



Lecture: 33

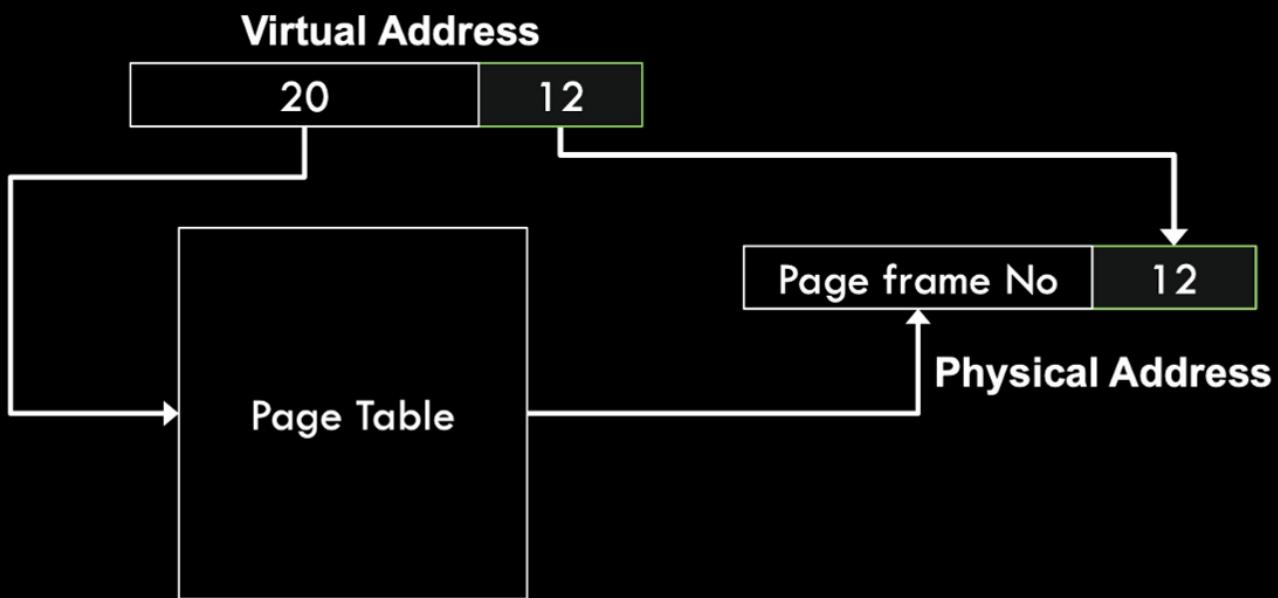
→ TLB

- dynamic memory allocation strategies
 - ↳ heap/ segmentation (variable size partitions)

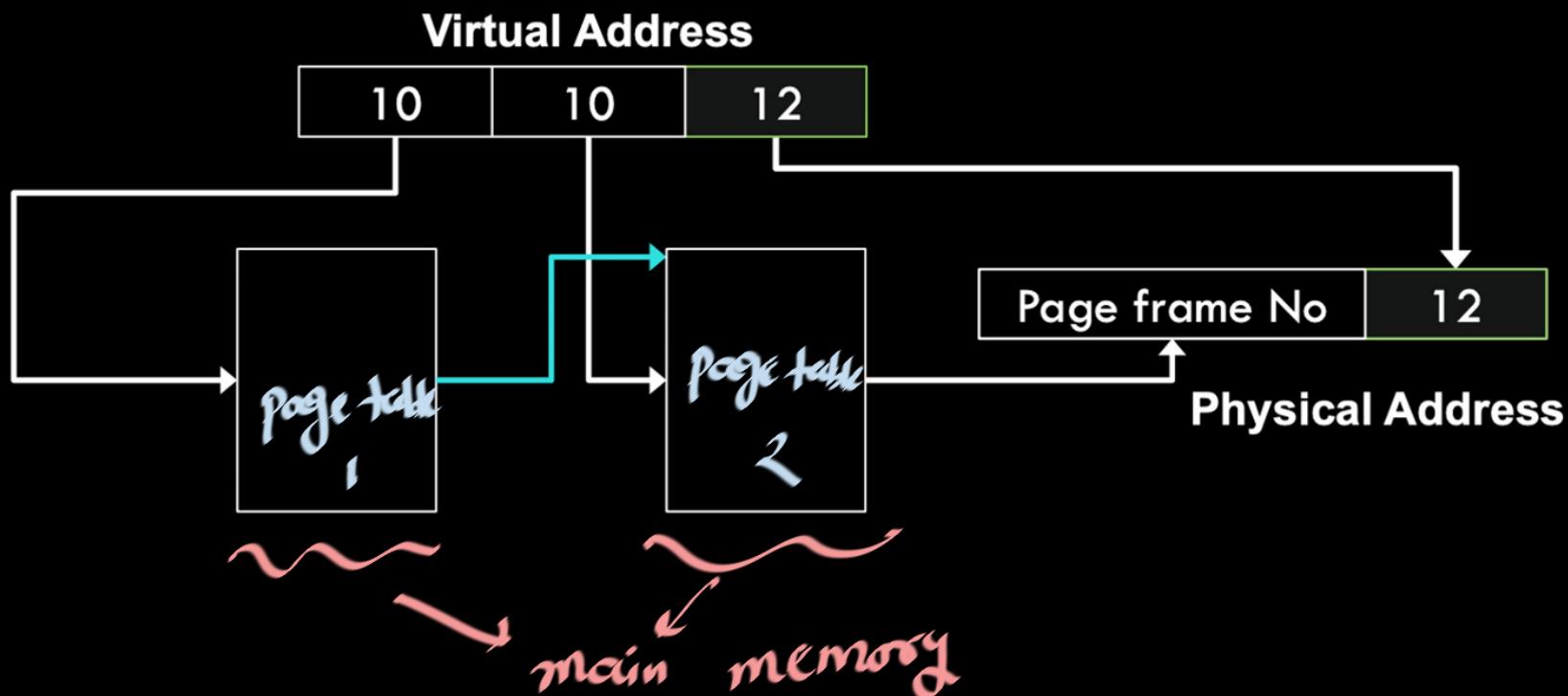


TLB: Translation Look aside Buffer

Address Translation

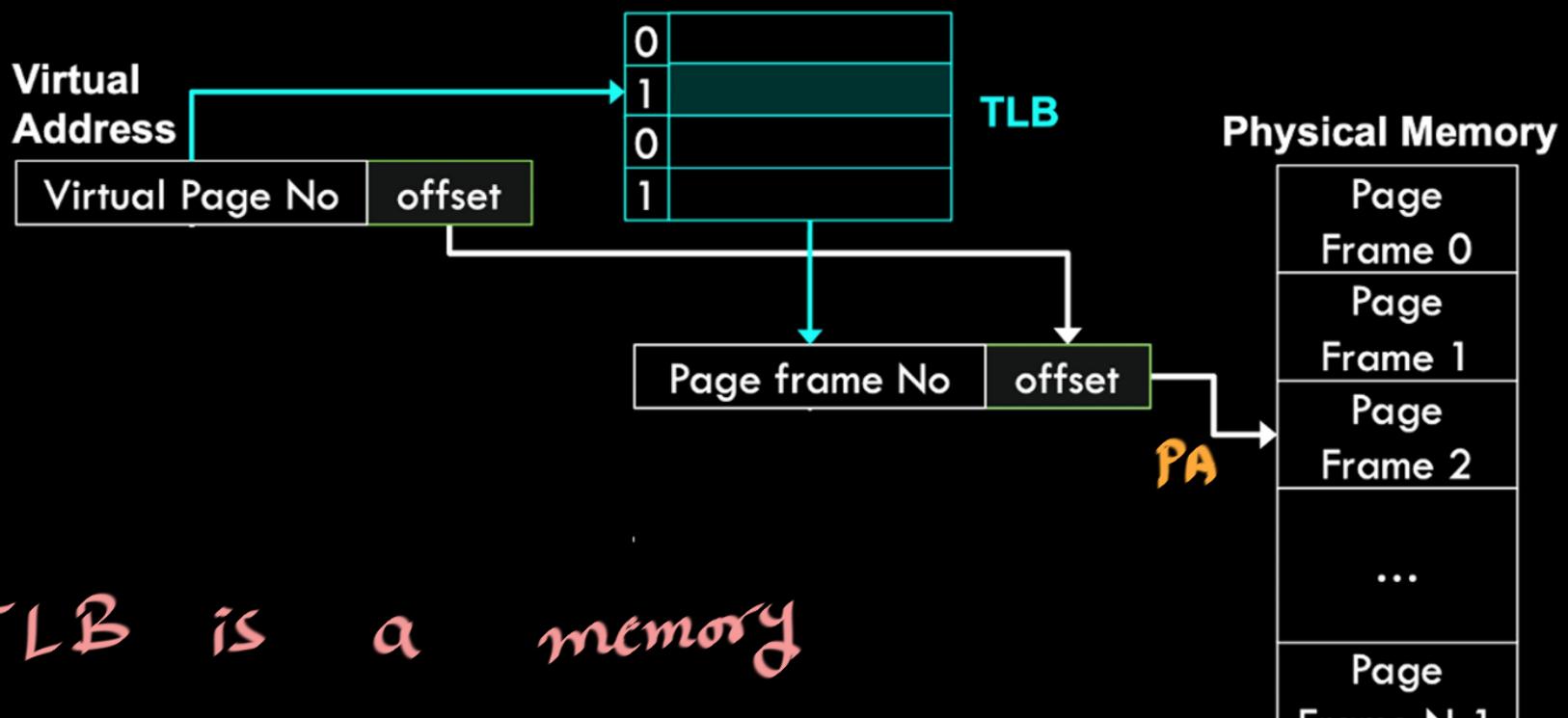


Multi-Level Page Table Address Translation

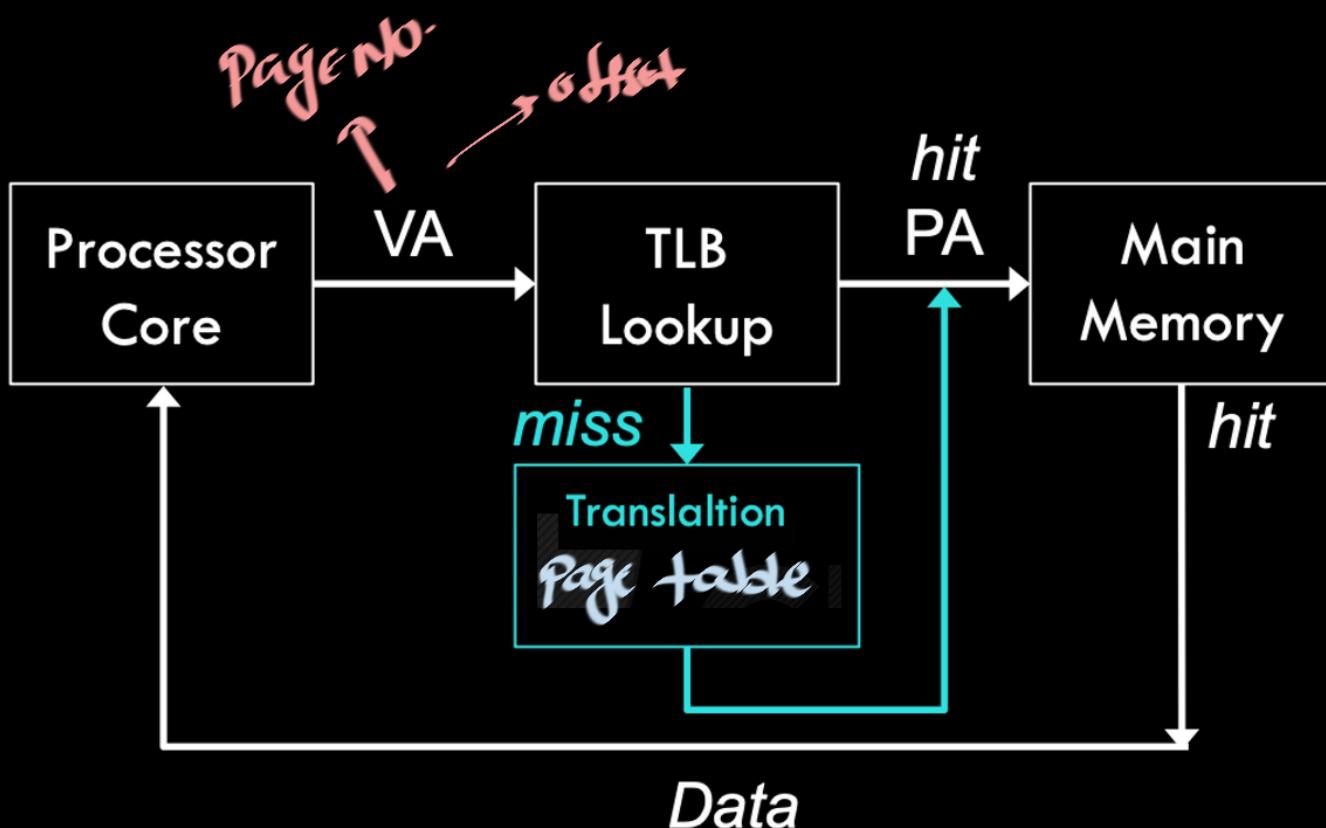


TLB has few page table entries





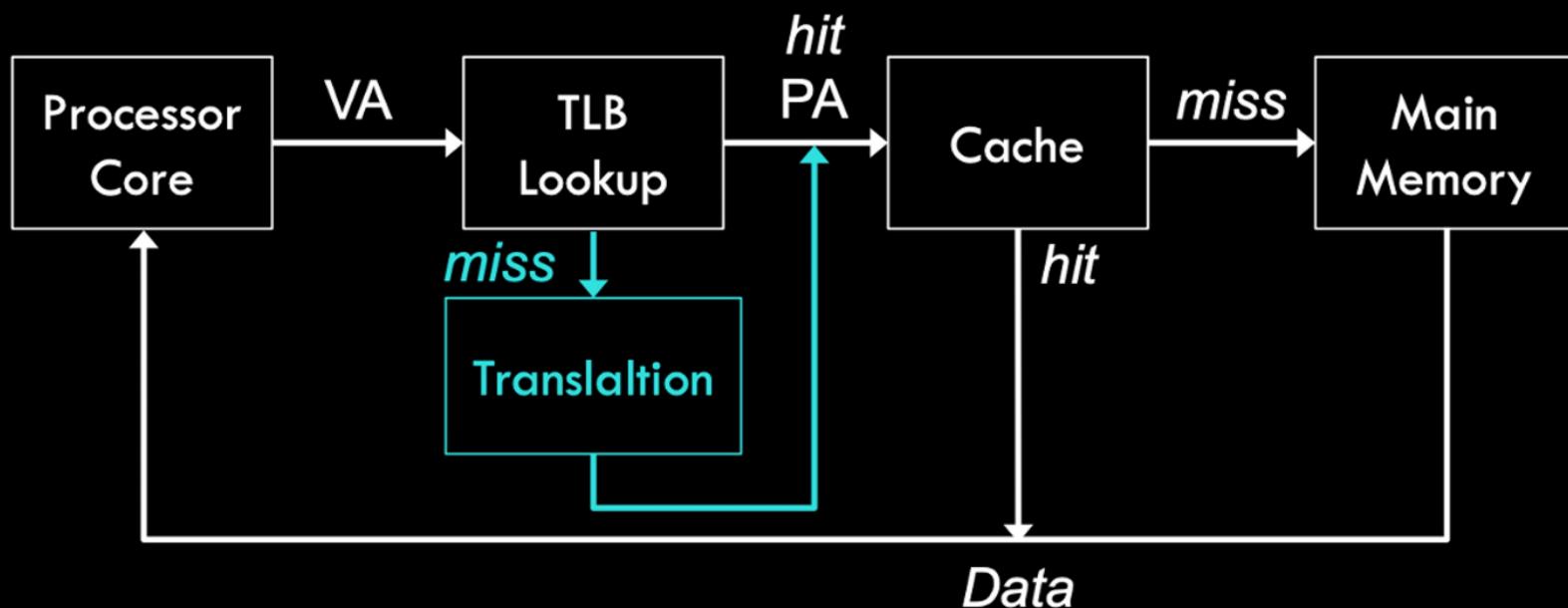
TLB is a memory
which is way faster than
main memory.



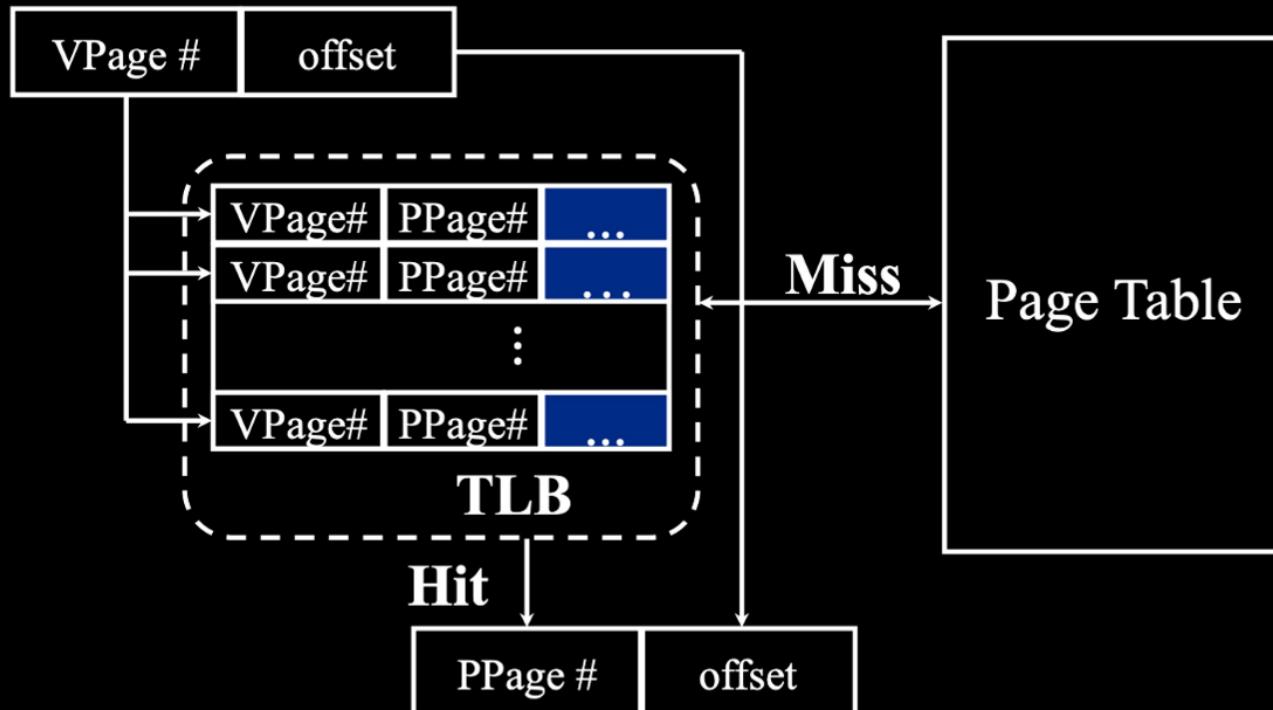


Operating Systems

Optional



Virtual address



<https://www.cs.princeton.edu/courses/archive/fall10/cos318/lectures/VM Paging.pdf>



Why TLB Works ?





Why TLB Works ?

- **Locality:** Programs tend to use data and instructions with addresses near or equal to those they have used recently

CLASSES



Why TLB Works ?

- **Locality:** Programs tend to use data and instructions with addresses near or equal to those they have used recently
 - Temporal locality
 - Spatial locality



Why TLB Works ?

■ Temporal locality:

- Recently referenced items are *likely* to be referenced again in the near future



memory
location



Why TLB Works ?

■ Temporal locality:

- Recently referenced items are *likely* to be referenced again in the near future



■ Spatial locality:

- Items with nearby addresses *tend* to be referenced close together in time
- How do caches take advantage of this?





Example: Locality?

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

Which of these memory locations has temporal locality in this code ?

- i
- sum
- Elements of the array **a[0], a[1]**



Example: Locality?

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

Which of these memory locations has temporal locality in this code ?

- i
- sum
- Elements of the array *a[0], a[1]*



Example: Locality?

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

Which of these memory locations has spatial locality in this code ?

- i
- sum
- Elements of the array



Example: Locality?

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

Which of these memory locations has spatial locality in this code ?

- i
- sum
- Elements of the array **a[0] a[i]**



Operating Systems

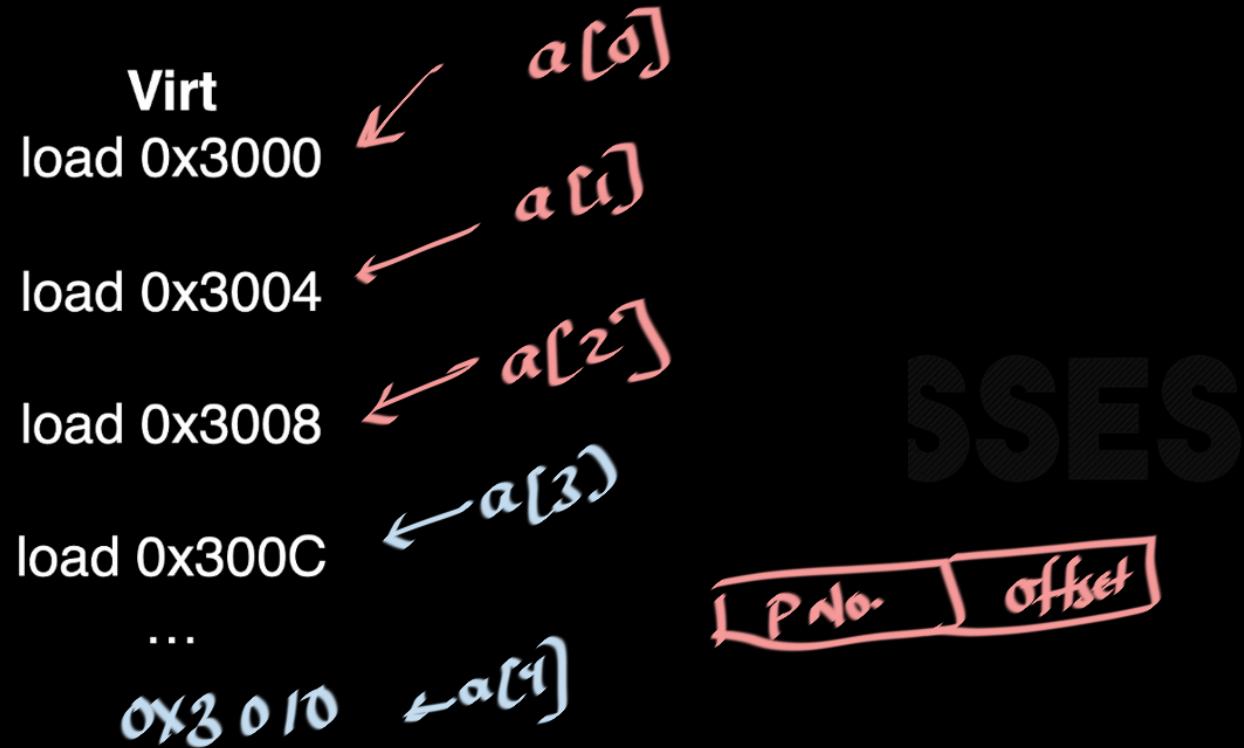
- A simple code snippet in array.c

```
int sum = 0;  
for (i=0; i<N; i++) {  
    sum += a[i];  
}
```

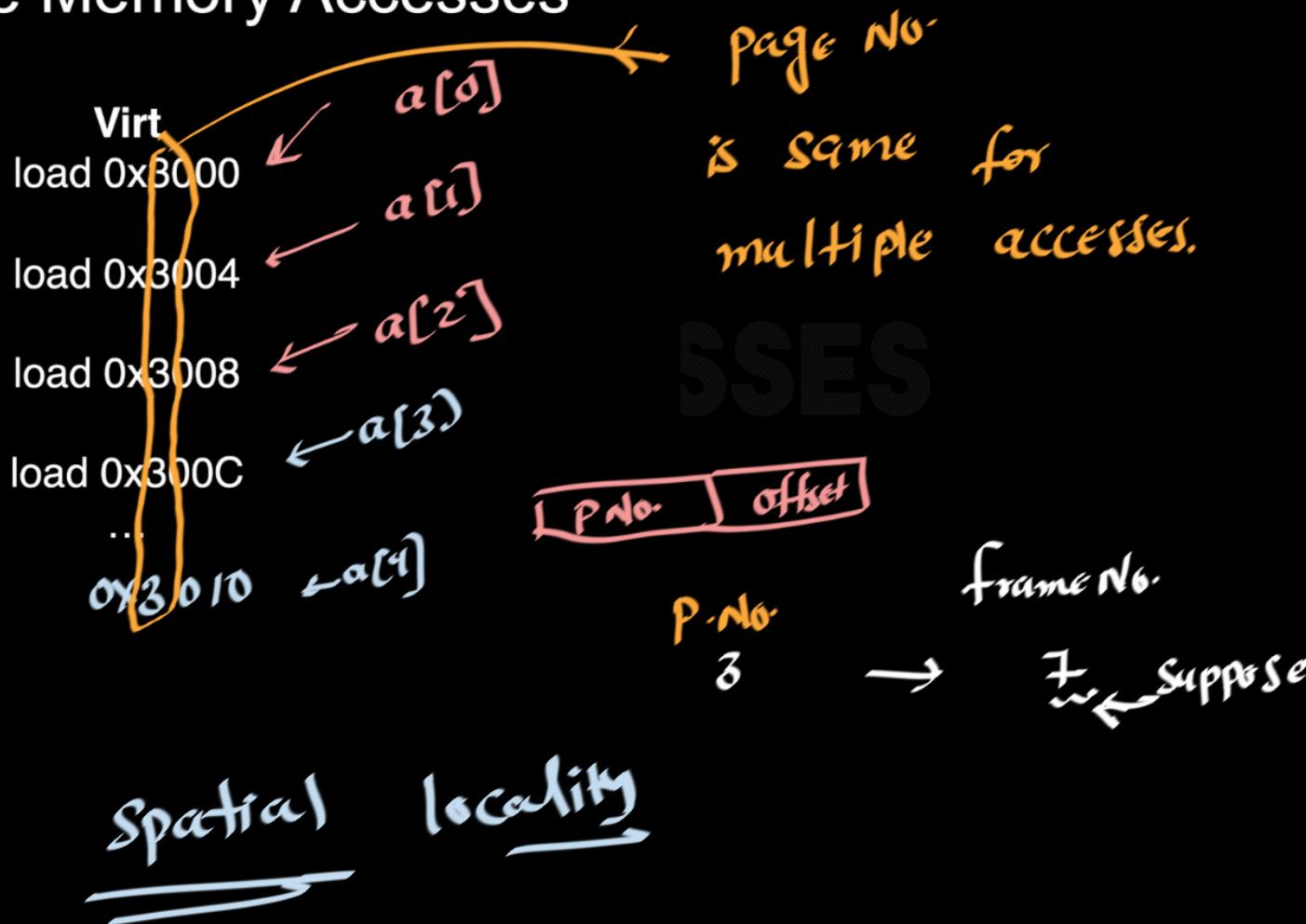
○ SES
CLAS:



Trace the Memory Accesses



Trace the Memory Accesses



TLB	
P.No	frame No.
3	7



How Many TLB Lookups

Assume 4KB pages

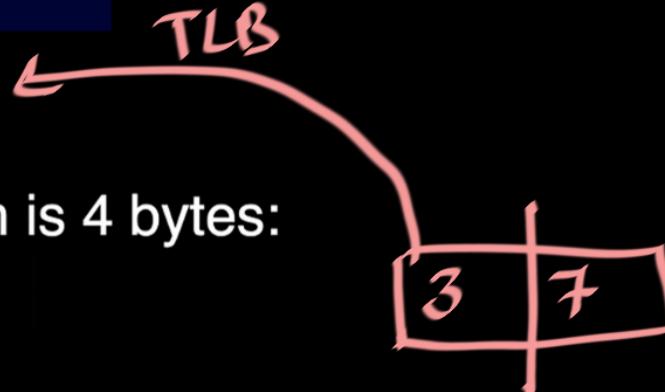
$$4 \times 2^{10} = 2^{12} B$$

```
int sum = 0;  
for (i=0; i<1024; i++) {  
    sum += a[i];  
}
```

Size of array = 4×1024

Array a[] has 1024 items, each item is 4 bytes:

Num of TLB miss = $\frac{1}{1024}$



$$\begin{aligned} &= 4096 B \\ &= 2^{12} B \end{aligned}$$

How Many TLB Lookups

Assume 4KB pages

$$4 \times 2^{10} = 2^{12} \text{ B}$$

```
int sum = 0;
for (i=0; i<1024; i++) {
    sum += a[i];
}
```

Array a[] has 1024 items, each item is 4 bytes:

Num of TLB miss =

1, 2
~

TLB

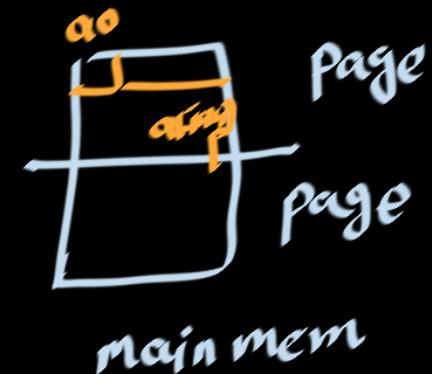
Size of array = 4×1024

$= 4096 \text{ B} = \text{Page Size}$

$$= 2^{12} \text{ B}$$



if data is spanned across **two** pages





How Many TLB Lookups

Assume 4KB pages

```
int sum = 0;  
for (i=0; i<1024; i++) {  
    sum += a[i];  
}
```

SES

Array a[] has 1024 items, each item is 4 bytes:

Size(a) = 4096

Num of TLB miss =



How Many TLB Lookups

Assume 4KB pages

```
int sum = 0;  
for (i=0; i<1024; i++) {  
    sum += a[i];  
}
```

SES

Array a[] has 1024 items, each item is 4 bytes:

Size(a) = 4096

Num of TLB miss: $4096/4096 = 1 \text{ or } 2$



How Many TLB Lookups

Assume 4KB pages

```
int sum = 0;  
for (i=0; i<1024; i++) {  
    sum += a[i];  
}
```

Array a[] has 1024 items, each item is 4 bytes:

$$\text{Size}(a) = 4096$$

Best case: Num of TLB miss: $4096/4096 = 1$

TLB miss rate: $1/1024 = 0.09\%$

TLB hit rate : 99.91% (almost 100%)

CLASSES







Translation Lookaside Buffer

- Just like any other cache, the TLB can be organized as fully associative, set associative, or direct
- TLB access is typically faster than cache access
 - Because TLBs are much smaller than caches
 - TLBs are typically not more than 128 to 256 entries even on high-end machines

mostly it
SES is fully associative

V	Virtual Page #	Physical Page #	Dirty	Status



Question

VAS

Consider a machine with 64-bit addresses and an 8KB page size.

Suppose our TLB has 512 entries, and our machine has 4GB of RAM. What fraction of physical memory can be accessed without suffering a TLB miss?

$$2^2 \times 2^{30} = 2^{32}$$





Question





$$2^2 \times 2^{30} = 2^{32}$$



Consider a machine with 64-bit addresses and an 8KB page size.

Suppose our TLB has 512 entries, and our machine has 4GB of RAM. What fraction of physical memory can be accessed without suffering a TLB miss?



of different
pages we
can access = 512 pages

$$2^9 \text{ pages} \times 2^5 = 2^{14} \text{ pages}$$

https://www.cs.rochester.edu/courses/252/spring2014/exams/final_2009.pdf

$$\frac{2^{14} \text{ pages}}{2^{32} \text{ pages}}$$

$$= \frac{1}{2^8}$$

fraction
of MM without
TLB miss

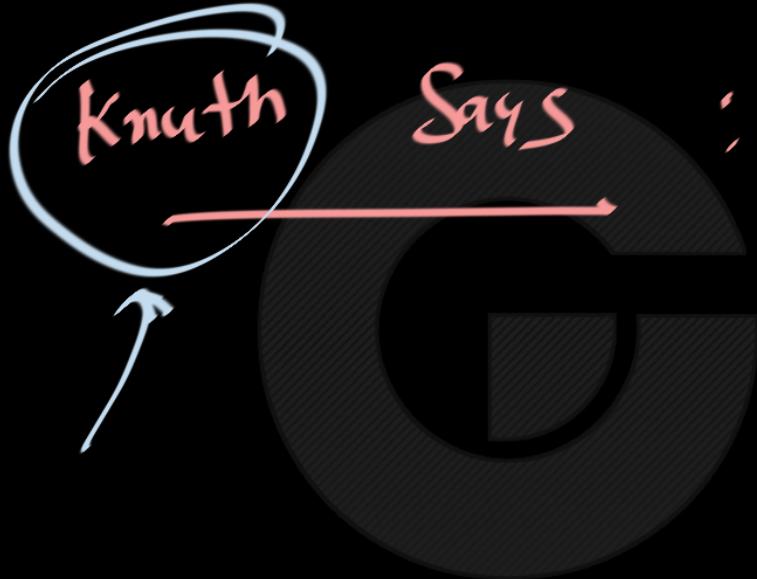


Solution

Consider a machine with 64-bit addresses and an 8KB page size.

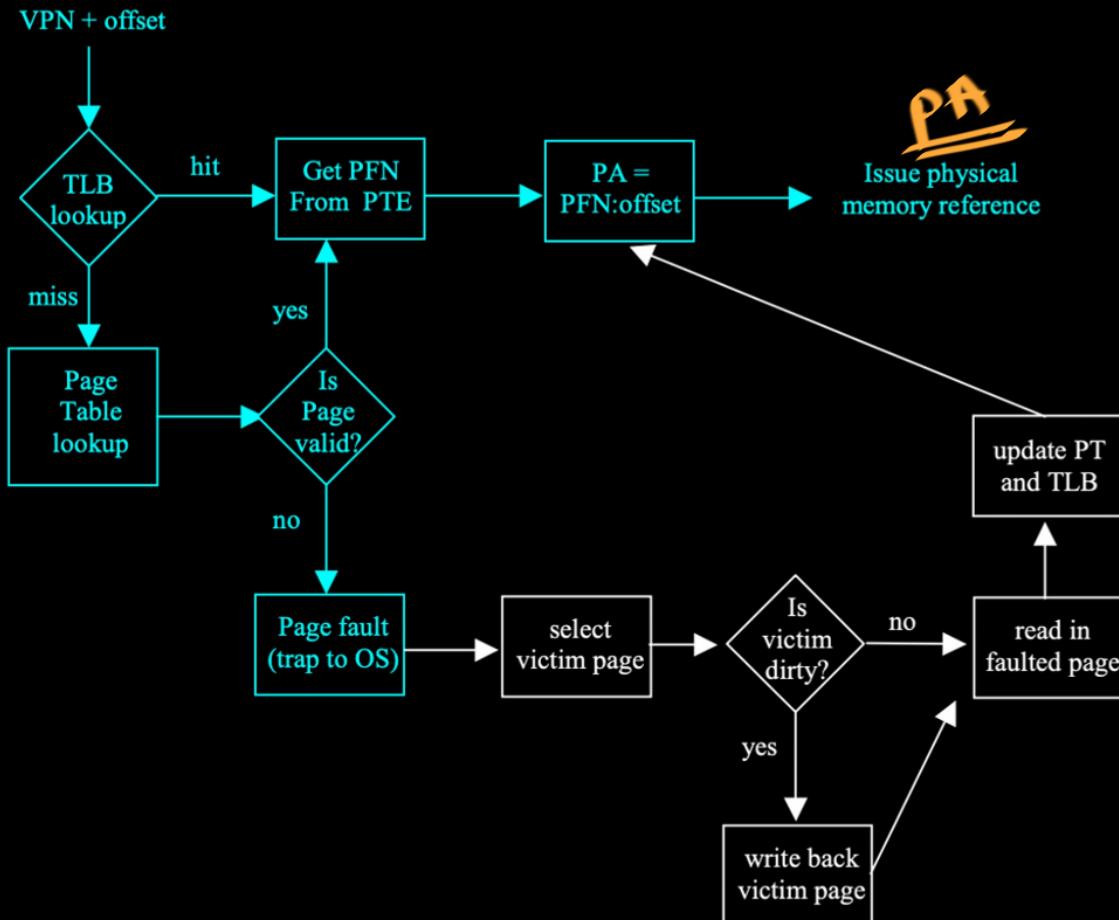
Suppose our TLB has 512 entries, and our machine has 4GB of RAM. What fraction of physical memory can be accessed without suffering a TLB miss?

Answer: 512 TLB entries can cover 4MB of memory, which is $1/1024$ -th of the physical memory of the machine. (Obviously we need a lot of temporal locality to use the TLB well.)



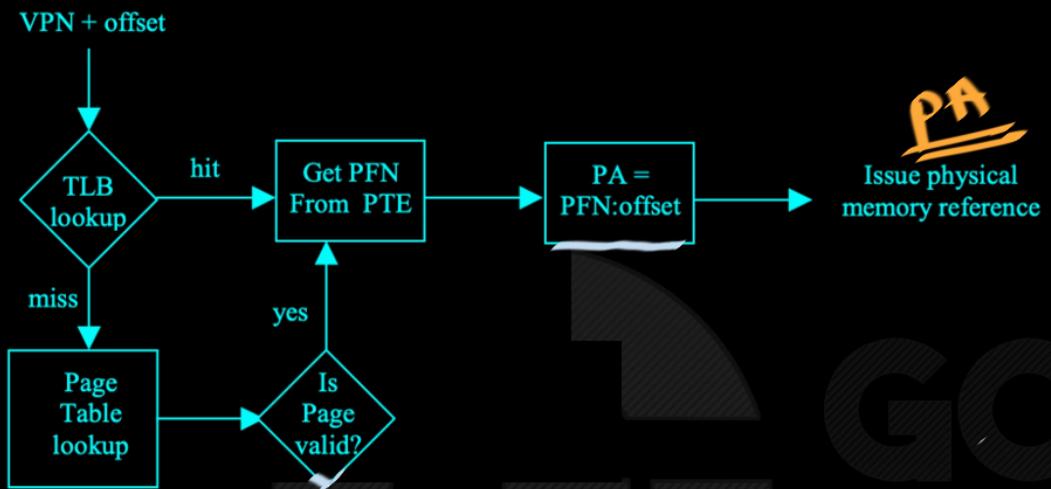
Knuth says: 98% of program's time
get spent in 1% of code.

Operating Systems

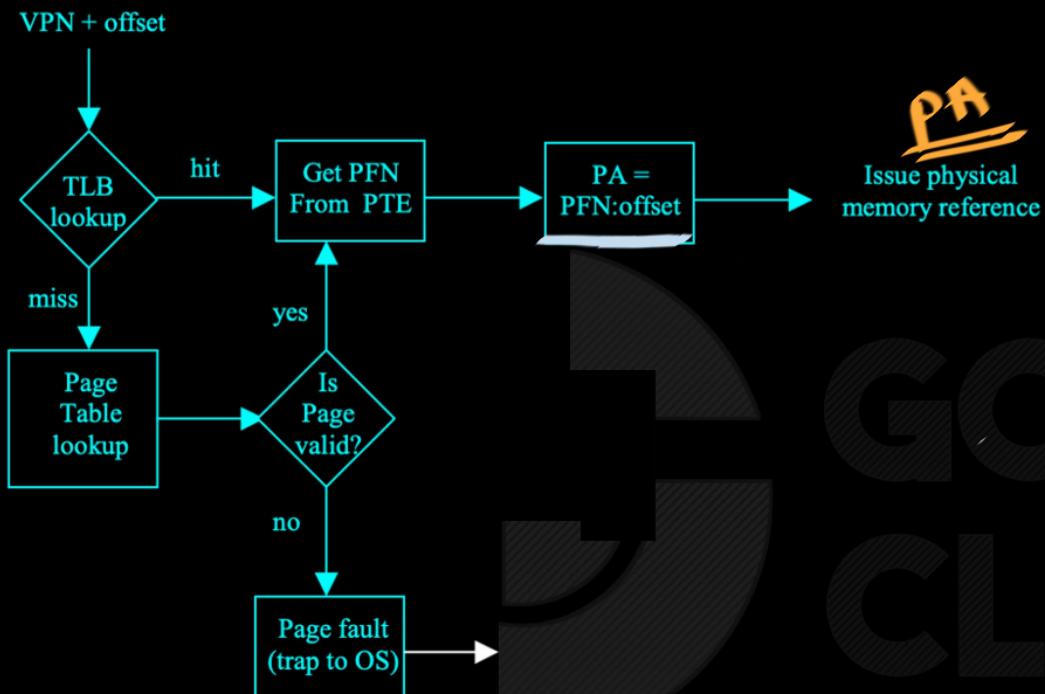


SSES

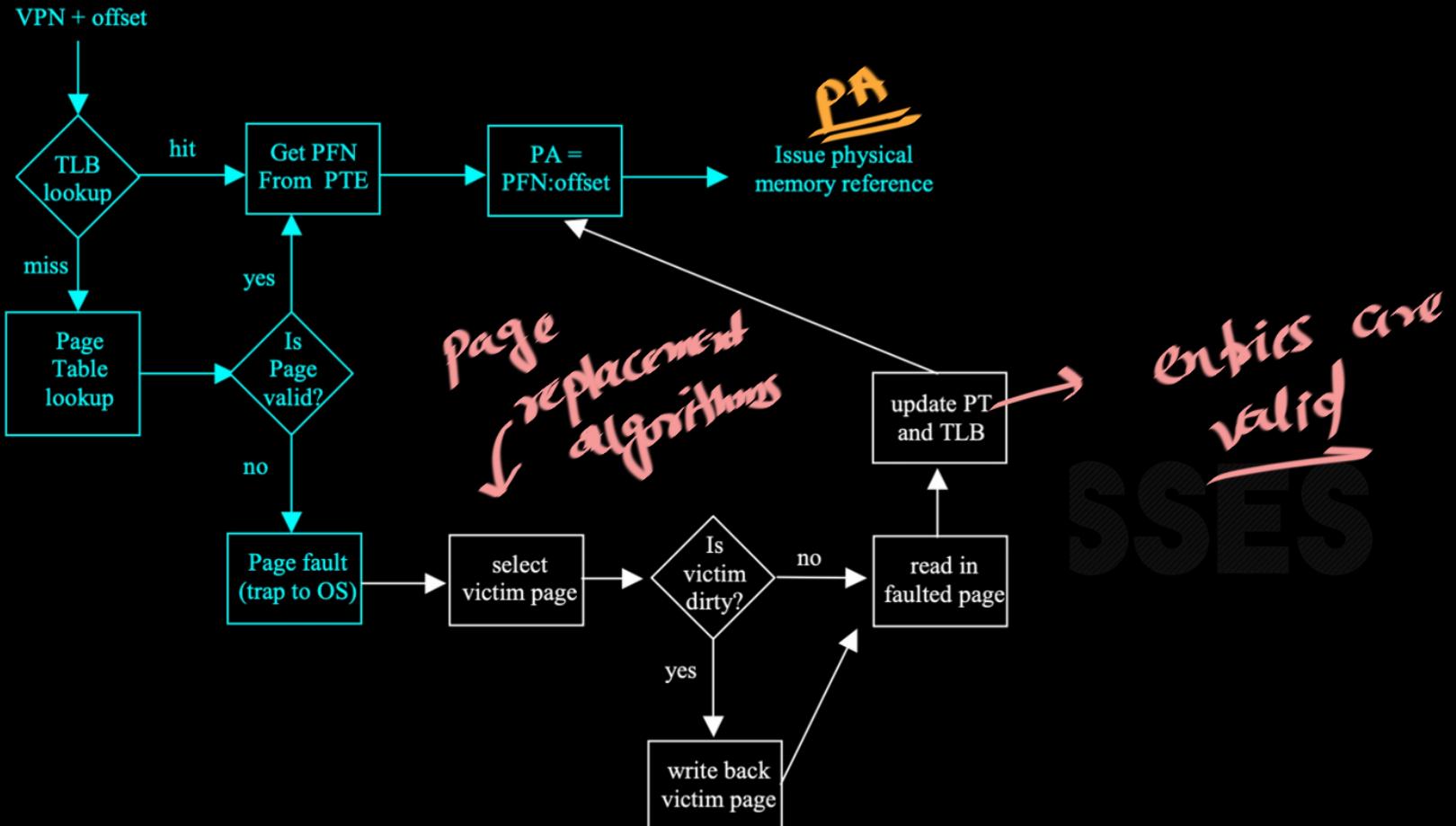
Operating Systems



Operating Systems



Operating Systems





GATE CSE 2014 Set 3 | Question: 33



36



Consider a paging hardware with a TLB . Assume that the entire page table and all the pages are in the physical memory. It takes 10 milliseconds to search the TLB and 80 milliseconds to access the physical memory. If the TLB hit ratio is 0.6, the effective memory access time (in milliseconds) is _____.

gatecse-2014-set3

operating-system

virtual-memory

numerical-answers

normal

$$\begin{aligned} \text{TLB access time} &= 10 \text{ ms} & TLB \text{ hit} &= 0.6 \\ \text{MMU } " &= 80 \text{ ms} & \text{TLB miss} &= 80 \text{ ms} \\ E_{MAT} &= 0.6 \times (10 + 80) + 0.4 \times (\text{data read/write}) \end{aligned}$$

<https://gateoverflow.in/2067/gate-cse-2014-set-3-question-33>

$$0.6 \left(\overbrace{10}^{\text{TLB}} + \overbrace{80}^{\text{data access}} \right) = 80 \times 0.6 + 80 \times 0.4 = 80$$

$$0.4 \left(\overbrace{10}^{\text{TLB}} + \overbrace{80}^{\text{page table}} + \overbrace{80}^{\text{data access}} \right) = 80 \times 0.4 + 80 \times 0.4 + 80 \times 0.4 = 80$$

$$0.6 \left(\overbrace{10}^{\text{TLB}} + \overbrace{80}^{\text{data access}} \right) = 80 \times 0.6 + 80 \times 0.4$$

$$0.4 \left(\overbrace{10}^{\text{TLB}} + \overbrace{80}^{\text{page table}} + \overbrace{80}^{\text{data access}} \right) = 8$$

$$\underline{10 + 80} + \underline{\underline{0.4 \times 80}} = 122 \text{ ms}$$

VA → PA + data access

0.6(10) + 0.4(10 + 80) + 80

32 + 10 + 80 = 122 ms

$$\begin{aligned}
 & \underbrace{\text{VA} \rightarrow \text{PA}}_{\downarrow} + \underbrace{\text{Data access time}}_{= 80 \text{ ms}} \\
 & 10 + 0.4 \times 80 \\
 & = 10 + 32 + 80 = \underline{\underline{122}} \text{ ms}
 \end{aligned}$$

~~**EXTRA**~~

earlier

$$\underbrace{VA \rightarrow PA}_{\downarrow} + \frac{\text{Data access time}}{= 80 \text{ ms}}$$

$$10 + \frac{0.4 \times 80}{= 10 + 32 + 80 = } + 80$$

$$\underline{\underline{122 \text{ ms}}}$$

$$0.6 \left(\underline{10} \right) + 0.4 \left(\underline{10 + 80} \right) + 80$$

$$\underline{\underline{10(0.6+0.4)}} + 0.4(80) + 80$$

$$= 10 + 0.4(80)$$

$$VA \rightarrow PA + \text{Data access time}$$

0.4×60
+
 0.6×90



Diagram illustrating the components of data access time:

- Read time = 60 ms
- Write time = 90 ms
- Latency time read
- Latency time write

A curly brace groups the read and write times with the latency times.



58



EMAT=TLB hit × (TLB access time + memory access time) + TLB miss(TLB access time + page table access time+memory access time)

$$= 0.6(10 + 80) + 0.4(10 + 80 + 80)$$



$$= 54 + 68$$

Best answer

$$= 122 \text{ msec}$$

answered Oct 28, 2014 • edited Jun 18, 2021 by Lakshman Bhaiya

edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



neha pawar



GATE CSE 2019 | Question: 33



54



Assume that in a certain computer, the virtual addresses are 64 bits long and the physical addresses are 48 bits long. The memory is word addressable. The page size is 8 kB and the word size is 4 bytes. The Translation Look-aside Buffer (TLB) in the address translation path has 128 valid entries. At most how many distinct virtual addresses can be translated without any TLB miss?

- A. 16×2^{10}
- B. 256×2^{10}
- C. 4×2^{20}
- D. 8×2^{20}

gatecse-2019

operating-system

virtual-memory

2-marks

TLB \Rightarrow 128 Valid entries

$$VA = 64 \text{ bit} = 2^{64} \text{ words}$$

$$PA = 48 \text{ bit}$$

$$\begin{aligned} \text{Page Size} &= 2^3 \times 2^{10} \text{ B} \\ &= 2^{11} \text{ words} \end{aligned}$$

<https://gateoverflow.in/302815/gate-cse-2019-question-33>



GATE CSE 2019 | Question: 33



54



Assume that in a certain computer, the virtual addresses are 64 bits long and the physical addresses are 48 bits long. The memory is word addressable. The page size is 8 kB and the word size is 4 bytes. The Translation Look-aside Buffer (TLB) in the address translation path has 128 valid entries. At most how many distinct virtual addresses can be translated without any TLB miss?

- A. 16×2^{10}
- B. 256×2^{10}
- C. 4×2^{20}
- D. 8×2^{20}

gatecse-2019

operating-system

virtual-memory

2-marks

TLB \Rightarrow 128 Valid entries

\Rightarrow 128 pages

$\Rightarrow 2^7 \times 2^9$ words

$$= 2^8 \times 2^{10} \text{ words}$$

$$= 2^{16} \times 2^{10}$$

$$VA = 64 \text{ bit} = 2^{64} \text{ words}$$

$$PA = 48 \text{ bit}$$

$$\text{Page Size} = 2^3 \times 2^{10} \text{ B}$$

$$= 2^{11} \text{ words}$$



TLB Entry:

Page Number	Frame Number
-------------	--------------

83

Memory is word addressable.



Best answer

- Word size = 4 Bytes
- Page size = 8 KB = 2^{11} words
- Virtual Memory size = 2^{64} words
- Number of pages possible = 2^{53}
- Number of bits required for Page number = 53 bits
- Number of bits required for Page offset = $64 - 53 = 11$ bits

At a time TLB contains $128 = 2^7$ distinct page numbers.

If a page number is found in TLB then there will be a hit for all the words (Word addresses) of that Page.

1 - page hit implies 2^{11} distinct virtual address hits.

So 2^7 page hit implies $2^7 * 2^{11} = 2^8 * 2^{10} = 256 * 2^{10}$ virtual address hits

Option B. At most, $256 * 2^{10}$ distinct virtual addresses can be translated without any TLB miss.

S

GO Classes

SES

answered Feb 7, 2019 • edited Jun 10, 2021 by Lakshman Bhaiya

edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



Soumya29



GATE IT 2008 | Question: 16



32



A paging scheme uses a Translation Look-aside Buffer (TLB). A TLB-access takes 10 ns and the main memory access takes 50 ns. What is the effective access time(in ns) if the TLB hit ratio is 90% and there is no page-fault?

- A. 54
- B. 60
- C. 65
- D. 75

gateit-2008

operating-system

virtual-memory

normal

TLB \Rightarrow 10 ns

MM \Rightarrow 50 ns

$$EAT = VA \rightarrow PA + \text{data access}$$

<https://gateoverflow.in/3276/gate-it-2008-question-16>



GATE IT 2008 | Question: 16



32



A paging scheme uses a Translation Look-aside Buffer (TLB). A TLB-access takes 10 ns and the main memory access takes 50 ns. What is the effective access time(in ns) if the TLB hit ratio is 90% and there is no page-fault?

- A. 54
- B. 60
- C. 65
- D. 75

gateit-2008

operating-system

virtual-memory

normal

TLB \Rightarrow 10 ns

MM \Rightarrow 50 ns

$$EAT = \underbrace{10 + 0.1 \times 50}_{VA \rightarrow PA} + \underbrace{50}_{\text{data access}} = 65$$

<https://gateoverflow.in/3276/gate-it-2008-question-16>



Effective access time = hit ratio \times time during hit + miss ratio \times time during miss

45



In both cases TLB is accessed and assuming page table is accessed from memory only when TLB misses.



$$= 0.9 \times (10 + 50) + 0.1 \times (10 + 50 + 50)$$

Best answer

$$= 54 + 11 = 65$$

Correct Answer: C

answered Nov 7, 2014 • edited May 23, 2019 by Naveen Kumar 3

edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



Arjun



More Questions on access time dealing with
page faults

(Not related to TLB)



GATE CSE 1998 | Question: 2.18



If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instruction is:

43



A. $i + \frac{j}{k}$

B. $i + (j \times k)$

C. $\frac{i+j}{k}$

D. $(i+j) \times k$

gate1998

operating-system

virtual-memory

easy

ugcnetcse-june2012-paper3

" i "

page fault EXTRA
"j"

Page fault rate = $\frac{1}{k}$

<https://gateoverflow.in/1691/gate-cse-1998-question-2-18-ugcnet-june2012-iii-48>



GATE CSE 1998 | Question: 2.18



If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instruction is:

43



A. $i + \frac{j}{k}$

B. $i + (j \times k)$

C. $\frac{i+j}{k}$

D. $(i+j) \times k$

gate1998

operating-system

virtual-memory

easy

ugcnetcse-june2012-paper3

" i "

page fault EXTRA
"j"

$$\left(\frac{\text{No page}}{\text{fault}} \right) i + \left(\frac{\text{Page fault}}{\text{instruction}} \right) (i + j)$$

$$= i + \frac{1}{k} (j)$$

$$\text{Page fault rate} = \frac{1}{k}$$



58



$$\text{Page fault rate} = \frac{1}{k}$$

$$\text{Page hit rate} = 1 - \frac{1}{k}$$

$$\text{Service time} = i$$

$$\text{Page fault service time} = i + j$$



Best answer

Effective memory access time,

$$= \frac{1}{k} \times \underline{(i + j)} + \left(1 - \frac{1}{k}\right) \times \underline{i}$$

$$= \frac{(i + j)}{k} + i - \frac{i}{k}$$

$$= \frac{i}{k} + \frac{j}{k} + i - \frac{i}{k}$$

$$= i + \frac{j}{k}$$

So, option (A) is correct.



GATE CSE 2000 | Question: 2.22



33



Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

- A. 1.9999 milliseconds
- B. 1 millisecond
- C. 9.999 microseconds
- D. 1.9999 microseconds

gatecse-2000

operating-system

easy

virtual-memory

hard disk \rightarrow ms

+
data access
ms

Page fault Service time = 10 ms = 10×10^{-3} s
 $mm = 1 \times 10^{-6}$ s

<https://gateoverflow.in/669/gate-cse-2000-question-2-22>



GATE CSE 2000 | Question: 2.22



33



Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

- A. 1.9999 milliseconds
- B. 1 millisecond
- C. 9.999 microseconds
- D. 1.9999 microseconds

gatecse-2000

operating-system

easy

virtual-memory

hard disk \rightarrow ms

+
data access
ms

Page fault Service time = 10 ms = 10×10^{-3} s

$$\text{mm} = 1 \times 10^{-6} \text{ s}$$

$$\frac{99.99}{100} \times 10^{-6} + \left(1 - \frac{99.99}{100}\right) \left(10 \times 10^{-3} \right)$$



52



Since nothing is told about page tables, we can assume page table access time is included in memory access time.



Best answer

$$\begin{aligned} &= .9999 \times 1 + 0.0001 \times 10,000 \\ &= 0.9999 + 1 \\ &= 1.9999 \text{ microseconds} \end{aligned}$$

Correct Answer: D



GATE CSE 2011 | Question: 20, UGCNET-June2013-II: 48



37



Let the page fault service time be 10 milliseconds(ms) in a computer with average memory access time being 20 nanoseconds (ns). If one page fault is generated every 10^6 memory accesses, what is the effective access time for memory?

- A. 21 ns
- B. 30 ns
- C. 23 ns
- D. 35 ns

gatecse-2011

operating-system

virtual-memory

normal

ugcnetcse-june2013-paper2

<https://gateoverflow.in/2122/gate-cse-2011-question-20-ugcnet-june2013-ii-48>



Open slides 12 – 13 to check :

65

<http://web.cs.ucla.edu/~ani/classes/cs111.08w/Notes/Lecture%2016.pdf>



Best answer

$$\begin{aligned} \text{EMAT} &= \frac{1}{10^6} \times 10 \text{ ms} + \left(1 - \frac{1}{10^6}\right) \times 20 \text{ ns} \\ &= 29.99998 \text{ ns} \\ &\approx 30 \text{ ns} \end{aligned}$$

Answer = option B

answered Oct 23, 2015 • edited Jun 21, 2021 by Lakshman Bhaiya

edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



amarVashishth



GATE CSE 2018 | Question: 10



19



Consider a process executing on an operating system that uses demand paging. The average time for a memory access in the system is M units if the corresponding memory page is available in memory, and D units if the memory access causes a page fault. It has been experimentally measured that the average time taken for a memory access in the process is X units.

Which one of the following is the correct expression for the page fault rate experienced by the process.

- A. $(D - M)/X - M$
- B. $(X - M)/D - M$
- C. $(D - X)/D - M$
- D. $(X - M)/D - X$

SES
HCO
= = =

gatecse-2018

operating-system

virtual-memory

normal

1-mark

<https://gateoverflow.in/204084/gate-cse-2018-question-10>



Let P be the page fault rate.

55

Average memory access time = $(1 - \text{page fault rate}) \times \text{memory access time when no page fault} + \text{Page fault rate} \times \text{Memory access time when page fault.}$



$$X = (1 - P)M + P D$$

Best answer

$$X = M + P(D - M)$$

$$P = (X - M)/(D - M)$$

(B) is the answer.

answered Feb 14, 2018 • edited Jun 29, 2018 by **kenzou**

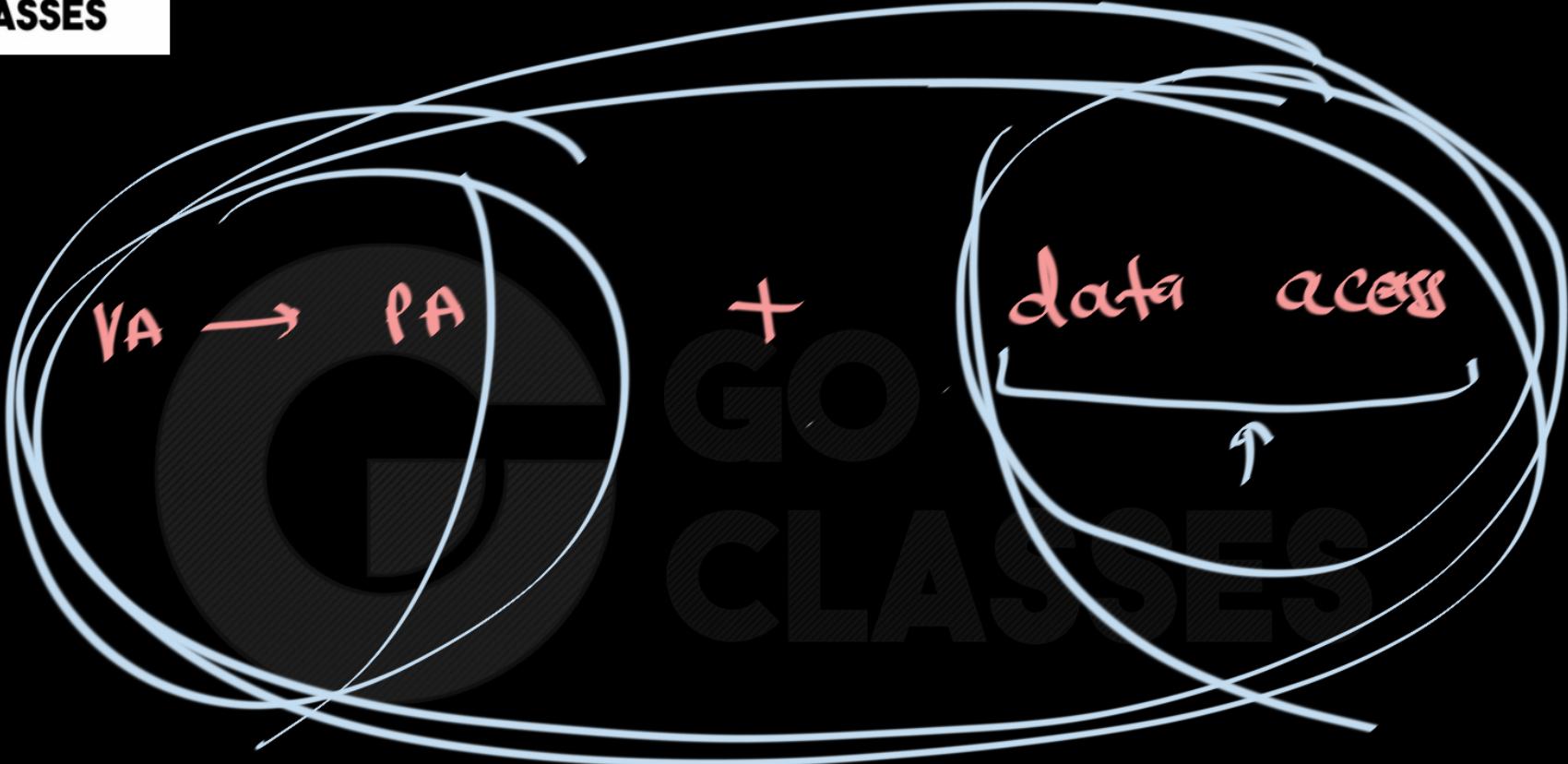
edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



Hemant Parihar





3 remaining questions in TLB can be solved later after cache topic in COA



Translation Lookaside Buffer

- Just like any other cache, the TLB can be organized as fully associative, set associative, or direct
- TLB access is typically faster than cache access
 - Because TLBs are much smaller than caches
 - TLBs are typically not more than 128 to 256 entries even on high-end machines

SES

V	Virtual Page #	Physical Page #	Dirty	Status



GATE CSE 2006 | Question: 62, ISRO2016-50

44 A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. The minimum size of the TLB tag is:

- A. 11 bits
- B. 13 bits
- C. 15 bits
- D. 20 bits

gatecse-2006

operating-system

virtual-memory

normal

isro2016

easy

<https://gateoverflow.in/1840/gate-cse-2006-question-62-isro2016-50>



The page size of 4 KB. So, offset bits are 12 bits.

57

So, the remaining bits of virtual address, $32 - 12 = 20$ bits, will be used for indexing.



Number of sets = $128/4 = 32$ (4-way set) \Rightarrow 5 bits.



So, tag bits = $20 - 5 = 15$ bits.

Best answer

Correct option C.

answered Jan 12, 2015 • edited Jun 3, 2021 by Lakshman Bhaiya

edit flag hide comment Follow

Pip Box Delete with Reason Wrong Useful

share this



Vicky Bajoria



GATE CSE 2015 Set 2 | Question: 25

- 51 A computer system implements a 40 -bit virtual address, page size of 8 kilobytes, and a 128-entry translation look-aside buffer (TLB) organized into 32 sets each having 4 ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is ____.

gatecse-2015-set2

operating-system

virtual-memory

easy

numerical-answers

<https://gateoverflow.in/8120/gate-cse-2015-set-2-question-25>



Operating Systems

GATE CSE 2020 | Question: 53

asked in **Operating System** Feb 12, 2020 • retagged Dec 1, 2022 by **Lakshman Bhaiya**

34,079 views

- 39 Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible. The average memory access time in ns (round off to 1 decimal places) is _____

gatecse-2020 numerical-answers operating-system virtual-memory 2-marks

answer comment Follow share this



Arjun

little hard

SES

aftor

cDA

<https://gateoverflow.in/333178/gate-cse-2020-question-53>



Dynamic allocation strategies (Heap allocation strategies)

for stack :

Very Simple

it will grow in just
one direction or shrink
in other.



Operating Systems

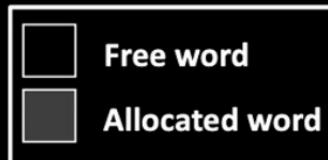


Allocated block
(4 words)

= 16B

Free block
(3 words)

= 3B



pointer to newly allocated block
of at least that size

number of contiguous bytes required

void* malloc(size_t size);

pointer to allocated block to free

void free(void* ptr);



Operating Systems





Operating Systems

- Easy without free: set a pointer to the beginning of some big chunk of memory (“heap”) and increment on each allocation:



CLASSES

that's almost same happens in stack.



Operating Systems

- Easy without free: set a pointer to the beginning of some big chunk of memory (“heap”) and increment on each allocation:



- Problem: free creates holes (“fragmentation”) Result? Lots of free space but cannot satisfy request!





□ Best fit

- Search the whole list on each allocation
- Choose the smallest block that can satisfy request
- Can stop search if exact match found

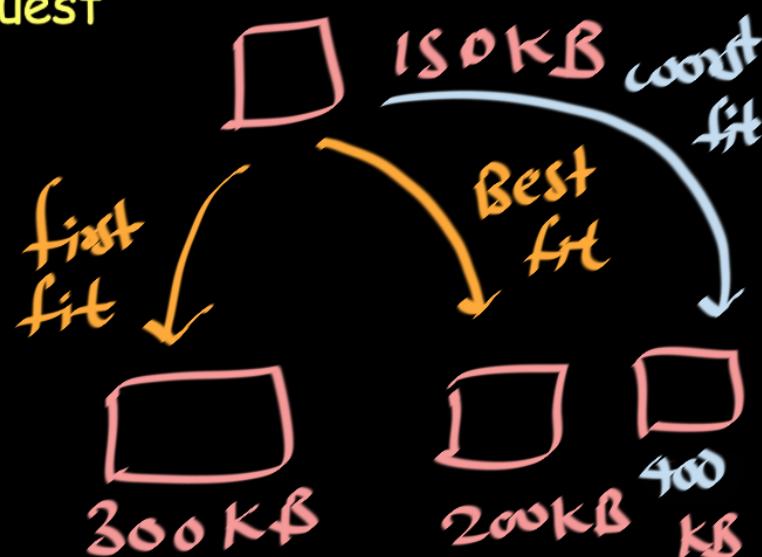
→ free space list

□ First fit

- Choose first block that can satisfy request

□ Worst fit

- Choose largest block (most leftover space)





8.3 Contiguous Memory Allocation 363

- **First fit.** Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended. We can stop searching as soon as we find a free hole that is large enough.
- **Best fit.** Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.
- **Worst fit.** Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.



→ in case of heap we can use
these strategies.

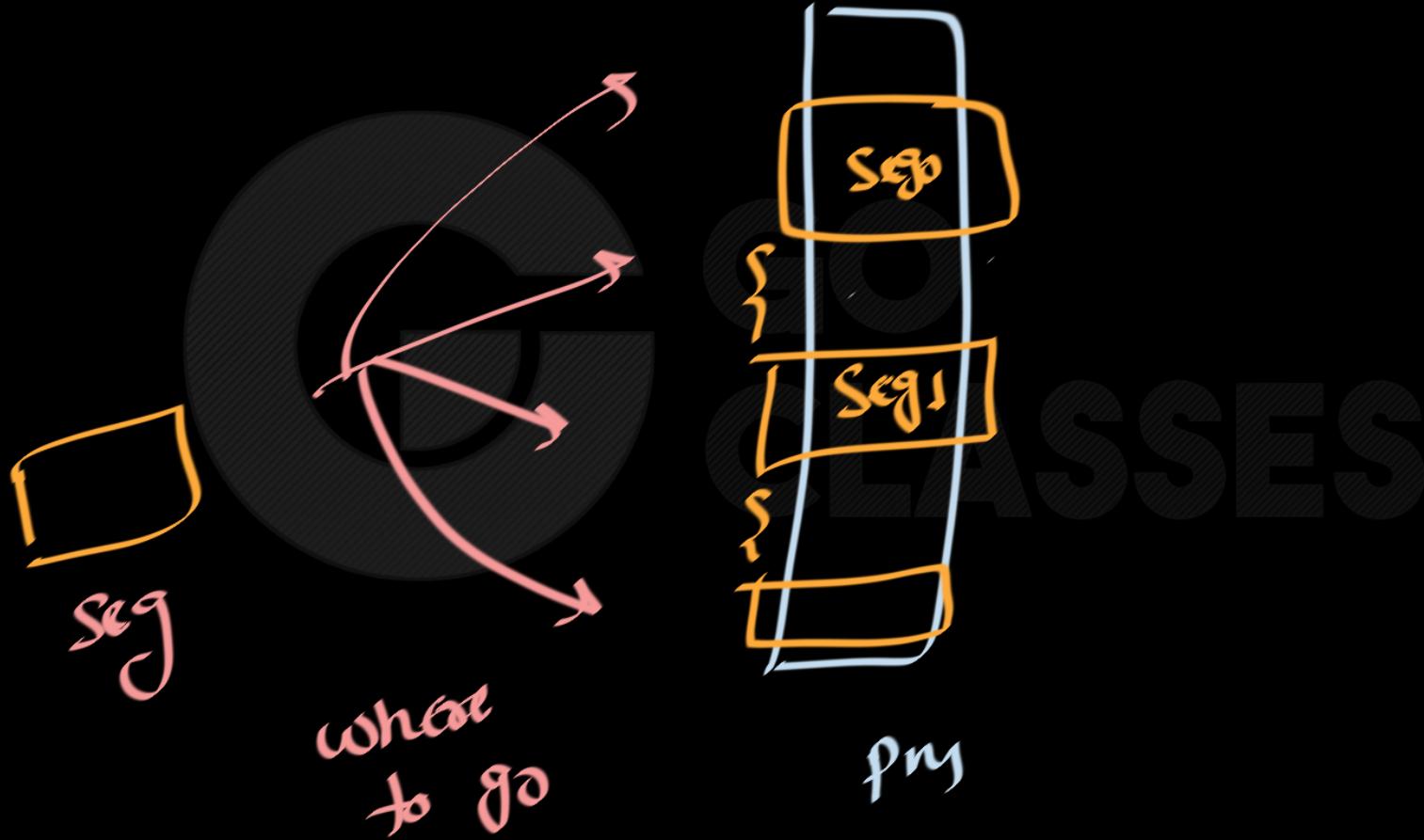
where else we can make use of these
strategies ? ⇒ Segmentation

→ in case of heap we can use
these strategies.

where else we can make use of these
strategies ?

⇒ Segmentation

↳ whenever we have holes scattered
of different size

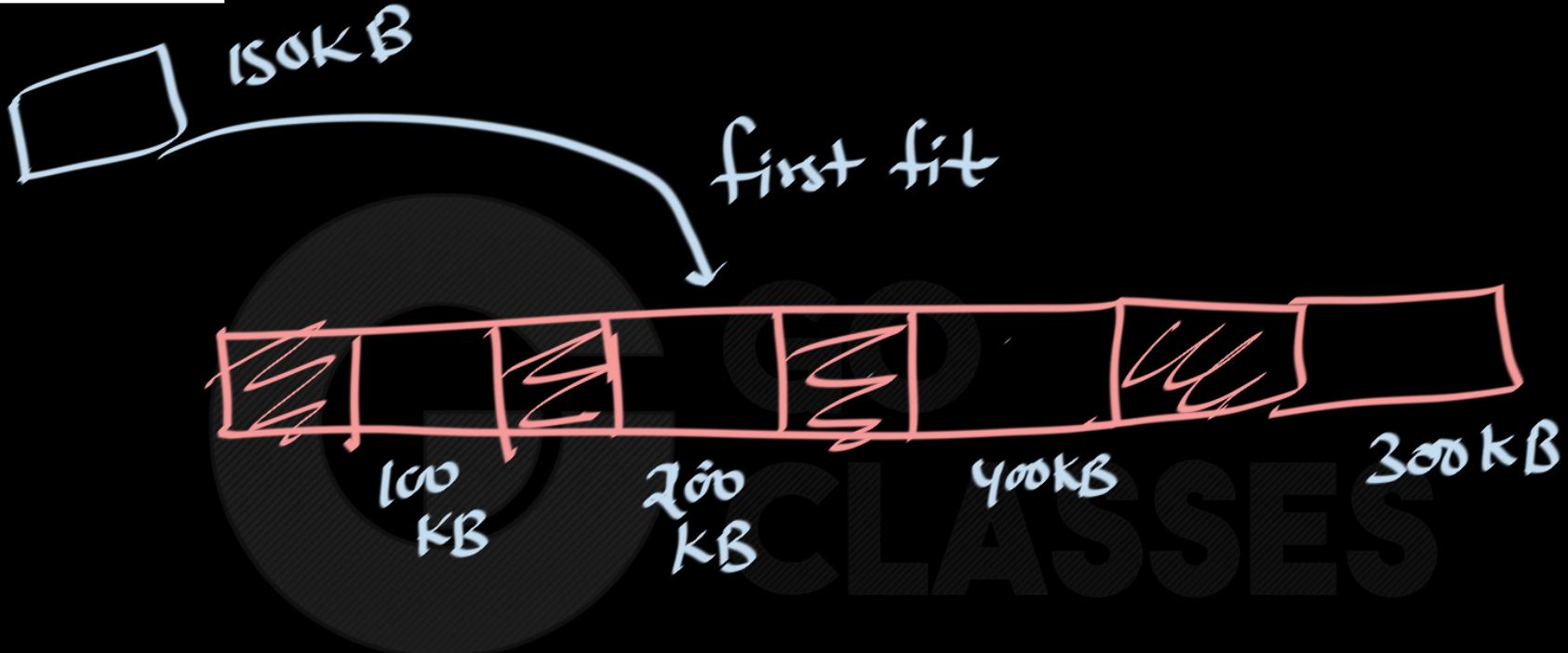


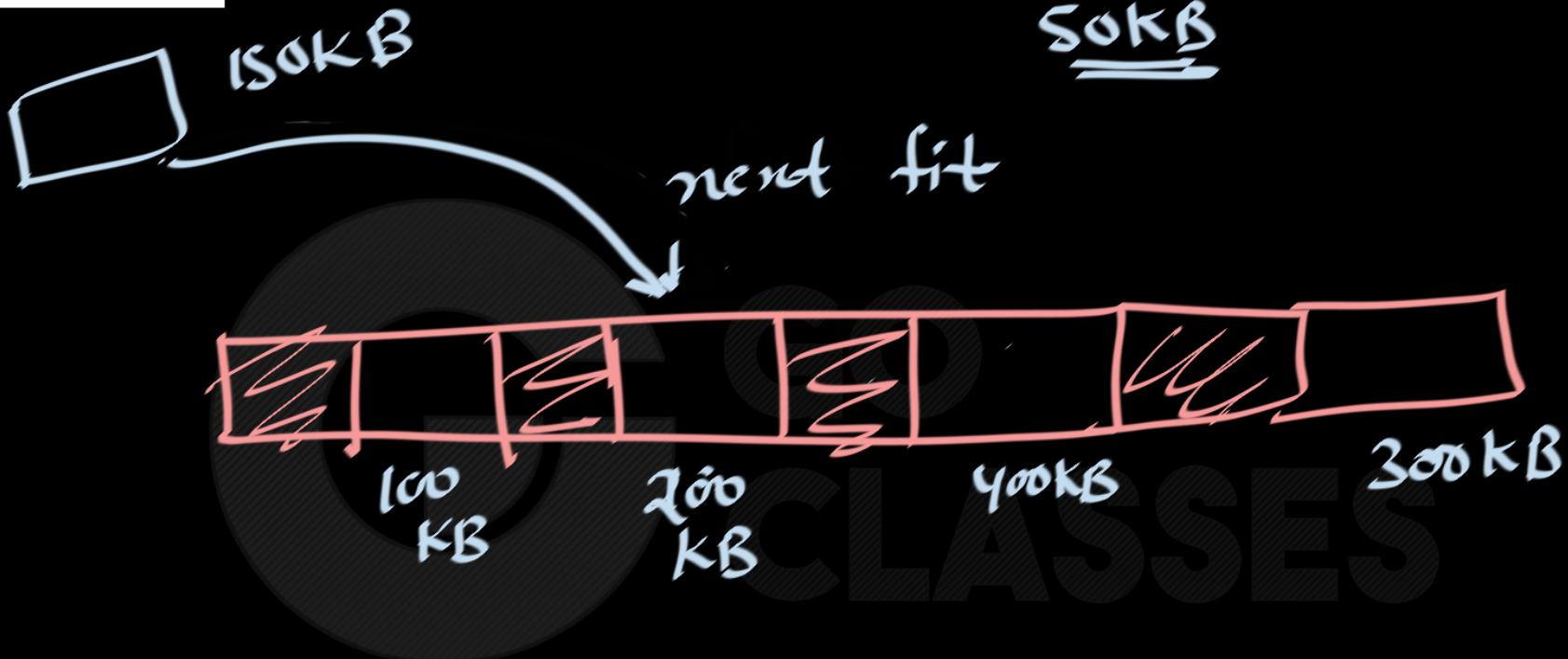


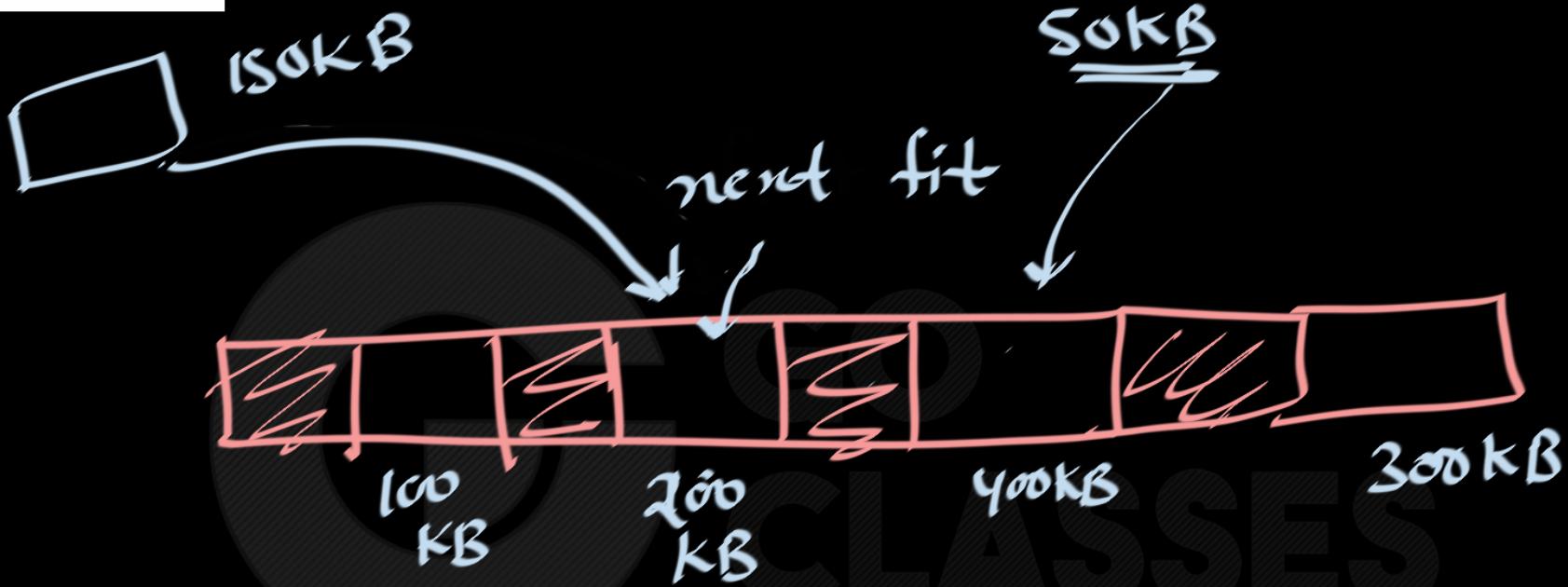
Segmentation: dynamic storage-allocation problem

- The OS accounts for both the assigned and free memory. After releasing a block of memory, adjacent free blocks are collapsed into a larger free block.
- How can the OS satisfy a request of size n from a list of free holes?
 - **first fit** Allocate the first hole that is big enough. Fast. Small holes appear in low areas of memory and large holes in high areas, assuming the search for holes start in the low areas.
 - **next fit** Allocate the next hole large enough, searching from the last allocated block
 - **best fit** Allocate the smallest hole that is big enough; must search entire list, unless ordered by size. Produces the smallest leftover hole.
 - **worst fit** Allocate the largest hole; must also search entire list. Produces the largest leftover hole
- First-fit and best-fit are found to operate better than worst-fit in terms of speed and storage utilization

Same as first fit but instead of starting from start, start from last pointed





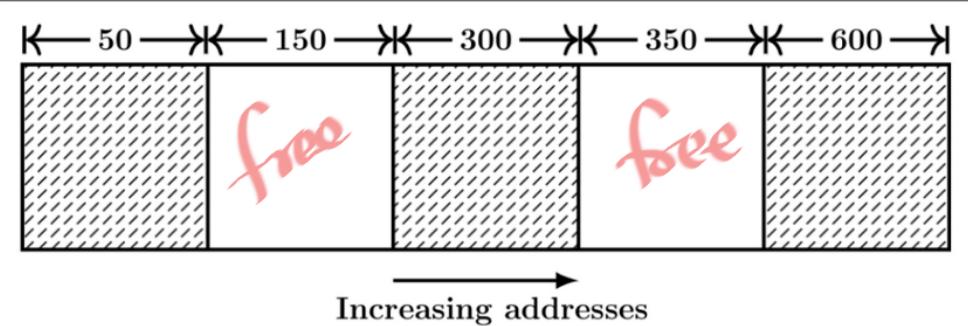


GATE CSE 1994 | Question: 1.24



Consider the following heap (figure) in which blank regions are not in use and hatched region are in use.

23



The sequence of requests for blocks of sizes 300, 25, 125, 50 can be satisfied if we use

- A. either first fit or best fit policy (any one)
- B. first fit but not best fit policy
- C. best fit but not first fit policy
- D. None of the above

ASSESSES
300, 25, 125, 50

gate1994

operating-system

page-replacement

normal



35



Best answer

In the first fit, block requests will be satisfied from the first free block that fits it.

- The request for 300 will be satisfied by a 350 size block reducing the free size to 50.
- Request for 25, satisfied by 150 size block, reducing it to 125.
- Request for 125 satisfied by 125 size block.
- And request for 50 satisfied by the 50 size block.

So, all requests can be satisfied.

In the best fit strategy, a block request is satisfied by the smallest block that can fit it.

- The request for 300 will be satisfied by a 350 size block reducing the free size to 50.
- Request for 25, satisfied by 50 size block as its the smallest size that fits 25, reducing it to 25.
- Request for 125, satisfied by 150 size block, reducing it to 25.

Now, the request for 50 cannot be satisfied as the two 25 size blocks are not contiguous.

So, answer (B).

answered Nov 17, 2014 • edited Apr 23, 2021 by Lakshman Bhaiya

comment Follow share this



Arjun

SES



GATE CSE 2015 Set 2 | Question: 30



29



Consider 6 memory partitions of sizes 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB, 491 KB in that order. If the best-fit algorithm is used, which partitions are NOT allotted to any process?

- A. 200 KB and 300 KB
- B. 200 KB and 250 KB
- C. 250 KB and 300 KB
- D. 300 KB and 400 KB

<https://gateoverflow.in/8145/gate-cse-2015-set-2-question-30>



31



Best answer

Option (A) is correct because we have 6 memory partitions of sizes 200 KB , 400 KB , 600 KB , 500 KB , 300 KB and 250 KB and the partition allotted to the process using best fit is given below:

- 357 KB process allotted at partition 400 KB .
- 210 KB process allotted at partition 250 KB
- 468 KB process allotted at partition 500 KB
- 491 KB process allotted at partition 600 KB

So, we have left only two partitions 200 KB and 300 KB

answered Feb 12, 2015 • edited Jun 19, 2021 by Lakshman Bhaiya

comment Follow share this



Anoop Sonkar



GATE CSE 1998 | Question: 7-b



44



In a computer system where the 'best-fit' algorithm is used for allocating 'jobs' to 'memory partitions', the following situation was encountered:

Partitions size in KB	4K 8K 20K 2K
Job sizes in KB	2K 14K 3K 6K 6K 10K 20K 2K
Time for execution	4 10 2 1 4 1 8 6

When will the 20K job complete?

gate1998

operating-system

process-scheduling

normal

<https://gateoverflow.in/12963/gate-cse-1998-question-7-b>



GATE CSE 1995 | Question: 5



A computer installation has $1000k$ of main memory. The jobs arrive and finish in the following sequences.

21



Job 1 requiring 200k arrives
Job 2 requiring 350k arrives
Job 3 requiring 300k arrives
Job 1 finishes
Job 4 requiring 120k arrives
Job 5 requiring 150k arrives
Job 6 requiring 80k arrives

A. Draw the memory allocation table using Best Fit and First Fit algorithms.

B. Which algorithm performs better for this sequence?

gate1995

operating-system

memory-management

normal

descriptive

<https://gateoverflow.in/2641/gate-cse-1995-question-5>



GATE CSE 1996 | Question: 2.18



42



A 1000 Kbyte memory is managed using variable partitions but no compaction. It currently has two partitions of sizes 200 Kbyte and 260 Kbyte respectively. The smallest allocation request in Kbyte that could be denied is for

- A. 151
- B. 181
- C. 231
- D. 541

gate1996

operating-system

memory-management

normal

<https://gateoverflow.in/2747/gate-cse-1996-question-2-18>