Secure the Quality of OS in Spacecraft

by Team Wicked Crew

Team Information

- **Group Leader:** Ashutosh Khadse
- **Research & Reporting:** Anastasia Baranova, Thant Kyaw Min
- **Development & Testing:** Harishik Dev Singh Jamwal, Ashutosh Khadse
- **Presentation Maker:** Harishik Dev Singh Jamwal
- **Presentation:** Shohujakhon Muhsin Ugli Zikirov, Ashutosh Khadse

Instruction Professor

PROFESSOR KIM YOUNG
AI & BIGDATA

CONTENTS

- Project Goal
- Key Achievements
- Limitations
- Project Summary/Recommendation
- Conclusion
- Demonstration

Understanding FreeRTOS Task Management

1 **Task Creation**

Efficient task creation and initialization

2 **Task Scheduling**

Reliable and responsive scheduling of APIs with each other

3 **Task Synchronization**

Seamless synchronization between concurrent tasks

4 **Completing 55 APIs**

Optimal distribution of API between Team

Key Achievements

Completed 54/55 API

One API need Arduino any micro controller supporting FreeRTOS to implement.

Created a Mock Environment

To run few APIs, we created a mock (virtual) environment.

Solving Errors

Solved the errors by using different ways.

`

Project Limitations

Error Indication and Solution

Error: vTaskResumeFromISR undefined
Solution: Setting Correct Configuration

Hardware Limitations

Need a micro controller to test few APIs without the mock environment.

Stack Depth Error

Static memory being barrier, need to use **dynamic memory** allocation using mock environment

Scalability

Supporting different operating systems **(like windows and MacOS).**

Project Purpose

1

Verify Correct Functionality

2

Measure Performance

3

Detect and Fix Bugs

4

Prepare Documentation

Assignment Of APIs to Test

Ashutosh APIs

1. xTaskAbortDelay
2. xTaskCallApplicationTaskHook
3. vTaskDelay
4. taskDISABLE_INTERRUPTS
5. taskENABLE_INTERRUPTS
6. xTaskGetCurrentTaskHandle
7. xTaskGetIdleTaskHandle
8. uxTaskGetNumberOfTasks
9. vTaskGetRunTimeStats
10. xTaskGetSchedulerState
11. uxTaskGetStackHighWaterMark
12. uxTaskGetSystemState
13. uxTaskPriorityGet
14. vTaskPrioritySet
15. taskENTER_CRITICAL
16. taskEXIT_CRITICAL
17. vTaskSetApplicationTaskTag()
18. xTaskGetApplicationTaskTag()
19. xTaskCreate()
20. vTaskStartScheduler()

Harishik APIs

1. xTaskCreate()
2. vTaskStartScheduler()
3. vTaskDelayUntil()
4. xTaskGetTickCount()
5. vTaskGetInfo()
6. vTaskSuspendAll()
7. xTaskResumeAll()
8. vTaskSuspend()
9. vTaskResume()
10. xTaskNotifyGive()
11. xTaskNotifyStateClear()
12. vTaskDelete()
13. taskYIELD()
14. ulTaskNotifyTake()
15. vApplicationStackOverflowHook()

Minh APIs

1. xTaskGetTickCountFromISR()
2. XTaskList()
3. XTaskNotify()
4. xTaskNotifyAndQuery()
5. xTaskNotifyAndQueryFromISR()
6. xTaskNotifyFromISR()
7. xTaskResumeFromISR()
8. xTaskSetTimeOutState()
9. vTaskStepTick()

Anastasia APIs

1. vTaskAllocateMPURegions()
2. xTaskCheckForTimeOut()
3. vTaskSetTimeOutState()
4. xTaskCreateStatic()
5. taskENTER_CRITICAL_FROM_ISR()

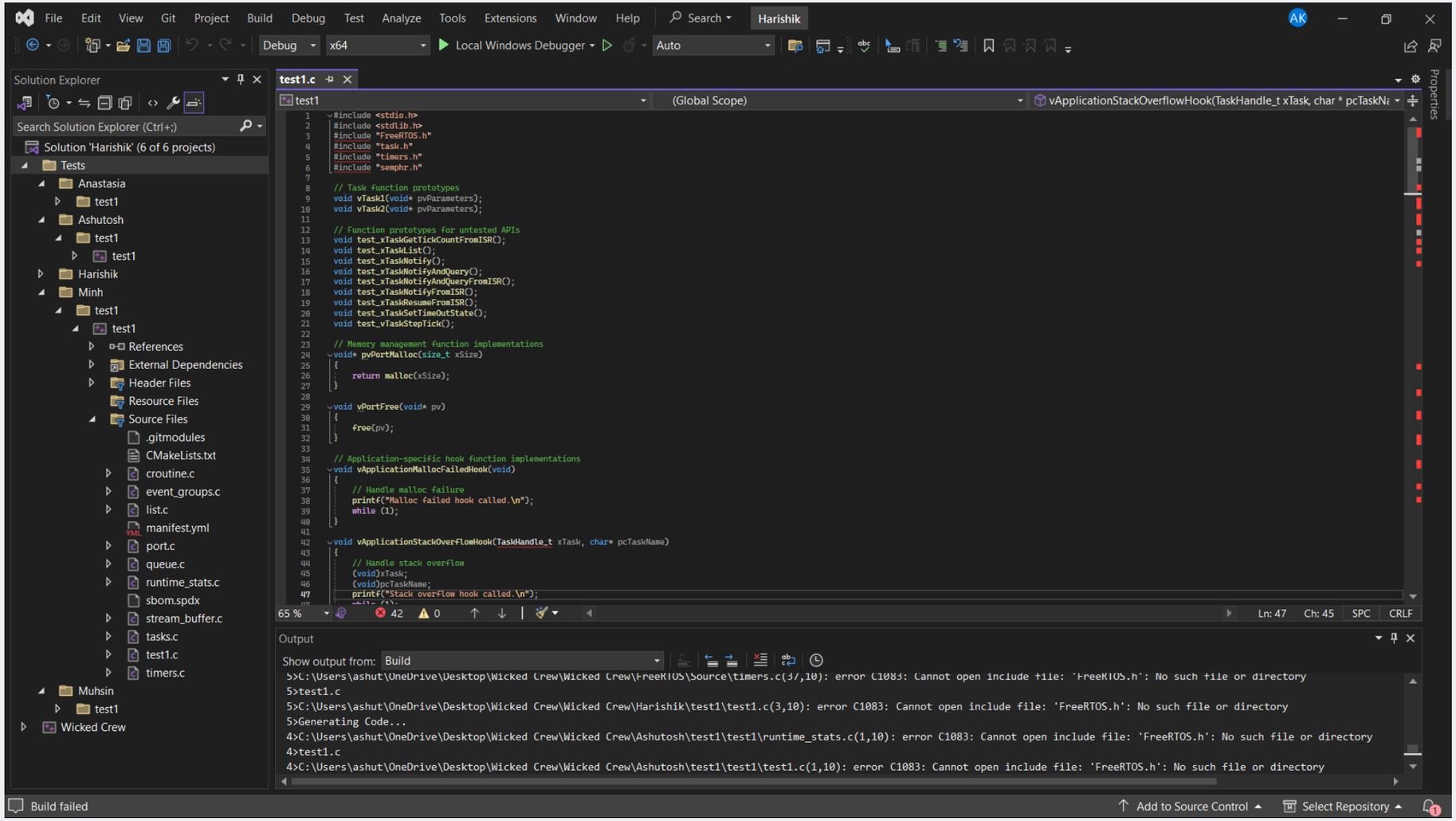
Muhsin APIs

1. taskEXIT_CRITICAL_FROM_ISR()
2. xTaskGetHandle()
3. vTaskGetRunTimeStats()
4. eTaskGetState()
5. pcTaskGetName()

Description of Final Design

Built a Project

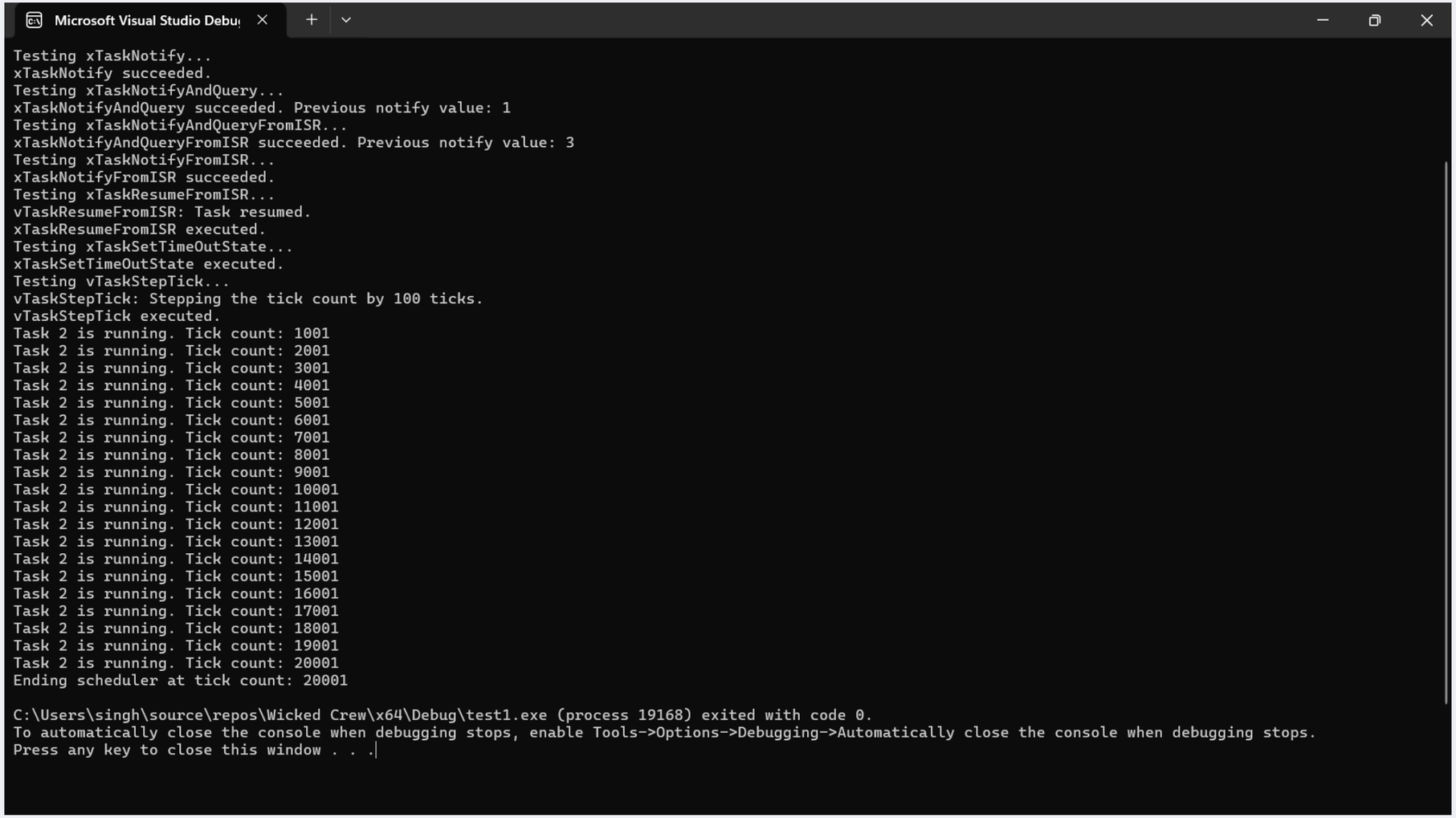
- Folder for each Team member.



Implementation of Final Design

Output Window

- xTaskNotify
- xTaskNotifyAndQuery
- xTaskNotifyFromISR()
- xTaskResumeFromISR()
- xTaskSetTimeOutState()
- xTaskGetTickCountFromISR()
- xTaskList()
- vTaskStepTick()



Implementing Enhancements & Demonstrating Improvements

1

Simulation & Validation

Verify performance and compliance

2

Rigorous Testing

Ensure reliability and robustness

3

Comprehensive Documentation

Facilitate understanding and adoption

Final Conclusions & Suggestions

Recommendation	Benefit
Implement Dynamic Scheduling Algorithms	Improve task prioritization and responsiveness
Enhance Memory Management Strategies	Optimize resource utilization for embedded systems
Develop Comprehensive Error Handling	Ensure reliable and fault-tolerant operation
Facilitate Cross-Team Collaboration	Align with Space-X requirements and objectives

What If we had two more weeks

Detailed Documentation

Creating detailed documentation of APIs

Advanced Memory Management using Hardware

Try to implement dynamic memory allocation using Arduino

More qualified Test

Combining APIs to make code more efficient

Conclusion

By addressing the task management challenges in FreeRTOS, we have optimized its performance and reliability, ensuring that it meets the stringent requirements of Space-X's manned spacecraft.