

Memory

Course: Systems Architecture

Lecturer: Gleb Lobanov

February 25, 2024

Contents

01 Memory

02 Memory allocation

03 Architecture ideas

04 Registers

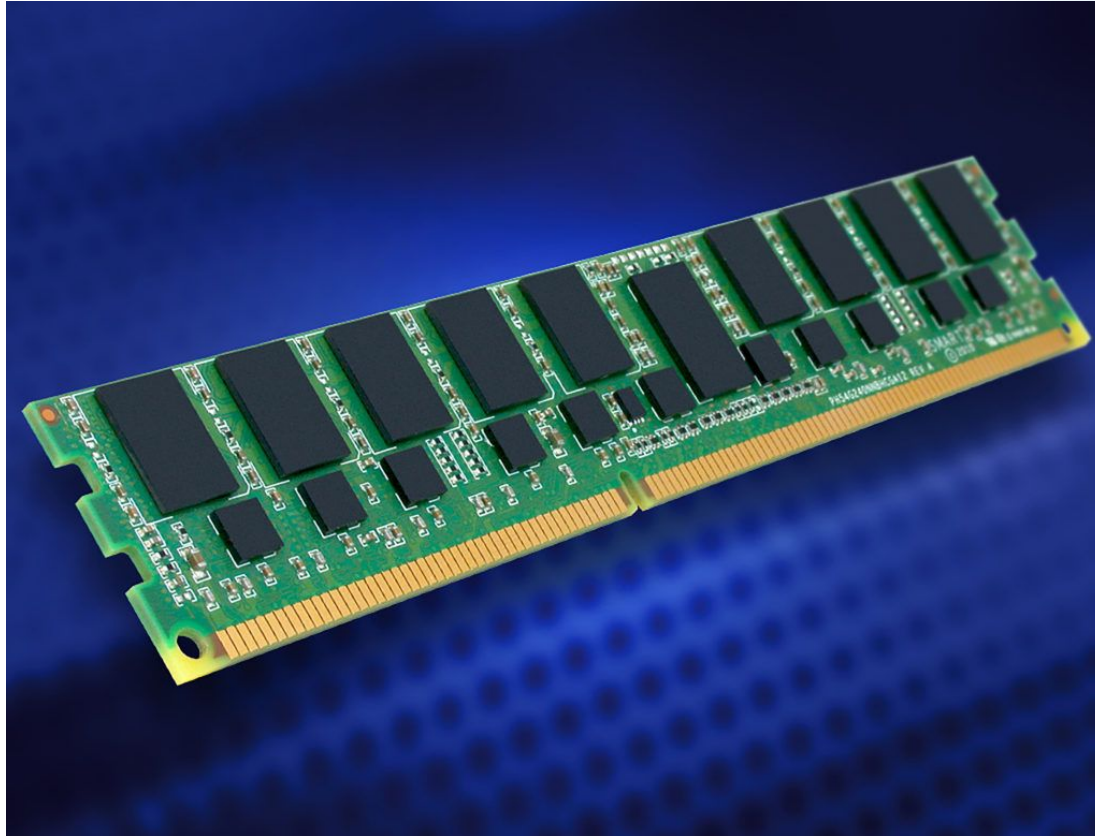
05 Caches

01

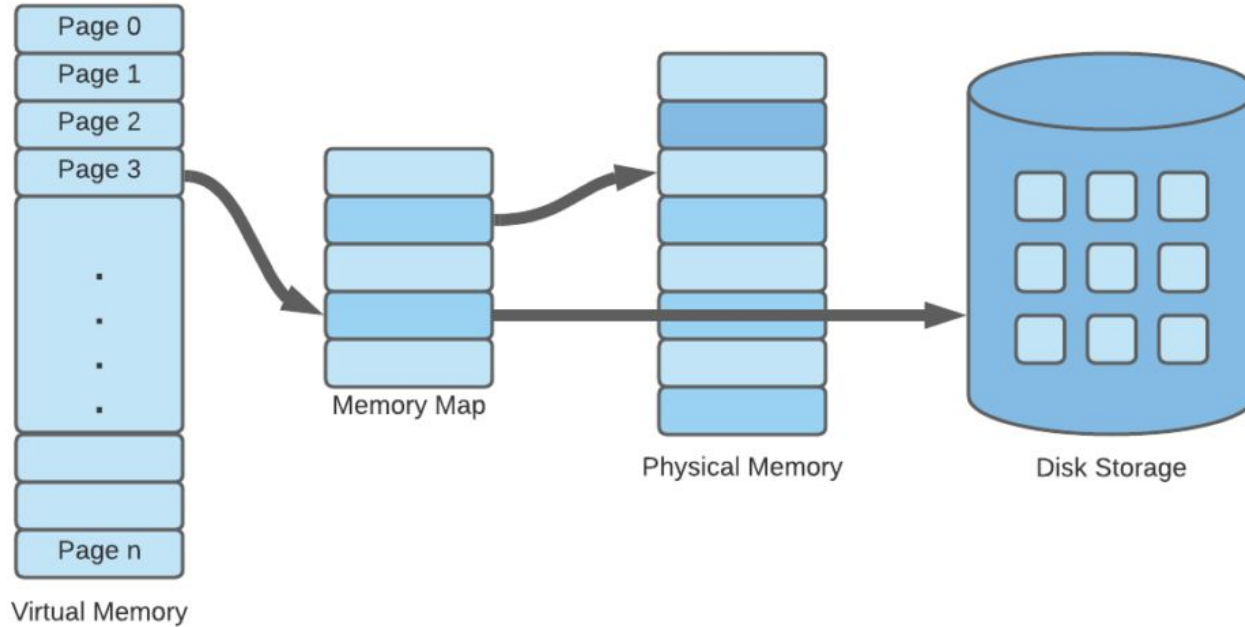
Memory

In this section, we will discuss what virtual memory and real memory are and what is the relationship between them.

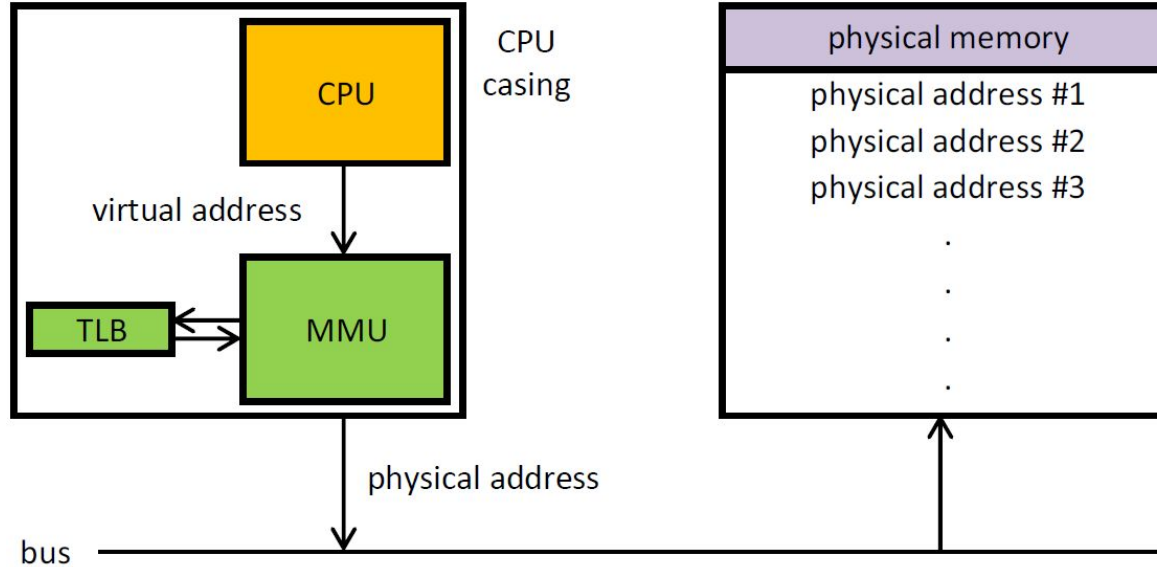
Memory



Also memory

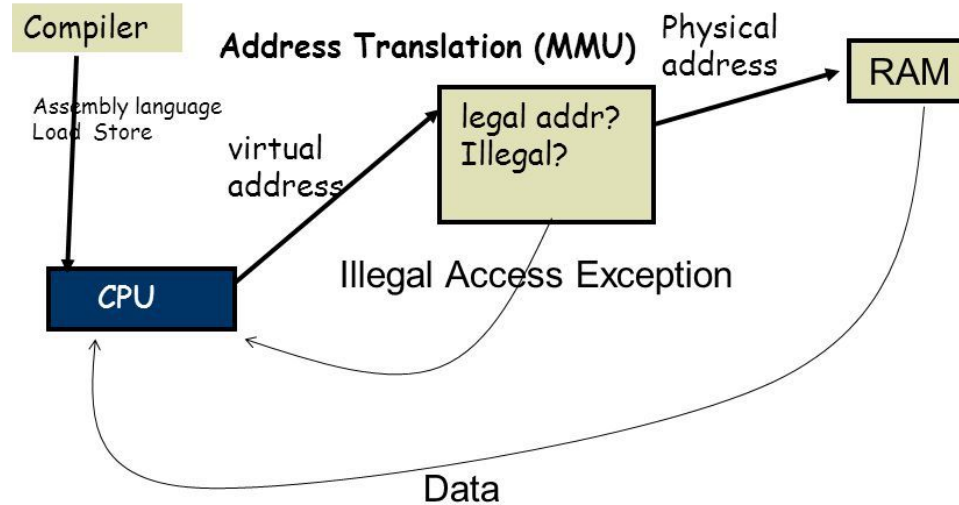


Memory management unit

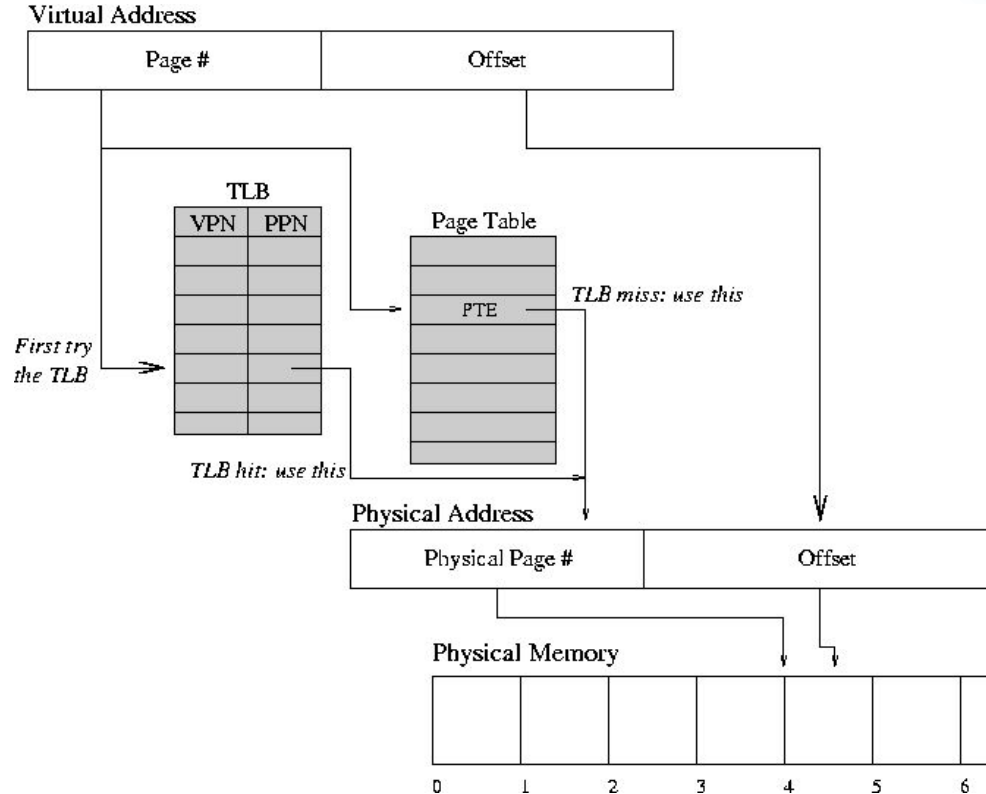


CPU: Central Processing Unit
MMU: Memory Management Unit
TLB: Translation lookaside buffer

Memory management unit

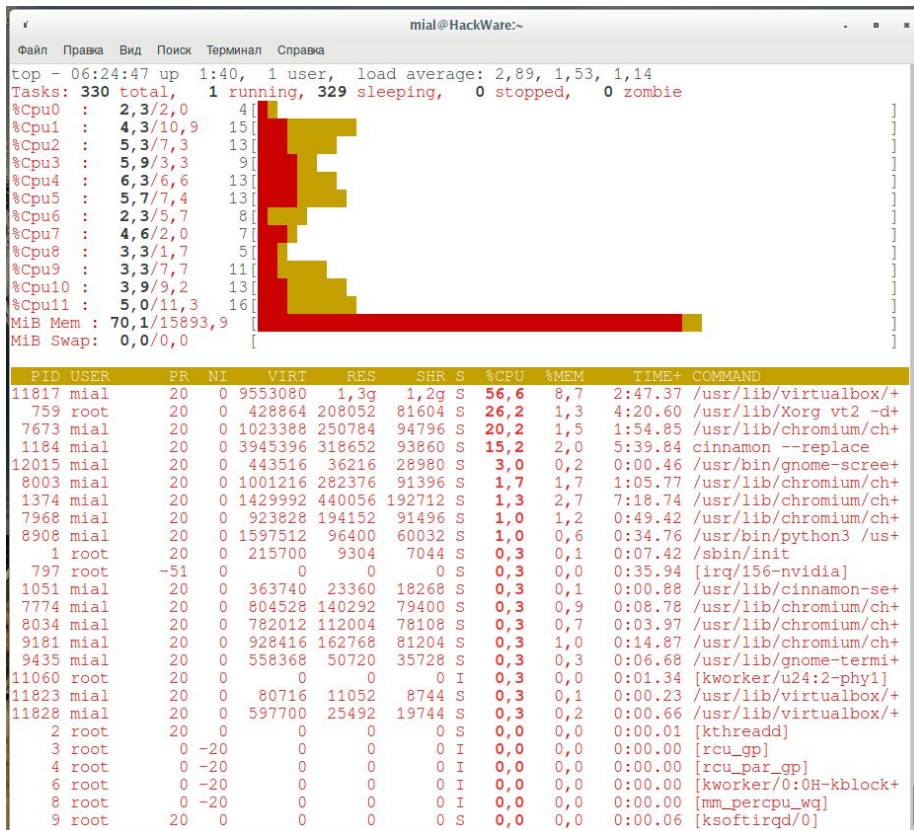


Translation Lookaside Buffer (TLB)



**EPIC**Institute of Technology
Powered by epari

Memory

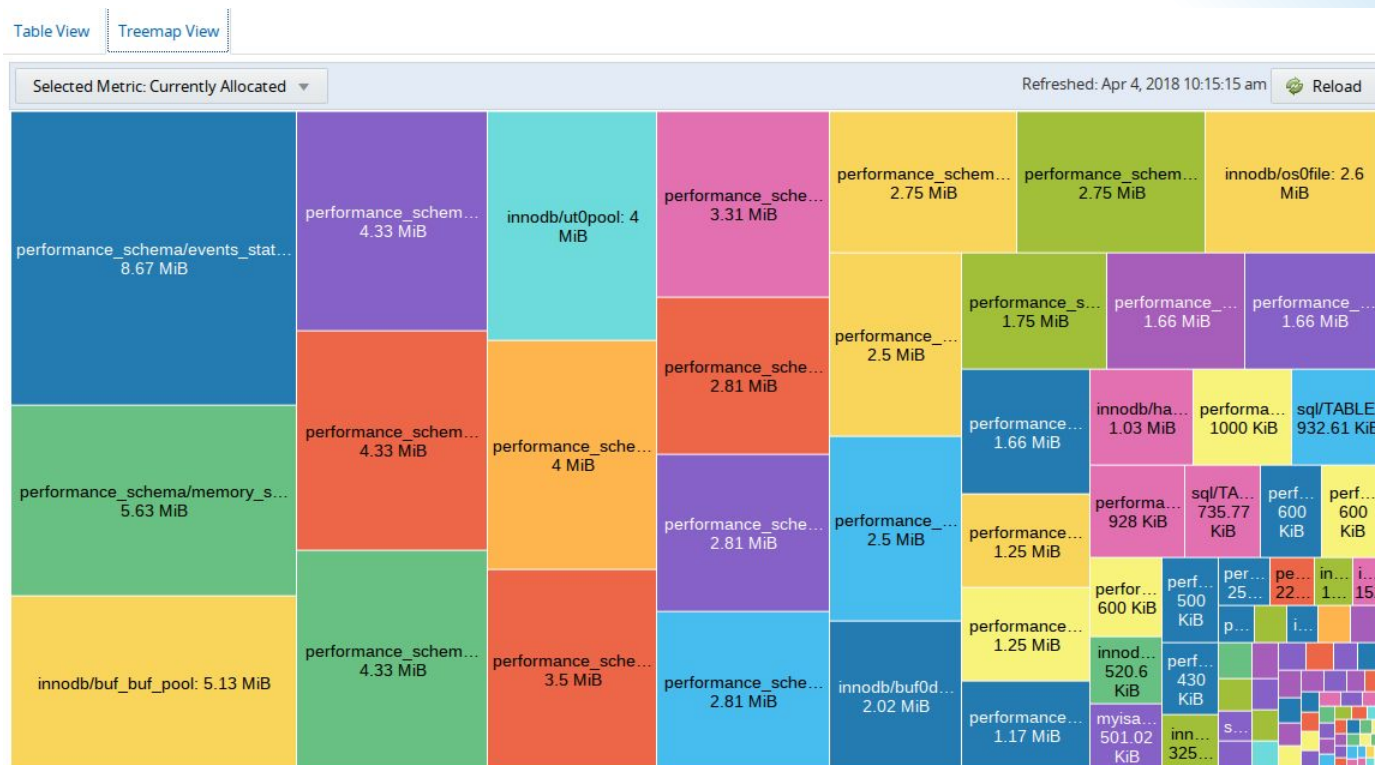




EPIC

Institute of Technology
Powered by 

Memory

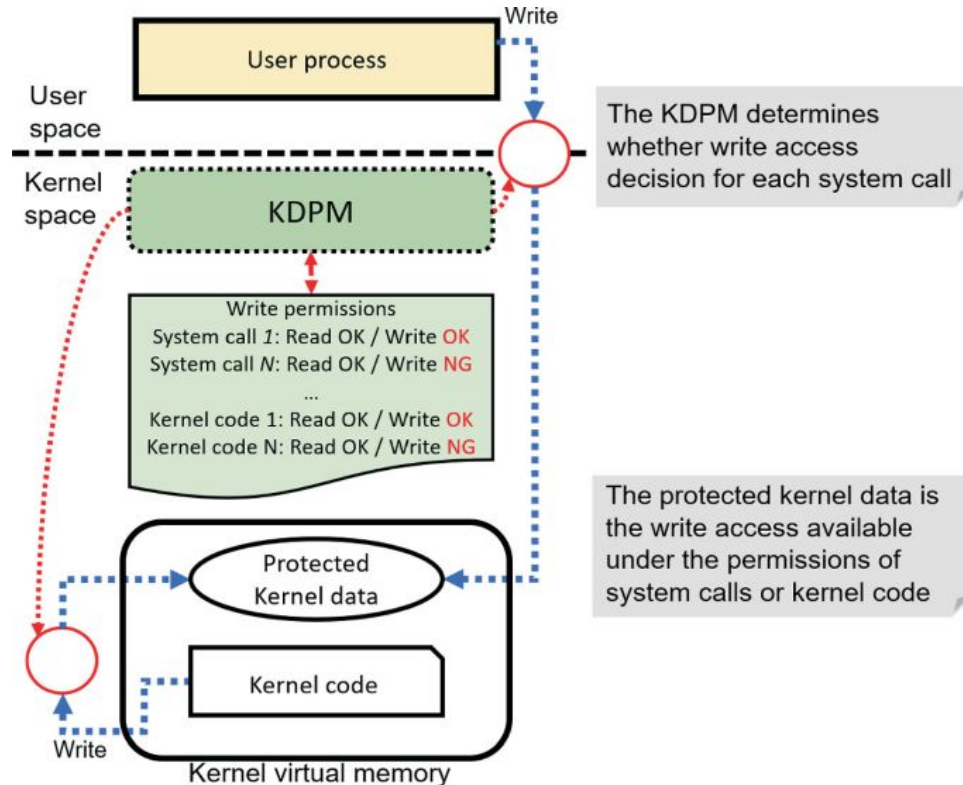




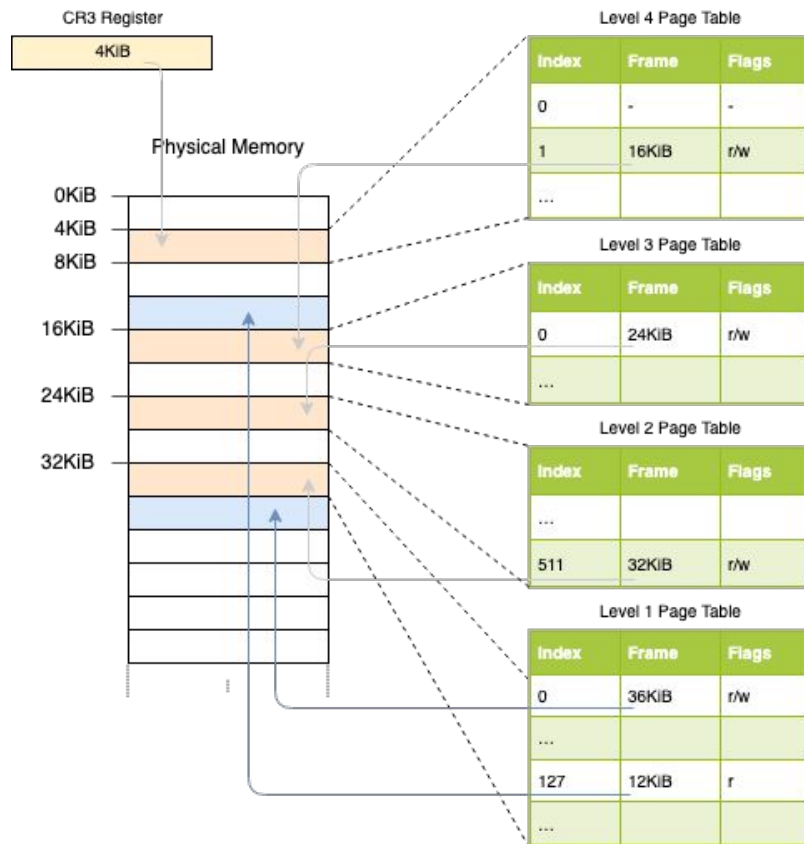
EPIC

Institute of Technology
Powered by ePam

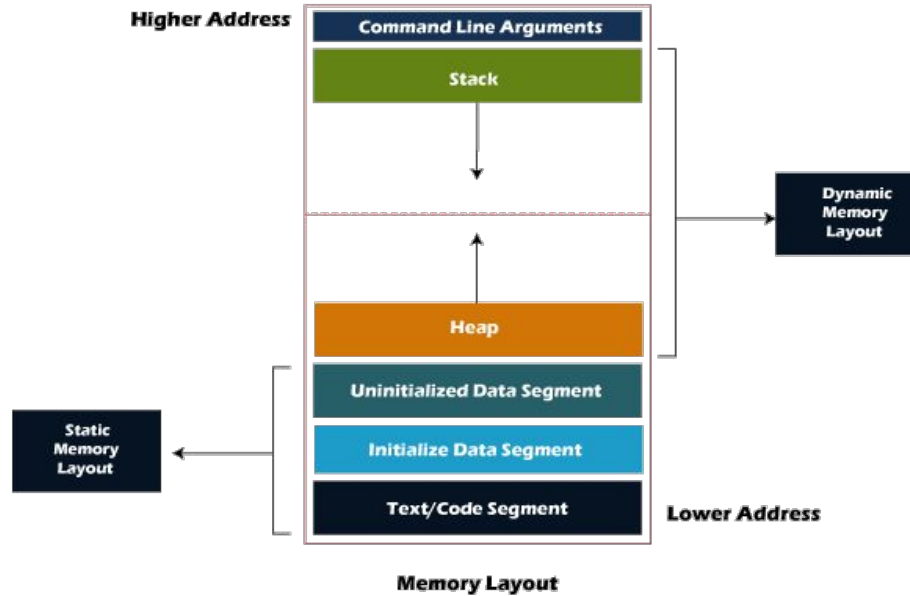
Kernel DMA Protection



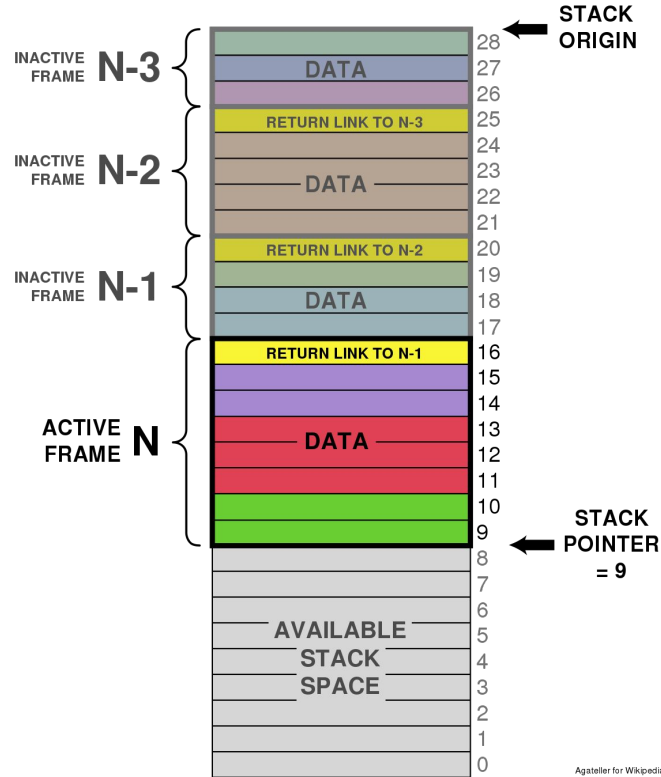
Pages



Memory structure



Stack



02

Memory allocation

In this section, we will discuss what memory allocation is and what allocation algorithms exist.



EPIC

Institute of Technology
Powered by epan

Allocation

cgroup-1

Memory



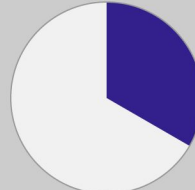
25%

CPU



25%

Network



33%

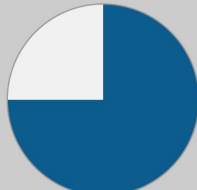
I/O



50%

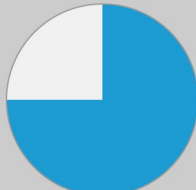
System Remainder After cgroup Allocation

Memory



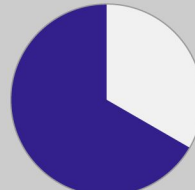
75%

CPU



75%

Network



67%

I/O



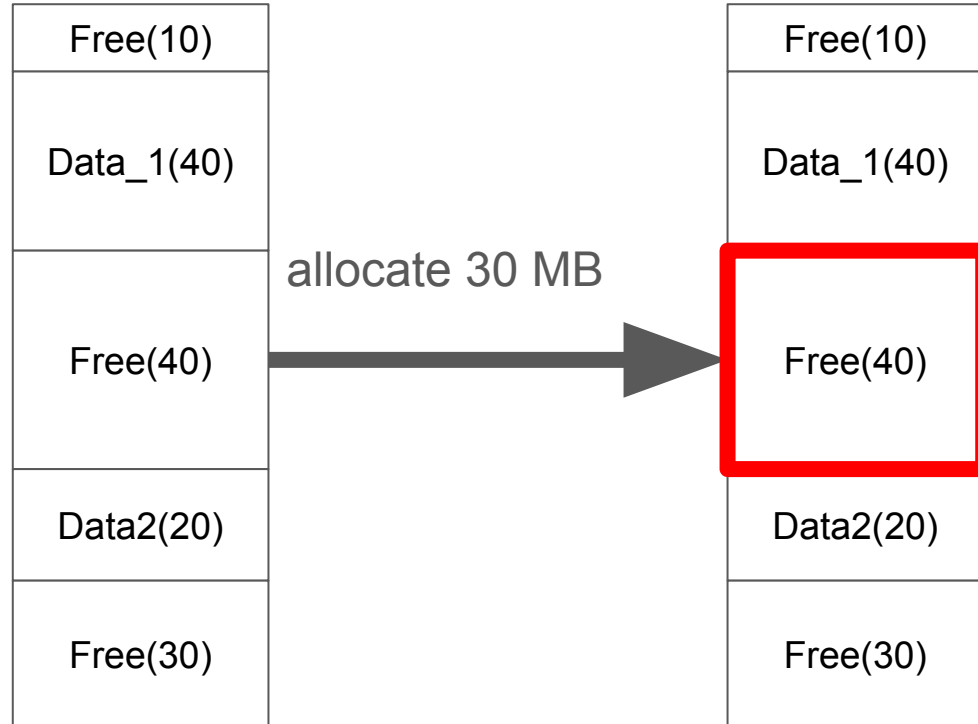
50%

How to alloc memory?

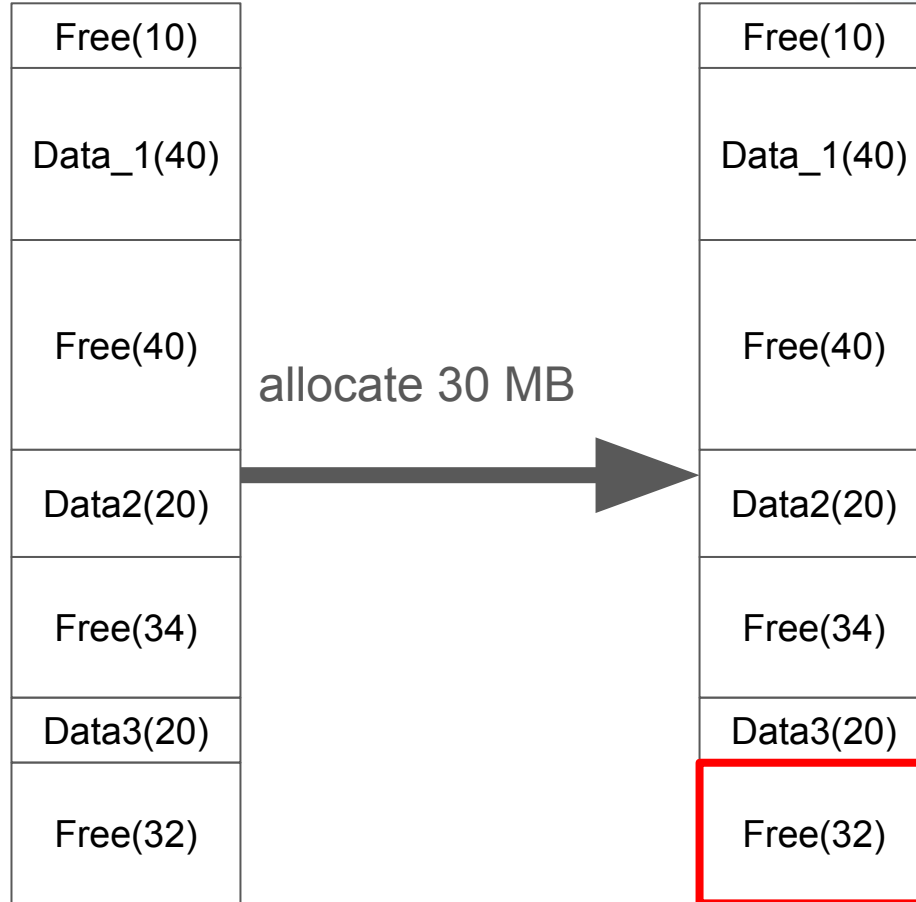
- 1) First-Fit
- 2) Best-Fit
- 3) Worst-Fit
- 4) Next-Fit
- 5) Quick-Fit
- 6) Slab Allocation*
- 7) Buddy Allocation*

maybe some others

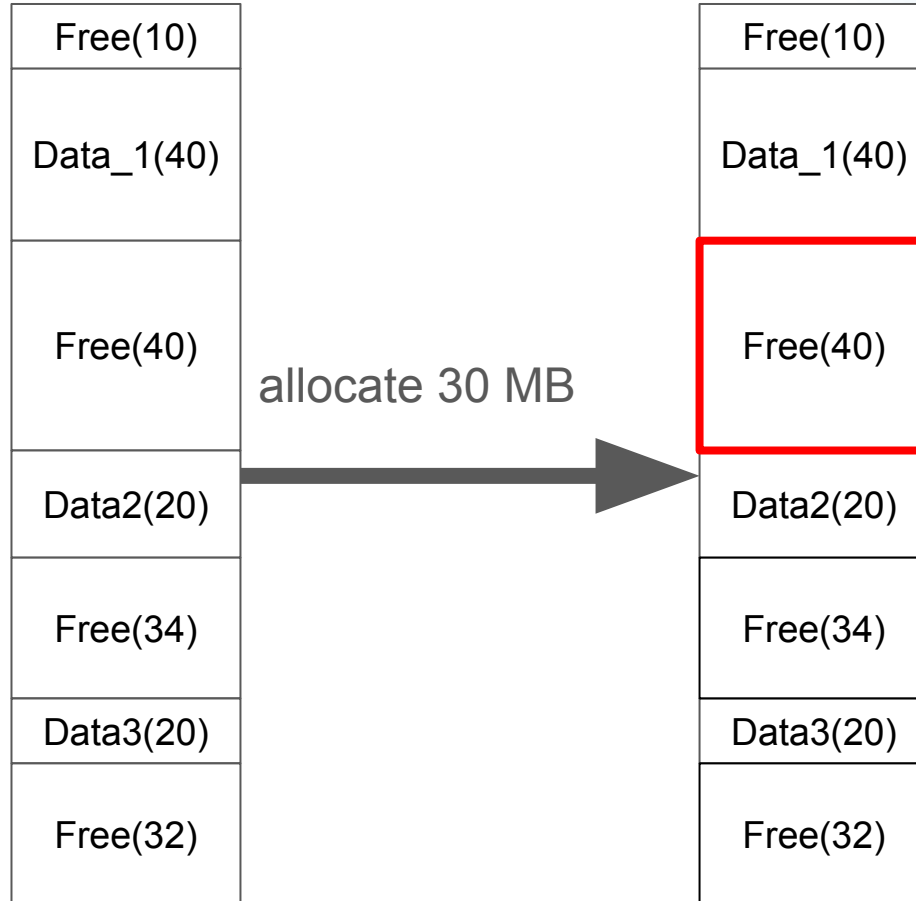
First-Fit Allocation



Best-Fit Allocation

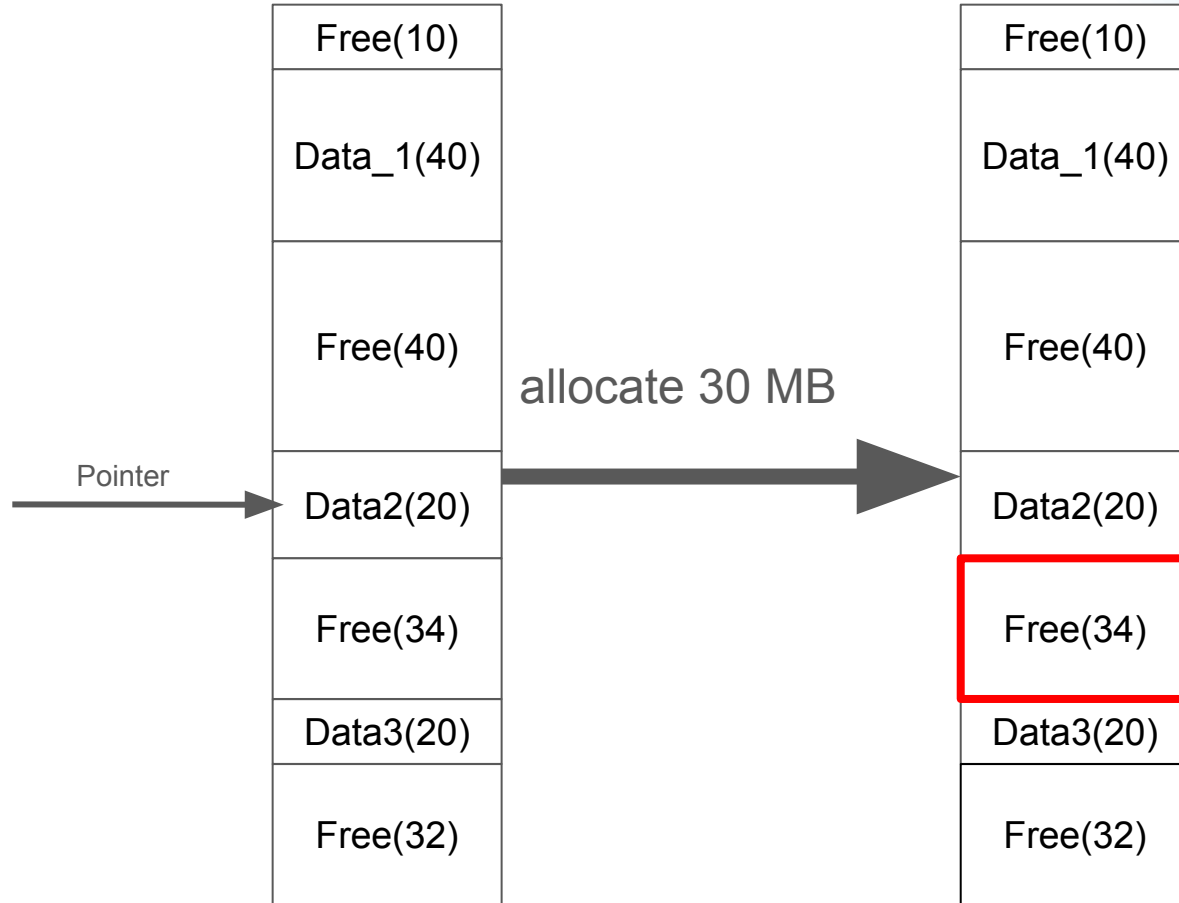


Worst-Fit Allocation

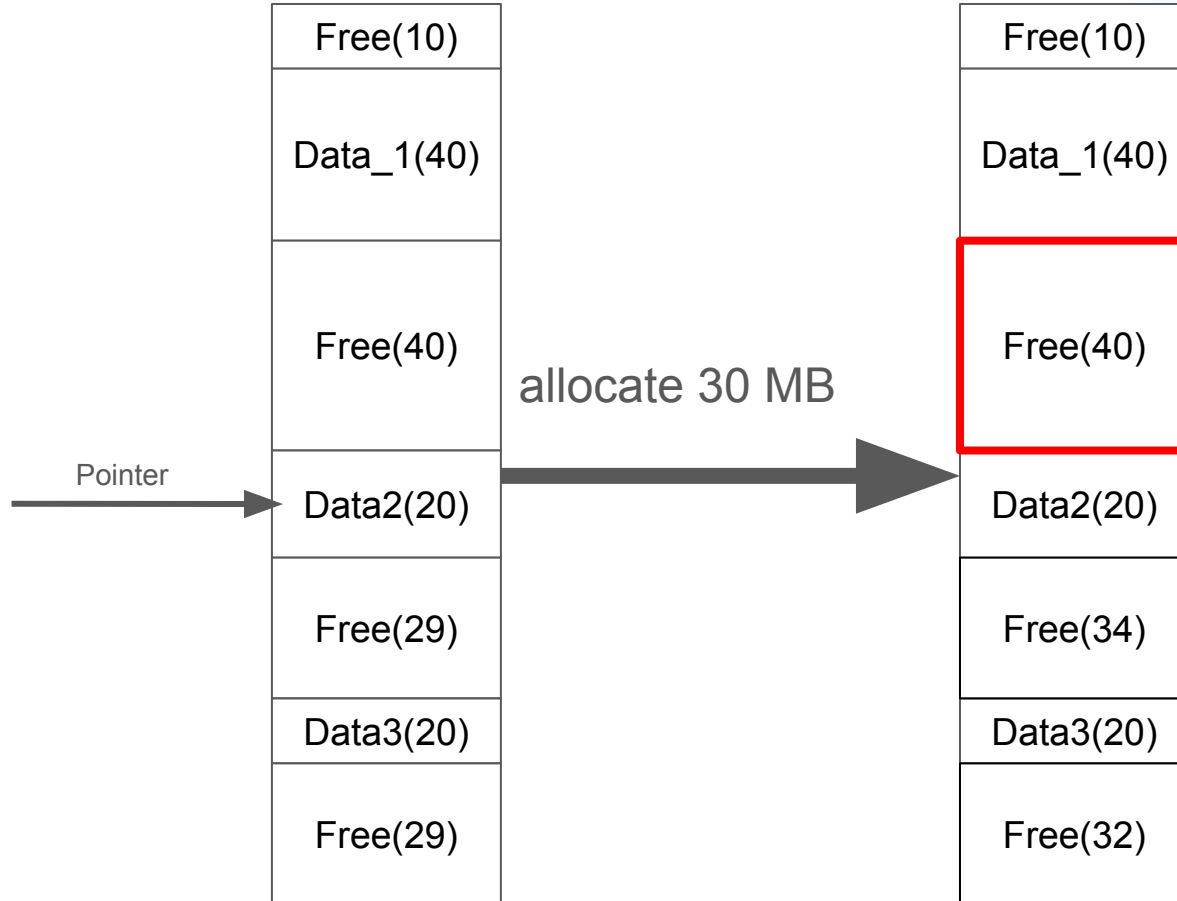


**EPIC**Institute of Technology
Powered by epan

Next-Fit Allocation

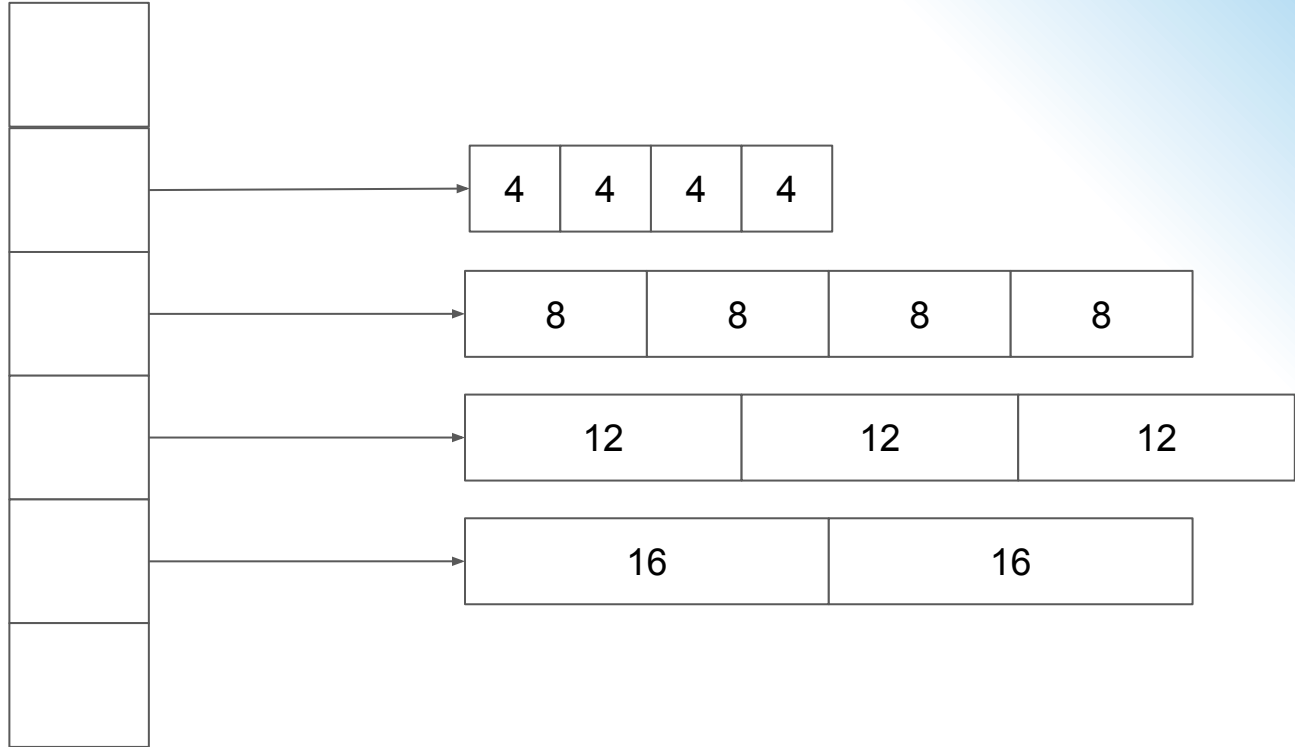


Important!!!



**EPIC**Institute of Technology
Powered by epan

Quick-Fit Allocation





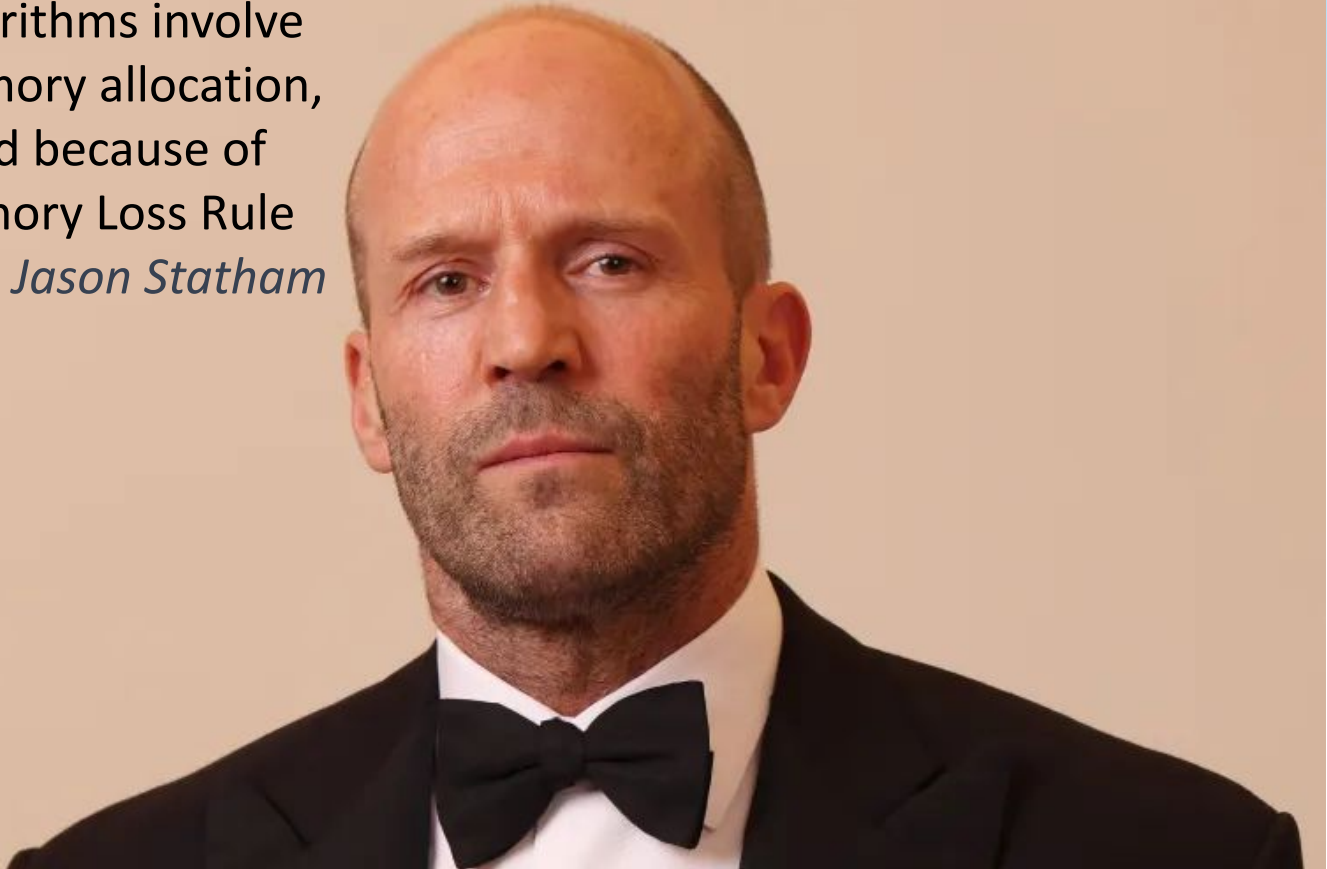
EPIC

Institute of Technology
Powered by 

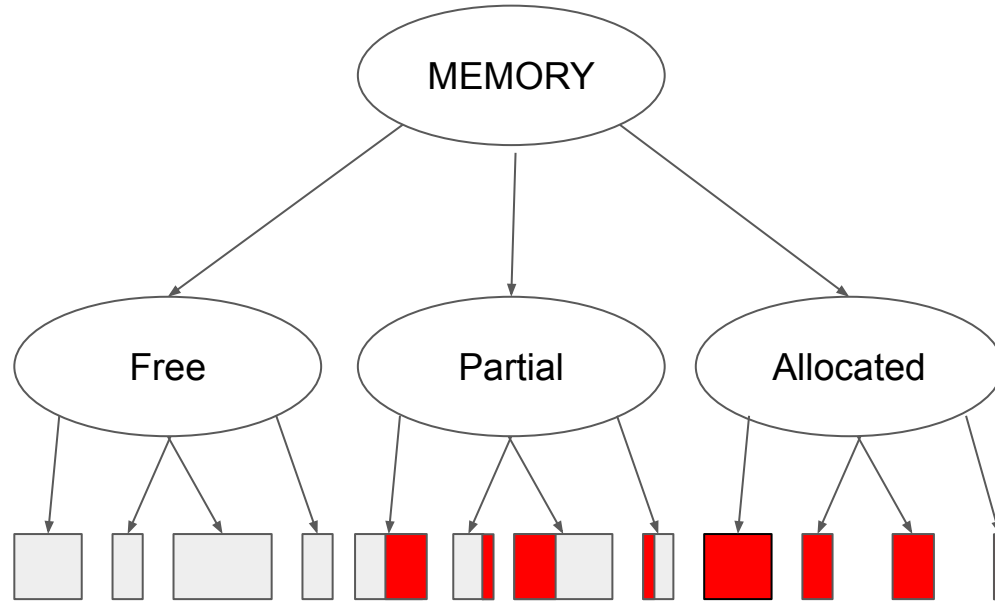
Why none of them?

All these algorithms involve
dynamic memory allocation,
and this is bad because of
the 50% Memory Loss Rule

- Jason Statham



Slab Allocation

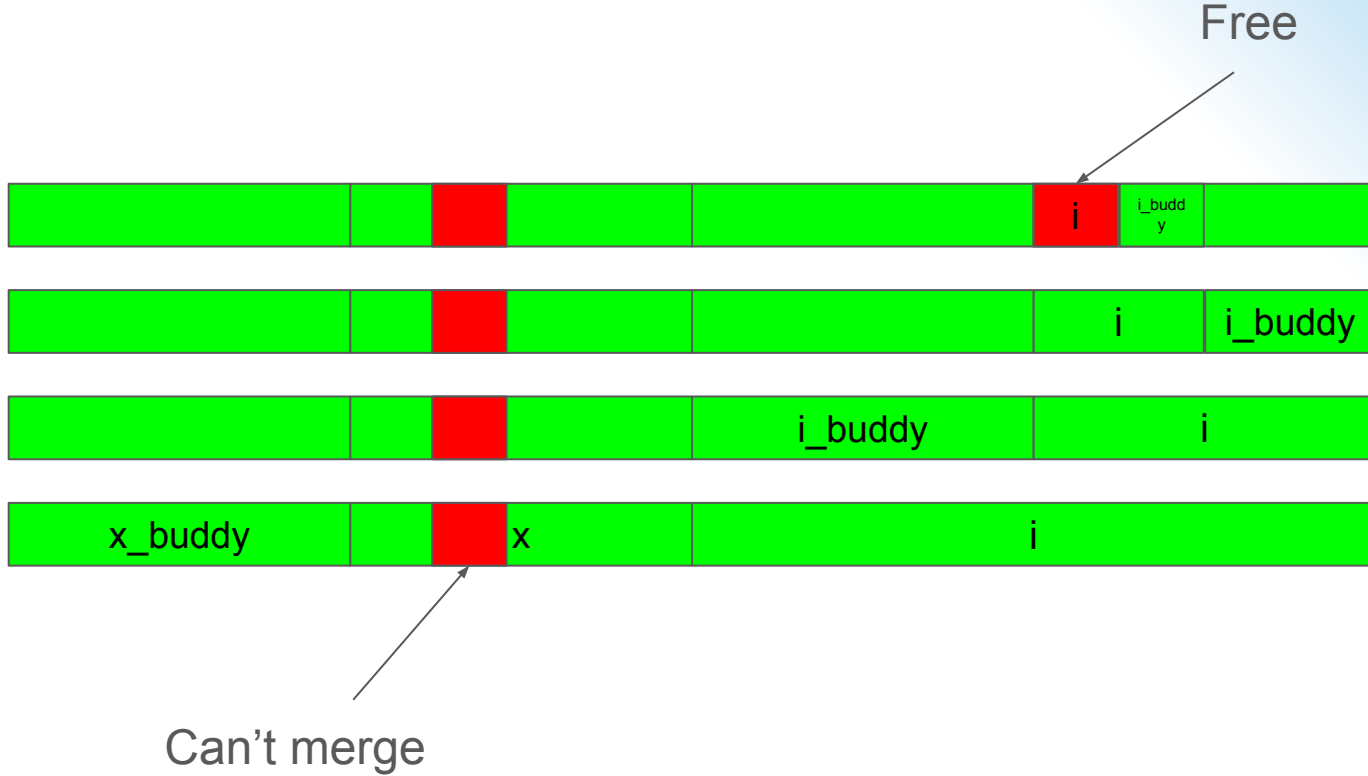




EPIC

Institute of Technology
Powered by epan

Buddy Allocation



03

Architecture ideas

In this section, we will discuss Von Neumann and Harvard architecture.

Von Neumann architecture

- 1) Memory and processor separation
- 2) Principle of memory addressing
- 3) Principle of command control
- 4) Principle of binary encoding

Harvard architecture

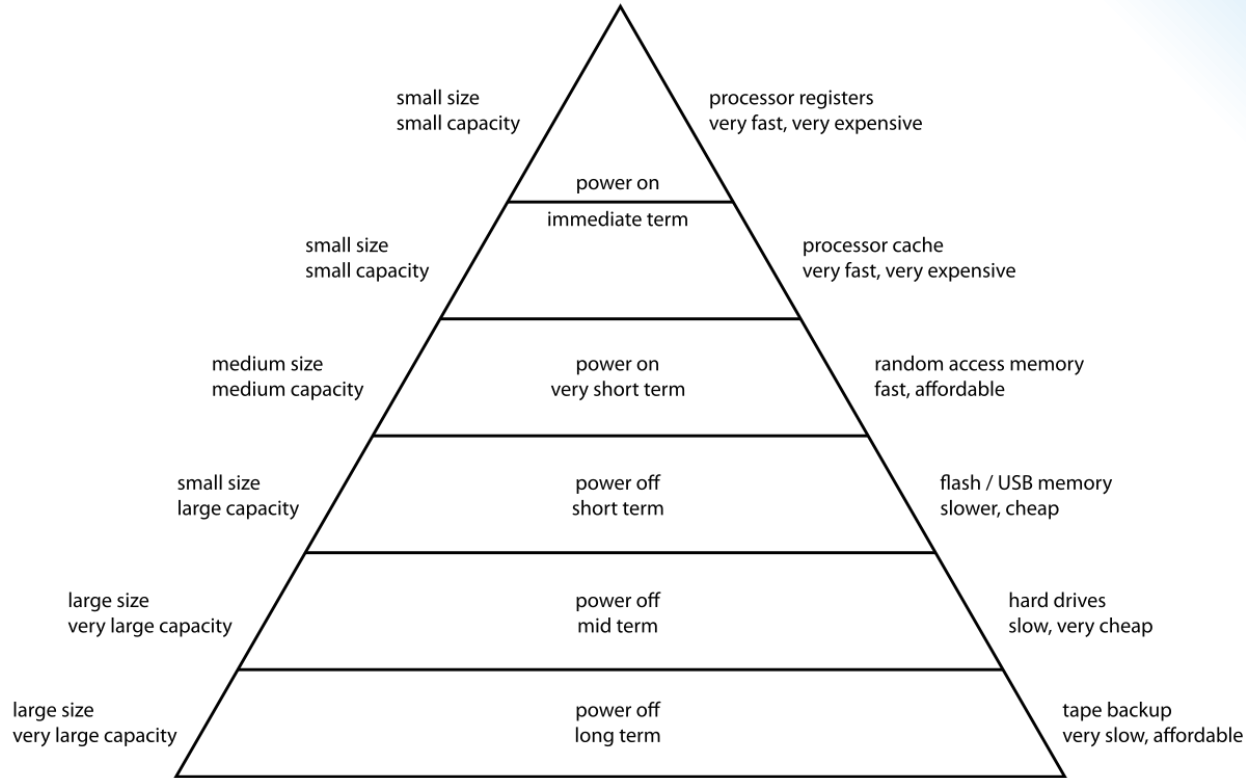
- 1) Instructions memory
- 2) Data memory



EPIC

Institute of Technology
Powered by <epam>

Computer Memory Hierarchy



SRAM





EPIC

Institute of Technology
Powered by <epam>

DRAM



**EPIC**Institute of Technology
Powered by <epam>

Comparison

Static Random Access Memory

SRAM is a type of RAM where each memory cell is made up of a set of transistors that store data in the form of charge.

It is more expensive and takes up more space on the chip than DRAM.

Dynamic Random Access Memory

Consists of a capacitor and a transistor that stores data in the form of charge on the capacitor.

It is cheaper and has a higher packing density but requires constant refreshing of the charge to maintain data.

DRAM is slower than SRAM.

**EPIC**Institute of Technology
Powered by <epam>

When to use?

SRAM

SRAM is used in cache memory and other places where fast access to data is required.

DRAM

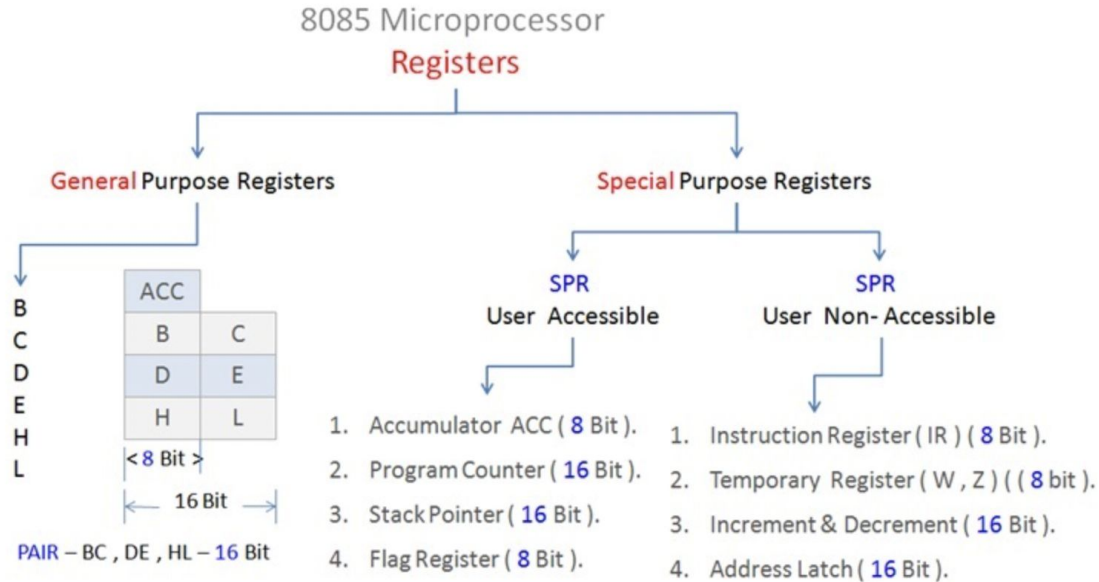
DRAM is used in the main computer memory and other devices where a large amount of memory is required at an affordable price.

04

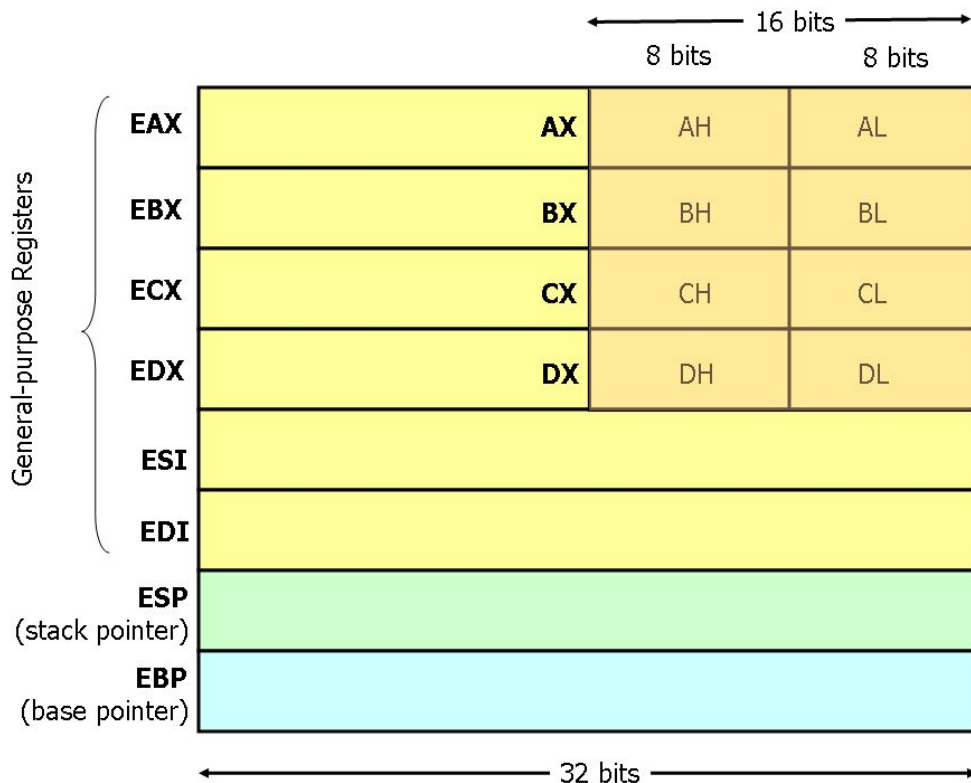
Registers

In this section, we will discuss registers.

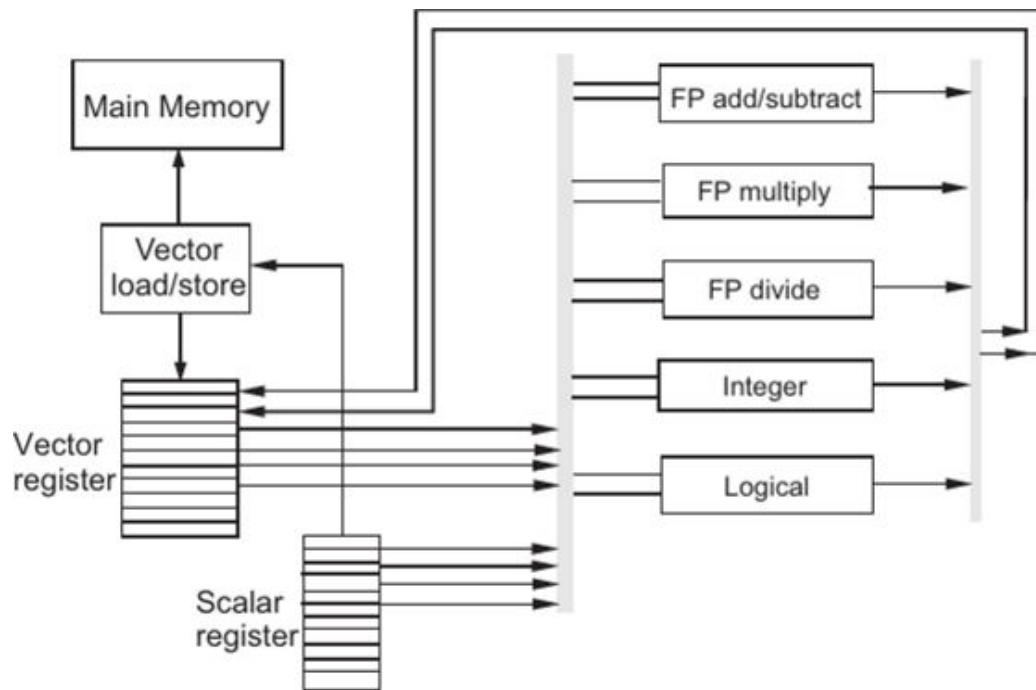
Registers



Registers



AVX

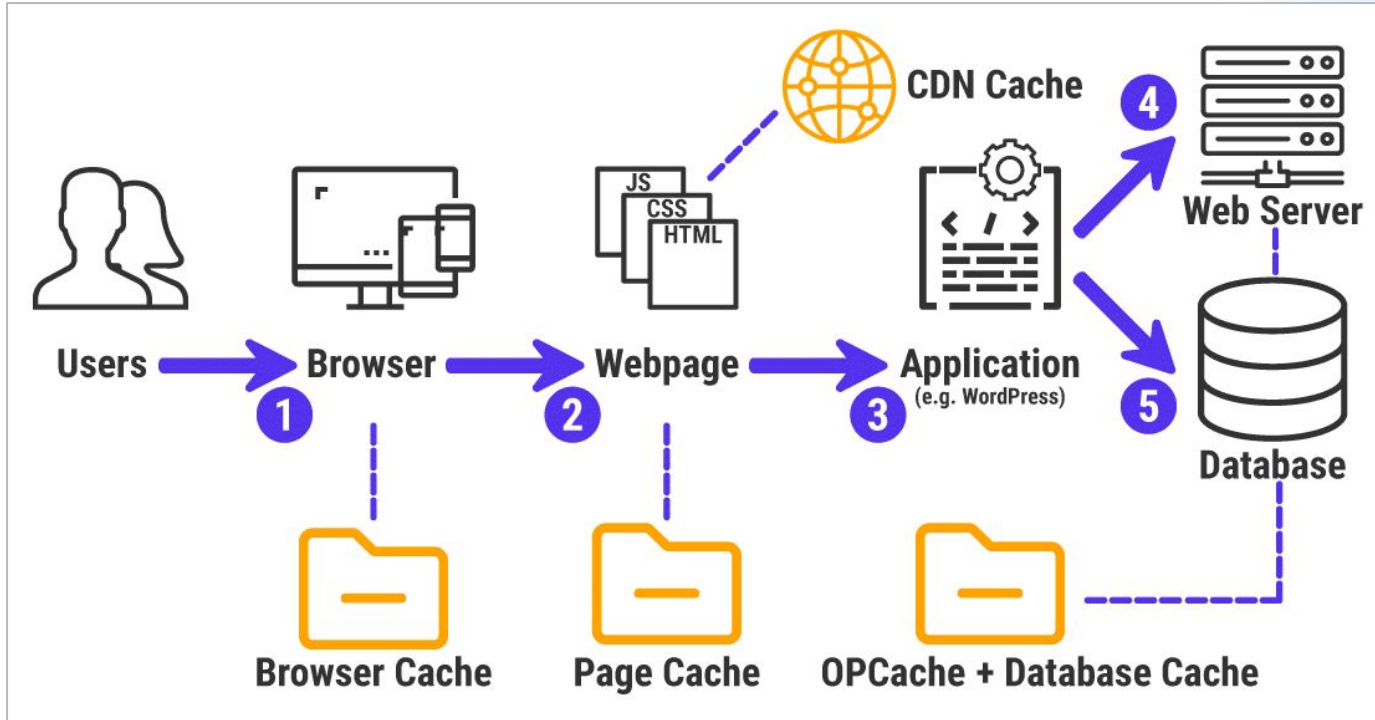


05

Caches

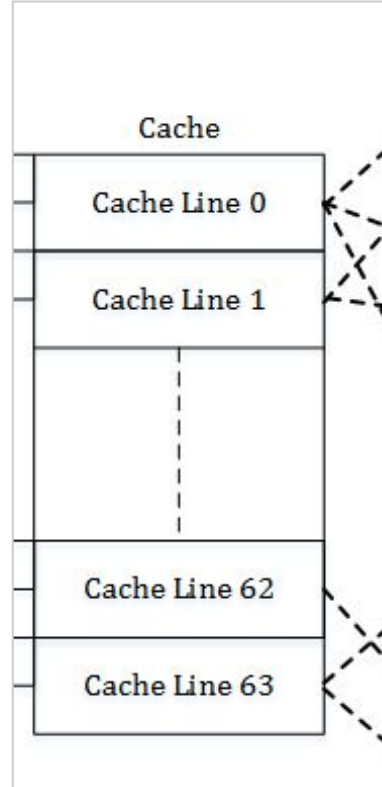
In this section, we will discuss caches.

Caches



Cache line

Usually like
system



Why cache is working?

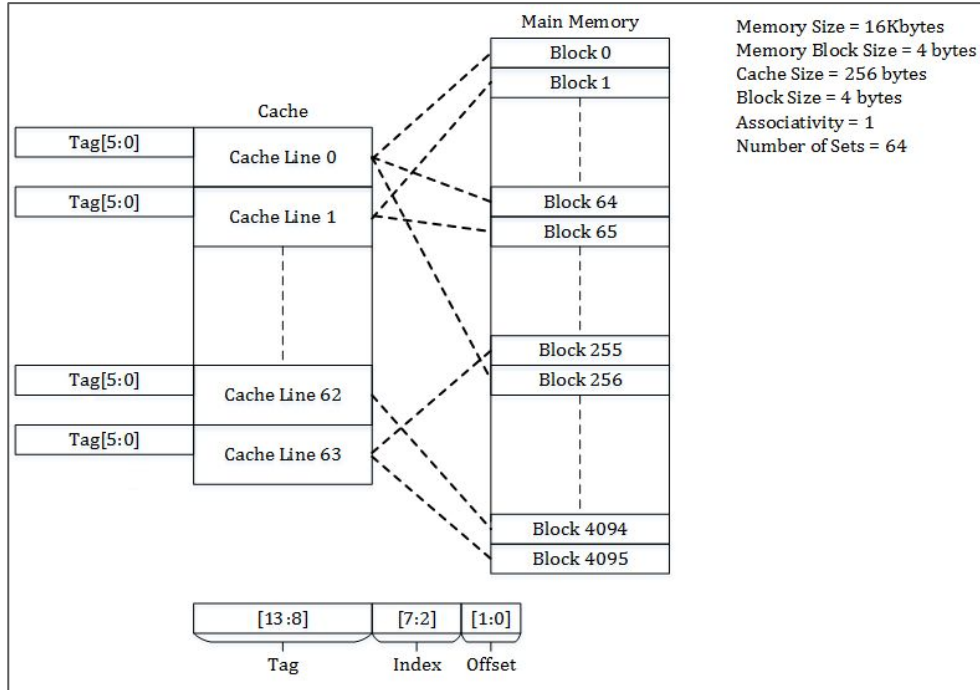
- 1) Time locality
- 2) Memory locality

How to cache?

- 1) Direct-mapped cache
- 2) Fully associative cache
- 3) Set-associative cache
- 4) Two-way skewed associative cache*
- 5) Pseudo-associative cache*

maybe some others

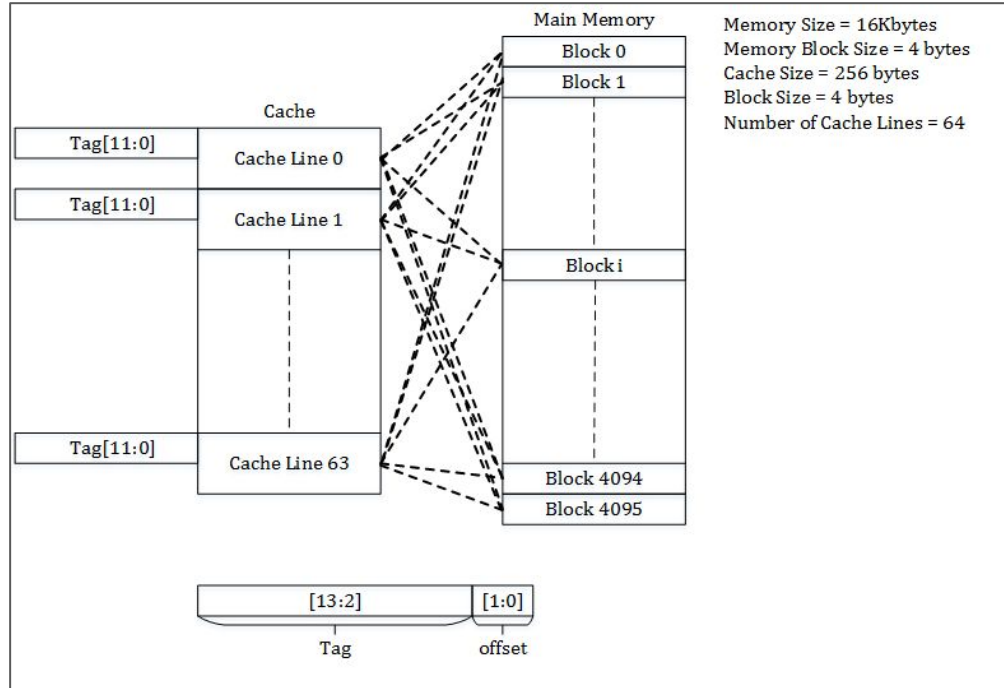
Direct-mapped cache



search 300 block:

- 1) check cache line with id $(300 \bmod 64) = 44$
- 2) check tag of cache line with id 44
- 3) if tag = 4 then take element from cache

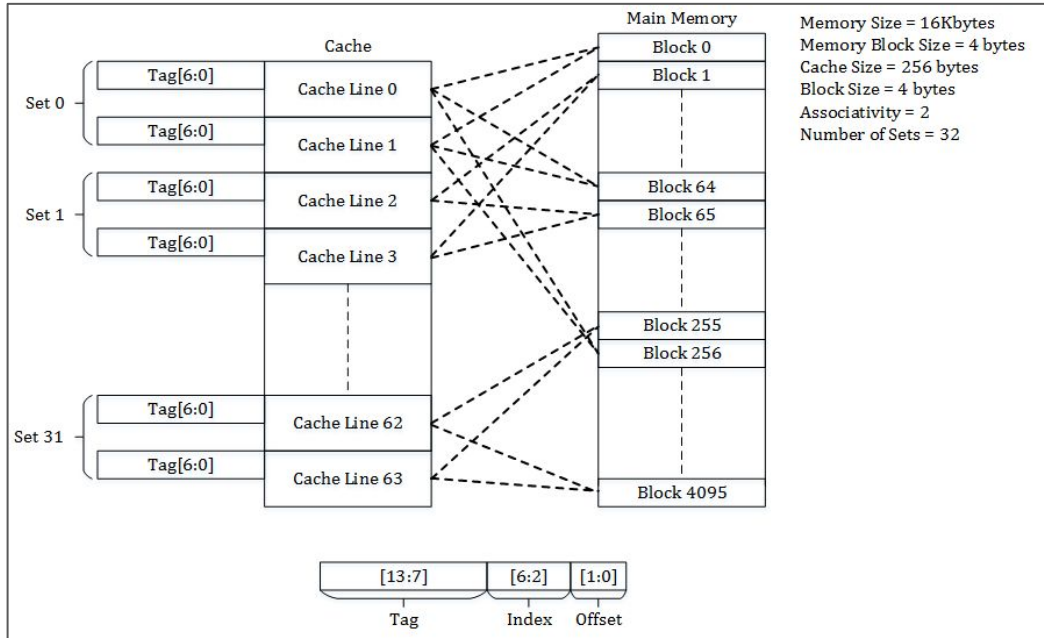
Fully associative cache



search 300 block:

- 1) check all tags
- 2) if we have our block number return result

Set-associative cache



search 300 block:

- 1) check set with id $(300 \bmod 32) = 12$
- 2) check both tags in it
- 3) if find one -> return

n-way cache

1-way

8-sets
1 block each

| |
|---|
| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |

direct-map
ped

2-way

4-sets
2 blocks each

| |
|---|
| 0 |
| 1 |
| 2 |
| 3 |

4-way

2-sets
4 blocks each

| |
|---|
| 0 |
| 1 |

8-way

1-set
8 blocks each

| |
|---|
| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |

fully
associative

Real Life



CPU-Z Ver. 2.05.1.x64

CPU | Mainboard | Memory | SPD | Graphics | Bench | About

Processor

| | | | |
|------------|----------------------|----------|---------|
| Name | AMD Ryzen 9 7950X | | |
| Code Name | Raphael | Max TDP | 170.0 W |
| Package | Socket AM5 (LGA1718) | | |
| Technology | 5 nm | Core VID | 1.319 V |

Specification

AMD Ryzen 9 7950X 16-Core Processor

| | | | | | |
|-------------|----|------------|----|----------|--------|
| Family | F | Model | 1 | Stepping | 2 |
| Ext. Family | 19 | Ext. Model | 61 | Revision | RPL-B2 |

Instructions MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, AVX512F, FMA3, SHA

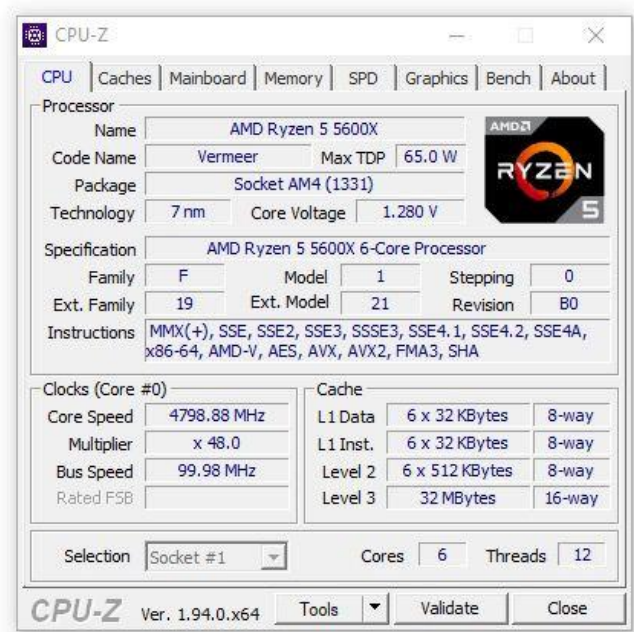
Clocks (Core #0)

| | |
|------------|---------------------|
| Core Speed | 3590.25 MHz |
| Multiplier | x 36.0 (4.0 - 58.5) |
| Bus Speed | 99.73 MHz |
| Rated FSB | |

Cache

| | | |
|----------|----------------|--------|
| L1 Data | 16 x 32 KBytes | 8-way |
| L1 Inst. | 16 x 32 KBytes | 8-way |
| Level 2 | 16 x 1 MBytes | 8-way |
| Level 3 | 2 x 32 MBytes | 16-way |

Selection: Socket #1 | Cores: 16 | Threads: 32



CPU-Z Ver. 1.94.0.x64

CPU | Caches | Mainboard | Memory | SPD | Graphics | Bench | About

Processor

| | | | |
|------------|-------------------|--------------|---------|
| Name | AMD Ryzen 5 5600X | | |
| Code Name | Vermeer | Max TDP | 65.0 W |
| Package | Socket AM4 (1331) | | |
| Technology | 7 nm | Core Voltage | 1.280 V |

Specification

AMD Ryzen 5 5600X 6-Core Processor

| | | | | | |
|-------------|----|------------|----|----------|----|
| Family | F | Model | 1 | Stepping | 0 |
| Ext. Family | 19 | Ext. Model | 21 | Revision | B0 |

Instructions MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, FMA3, SHA

Clocks (Core #0)

| | |
|------------|-------------|
| Core Speed | 4798.88 MHz |
| Multiplier | x 48.0 |
| Bus Speed | 99.98 MHz |
| Rated FSB | |

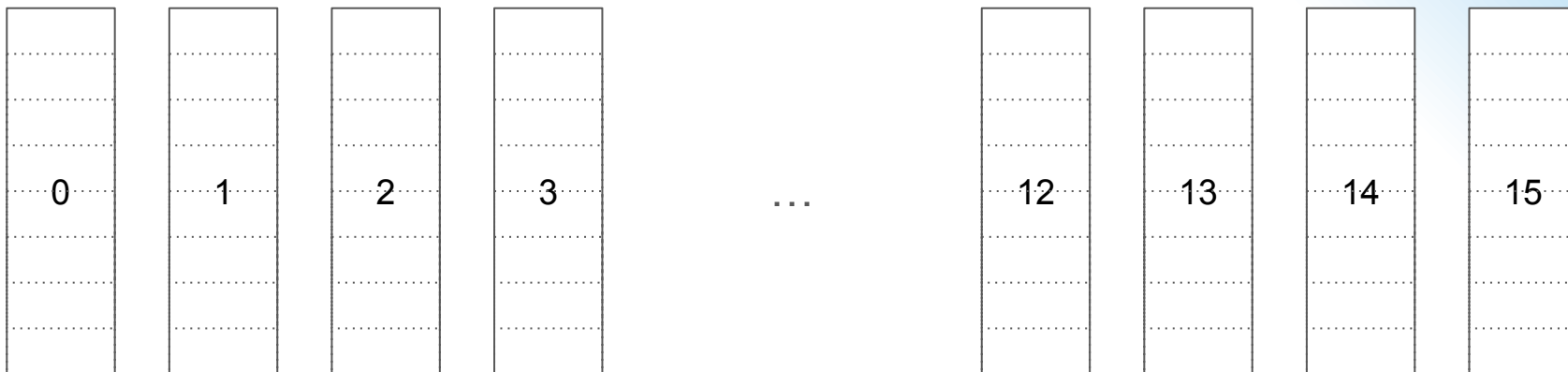
Cache

| | | |
|----------|----------------|--------|
| L1 Data | 6 x 32 KBytes | 8-way |
| L1 Inst. | 6 x 32 KBytes | 8-way |
| Level 2 | 6 x 512 KBytes | 8-way |
| Level 3 | 32 MBytes | 16-way |

Selection: Socket #1 | Cores: 6 | Threads: 12

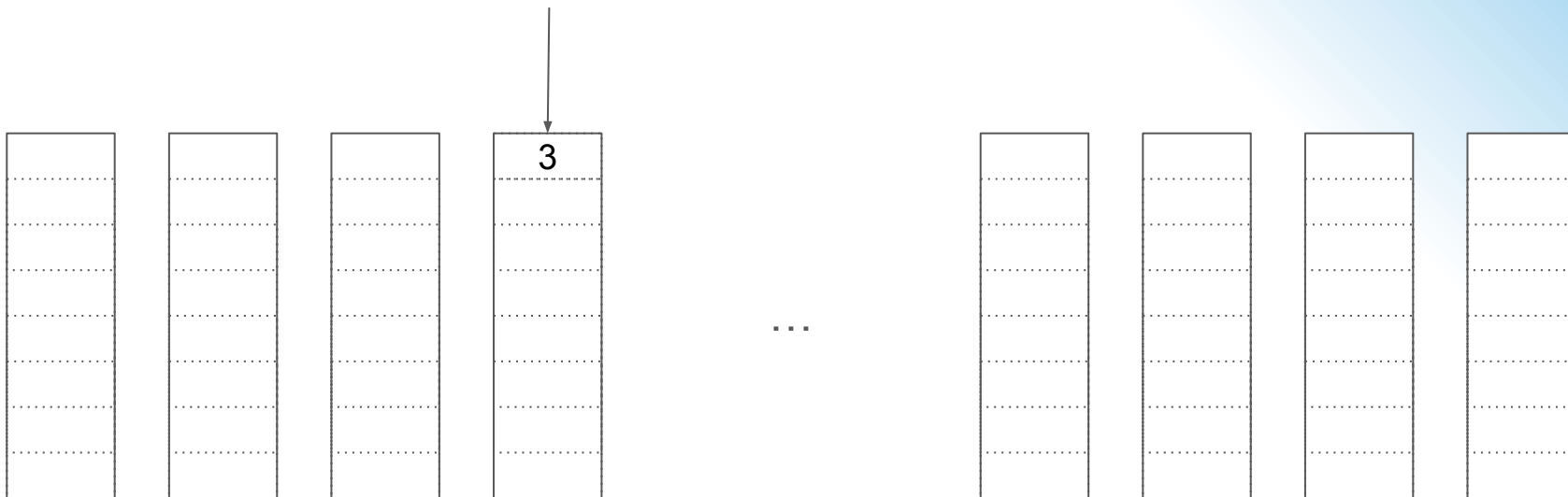
Explanation(L1)

line size = 4 kb
32-bit



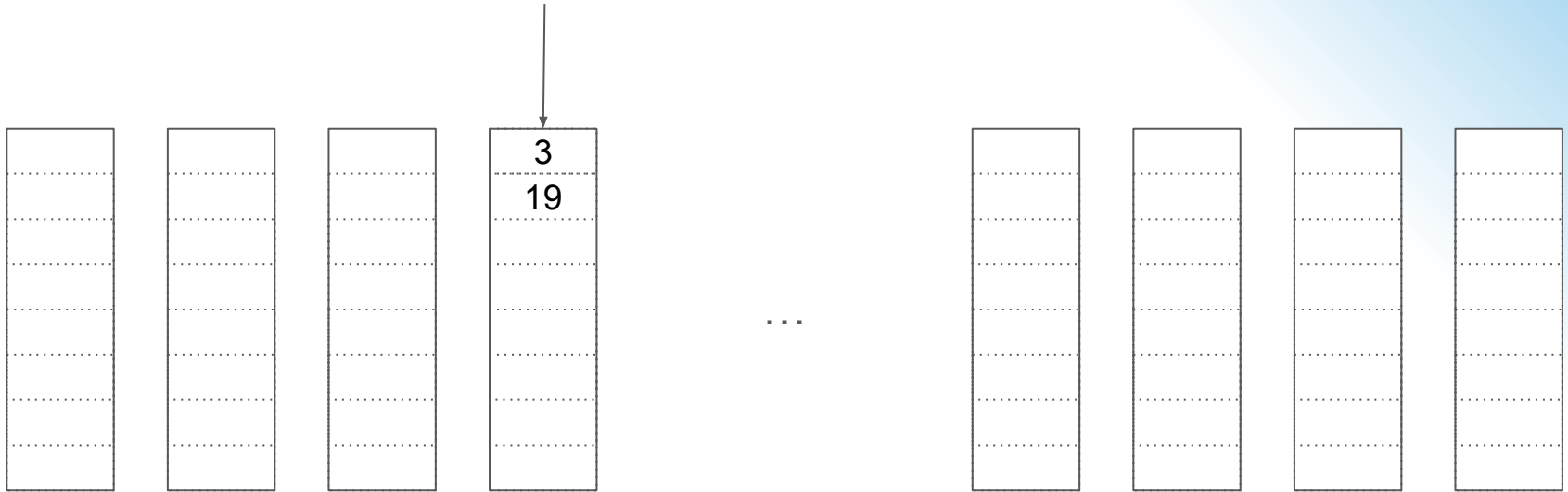
Explanation(L1)

Insert 3
 $3 \bmod 16 = 3$



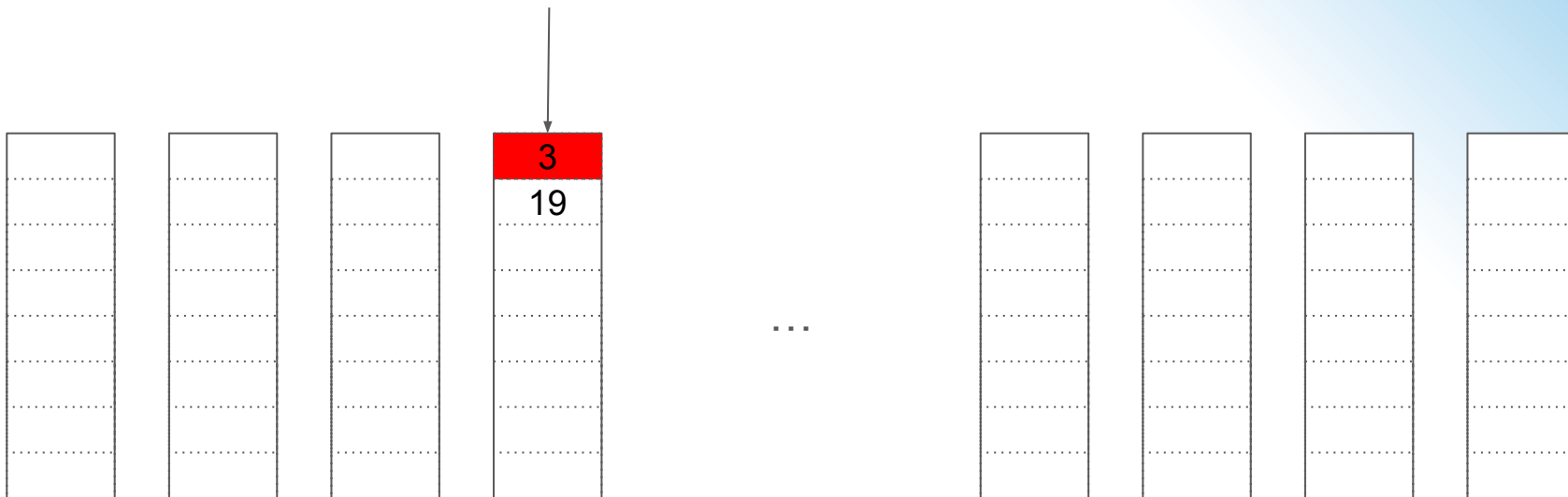
Explanation(L1)

Insert 19
 $3 \bmod 16 = 3$



Explanation(L1)

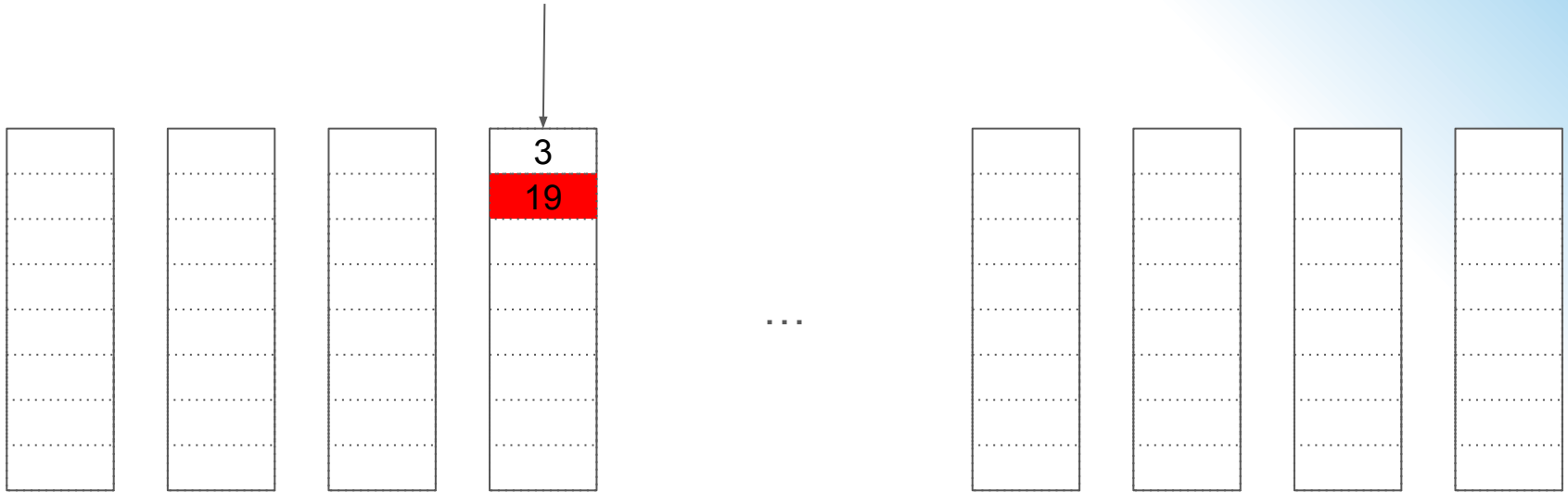
Find 35
 $35 \bmod 16 = 3$



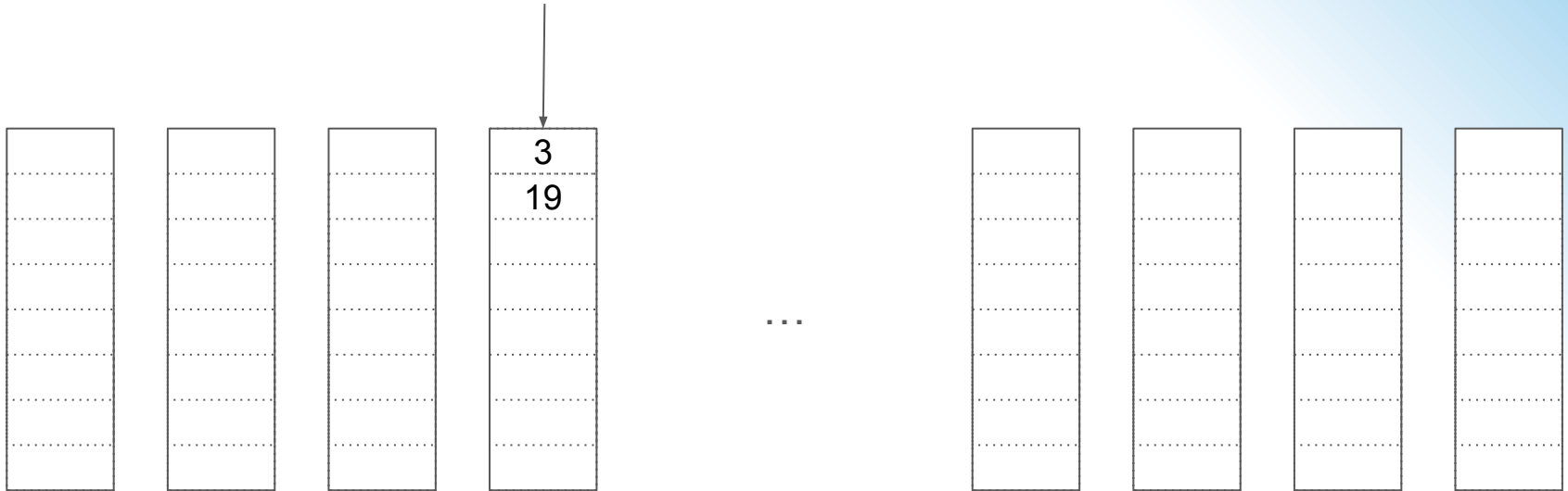


Explanation(L1)

Find 35
 $35 \bmod 16 = 3$



Explanation(L1)



What to keep in cache?

- 1) LRU (Least Recently Used)
- 2) FIFO (First-In, First-Out)
- 3) LFU (Least Frequently Used)
- 4) MRU (Most Recently Used)
- 5) Random
- 6) LRU-k

maybe some others

**EPIC**Institute of Technology
Powered by epan>

LRU

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 1 | 3 | 1 | 6 | 3 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 2 |
| | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | | | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| M | M | M | M | M | M | H | H | M | H | M | H |
|---|---|---|---|---|---|---|---|---|---|---|---|

M = Miss**H = Hit**



EPIC

Institute of Technology
Powered by ePam

FIFO

Page reference stream:

1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3

1 1 1 1 1 2 2 3 5 1 6 6 2 5 5 3 3 1 6 2

2 2 2 2 3 3 5 1 6 2 2 5 3 3 1 1 6 2 4

3 3 3 5 5 1 6 2 5 5 3 1 1 6 6 2 4 3

* * * * * * * * * * * * *

FIFO

Total 14 page faults

LFU

Least Frequently Used (LFU)

X X

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|
| 7 | 7 | 7 | 2 | 2 | 2 | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | | | | | | | | | |
| | | 1 | 1 | 1 | 3 | | | | | | | | | |
| F | F | F | F | | F | | | | | | | | | |

Frequencies :

7 = 0 0 = 2 1 = 0 2 = 1 3 = 1 4 = 0

**EPIC**Institute of Technology
Powered by epan

LRU-k

