

①

~~Therapy~~

Segmented Memory Mgmt.

Paging has one disadvantage

↓
It destroys the logical modularity
of a program.

↓
This is an issue when main memory
space is small (to be covered compre-
hensively in Virtual Memory).

To avoid breaking modularity,
we follow the segmented memory
mgmt technique.

Modular program

(2)

```
int Main()
```

```
{  
    sqst()
```

```
    sum()
```

```
    average()
```

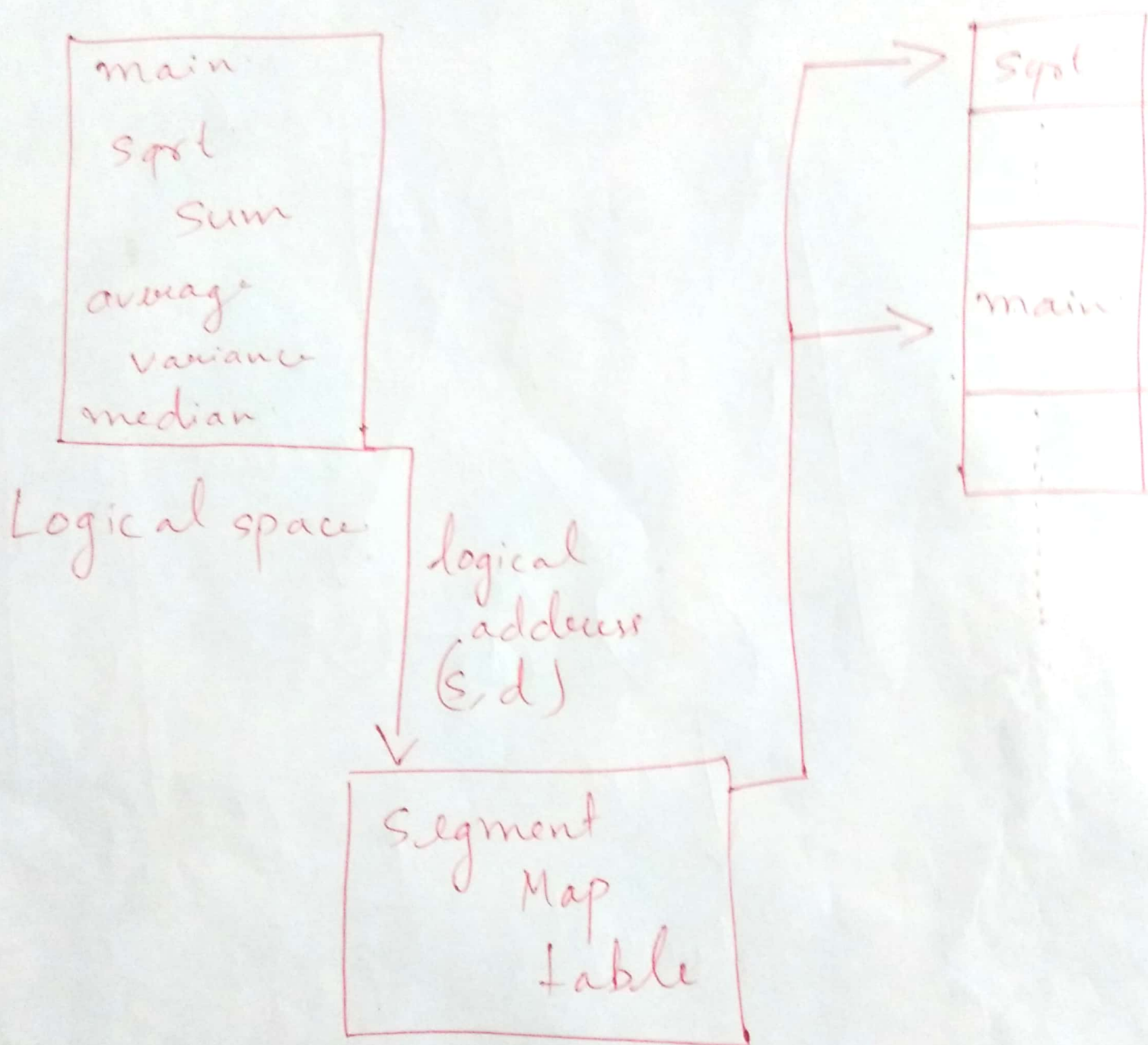
```
    variance()
```

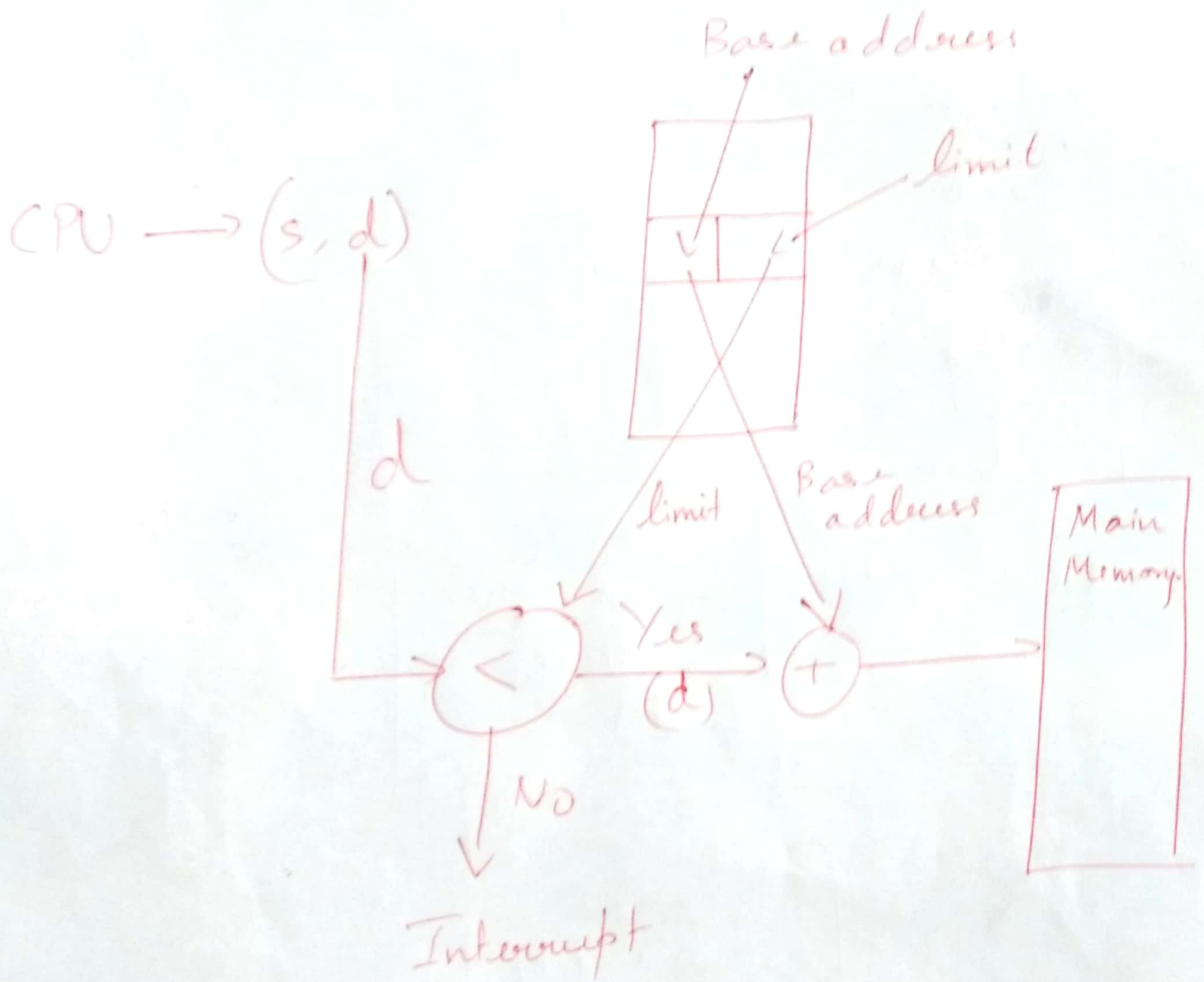
```
    median()  
}
```

All these
are
different modules.

We break the logical space into different partitions based on modules in the program.

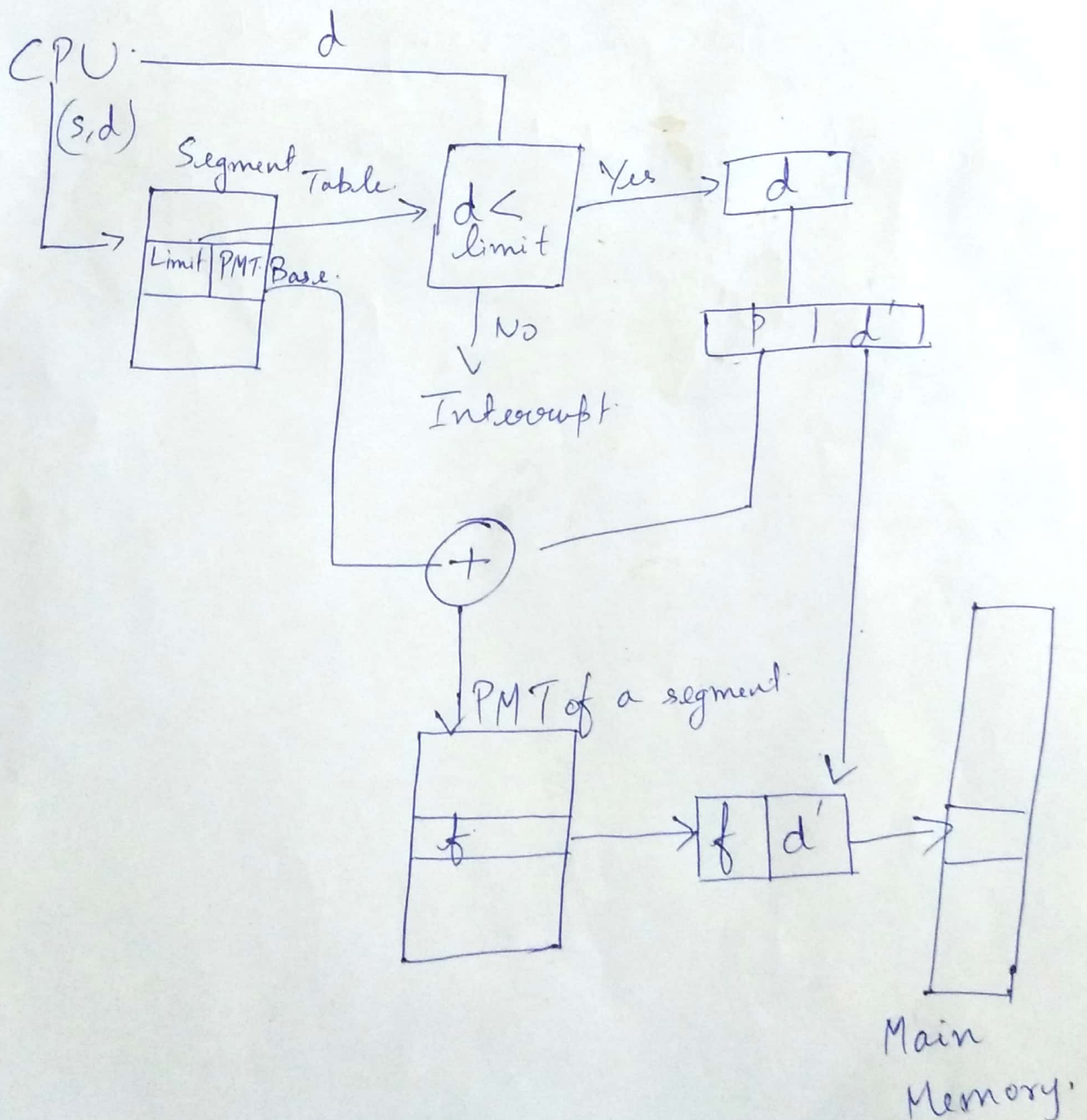
lets call all these partitions as segments.





Paged-Segmented Memory Mgmt.

→ Mixture of segmented and paged memory management.



Note: In PSM, the offset (d) is used to access the page number of the segment and the offset in the page.

We are, in this case, using both the ideas of segmentation and paging.

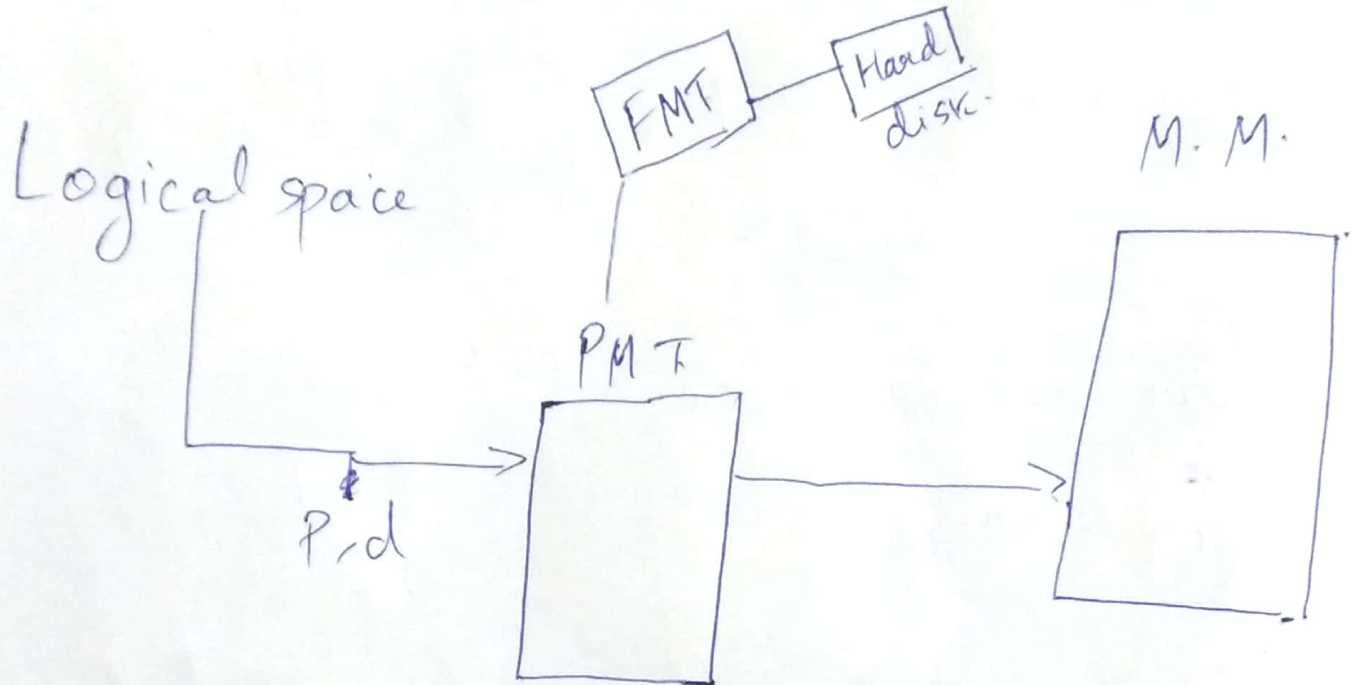
Demand Paging.

- Originally in all previous memory management technique all pages/segments must be available in the memory.
- However, a program could be sometimes larger than the main memory.

In that case, we use demand paging.

Core Idea: Load only one page and load other pages when they are needed.

Demand paging is also called as virtual memory management.



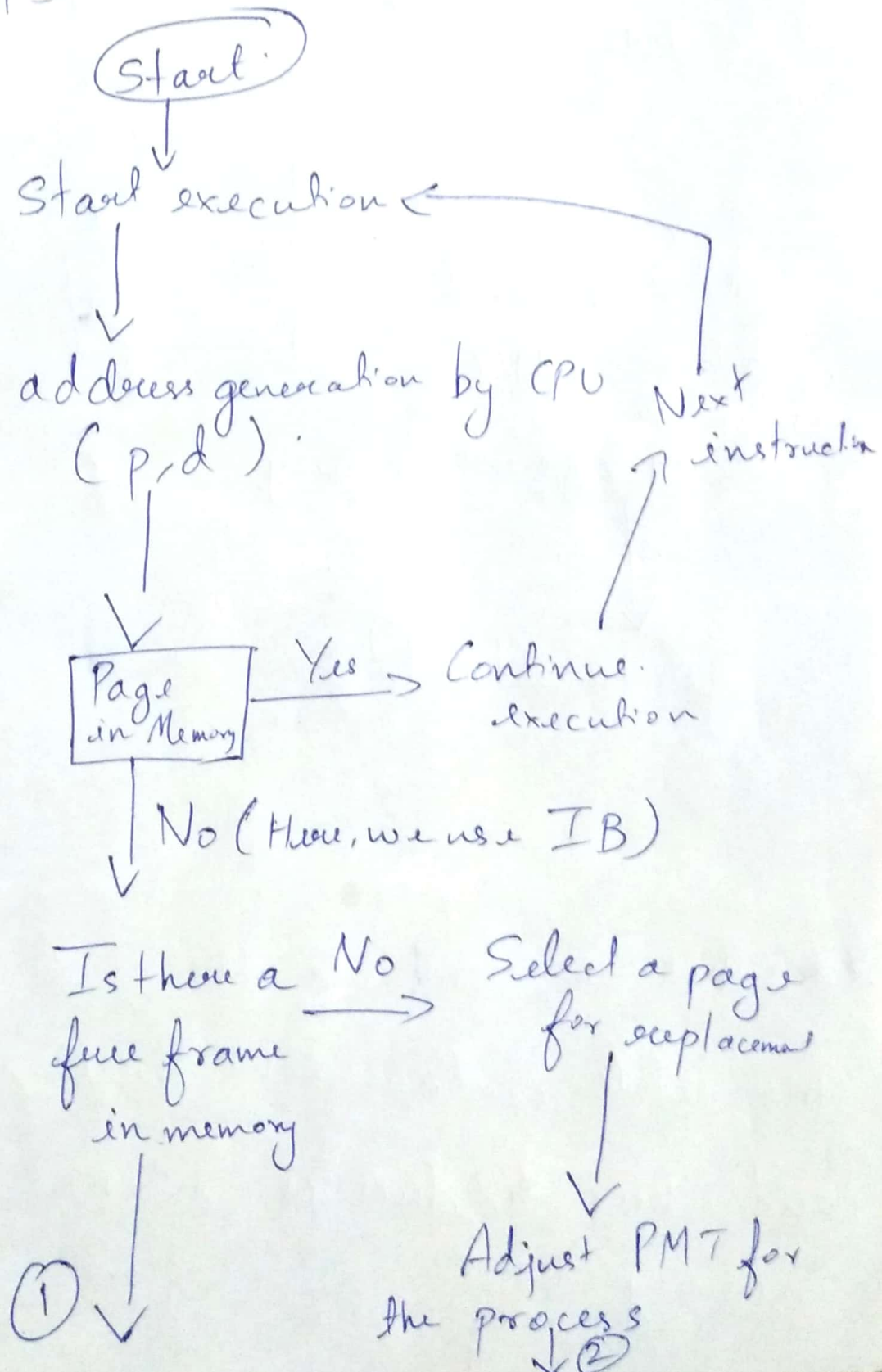
In demand paging, we also add one additional field (Interrupt bit)

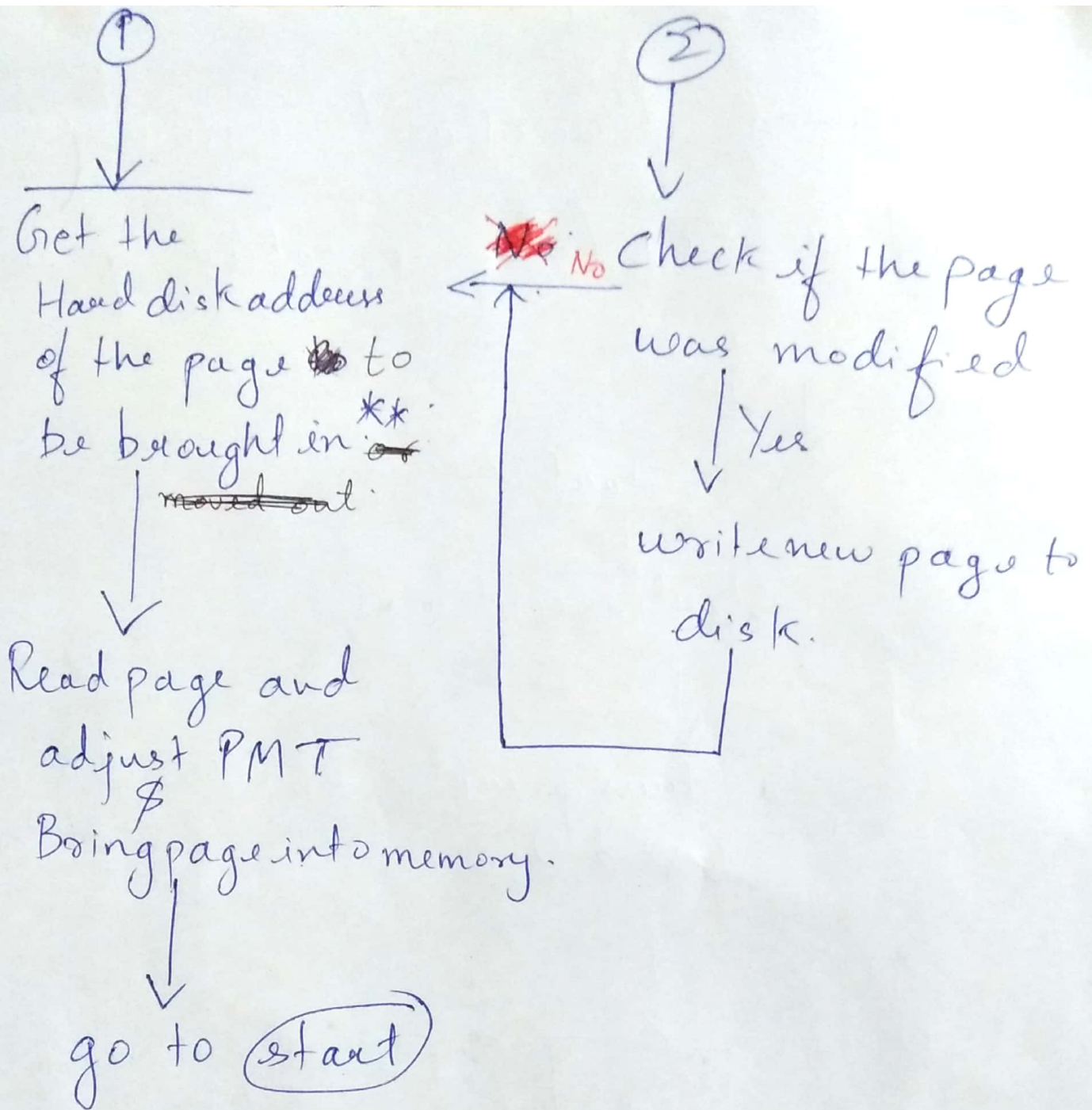
IB

If IB: 0 (page is in memory)

IB: 1 (page is not memory - it should be brought in)

The following flowchart summarizes the entire idea (It is taken from IIT kgp).





****** This is done via the File Map table. This table contains the info on pages and their addresses in the secondary memory.

Now, in the previous flow, we replaced pages.

Question : How to decide which page should be replaced??

Three techniques are used here :

- ① First in first out (page that was brought earlier must go).
- ② Optimal : Remove page that will be referenced very late
- ③ Least recently used technique:
Remove pages that were accessed recently.