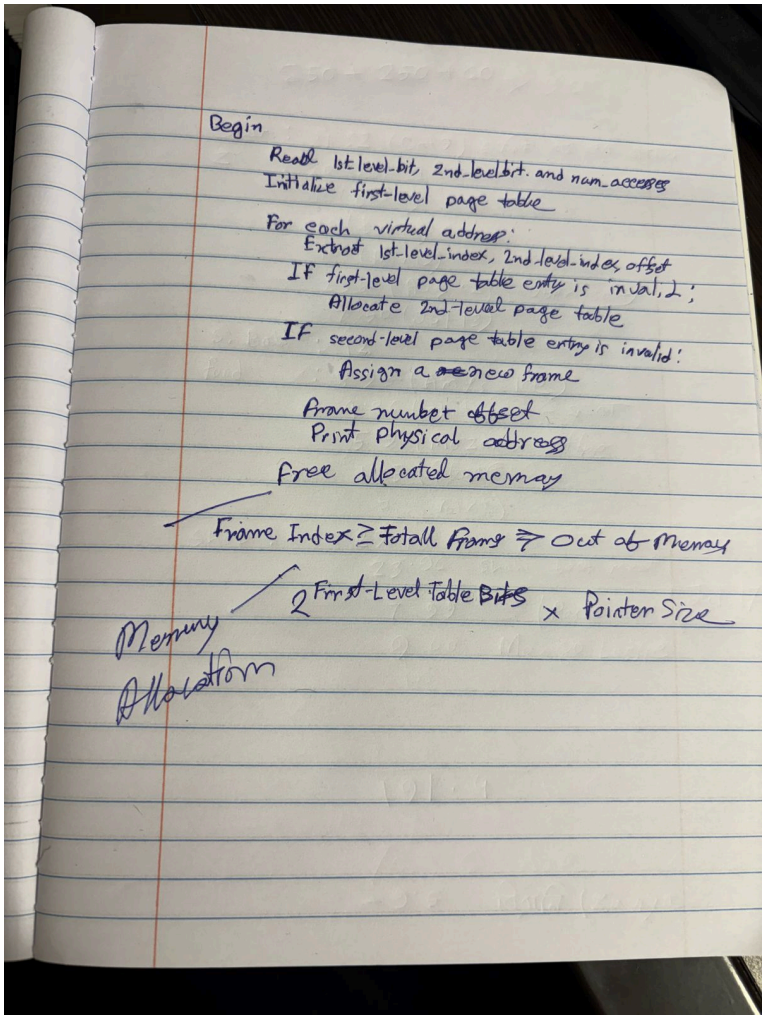
Planning :


Begin
Read 1st_level_bit, 2nd_level_bit, and num_accesses
Initialize first-level page table
For each virtual address:
Extract 1st_level_index, 2nd_level_index, offset
IF first-level page table entry is invalid:
Allocate 2nd-level page table
IF second-level page table entry is invalid:
Assign a new frame
Frame number
Print physical address
free allocated memory
Frame Index ≥ Total Frame ⇒ Out of Memory
2 First-Level Table × Pointer Size
Memory
Allocation

Breakdown:

Access #	Virtual Address (Hex)	1st Level Index	2nd Level Index	Offset	Frame Assigned	Physical Address (Hex)
1	**0x1A34**	1	10	0x34	0	**0x034**
2	**0x1A56**	1	10	0x56	0	**0x056**
3	**0x2B78**	2	11	0x78	1	**0x178**
4	**0x2B12**	2	11	0x12	1	**0x112**
5	**0x3C9A**	3	12	0x9A	2	**0x29A**
6	**0x3D4F**	3	13	0x4F	3	**0x34F**
7	**0x4E21**	4	14	0x21	4	**0x421**
8	**0x5F3C**	5	15	0x3C	5	**0x53C**

Access 1: Virtual Address = 0x1A34

Before Access:

- **1st Level Index = 1**
- **2nd Level Index = 10**
- **Offset = 0x34**
- **No second-level page table exists for index 1.**
- **Allocate a new second-level page table at index 1.**
- **Assign Frame 0.**

After Access:

- **Frame 0 assigned to virtual address 0x1A34.**
- **Physical Address: 0x034.**

Access 2: Virtual Address = 0x1A56

Before Access:

- **1st Level Index = 1 (already exists).**
- **2nd Level Index = 10 (already allocated).**
- **Offset = 0x56.**
- **No new allocation needed.**

After Access:

- **Same frame (Frame 0) used.**
- **Physical Address: 0x056.**

Access 3: Virtual Address = 0x2B78

Before Access:

- **1st Level Index = 2 (new).**
- **2nd Level Index = 11 (new).**
- **Offset = 0x78.**
- **No second-level page table exists for index 2.**
- **Allocate a new second-level page table at index 2.**
- **Assign Frame 1.**

After Access:

- **Frame 1 assigned.**

- **Physical Address: 0x178.**

Access 4: Virtual Address = 0x2B12

Before Access:

- **1st Level Index = 2 (exists).**
- **2nd Level Index = 11 (exists).**
- **Offset = 0x12.**
- **No new allocation needed.**

After Access:

- **Same frame (Frame 1) used.**
- **Physical Address: 0x112.**

Access 5: Virtual Address = 0x3C9A

Before Access:

- **1st Level Index = 3 (new).**
- **2nd Level Index = 12 (new).**
- **Offset = 0x9A.**
- **No second-level page table exists for index 3.**
- **Allocate new second-level page table at index 3.**
- **Assign Frame 2.**

After Access:

- **Frame 2 assigned.**
- **Physical Address: 0x29A.**

Access 6: Virtual Address = 0x3D4F

Before Access:

- **1st Level Index = 3 (exists).**
- **2nd Level Index = 13 (new).**
- **Offset = 0x4F.**
- **Allocate new second-level entry.**
- **Assign Frame 3.**

After Access:

- **Frame 3 assigned.**
- **Physical Address: 0x34F.**

Access 7: Virtual Address = 0x4E21

Before Access:

- **1st Level Index = 4 (new).**
- **2nd Level Index = 14 (new).**
- **Offset = 0x21.**
- **Allocate new second-level page table at index 4.**
- **Assign Frame 4.**

After Access:

- **Frame 4 assigned.**
- **Physical Address: 0x421.**

Access 8: Virtual Address = 0x5F3C

Before Access:

- **1st Level Index = 5 (new).**
- **2nd Level Index = 15 (new).**
- **Offset = 0x3C.**
- **Allocate new second-level page table at index 5.**
- **Assign Frame 5.**

After Access:

- **Frame 5 assigned.**
- **Physical Address: 0x53C.**

Follow Up :

1. 3 days. NA
2. Multilevel page tables and how memory is translated from virtual to physical addresses. Memory management techniques, including how page tables are allocated dynamically. The importance of planning before coding, especially for complex data structures.
3. Handling dynamic memory allocation and Ensuring correctness in address translation
4. Step:
    a. Planning: I first outlined the necessary data structures and how memory translation should work.

b. Understanding Address Breakdown: I studied how to extract first-level index, second-level index, and offset from a given virtual address.

c. Implementing the Page Table: I wrote the code to allocate and manage first-level and second-level page tables dynamically.

d. Testing Different Access Patterns: I tested redundant, disparate, and mixed accesses to verify correctness.

e. Tracing and Documentation: I recorded before and after states of the page table for different access patterns.

5. Yes. error fix also for Understanding multilevel paging concepts.