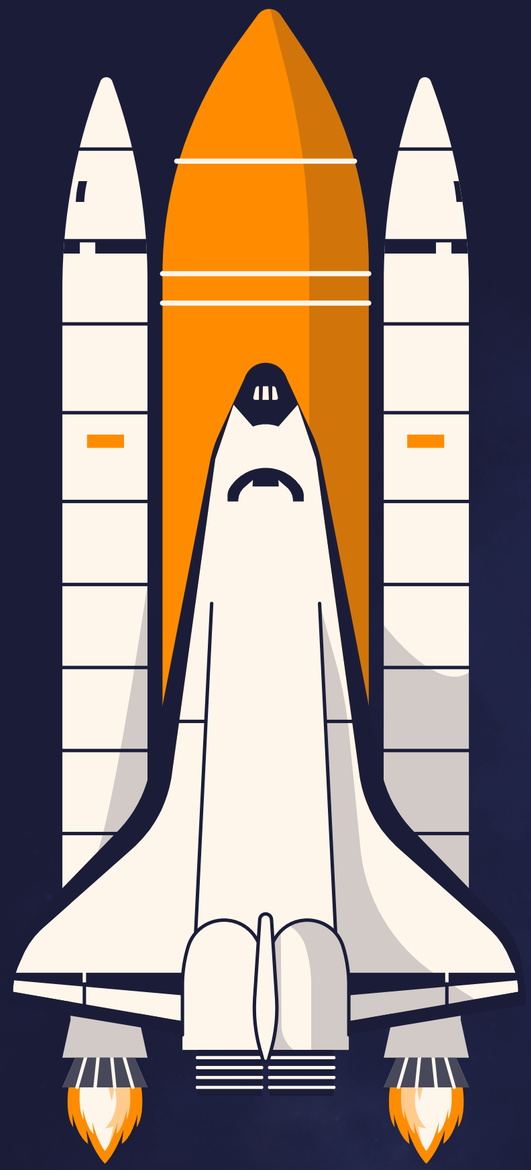


Secure the Quality of OS in Space Shuttle

PROJECT COMPLETION

Prepared by **TEAM OS**

amazon

WOOSONG UNIVERSITY

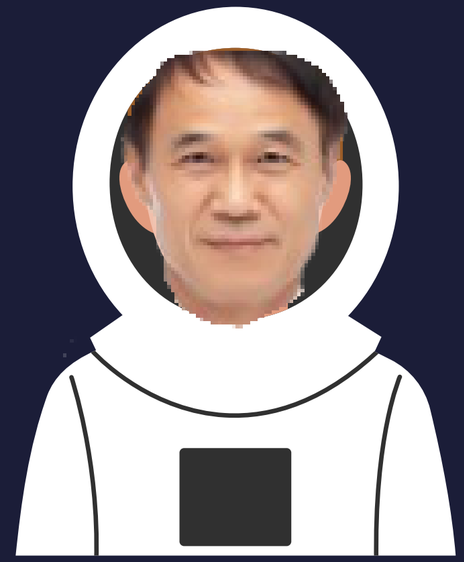
SPACEX

TABLE OF CONTENTS

TEAM INTRODUCTION

TEAM MEMBERS:	STUDENT ID:
Alisher Sadykov	202112123
최정욱	202010144
고권표	202010094
Sweety Kahal	202212051
SHAWON CHAKRABARTY	202212039

INSTRUCTION PROFESSOR

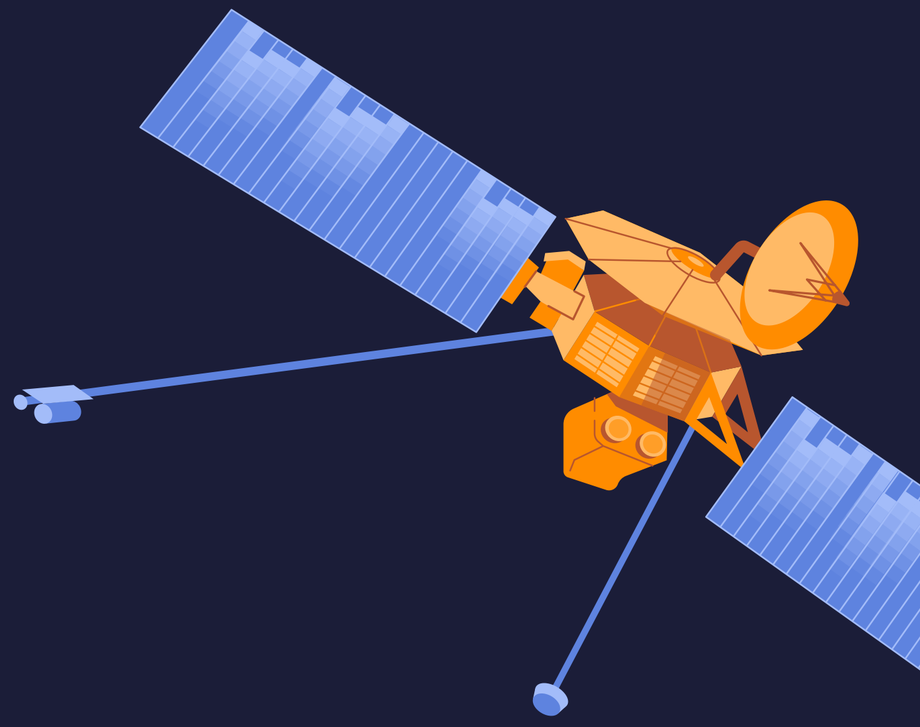


Professor

Kim Young Il

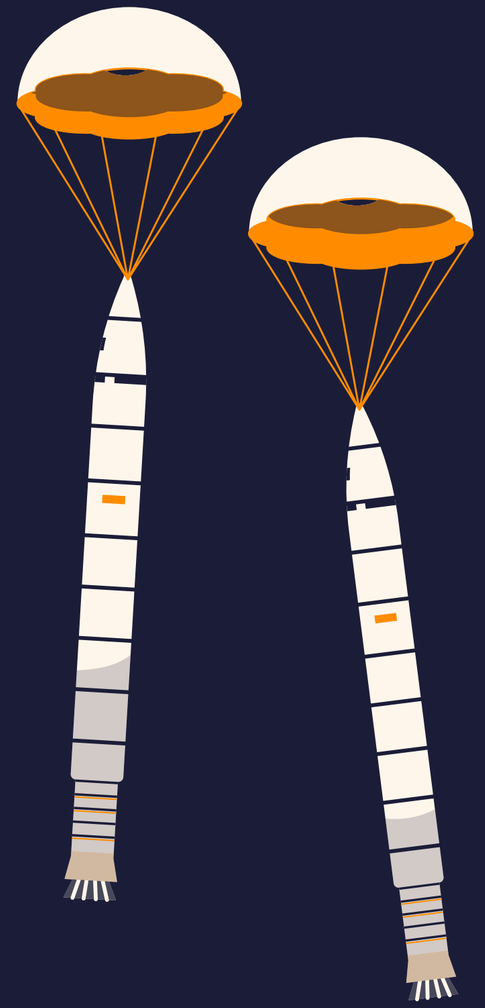
PROJECT GOALS

1. Examine all of the FreeRTOS APIs related to Task Management and Scheduling
2. Improve our C programming language and FreeRTOS skills
3. Improve Embedded Systems Development skills
4. Work as a team and follow ground rules



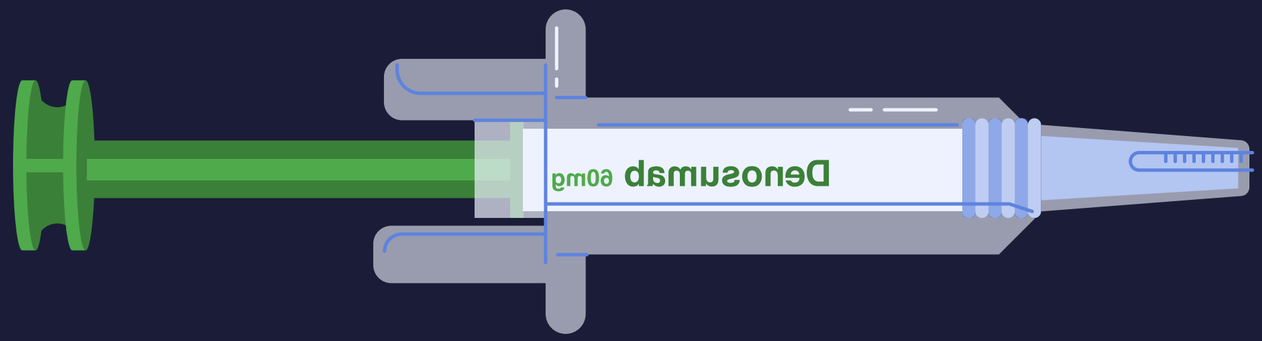
KEY ACHIEVEMENTS

1. Improved skills in C and FreeRTOS
2. Successfully tested 52 APIs
3. Conducted functionality tests
4. Wrote a proper Quality Test Report
5. Worked as a team. Nobody left behind
6. Developed scenario based program



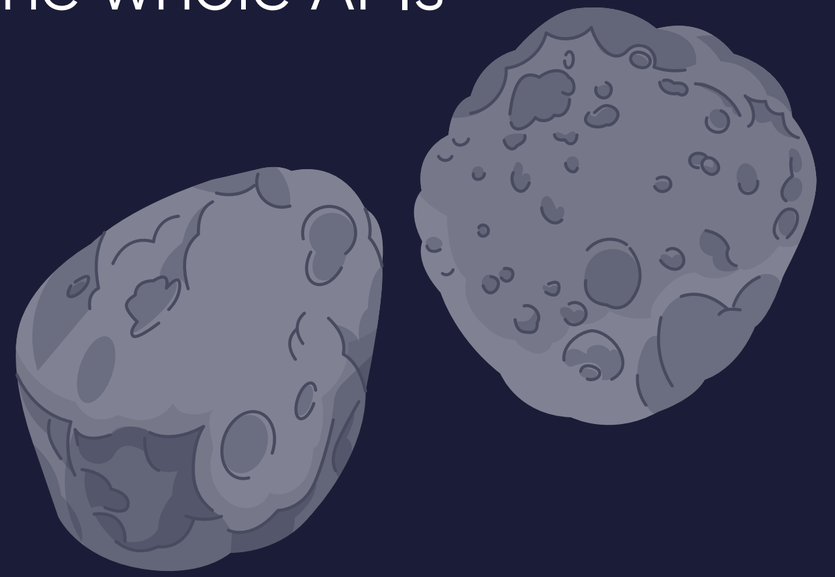
PROJECT LIMITATIONS

1. **Technical Knowledge and Expertise:** Lack of experience working with FreeRTOS and Embedded System Programming
2. **Hardware Requirements:** 3 of the APIs could not be tested because they require to use FreeRTOS MPU(2.1; 2.2; 2.8)
3. **Development Environment:** different OS systems across the team and testing&debugging tools
4. **Time constraints**



PROJECT BACKGROUND

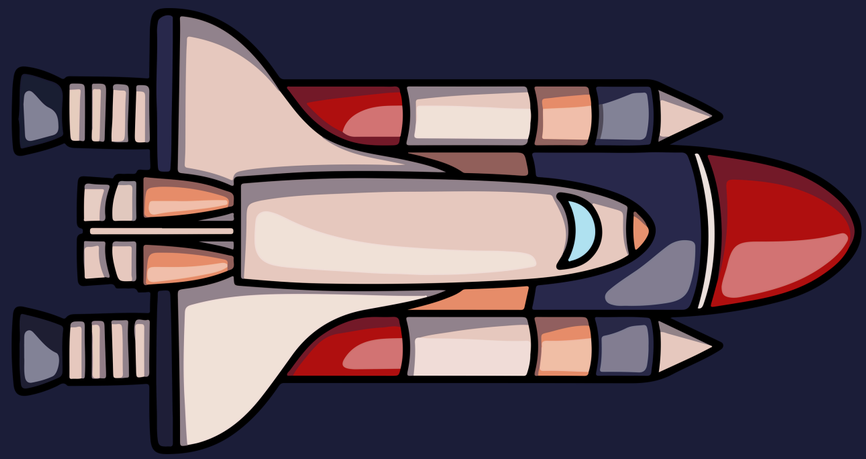
- Amazon developed a real time operating system - **FreeRTOS**
- Targeted usage of FreeRTOS - **Space X's manned spacecraft**
- Problem with the function of **Task Management** found
- Amazon asked Woosong University to examine whole APIs



PROJECT PURPOSE

1. Help **SpaceX** to secure OS quality in space shuttle by testing all the FreeRTOS APIs related with Task Management and Scheduling

2. Imitate real spacecraft functionality by creating a functional scenario-based tests



MILESTONES

April,30
Established team, received task

May,5
Tested 1st API (Homework)

May,14
Project Proposal Presentation

May,25
Tested 10 APIs

May,28
Project Progress Presentation

June,3
Tested 35 APIs

June,7
Finally tested ALL 55 APIs

June,10
Finished Quality Test Report

June,11
Final Presentation

TEST PROCEDURE

1. Review Documentation
2. Define API system requirements
3. Define Expected Test Results
4. Conduct Functional Testing
5. Examine Results
6. Document Test Results

The FreeRTOS™ Reference Manual

2.24 vTaskPrioritySet()

```
#include "FreeRTOS.h"
#include "task.h"

void vTaskPrioritySet( TaskHandle_t pxTask, UBaseType_t uxNewPriority );
```

Listing 82 vTaskPrioritySet() function prototype

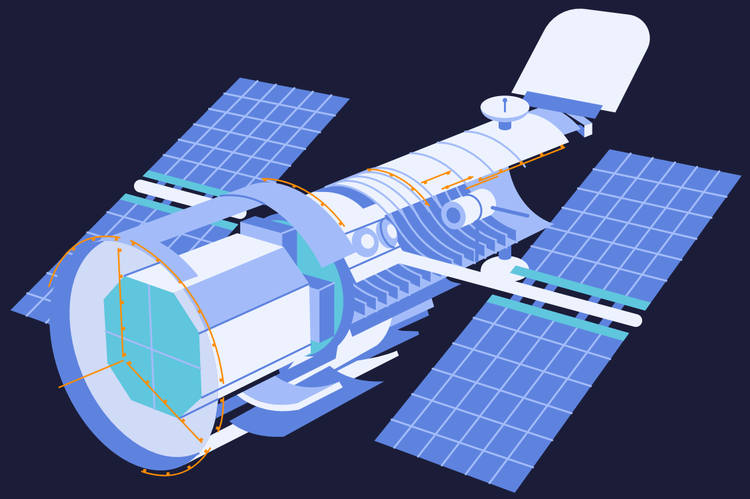
```
TempSensor task priority: 1
DataLogger task priority: 2
AlarmSystem task priority: 3

Changed DataLogger task priority from 2 to 3
Changed AlarmSystem task priority from 3 to 2

TempSensor task priority: 1
DataLogger task priority: 3
AlarmSystem task priority: 2
```

Quality Test Report

By TeamOS

TOUGH DECISIONS

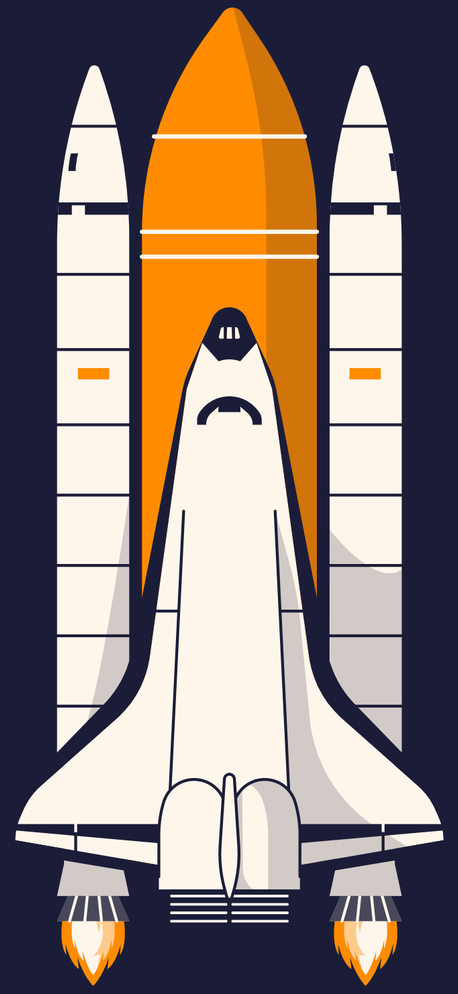
1. Dividing API -> Split and Help technique
2. FreeRTOS for MacOS -> UBUNTU Virtual Machine + POSIX Simulator
3. Tracking a progress -> Created progress tracker in Google Docs
4. Imbalanced skills within a team -> Do as much as you can



APIs Name	Status	Notes
xTaskNotify()	In progress	-
xTaskGetTickCount()	Approved	-
taskDISABLE_INTERRUPTS)	In progress	Work with ‘taskENABLE_INTERRUPTS()’
vTaskGetRunTimeStats()	Under review	Ok with the config and compiling but result is only showing 1,2 rep
xTaskGetSchedulerState()	Approved	-
uxTaskGetStackHighWaterMark()	Approved	-
eTaskGetState()	Approved	-
uxTaskGetSystemState()	Approved	#define config USE_TRACE_FACILITY 1
vTaskList()	In progress	-

CHALLENGE & SOLUTION

Challenge	Solution
1 How to test functionality of exact API?	1 Print message after API performed its functionality
2 Multiple Main Function	2 Test one-by-one or create functions and 1 main
3 Untestable 3 APIs	3 Documented the reason in QTR (FreeRTOS MPU)



PURPOSE ACCOMPLISHMENT

Purpose

1 Help SpaceX to Secure to secure the OS quality

1 Imitate real space shuttle functionality

How accomplished

2 Tested all testable 52 APIs with simple or functional test

2 Will see during the LIVE DEMO (Success)

CONCLUSION & SUGGESTIONS

Conclusion

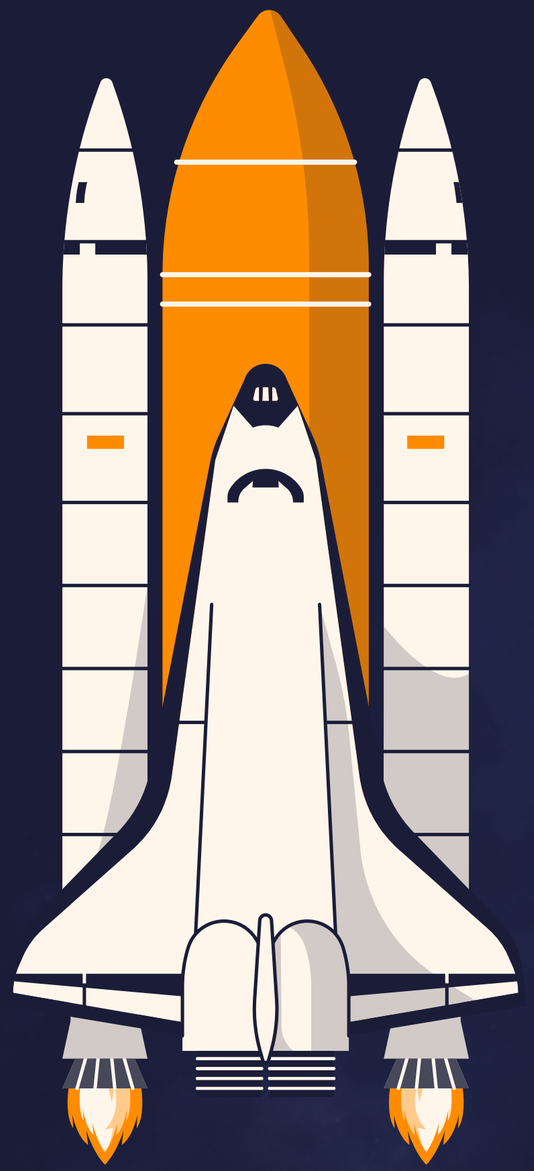
- Reached the goal of testing all APIs
- Accomplished purpose by conducting scenario-based tests
- Significantly improved skills in C
- Mastered FreeRTOS
- Nobody of our team left behind

Suggestions

- Use FreeRTOS MPU for test 3 untestable APIs
- Conduct more comprehensive functional tests
- Test performance using third-party services

OUTCOME WITH 2 MORE WEEKS

1. Conduct functional tests on all 55 APIs
2. Setup FreeRTOS MPU with hardware for testing 3 API
3. Develop more programs imitating real spacecraft
4. Use test benchmarks to increase volume of test results



THANK YOU FOR ATTENTION!

LIVE DEMO TIME