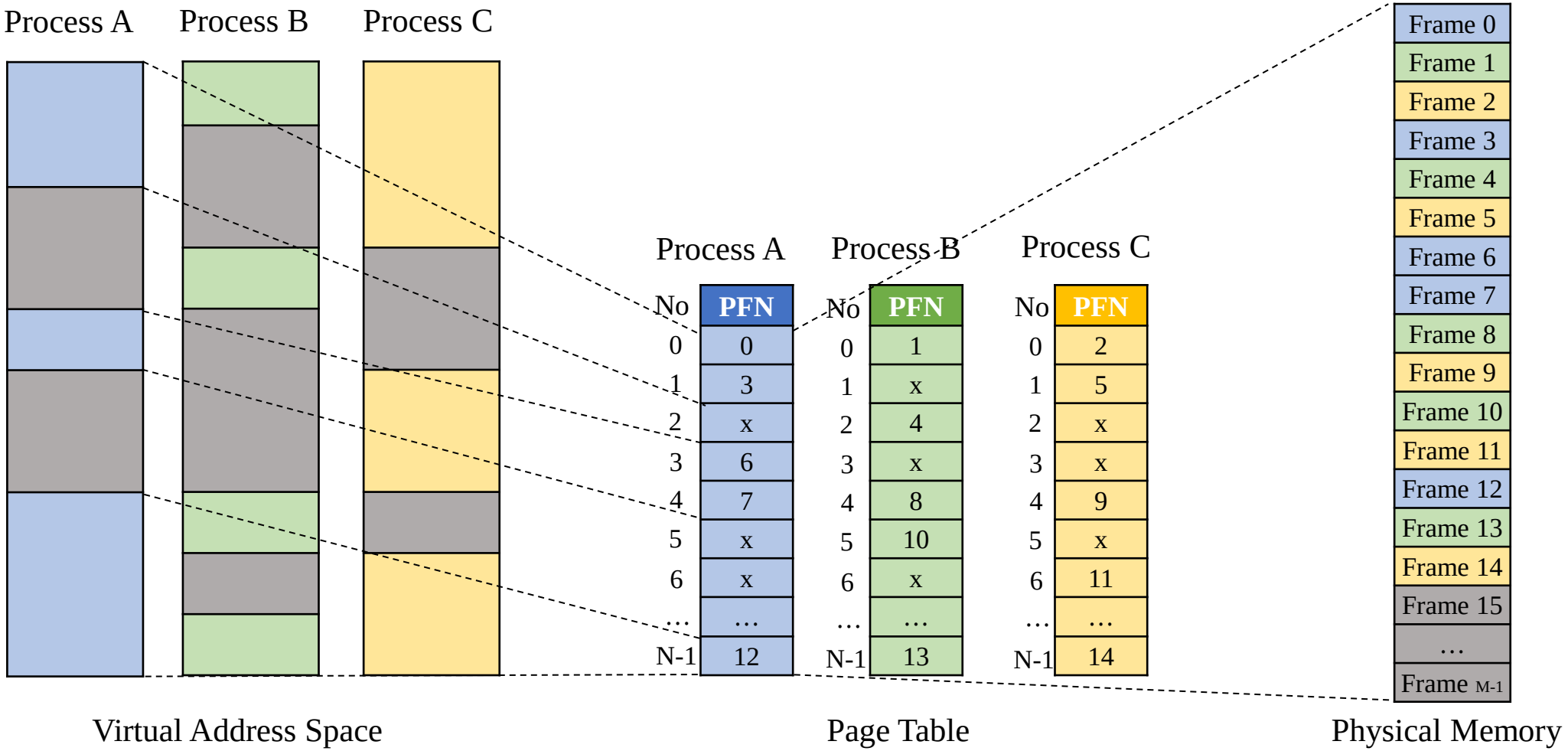
Memory Manager

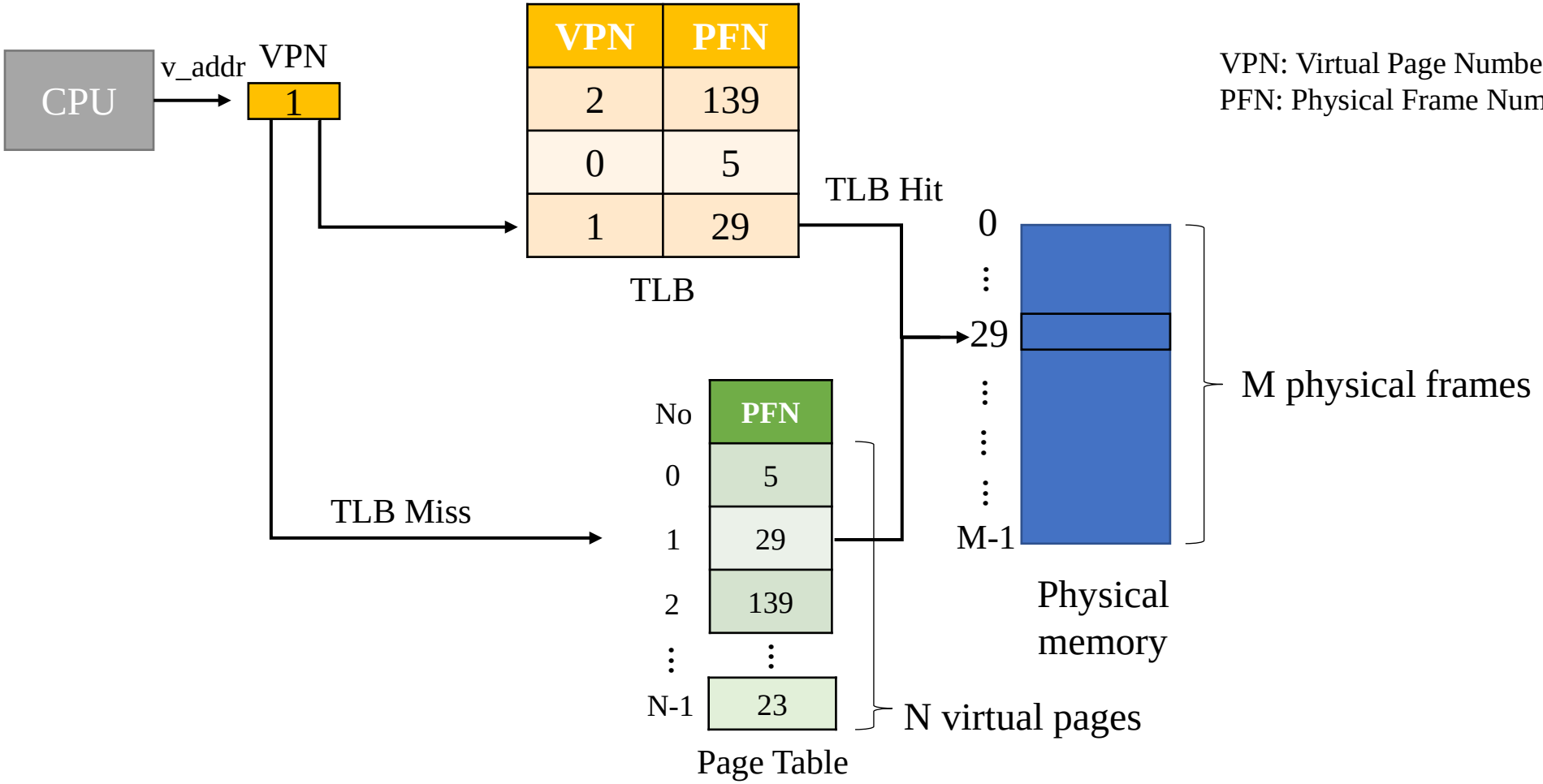
Background-Paging System with Multiple Processes



Overview of Memory Mapping with Multiple Processes

Done by hardware

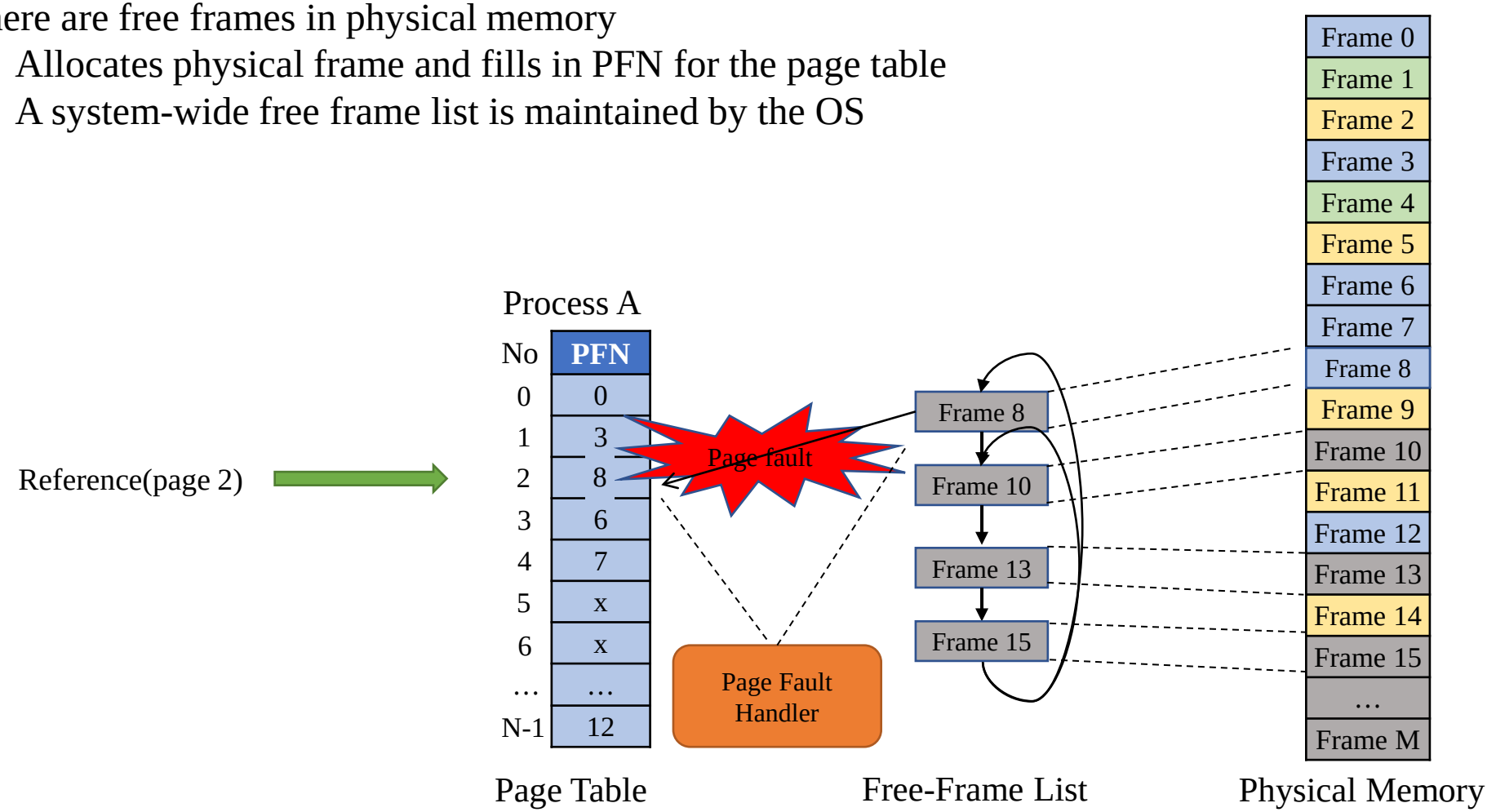
Overview-Address Translation



A paging based memory manager with TLB support

Overview-Page Fault Handler (1/2)

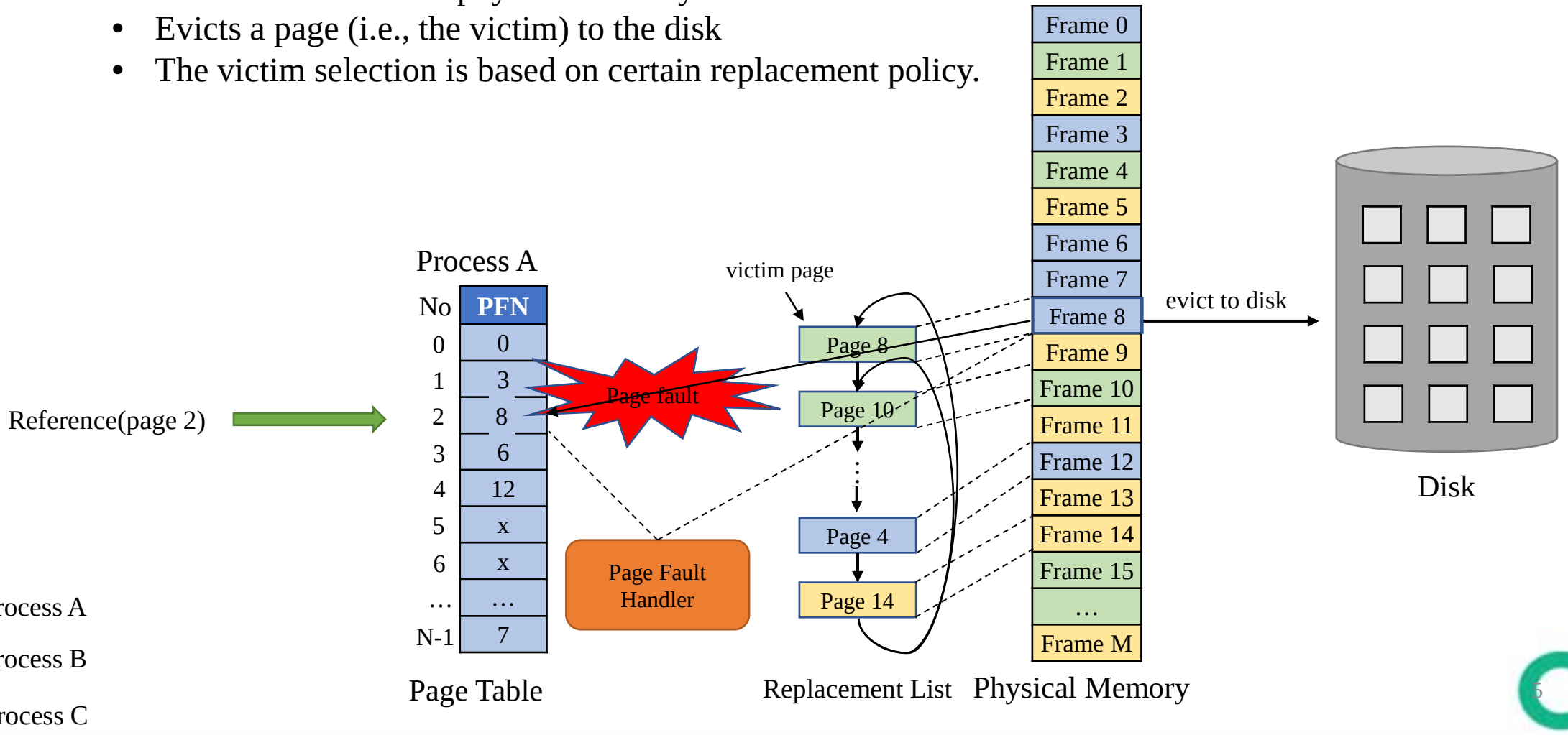
- Once a page fault occurs
  - If there are free frames in physical memory
    - Allocates physical frame and fills in PFN for the page table
    - A system-wide free frame list is maintained by the OS



Done by OS kernel

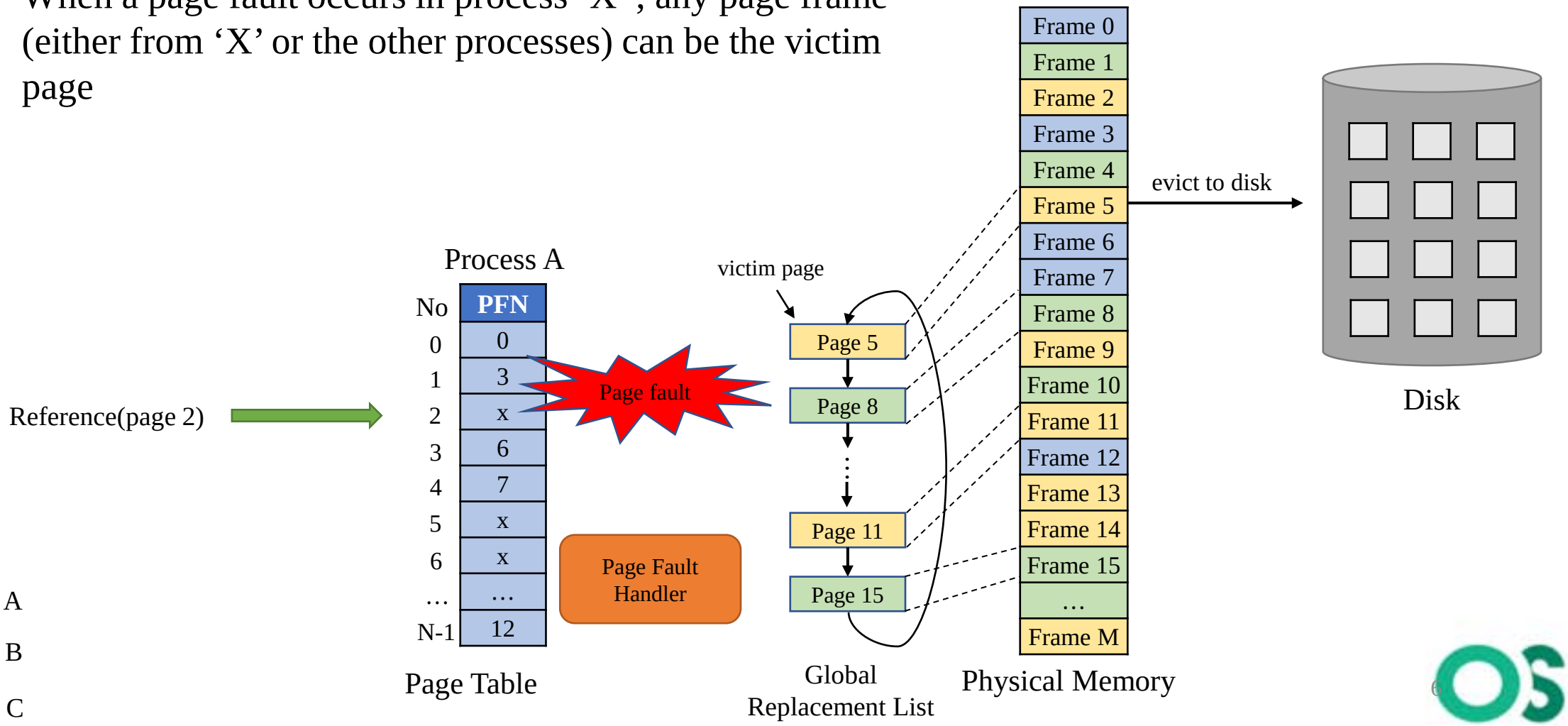
Overview-Page Fault Handler (2/2)

- Once a page fault occurs
  - If there is no free frame in physical memory
    - Evicts a page (i.e., the victim) to the disk
    - The victim selection is based on certain replacement policy.



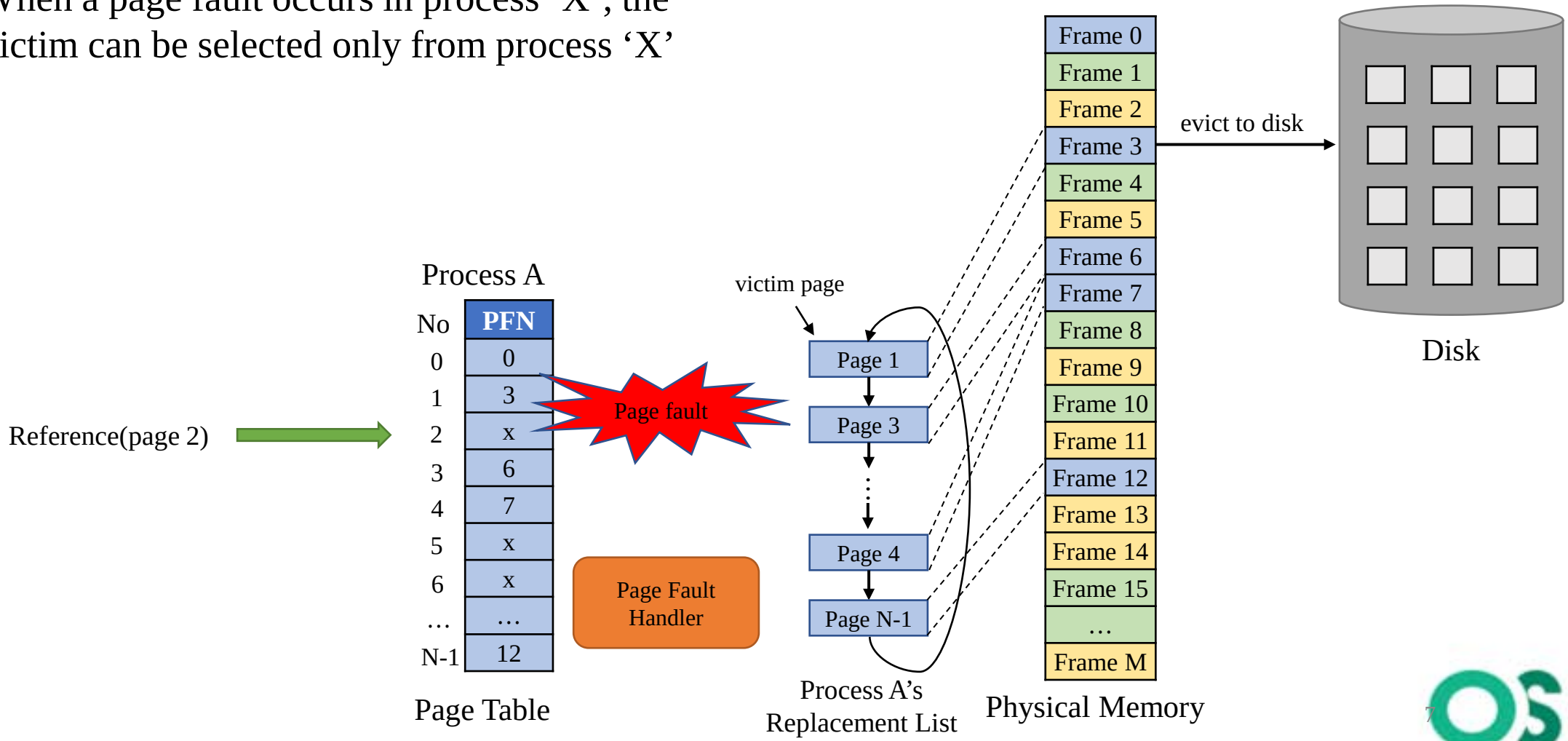
Overview-Allocation Policy (1/2)

- Global
  - When a page fault occurs in process ‘X’ , any page frame (either from ‘X’ or the other processes) can be the victim page

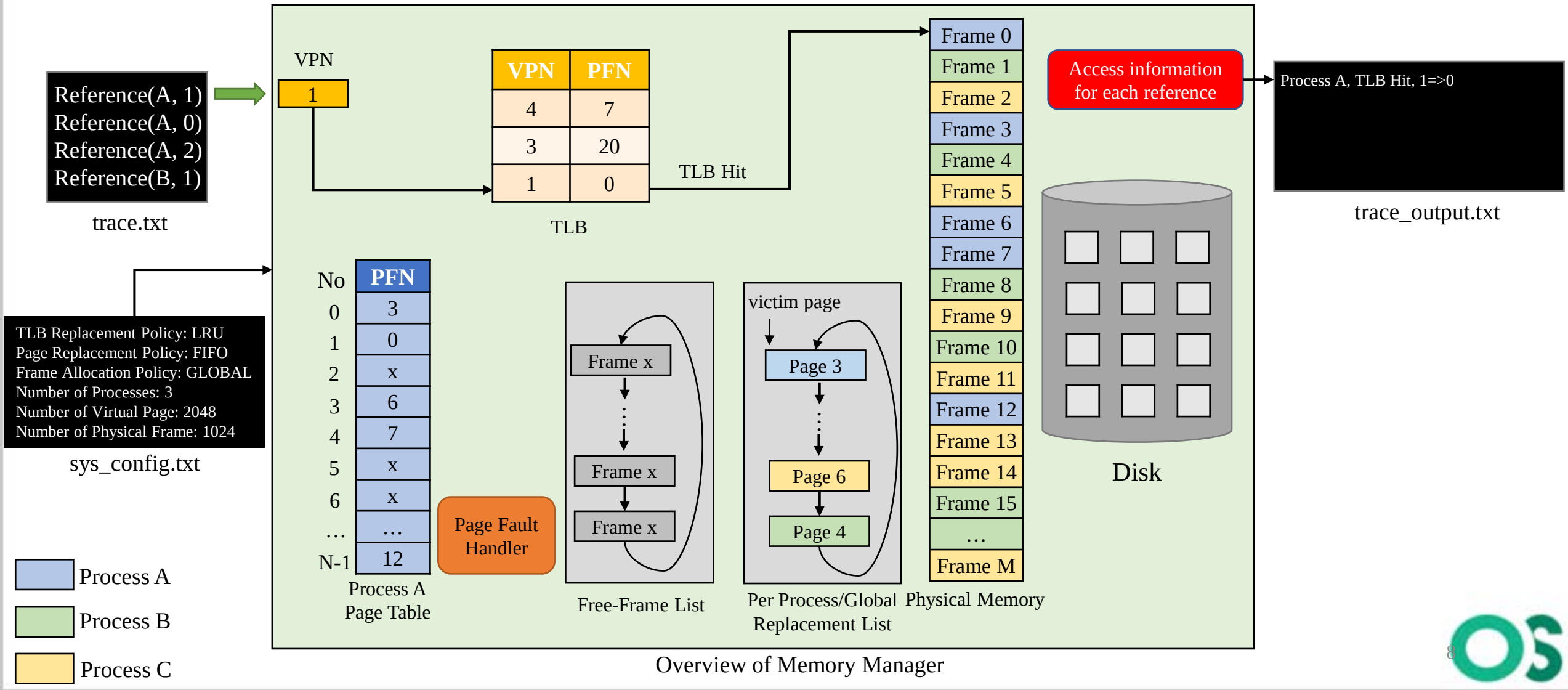


Overview-Allocation Policy (2/2)

- Local
  - When a page fault occurs in process ‘X’, the victim can be selected only from process ‘X’

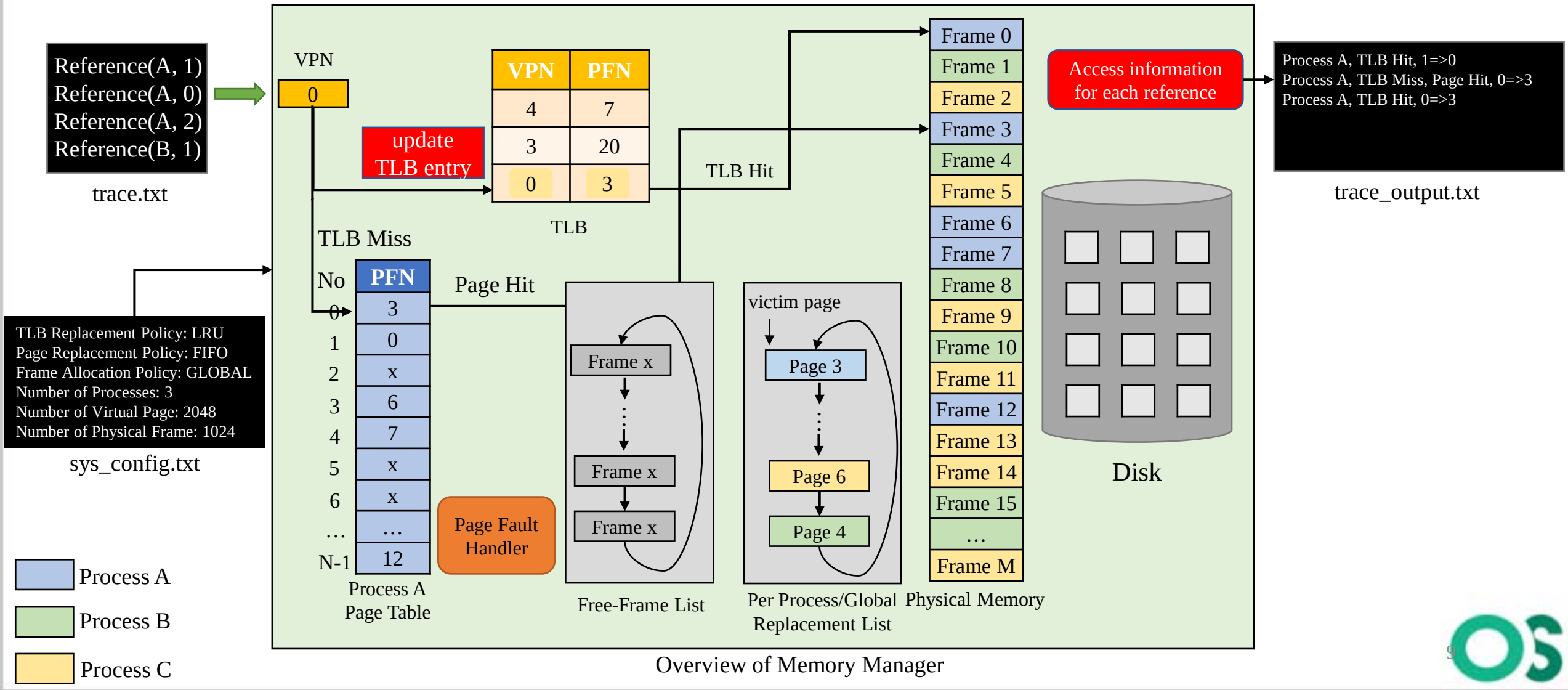


Overview-Memory Manager (1/5)

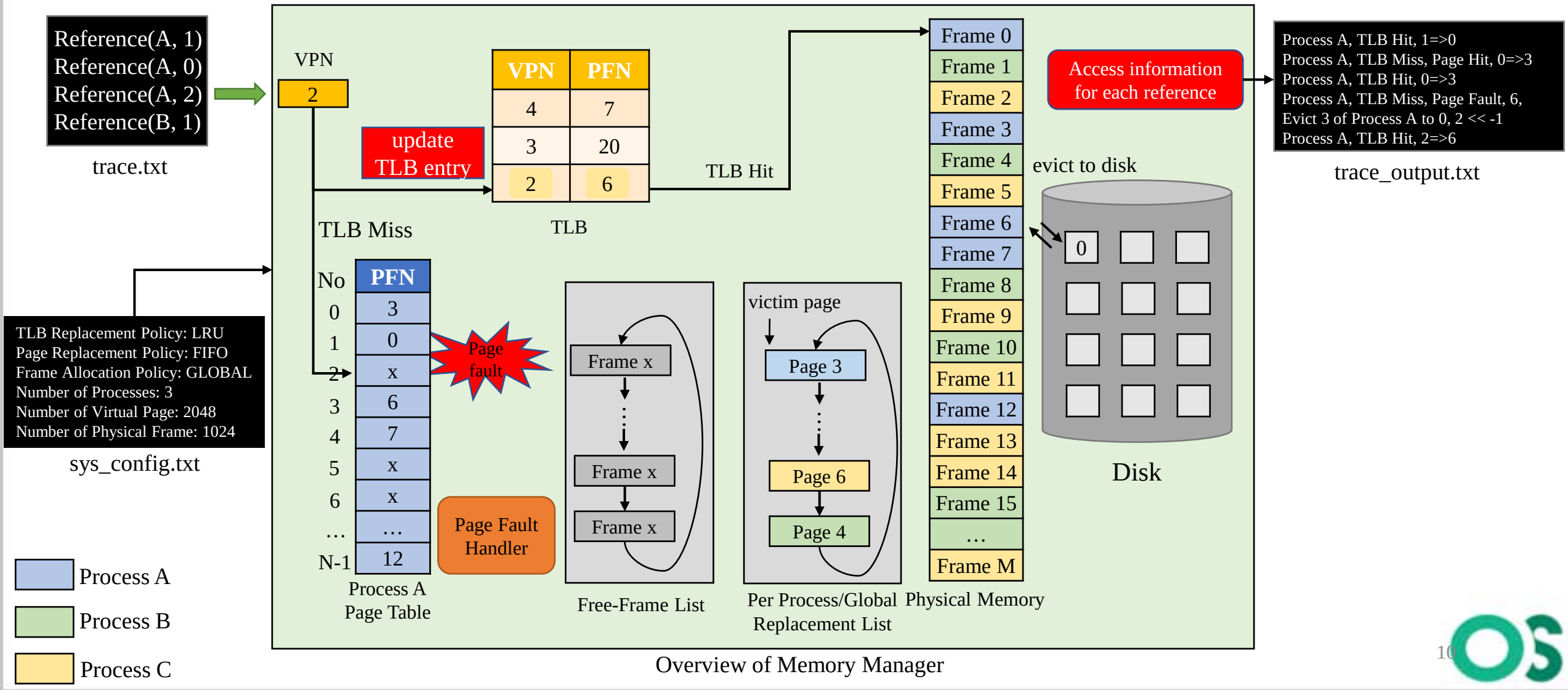


Overview of Memory Manager

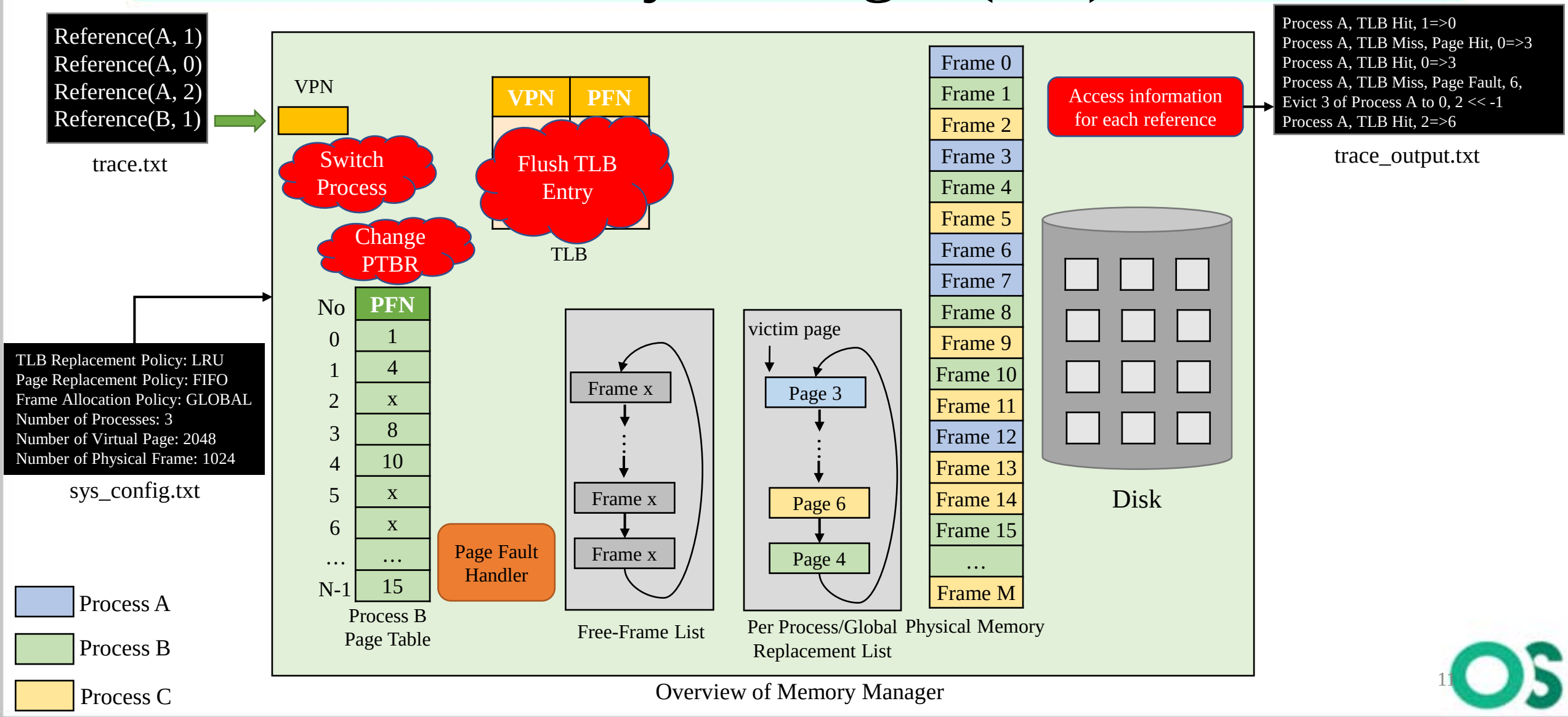
Overview-Memory Manager (2/5)



Overview-Memory Manager (3/5)

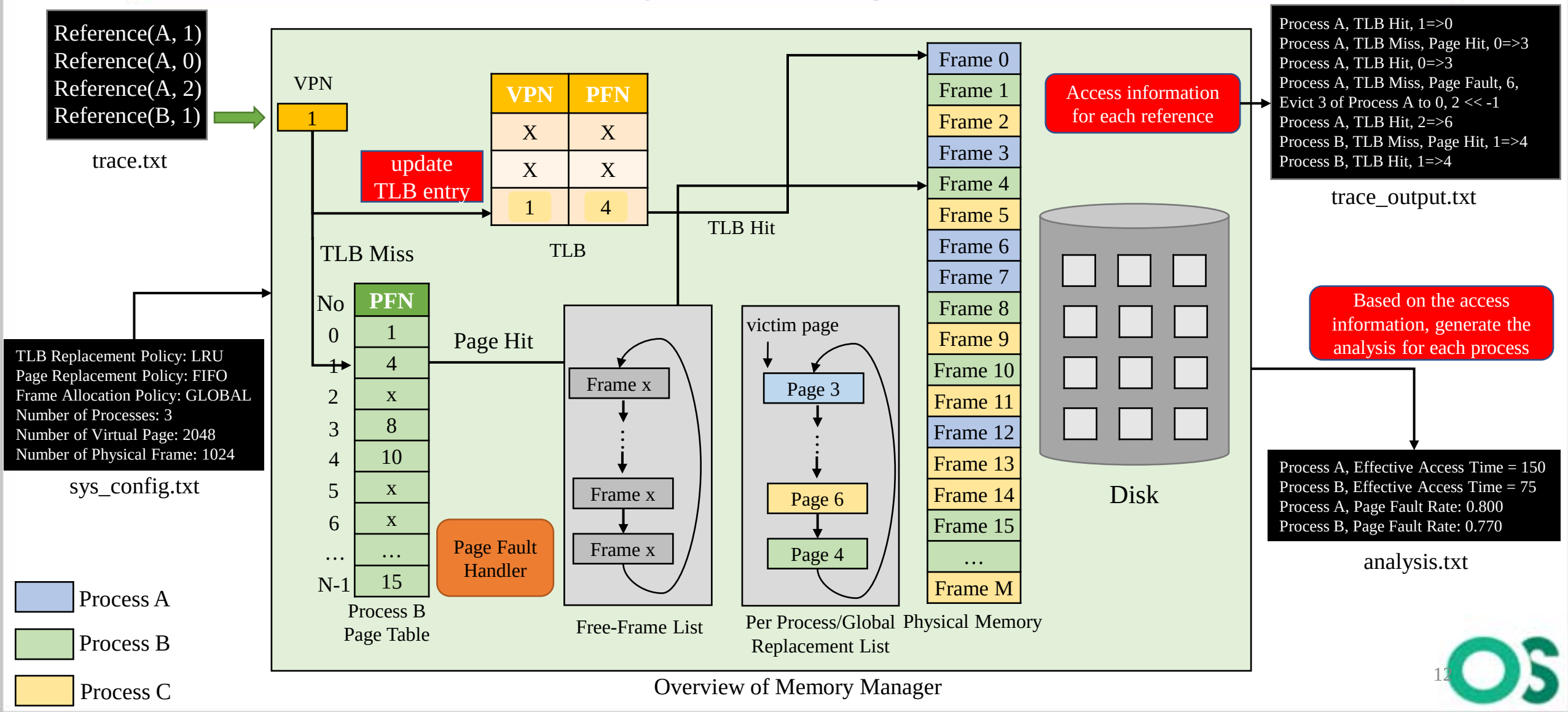


Overview-Memory Manager (4/5)



Overview of Memory Manager

Overview-Memory Manager (5/5)



Requirements (1/2)

- Implement a paging based memory manager with TLB support
  - Allocate/manage physical frames for multiple processes
  - Use a TLB to speed up address translation by software simulation
  - Use an one-level page table for mapping virtual pages to physical frames
- When the page table has been updated
  - Ensure that TLB/page table are consistent
- When the process has been switched
  - Flush TLB and change PTBR
- TLB Replacement Policy
  - Random
  - LRU
- Page Replacement Policy
  - FIFO
  - Clock
- Frame Allocation Policy
  - Global
  - Local

Requirements (2/2)

- Show the TLB miss/hit and related information for each reference in trace file on the output file.
  - If a TLB miss occurs, show the page hit/page fault and related information for each reference in trace file on the output file.
- Show the following information for each process under different policies
  - Effective Access Time
  - Page Fault Rate
- Write a document to show the pros and cons of each policy
  - Please describe your own opinions

Requirement-Page Table

- Reference Bit
  - **1**: the page table entry is referenced
  - **0**: the page table entry is not referenced
- Present Bit
  - **1**: the page is in physical memory
  - **0**: the page is not in physical memory, it is on disk
- When a page is page-out to disk block ***K***, the **PFN** field will be set as ***K***

VPN	PFN/DBI	Reference	Present
0	4	1	0
1	0	1	0
2	0	0	0
	…		
Z	2	1	1

VPN : virtual page number
PFI : physical frame number
DBI : disk block number

Assumptions

- The number of TLB entries is fixed to 32
  - There is no ASID support in this homework assignment
- There will be P processes, N virtual pages and M physical frames
  - P, N and M will be given in the trace file
  - N is greater than M
  - N and M are both power of 2
- Page fault handler only evicts a page to the disk when there is no free frame in physical memory
- An evicted page should be written back to the disk whether it is dirty or not
  - This is not the case in real world, but it simplifies the complexity of this homework
- The disk always has enough space for evicted pages
  - To page-out a page, select an free disk block with the smallest disk block number

Input File Format (1/2)

- Two input files: system configuration file and trace information file
- System configuration file
  - File name: “sys_config.txt”
  - Includes 6 lines
    - Which TLB Replacement Policy?
    - Which Page Replacement Policy?
    - Which Frame Allocation Policy?
    - Number of Process?
      - $20 \geqq P \geqq 1$
    - Number of Virtual Page N
      - $2048 \geqq N \geqq 2$
      - Power of 2
    - Number of Physical Frame M
      - $1024 \geqq M \geqq 1$
      - $N \geqq M$
      - Power of 2

```
1 TLB Replacement Policy: LRU | Random
2 Page Replacement Policy: FIFO | CLOCK
3 Frame Allocation Policy: LOCAL | GLOBAL
4 Number of Processes: P (>=1)
5 Number of Virtual Page: N (power of 2)
6 Number of Physical Frame: M (power of 2)
```

sys_config.txt

Input File Format (2/2)

- Trace information file
  - File name: “trace.txt”
  - Includes page reference information of the processes
  - Reference (X, Y): reference virtual page Y of Process X
  - X ranges from ‘A’ ~ ‘T’

```
1 Reference(A, 0)
2 Reference(A, 1)
3 Reference(A, 2)
4 Reference(B, 0)
5 Reference(B, 1)
6 Reference(B, 2)
7 Reference(C, 0)
8 Reference(C, 1)
9 Reference(C, 2)
…
2 Reference(B, 4)
…
Z Reference(C, 8)
```

trace.txt

Output File Format (1/2)

- Show the following information for each reference
  - Format for a TLB hit: **Process [X], TLB Hit, [VPN]=>[PFN]**
  - Format for a TLB miss:
    - Page hit: **Process [X], TLB Miss, Page Hit, [VPN]=>[PFN]**
    - Page fault: **Process [X], TLB Miss, Page Fault, [PFN], Evict [VPN] of Process [X] to [Destination], [VPN]<<[Source]**
      - PFN: frame index that is about to be replaced
      - Source: the block number of the page which is page-in from disk
      - Destination: the block number where the evicted page page-out
    - If there is no source/destination (e.g., first reference, no page is page-out) or no evicted VPN, set the value as -1
- Store as “trace_output.txt”

```
Process A, TLB Miss, Page Fault, 0, Evict -1 of Process A to -1, 6 << -1
Process A, TLB Hit, 6=>0
…
Process A, TLB Miss, Page Hit, 1=>6
…
Process B, TLB Miss, Page Hit, 2=>10
```

trace_output.txt

Output File Format (2/2)

- Show the **Effective access time** **for each process**
  - EAT = α(m+t) + (1- α)(2m+t)
  - Assume
    - m = 100ns
    - t = 20ns

TLB Lookup time = *t* time units
Memory cycle time = *m* time units
Hit ratio = *α*

- Show the **page fault rate** **for each process**
- Store as `analysis.txt`

```
Process A, Effective Access Time = 191.428
Process A, Page Fault Rate: 0.800
Process B, Effective Access Time = 160.725
Process B, Page Fault Rate: 0.770
```

analysis.txt

three decimal place accuracy