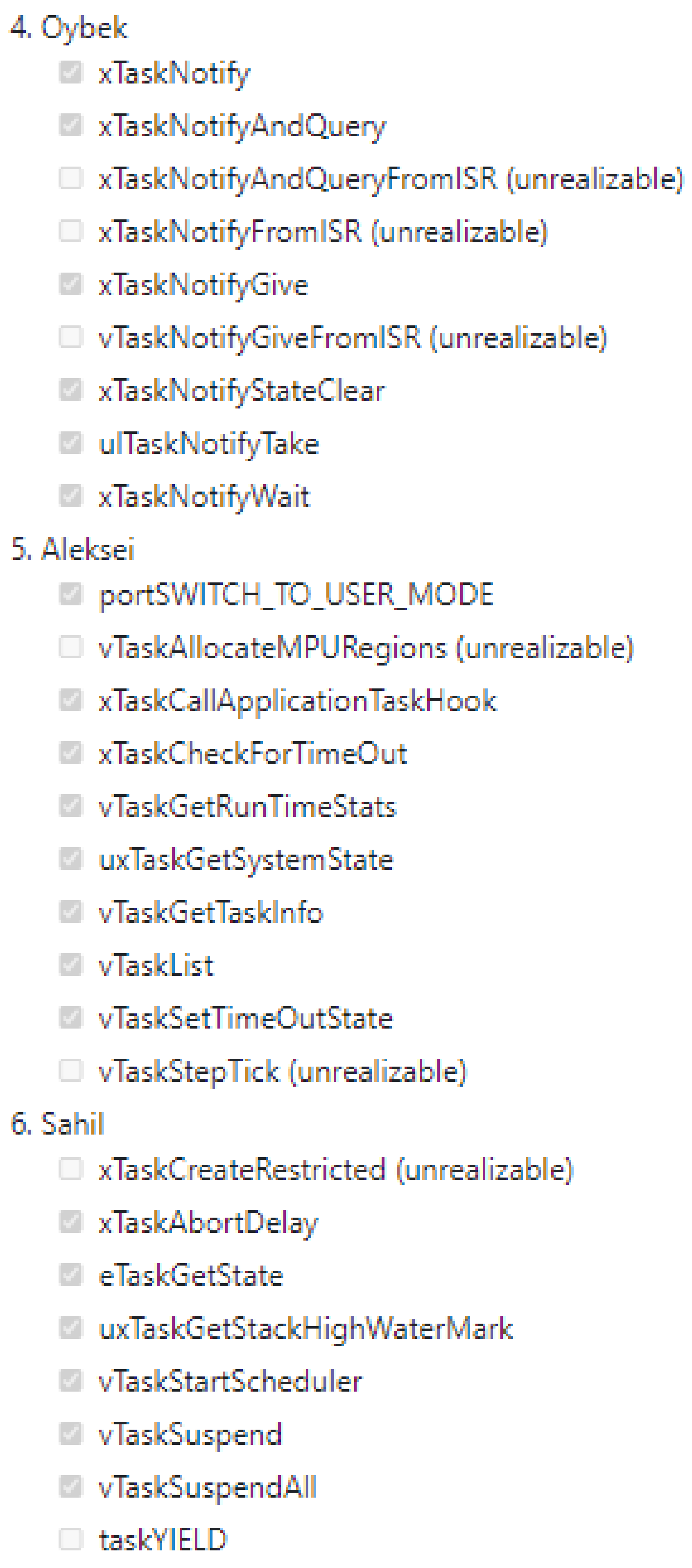
README.md

GalaxyX



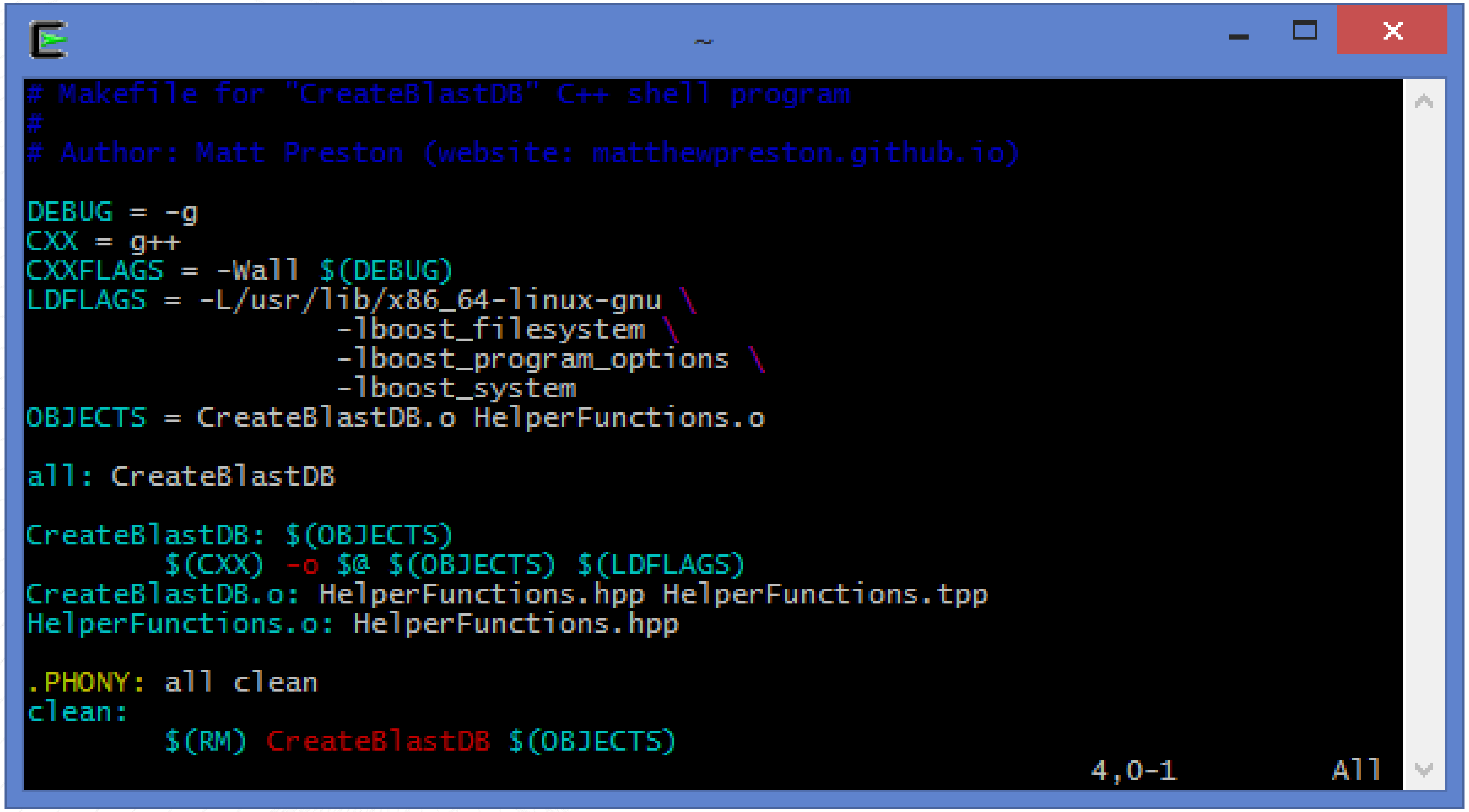
Assignment of API functions to test

1. 박철헌
   - ☑ xTaskCreate
   - □ xTaskCreateStatic
   - ☑ vTaskDelete
   - ☑ vTaskDelay
   - ☑ vTaskDelayUntil
   - □ taskDISABLE_INTERRUPTS
   - □ taskENABLE_INTERRUPTS
   - ☑ xTaskGetApplicationTaskTag
   - ☑ vTaskSetApplicationTaskTag
2. Joseph
   - ☑ taskENTER_CRITICAL
   - ☑ taskEXIT_CRITICAL
   - □ taskENTER_CRITICAL_FROM_ISR (unrealizable)
   - □ taskEXIT_CRITICAL_FROM_ISR (unrealizable)
   - □ xTaskGetCurrentTaskHandle
   - ☑ xTaskGetIdleTaskHandle
   - ☑ xTaskGetHandle
   - ☑ uxTaskGetNumberOfTasks
   - □ xTaskGetSchedulerState
3. 김주영
   - ☑ pvTaskGetThreadLocalStoragePointer
   - ☑ vTaskSetThreadLocalStoragePointer
   - ☑ pcTaskGetName
   - ☑ xTaskGetTickCount
   - ☑ xTaskGetTickCountFromISR
   - ☑ uxTaskPriorityGet
   - ☑ vTaskPrioritySet
   - ☑ vTaskResume
   - ☑ xTaskResumeAll
   - □ xTaskResumeFromISR (unrealizable)

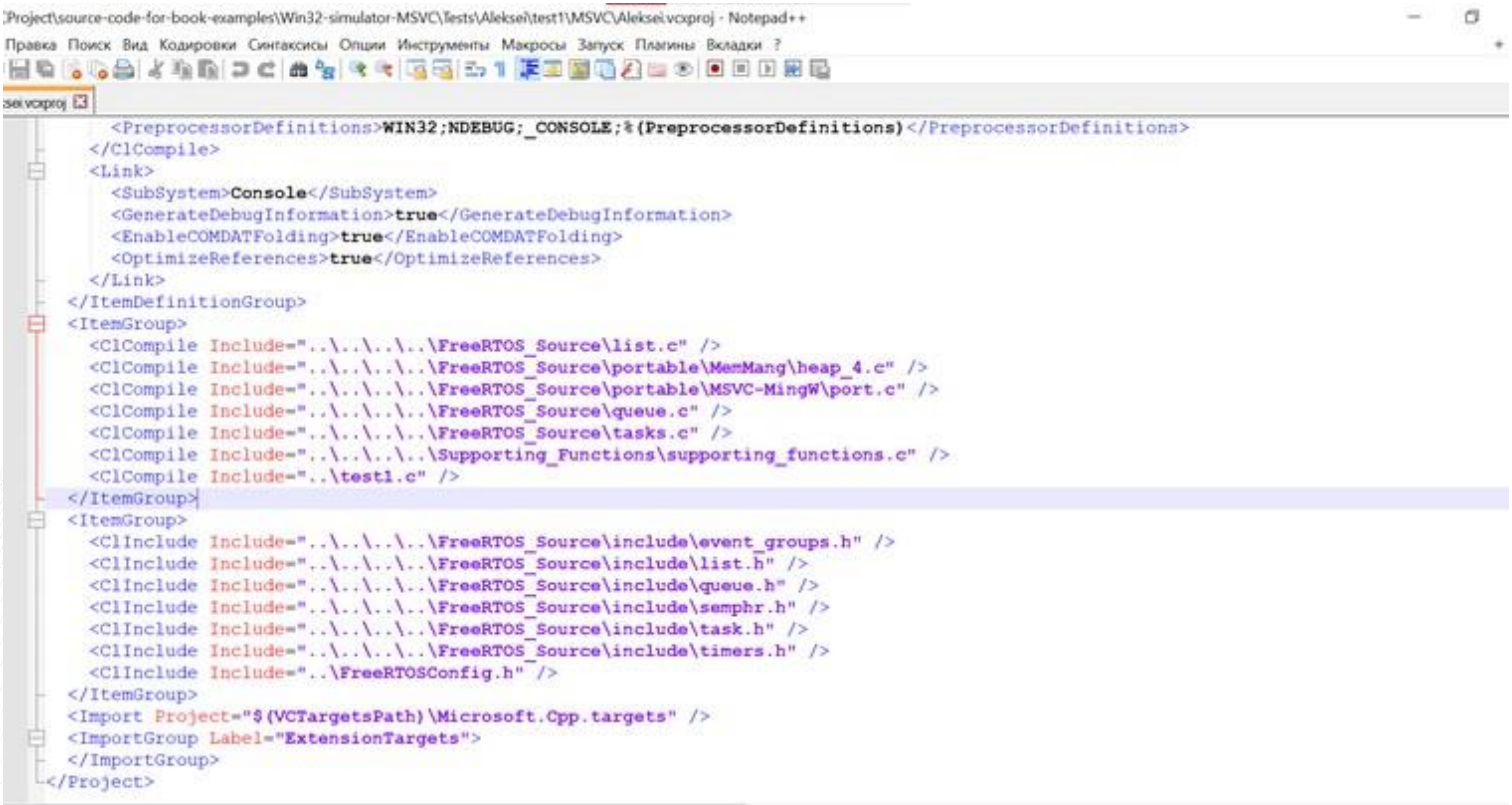


4. Oybek
   - ☑ xTaskNotify
   - ☑ xTaskNotifyAndQuery
   - □ xTaskNotifyAndQueryFromISR (unrealizable)
   - □ xTaskNotifyFromISR (unrealizable)
   - ☑ xTaskNotifyGive
   - □ vTaskNotifyGiveFromISR (unrealizable)
   - ☑ xTaskNotifyStateClear
   - ☑ ulTaskNotifyTake
   - ☑ xTaskNotifyWait
5. Aleksei
   - ☑ portSWITCH_TO_USER_MODE
   - □ vTaskAllocateMPURegions (unrealizable)
   - ☑ xTaskCallApplicationTaskHook
   - ☑ xTaskCheckForTimeOut
   - ☑ vTaskGetRunTimeStats
   - ☑ uxTaskGetSystemState
   - ☑ vTaskGetTaskInfo
   - ☑ vTaskList
   - ☑ vTaskSetTimeOutState
   - □ vTaskStepTick (unrealizable)
6. Sahil
   - □ xTaskCreateRestricted (unrealizable)
   - ☑ xTaskAbortDelay
   - ☑ eTaskGetState
   - ☑ uxTaskGetStackHighWaterMark
   - ☑ vTaskStartScheduler
   - ☑ vTaskSuspend
   - ☑ vTaskSuspendAll
   - □ taskYIELD

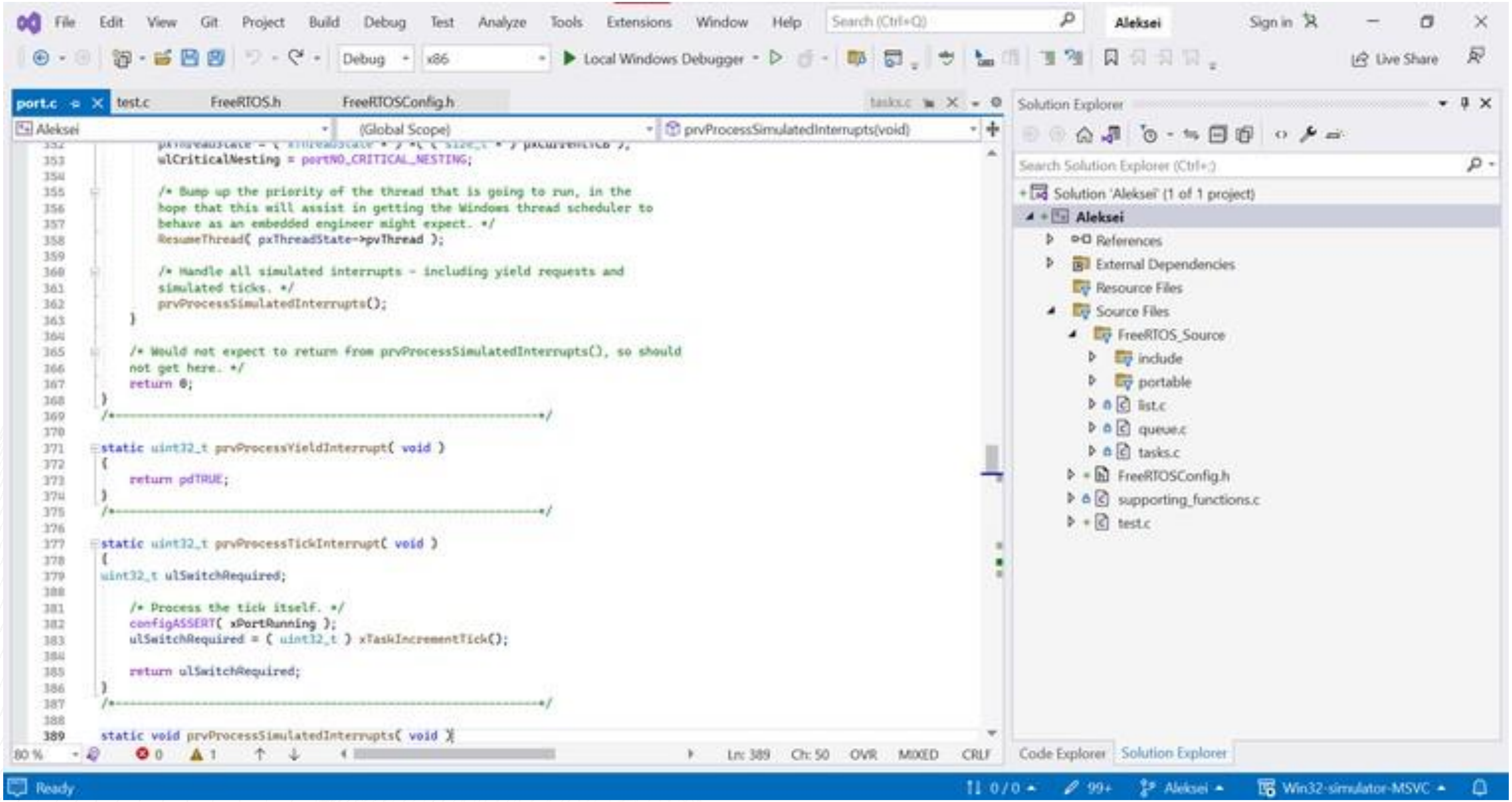
Makefile



```
# Makefile for "CreateBlastDB" C++ shell program
#
# Author: Matt Preston (website: matthewpreston.github.io)

DEBUG = -g
CXX = g++
CXXFLAGS = -Wall $(DEBUG)
LDFLAGS = -L/usr/lib/x86_64-linux-gnu \
                  -lboost_filesystem \
                  -lboost_program_options \
                  -lboost_system
OBJECTS = CreateBlastDB.o HelperFunctions.o

all: CreateBlastDB

CreateBlastDB: $(OBJECTS)
        $(CXX) -o $@ $(OBJECTS) $(LDFLAGS)
CreateBlastDB.o: HelperFunctions.hpp HelperFunctions.tpp
HelperFunctions.o: HelperFunctions.hpp

.PHONY: all clean
clean:
        $(RM) CreateBlastDB $(OBJECTS)
                                                              4,0-1         All
```


:Project\source-code-for-book-examples\Win32-simulator-MSVC\Tests\Aleksei\test1\MSVC\Aleksei.vcxproj - Notepad++
Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
sei.vcxproj

```
      <PreprocessorDefinitions>WIN32;NDEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDefinitions>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <GenerateDebugInformation>true</GenerateDebugInformation>
      <EnableCOMDATFolding>true</EnableCOMDATFolding>
      <OptimizeReferences>true</OptimizeReferences>
    </Link>
  </ItemDefinitionGroup>
  <ItemGroup>
    <ClCompile Include="..\..\..\..\FreeRTOS_Source\list.c" />
    <ClCompile Include="..\..\..\..\FreeRTOS_Source\portable\MemMang\heap_4.c" />
    <ClCompile Include="..\..\..\..\FreeRTOS_Source\portable\MSVC-MingW\port.c" />
    <ClCompile Include="..\..\..\..\FreeRTOS_Source\queue.c" />
    <ClCompile Include="..\..\..\..\FreeRTOS_Source\tasks.c" />
    <ClCompile Include="..\..\..\..\Supporting_Functions\supporting_functions.c" />
    <ClCompile Include="..\test1.c" />
  </ItemGroup>
  <ItemGroup>
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\event_groups.h" />
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\list.h" />
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\queue.h" />
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\semphr.h" />
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\task.h" />
    <ClInclude Include="..\..\..\..\FreeRTOS_Source\include\timers.h" />
    <ClInclude Include="..\FreeRTOSConfig.h" />
  </ItemGroup>
  <Import Project="$(VCTargetsPath)\Microsoft.Cpp.targets" />
  <ImportGroup Label="ExtensionTargets">
  </ImportGroup>
</Project>
```



```
            pxThreadState = ( xThreadState * ) *( ( size_t * ) pxCurrentTCB );
            ulCriticalNesting = portNO_CRITICAL_NESTING;

            /* Bump up the priority of the thread that is going to run, in the
            hope that this will assist in getting the Windows thread scheduler to
            behave as an embedded engineer might expect. */
            ResumeThread( pxThreadState->pvThread );

            /* Handle all simulated interrupts - including yield requests and
            simulated ticks. */
            prvProcessSimulatedInterrupts();
        }

        /* Would not expect to return from prvProcessSimulatedInterrupts(), so should
        not get here. */
        return 0;
    }
    /*-----------------------------------------------------------*/

    static uint32_t prvProcessYieldInterrupt( void )
    {
        return pdTRUE;
    }
    /*-----------------------------------------------------------*/

    static uint32_t prvProcessTickInterrupt( void )
    {
    uint32_t ulSwitchRequired;

        /* Process the tick itself. */
        configASSERT( xPortRunning );
        ulSwitchRequired = ( uint32_t ) xTaskIncrementTick();

        return ulSwitchRequired;
    }
    /*-----------------------------------------------------------*/

    static void prvProcessSimulatedInterrupts( void )
```

SOLUTION FOR CHALLENGING DESIGN PROBLEM

1. Virtual machines
2. GitHub to organize common code structure
3. MS VS to compile and execute the program

Implementation of the purpose of the project

1. 40 API functions were tested
2. Each member created branch with own tests on GitHub
3. Every test was isolated into folder to avoid configuration conflicts

CONCLUSION AND SUGGESTION

01. Tested 40 APIs

02. Knowledge of Task and Scheduler API

03. Basics of C

04. GitHub

WHAT IF WE HAD TWO MORE WEEKS?

- Would finish all 55 APIs
- More qualified tests
- Would test APIs in relax mode
- Finish all reports (weekly reports, quality test report)

THANK YOU