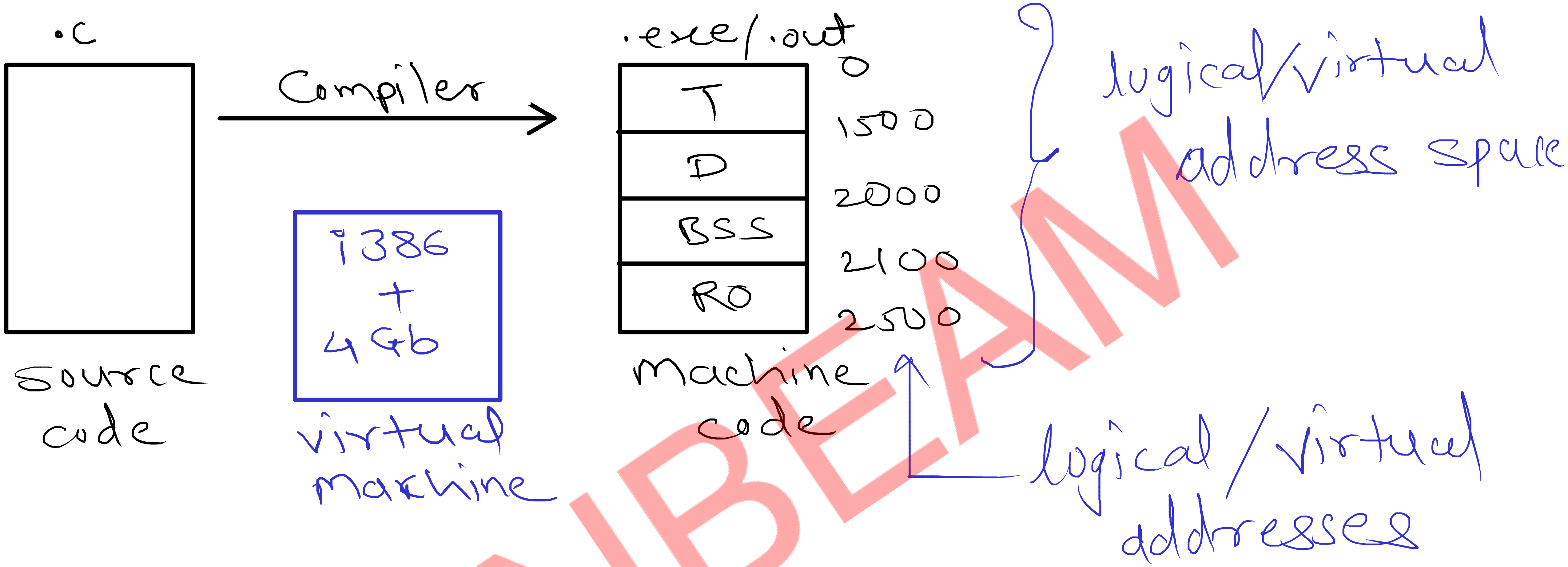
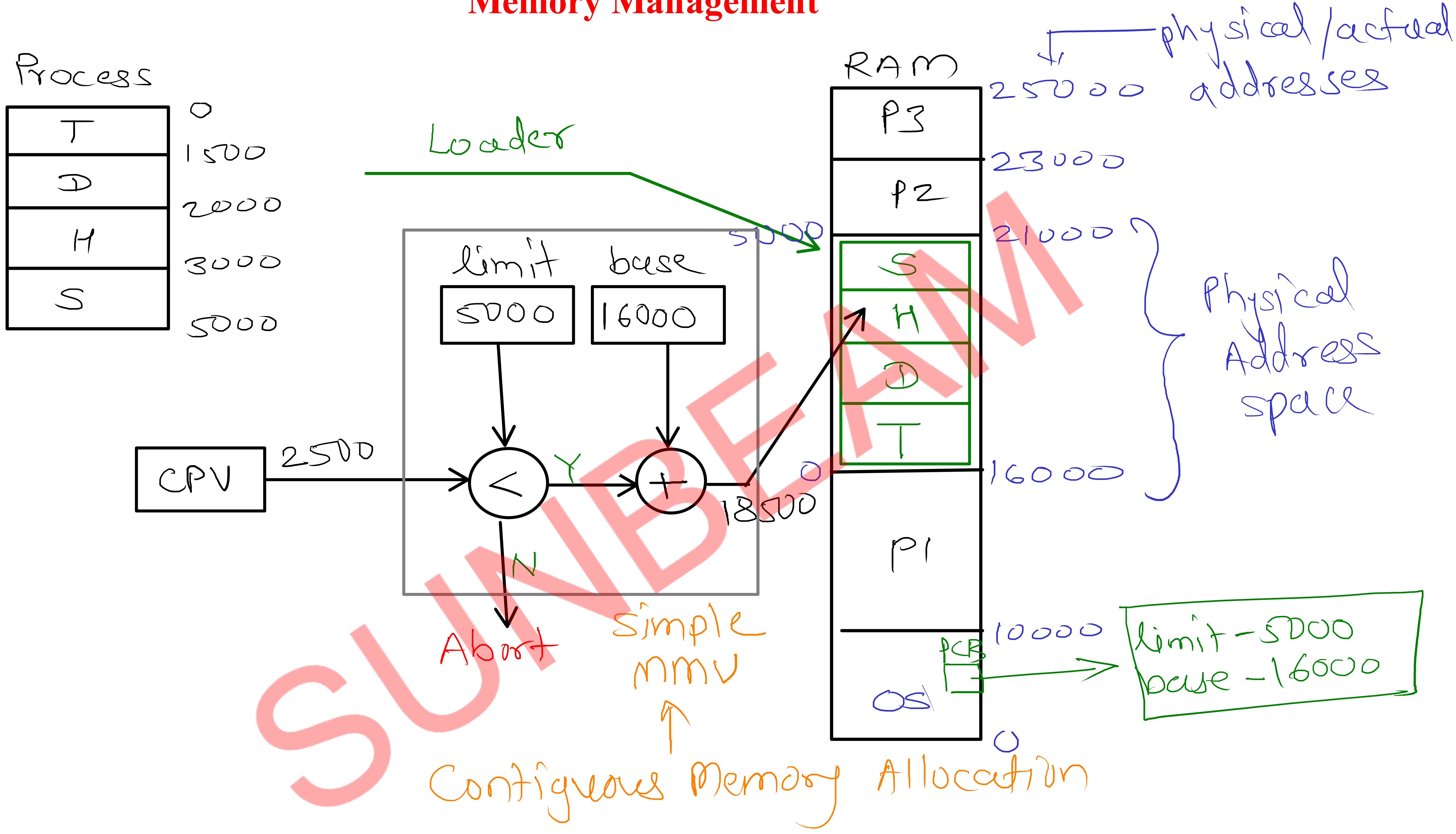
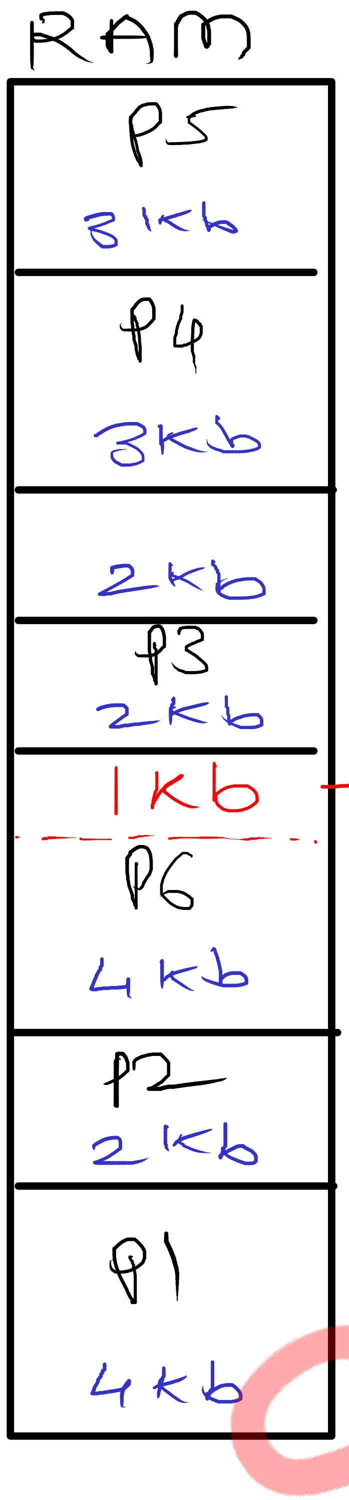
Memory Management



Memory Management



Fixed Partition



— Internal fragmentation

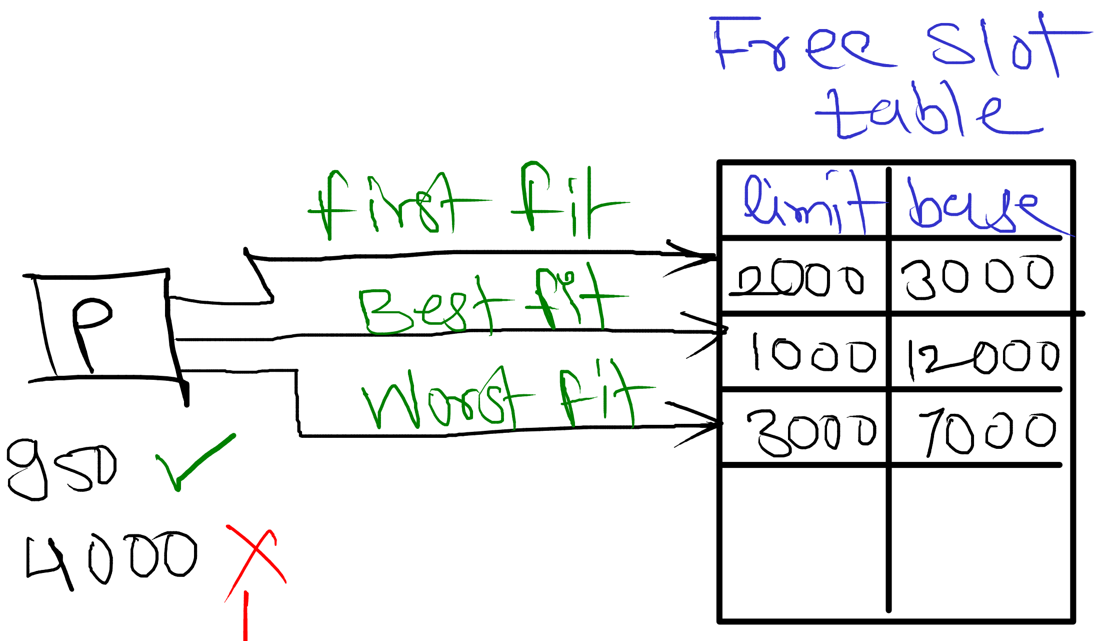
↳ if process is not utilizing complete partition which is allocated then some space is wasted

Limitations

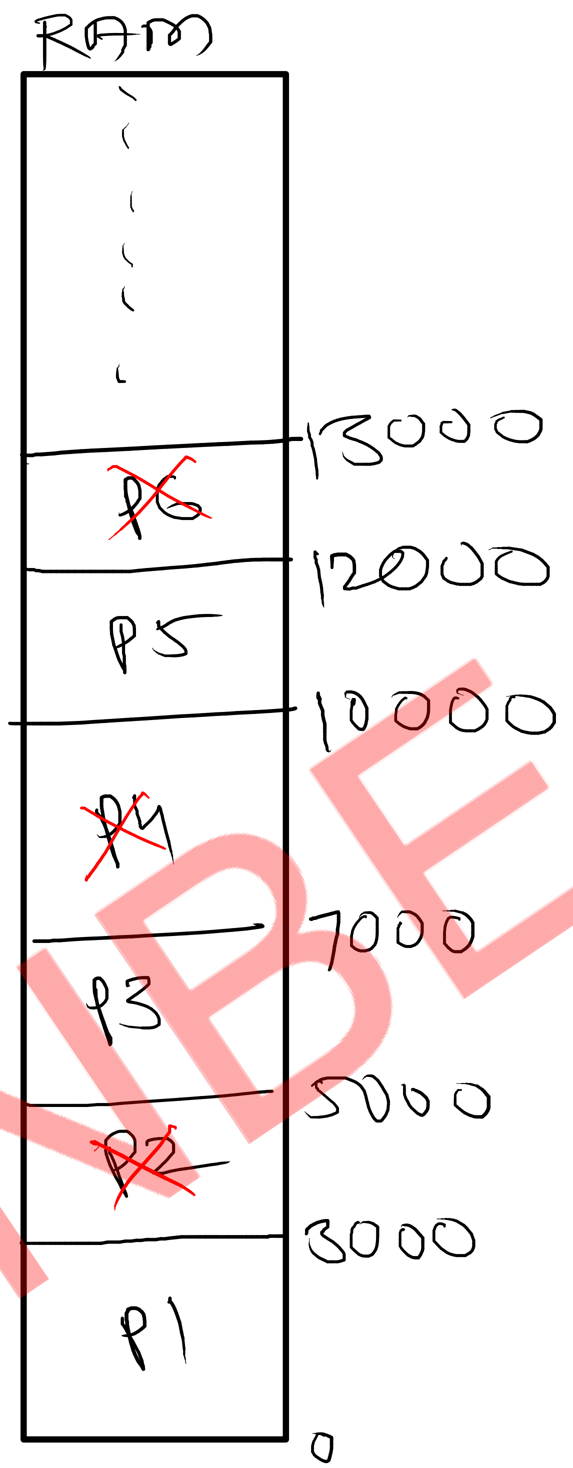
1) Max no. of processes = no. of partitions

2) Max size of processes = max. size of partition

Dynamic Partition



limit	base
2000	3000
1000	12000
3000	7000



External fragmentations

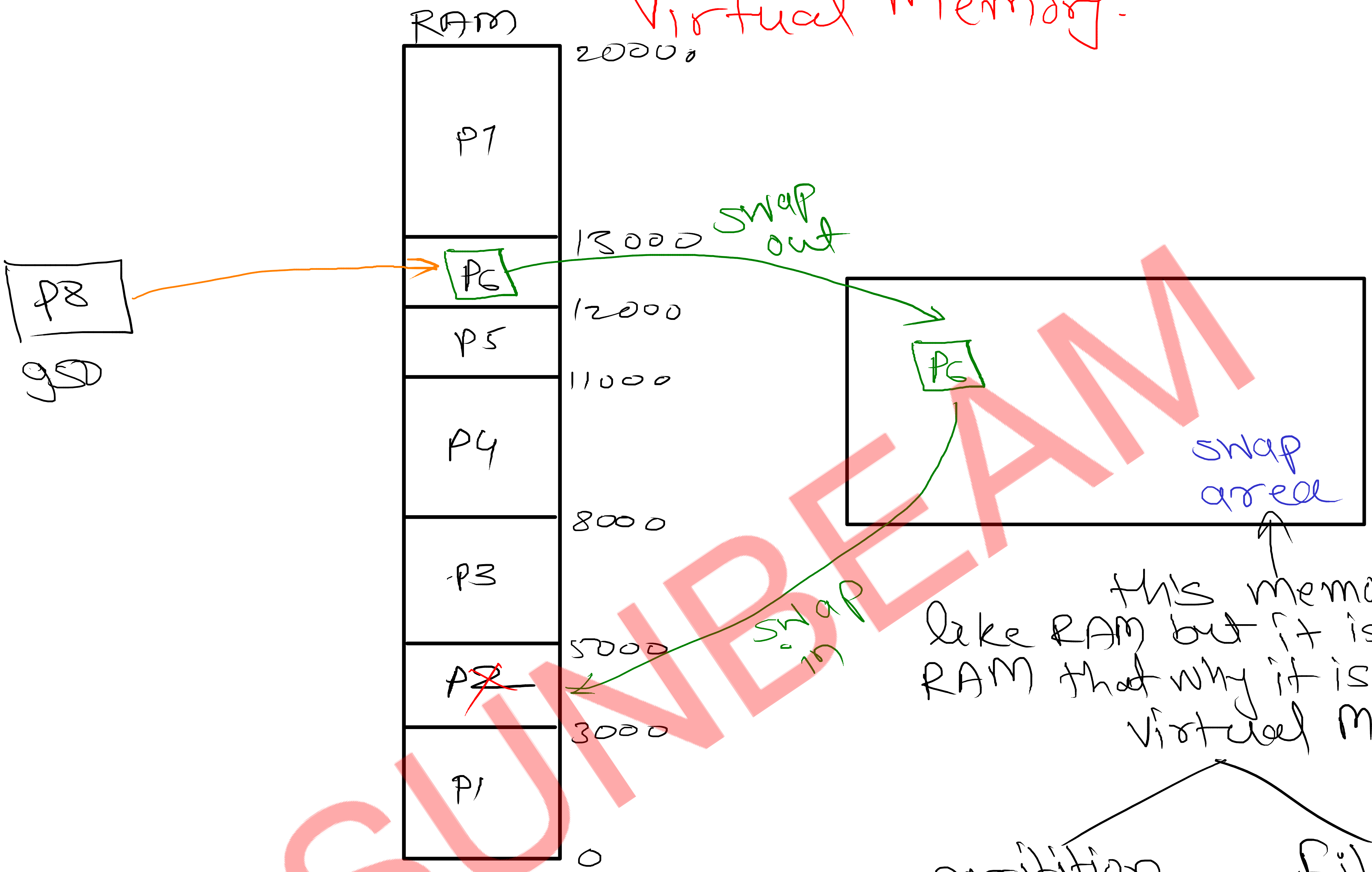
↓

due to unavailability of contiguous free space bigger process can not be loaded into memory

Compaction

↳ processes are moved into RAM to creat large contiguous free space

Virtual Memory.



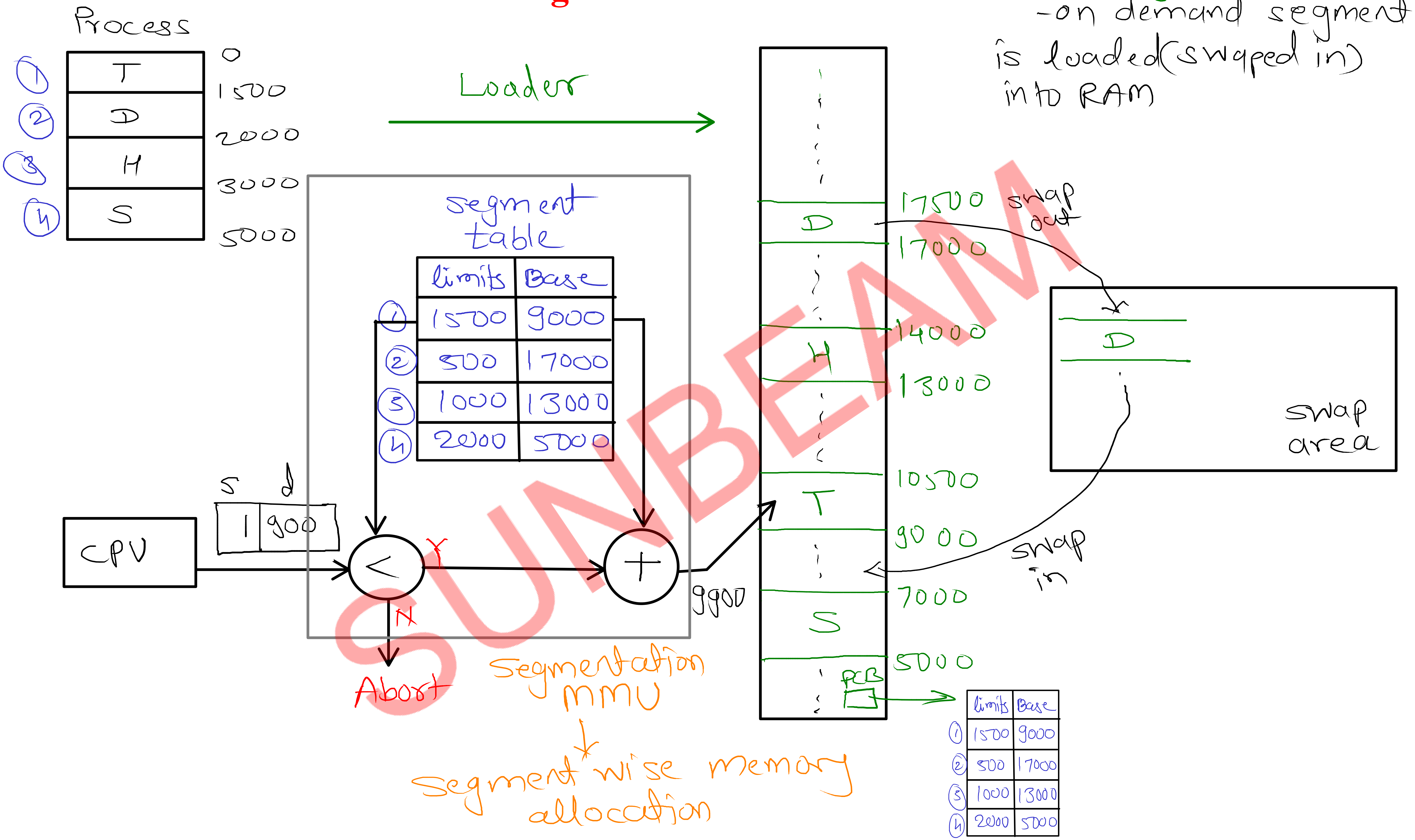
this memory looks like RAM but it is not actual RAM that why it is known as Virtual Memory

- paritition → linux
- file → windows → (pagefile.sys)

size = 2 * RAM size

Segmentation MMU

Demand segmentation
- on demand segment is loaded(swaped in) into RAM

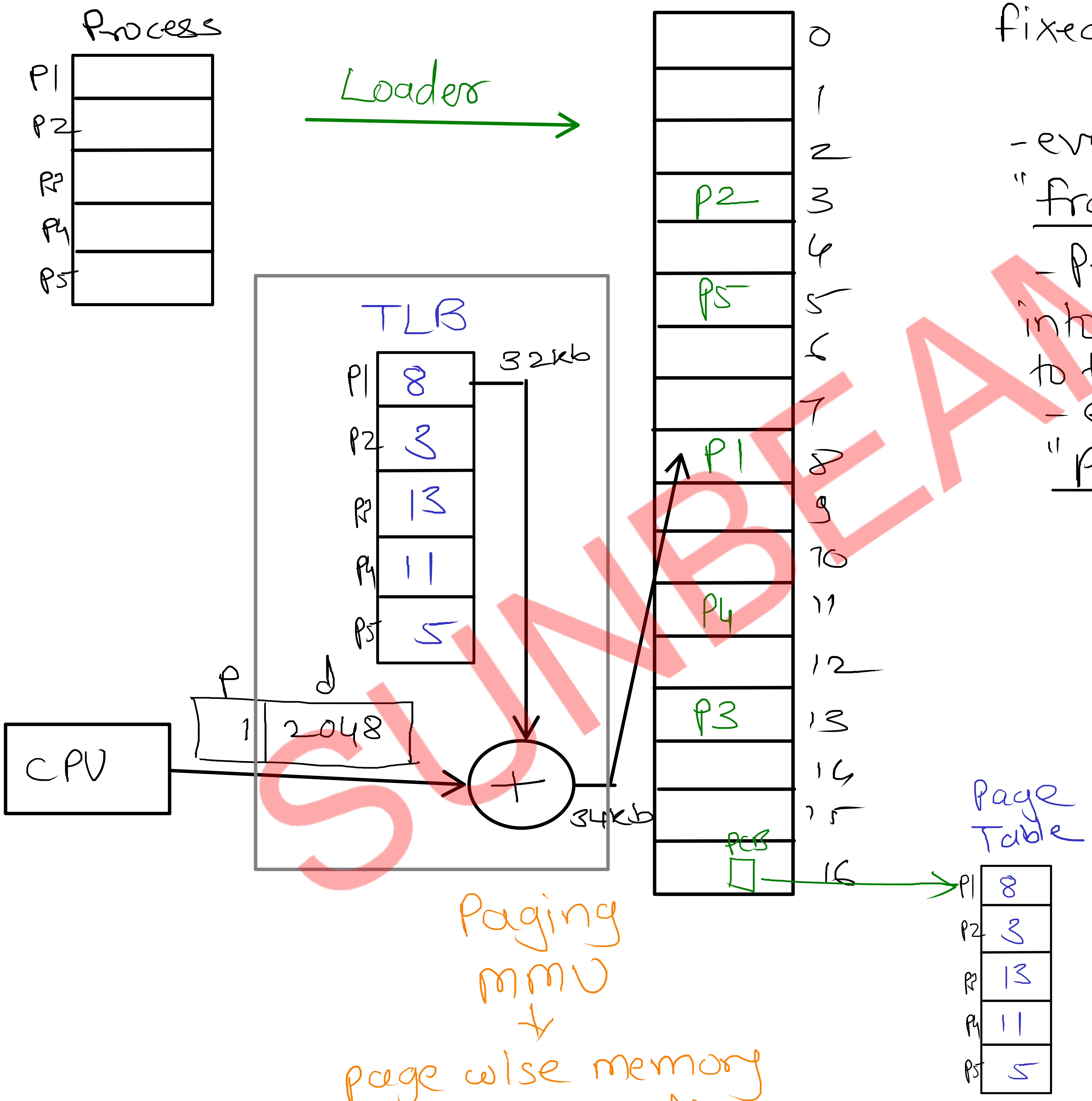


Segmentation MMU
↓
segment wise memory allocation

	limits	Base
1	1500	9000
2	500	17000
3	1000	13000
4	2000	5000

	limits	Base
1	1500	9000
2	500	17000
3	1000	13000
4	2000	5000

Paging MMU



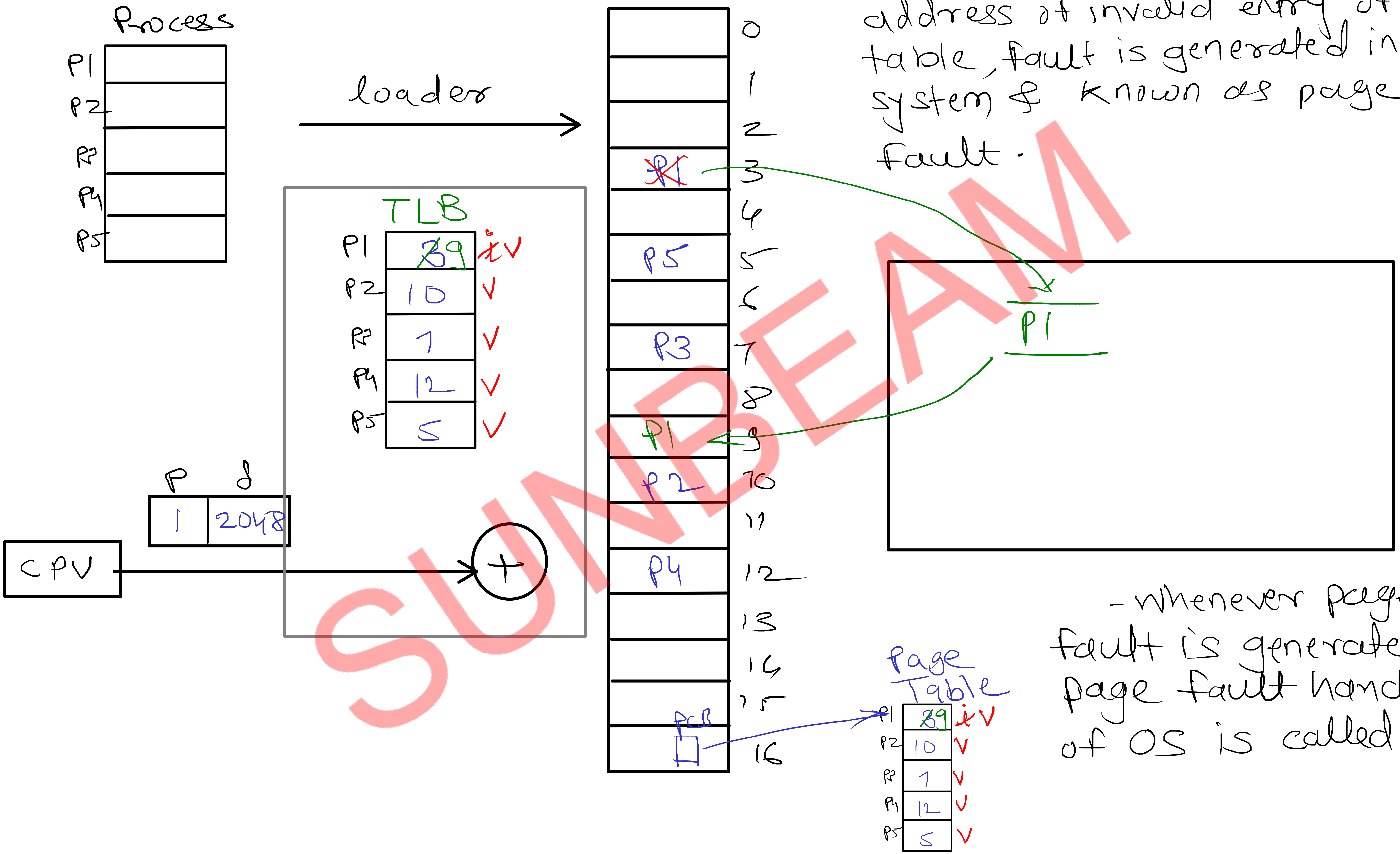
P1	8
P2	3
P3	13
P4	11
P5	5

Paging MMU
↓
page wise memory allocation

- RAM is divided into fixed & equal size partitions
  size = 4096 bytes (4 kb)
- every partition is known as "frame" / "physical page"
- Process is also divided into partitions of size equal to frame size
- every partition is known as "page" / "logical page"

Page fault:

- when CPU request for an address of invalid entry of page table, fault is generated into system & known as page fault.



TLB

P1	~~3~~9	~~i~~v
P2	10	v
P3	7	v
P4	12	v
P5	5	v

Page Table

P1	~~3~~9	~~i~~v
P2	10	v
P3	7	v
P4	12	v
P5	5	v

- whenever page fault is generated, page fault handler of OS is called.

```
page_fault_handler( ){
    1> check requested address is valid or not?
            if not valid abort the process
    2> check for read/write access
            if no access then abort the process
    3> find free frame to swap in the page which is
       on swap area
    4> swap in the requested page and update the
       entry of page table
    5> request the same address for which page
       fault was occured.
}
```

OS Booting

1) Power ON

ROM

2) POST / BIST — check for all hardware

3) Bootstrap loader - find bootable partition of harddisk

Boot sector

4) load & start bootloader - show menu to the users depending on user's choice load bootstrap program

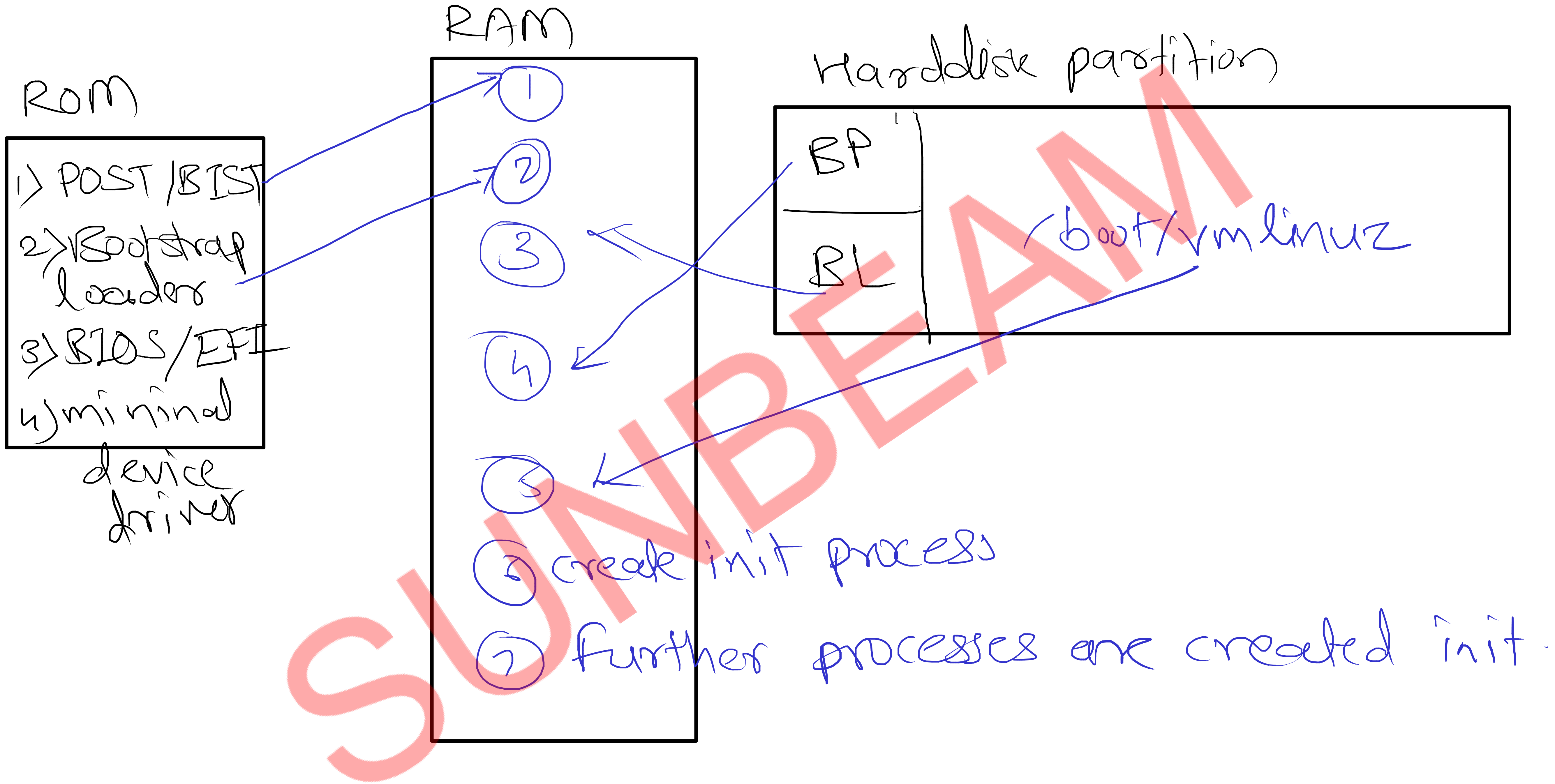
5) Bootstrap program → load kernel image into RAM

6) self extraction of kernel

7) start init / systemd process (first process)

8) remaining processes are created by this process and also it starts all the services

Booting Process



Types of kernel

1) Monolithik kernel — BSD UNIX
2) Micro kernel — Symbian, MACH
3) Modular kernel — Windows
4) Hybrid kernel — iOS — Darwin = BSD UNIX + MACH
5) Nano kernel — FreeRTOS

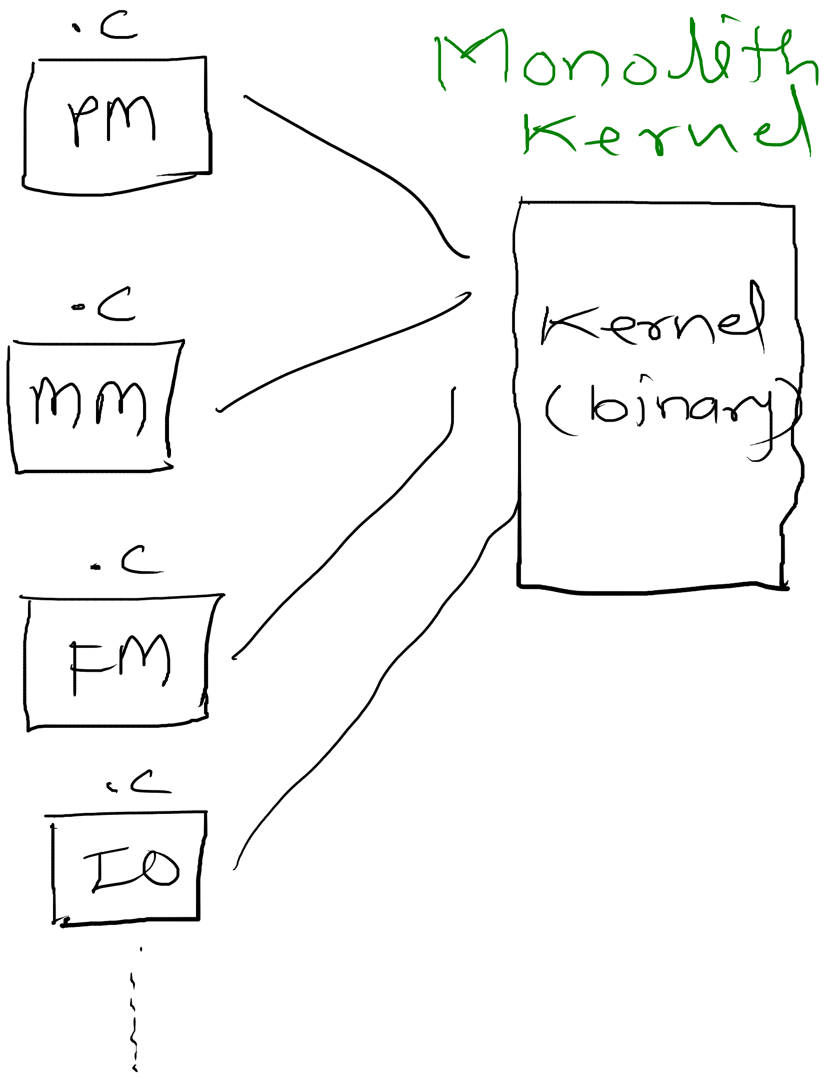
Linux Kernel = static component + Dynamic Component + xServer for UI

static component:
MM, PM, HAL, Sched, System calls, IO subsystem,
→ kernel vmlinuz

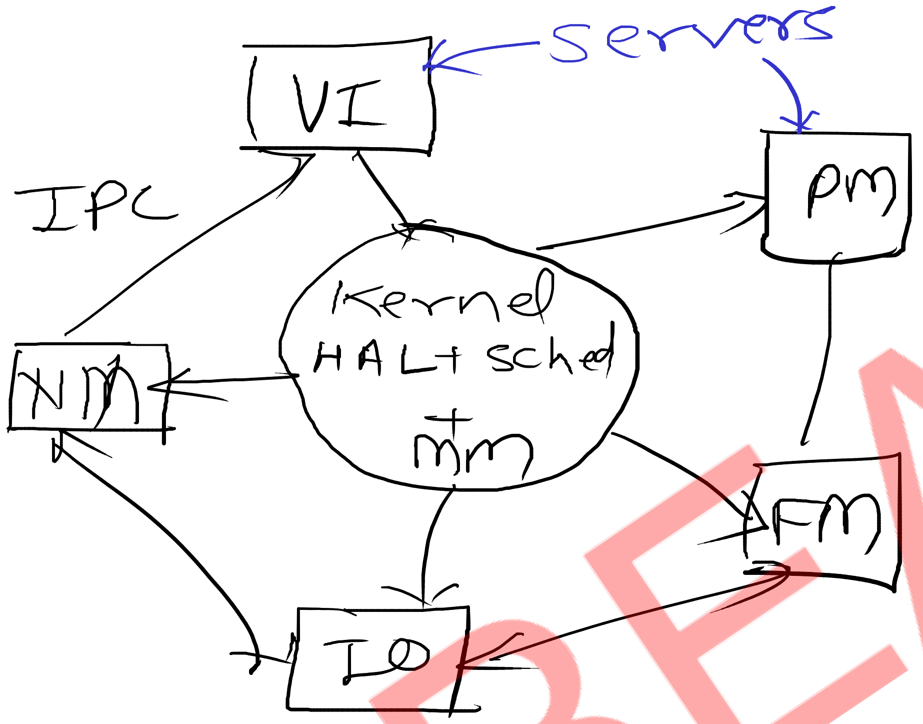
Dynamic Component:
File system mgr
device drivers
→ dynamically loadable modules (kernel objects)

Types of kernel

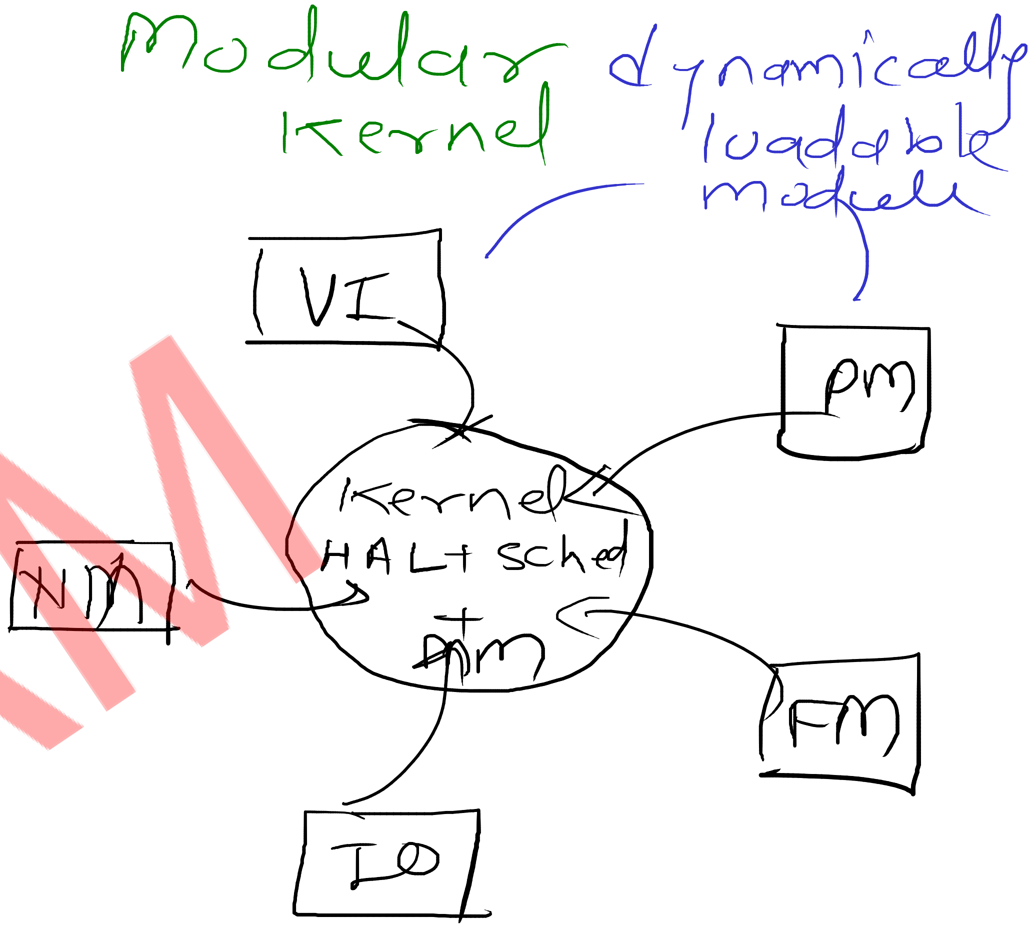
Monolithic Kernel



Micro Kernel



Modular Kernel


dynamically
loadable
module
VI
pm
NM
Kernel
HAL + sched
+
mm
FM
IO