

OS'25 Project

PART 0: ROADMAP



Agenda

- PART 0: ROADMAP
- PART I: PREREQUISITES
- PART II: GROUP MODULES
- PART III: INDIVIDUAL MODULES
- PART IV: OVERALL TESTING & BONUSES

Agenda

- **PART 0: ROADMAP**

- Registration
- Project Overview
- Supportive Materials
- Logistics & Delivery
- Advices

Registration

- Register your team info [4~6 members] at [**this form**](#)
- Due: FRI 17 OCT 23:59 isA
- Each member **MUST** select different **individual** module in **their order**
- Reminder: “**CAREFULLY SELECT TEAM - WORK AS A TEAM...**”

Project Overview

10
MARKS

[TW] Prerequisite Modules
Dyn. Alloc, Kern Heap, Fault Handl. 1 Placement

10
MARKS

[M#1]
Fault Handl. 2
(Replacement)

[M#2]
User Heap

[M#3]
Shared Memory

[M#4]
CPU Sched.

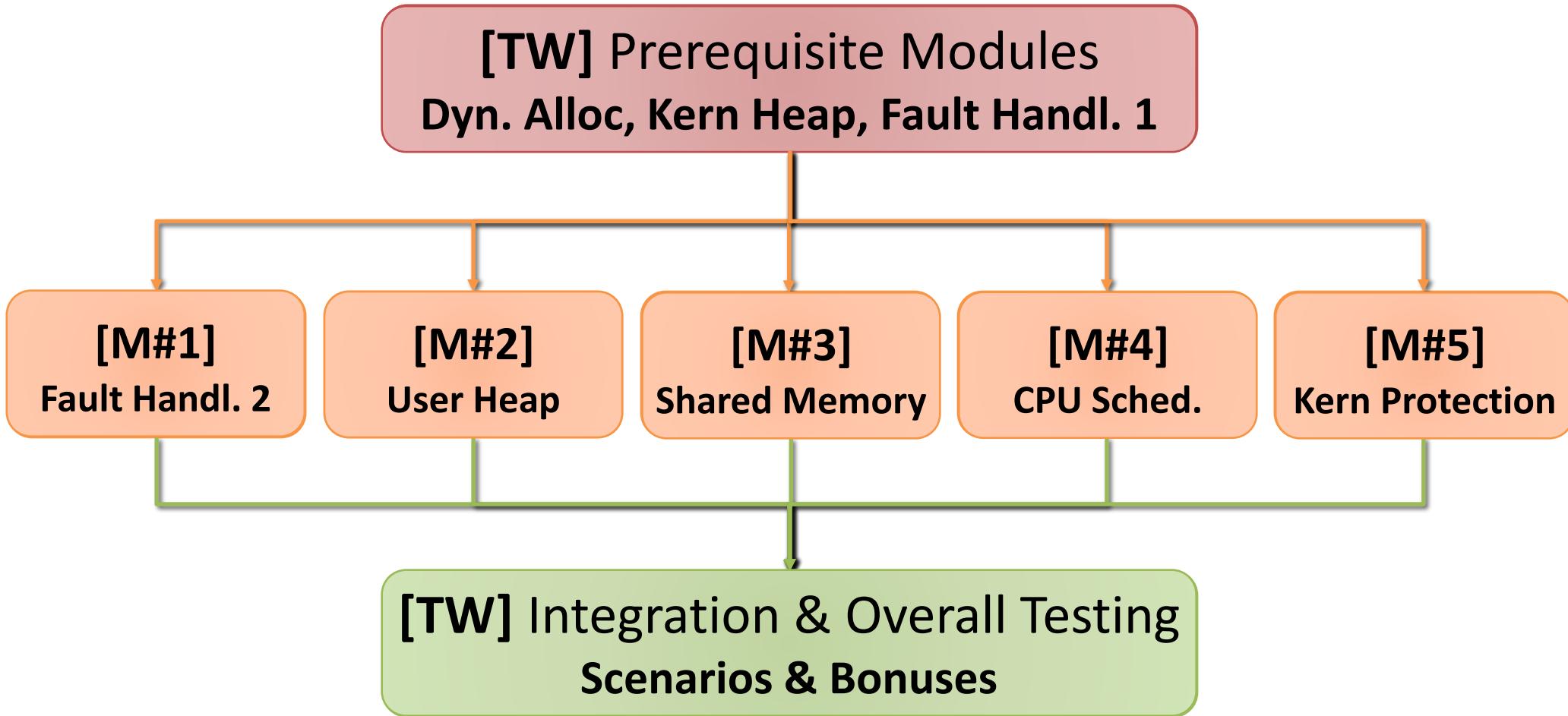
[M#5]
Kern Protection

5
MARKS

[TW] Integration & Overall Testing
Scenarios & Bonuses

Project Overview 4 members

10
MARKS



5
MARKS

Project Overview **6 members**

**10
MARKS**

[TW] Prerequisite Modules
Dyn. Alloc, Kern Heap, Fault Handl. 1

**10
MARKS**

[M#1]
Fault Handl. 2

[M#2]
User Heap

[M#3]
Shared Memory

[M#4]
CPU Sched.

[M#5]
Kern Protection

[M#6]
F. Hand 3

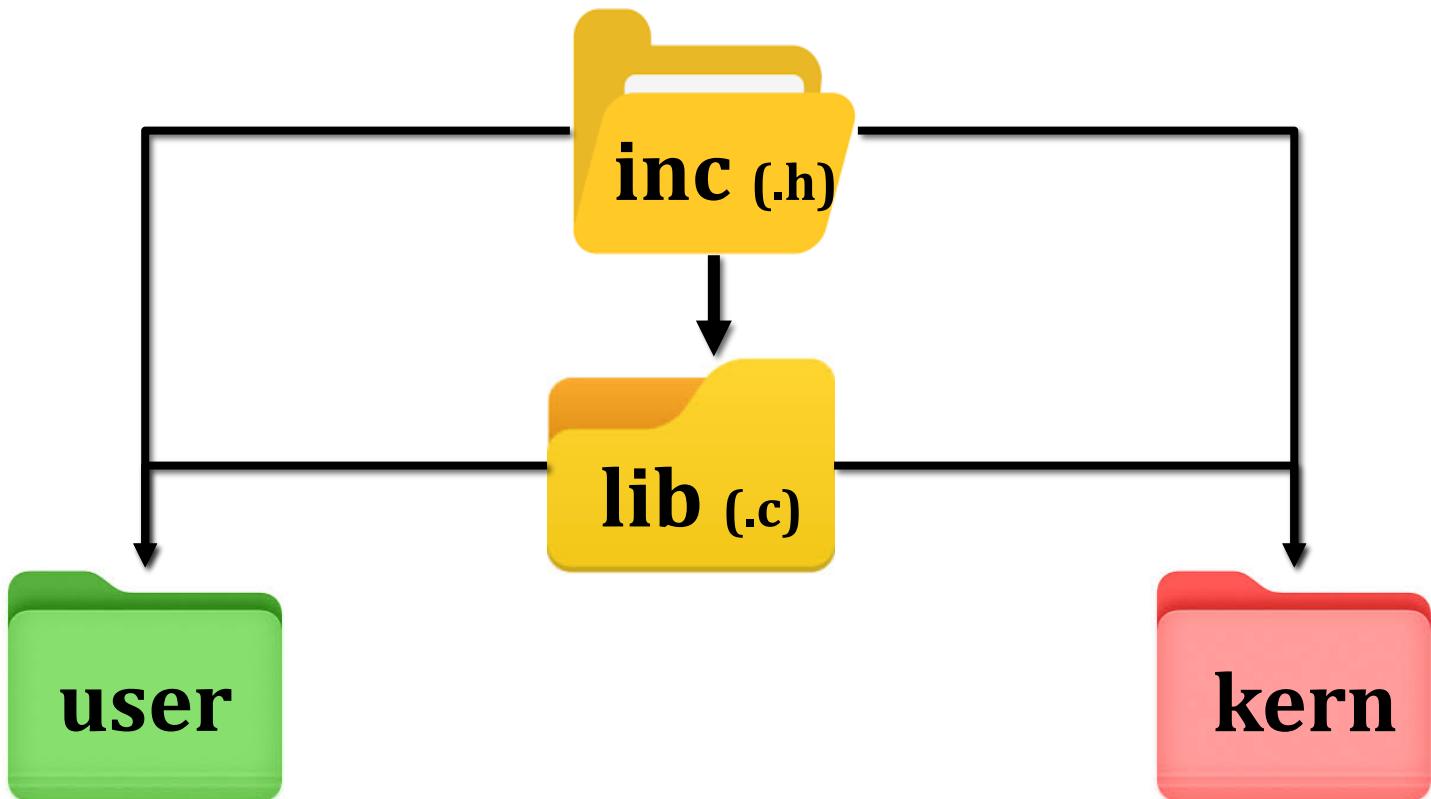
(Rep.)

(Rep.)

**5
MARKS**

[TW] Integration & Overall Testing
Scenarios & Bonuses

PROJECT BIG PICTURE



FOS_PROJECT_2022_TEMPLATE	
> Includes	
> boot	
> conf	
> inc	
> kern	
> cmd	Kern Protect.
> conc	
> cons	
> cpu	CPU Sched.
> disk	
> mem	Kheap, Share
> proc	
> tests	
> trap	Fault Handler
> entry.S	
> init.c	
> COPYRIGHT	
> kernel.id	
> Makefrag	
> lib	Dyn Alloc & UHeap
> user	

Agenda

- **PART 0: ROADMAP**

- Registration
- Project Overview
- **Supportive Materials**
- Logistics & Delivery
- Advices

Supportive Materials – **CODES**

APPENDICES:

- 1. STRING HELPER FUNCTIONS**
- 2. LISTS HELPER MACROS**
- 3. ENTRY MANIPULATION in TABLES and DIRECTORY**
- 4. MEMORY MANAGEMENT FUNCTIONS**
- 5. SHARED MEMORY STRUCTURE & HELPER FUNCTIONS**
- 6. PAGE FILE HELPER FUNCTIONS**
- 7. WORKING SET STRUCTURE & HELPER FUNCTIONS**
- 8. SCHEDULER STRUCTURE & HELPER FUNCTIONS**
- 9. PROTECTION PRIMITIVES**
- 10. READY-MADE COMMANDS**

Supportive Materials – **VIDEOS**

- How to import a new project into eclipse? [\[link\]](#)
- Debugging:
 1. Debug KERN Code via **breakpoints** (ECLIPSE) [\[link\]](#)
 2. Debug USER Code via **breakpoints** (ECLIPSE) [\[link\]](#)
 3. Debug any code via **printing** [\[link\]](#): 1st minute]
 4. Locate the line causing exception via **disassembly** [\[link\]](#)

Supportive Materials – TESTING

- Set of **modular tests** for each module [**seen** & **unseen**]
- Helps to discover some bugs

Successful Test ↳ 100% Correctness

Be LOGIC-DRIVEN... Not TEST-DRIVEN

Tests will be released later isA

Where should I write the Code?

There're shortcut links that direct you to the function definition

[1] Click on “Tasks” Tab

[2] Double Click on the required function

The screenshot shows a code editor interface with a top navigation bar containing tabs: Problems, Tasks, Console, Properties, Progress, and Search. The Tasks tab is highlighted with a red circle and a green arrow pointing to it. Below the navigation bar, a message '20 items' is displayed. A table lists 20 TODO items under the 'Description' column. The first five items are grouped together with an orange rounded rectangle and labeled 'GIVEN'. The next five items are grouped together with a red rounded rectangle and labeled 'REQUIRED'. The remaining ten items are listed individually. The right side of the interface shows a list of resources with their paths, line numbers, locations, and types.

Resource	Path	Location	Type
ubuddy.h	/Project2020_T...	line 7	C/C++ Task
ubuddy.h	/Project2020_T...	line 39	C/C++ Task
ubuddy.h	/Project2020_T...	line 20	C/C++ Task
environment_def...	/Project2020_T...	line 147	C/C++ Task
environment_def...	/Project2020_T...	line 30	C/C++ Task
trap.c	/Project2020_T...	line 486	C/C++ Task
memory_manag...	/Project2020_T...	line 770	C/C++ Task
uheap.c	/Project2020_T...	line 48	C/C++ Task
memory_manag...	/Project2020_T...	line 781	C/C++ Task

[3] Function body, at which you should write the code

The screenshot shows a code editor displaying a C function named 'allocateMem'. The code includes several TODO comments. Line 766 has a green arrow pointing to it with the text '[1] allocateMem'. Line 768 starts the function definition with 'void allocateMem(struct Env* e, uint32 virtual_address, uint32 size)'. Line 770 contains a TODO comment: '//TODO: [PROJECT 2020 - [2] User Heap] allocateMem() [Kernel Side]'. Line 771 contains a note: '// Write your code here, remove the panic and write your code'. Line 772 contains a panic statement: 'panic("allocateMem() is not implemented yet...!!");'. Lines 774 and 775 provide additional instructions: '//This function should allocate ALL pages of the required range in the PAGE FILE' and '//and allocate NOTHING in the main memory'. Line 776 ends the function definition with a closing brace '}'.

Agenda

- **PART 0: ROADMAP**

- Registration
- Project Overview
- Supportive Materials
- **Logistics & Delivery**
- Advices

Logistics

Startup Code:

- FOS_PROJECT_2025_template.zip
- Follow [these steps](#) to import the project folder into the eclipse

Delivery Method: **GOOGLE FORM**

- It's FINAL delivery
- MUST deliver the required tasks and ENSURE they're worked correctly

Delivery Dates:

- **Week #12 (BEG of DECEMBER) is A**
- Upload your code EARLY as NO EXCEPTION will be accepted.

Support:

- The support for teams will be through their **MENTORS ONLY (+Lecturer)** during via:
 1. MAIN METHOD: [weekly office hours](#).
 2. SECONDARY METHOD [OPTIONAL]: [other contact method](#)

Submission Rules

Read the following instructions as the code correction is done AUTOMATICALLY. Any violation in these rules will lead to 0 and, in this case, nothing could be happened.

First ensure the following that (READ CAREFULLY):

- You tested each function in a **FRESH RUN** and a congratulations message have been appeared.
- **NO CODE with errors WILL BE CORRECTED.** So, CLEAN & RUN your project several times before your submission.
- You submitted **BEFORE** the deadline by several hours to **AVOID** any internet problems.
- **DEADLINE: Week #12 (BEG of DECEMBER) isA**
- **NO DELAYED submissions WILL BE ACCEPTED.**
- **ONLY ONE person** from the team shall submit the code.
- The **TEAM # MUST BE CORRECT.**
- **DON'T take the FORM LINK FROM ANYONE.** OPEN the form from its **LINK ONLY**. Otherwise, your submission is AUTOMATICALLY CANCELLED by GOOGLE.
- You **MUST RECEIVE A MAIL FROM GOOGLE** with your submission after clicking submit. If nothing received, re-submit again to consider your submission.

Submission Steps

STEPS to SUBMIT:

- Step 1: Clean & run your code the last time to ensure that there are any errors.
- Step 2: Create a new folder and name it by your team number **ONLY**. Example **1** or **95**. [**ANY extra chars will lead to 0**].
- Step 3: **DELETE** the “obj” folder from the “FOS_PROJECT_2025_Template”
- Step 4: PASTE the “FOS_PROJECT_2025_Template” in the folder created in step #2.
- Step 5: Zip the created new folder. Its name shall be like **[num of your team.zip]**. [**ANY extra chars will lead to ZERO**].
- Step 6: Open the form from [**HERE**](#).
- Step 7: Fill your team's info .. Any wrong information will cancel your submission, revise them well.
- Step 8: Upload the zipped folder in step 5 to the form in its field.
- Step 9: MUST RECEIVE A MAIL from GOOGLE with your submission, otherwise re-submit again.



ADVICE#1: CAREFULLY SELECT TEAM

(ALL should learn...ALL should work)

ADVICE#2: WORK AS A TEAM

(TW Module: Avg # Tasks / Member \approx 2~3)

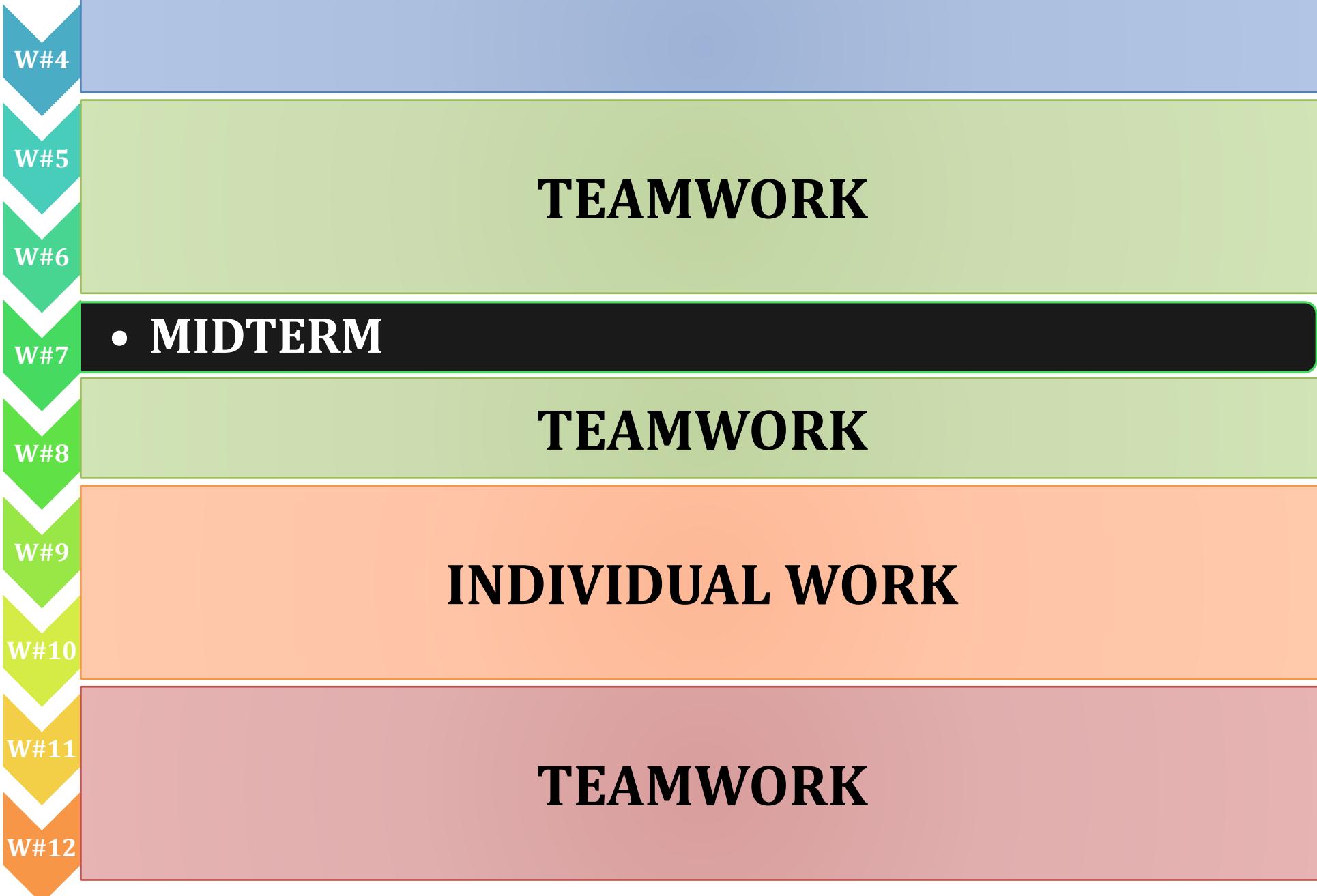
ADVICE#3: START IMMEDIATELY

(To resolve dependencies AND chance to understand, ask & fix bugs)

ADVICE#4: MUST READ MATERIALS

(Detailed steps AND helper ready-made functions)

Time Plan

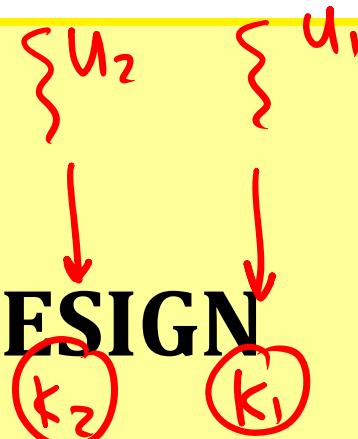


CAUTION

During your solution, any **SHARED** data **MUST** be
PROTECTED by critical section via **LOCKS**

1 - 1 *Thread*
ACTION

CORRECTNESS by DESIGN



Be **LOGIC-DRIVEN...** Not **TEST-DRIVEN**

LET'S START OUR JOURNEY...

☺ Enjoy **developing** your **own OS** ☺

