Operating Systems

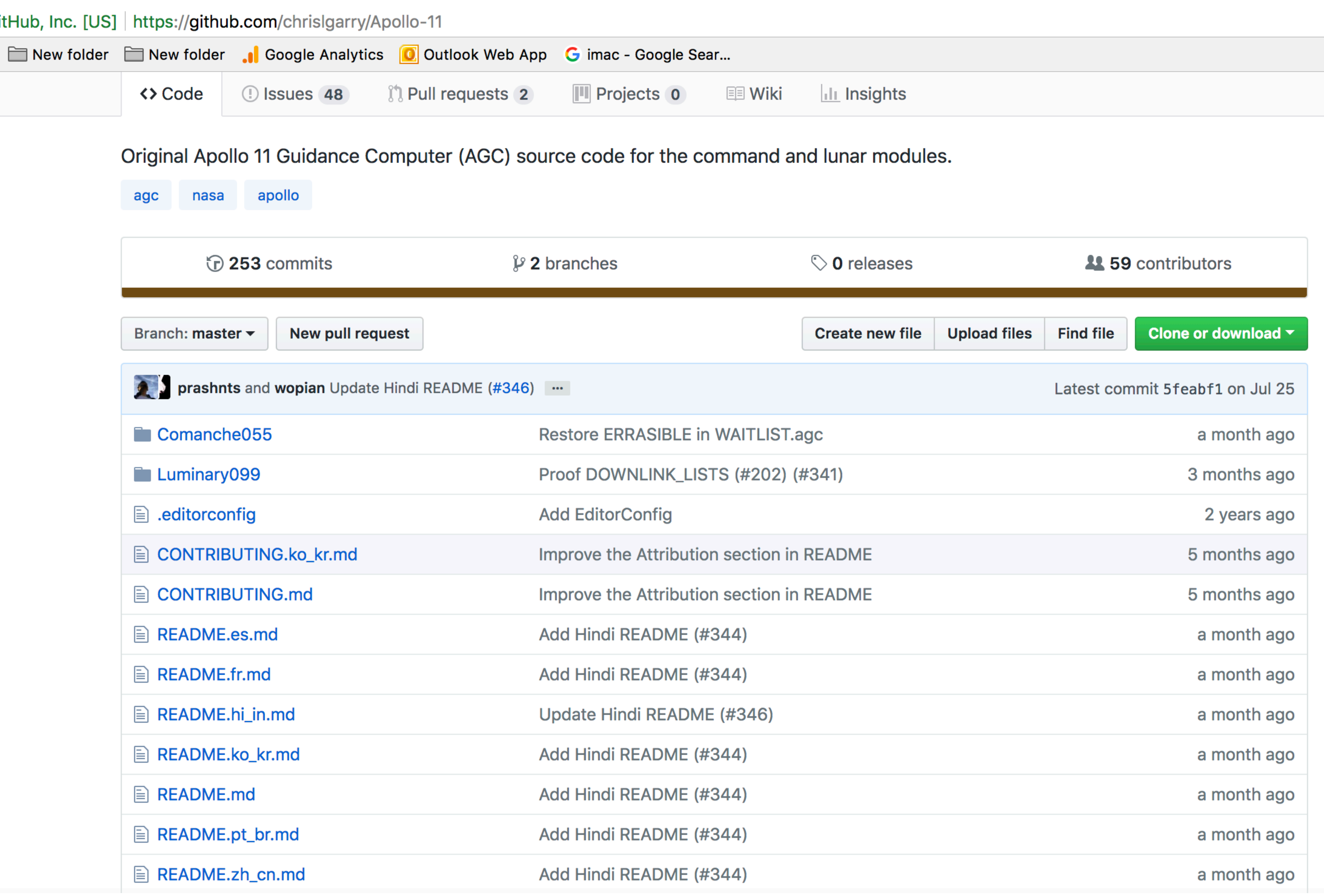
Lecture 12: Paging + TLB

Nipun Batra

Aug 28, 2018

CS stories

https://www.youtube.com/watch?v=kTn56jJW4zY


itHub, Inc. [US] https://github.com/chrislgarry/Apollo-11

Original Apollo 11 Guidance Computer (AGC) source code for the command and lunar modules.

253 commits · 2 branches · 0 releases · 59 contributors

prashnts and wopian Update Hindi README (#346) ... Latest commit 5feabf1 on Jul 25

Comanche055	Restore ERRASIBLE in WAITLIST.agc	a month ago
Luminary099	Proof DOWNLINK_LISTS (#202) (#341)	3 months ago
.editorconfig	Add EditorConfig	2 years ago
CONTRIBUTING.ko_kr.md	Improve the Attribution section in README	5 months ago
CONTRIBUTING.md	Improve the Attribution section in README	5 months ago
README.es.md	Add Hindi README (#344)	a month ago
README.fr.md	Add Hindi README (#344)	a month ago
README.hi_in.md	Update Hindi README (#346)	a month ago
README.ko_kr.md	Add Hindi README (#344)	a month ago
README.md	Add Hindi README (#344)	a month ago
README.pt_br.md	Add Hindi README (#344)	a month ago
README.zh_cn.md	Add Hindi README (#344)	a month ago

CS stories

clock speed of 4.077MHz. That’s 0.004077 GHz. The Apollo’s Guidance Computer was a snail-like 1.024 MHz in comparison, and it’s external signaling was half that.

Internally, the architecture of 8086 had 8 16-bit registers available to work with. It could keep track of eight registers, the Apollo Guidance Computer held just four.

The most amazing part that will blow you away isn’t so much the hardware, as the software they used to get to the Moon. In fact, the real-time operating system in the Apollo 11 spacecraft could multi-task eight jobs at a time, something we take entirely for granted today, but no small feat for the time it was developed.

Multi-tasking however, wasn’t quite as we now think of it. Our operating systems use pre-emptive -multitasking, where the operating system itself is in control of the execution and can stop any program at any time. The AGC relied on non-pre-emptive multi-tasking, whereby programs had to relinquish control back to the OS periodically.

Revision

Revision

1. Segmentation

Revision

1. Segmentation
    1. Registers containing:

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)
        4. … (save memory using identical code segment)

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)
        4. … (save memory using identical code segment)
    2. Segment = (VA & SEG_MASK) >>SEG_SHIFT

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)
        4. … (save memory using identical code segment)
    2. Segment = (VA & SEG_MASK) >>SEG_SHIFT
    3. Offset = VA & OFF_Mask

Revision

1. Segmentation
   1. Registers containing:
      1. Start VA
      2. Bounds
      3. …. (think Stack)
      4. … (save memory using identical code segment)
   2. Segment = (VA & SEG_MASK) >>SEG_SHIFT
   3. Offset = VA & OFF_Mask
   4. Segmentation cons:

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)
        4. … (save memory using identical code segment)
    2. Segment = (VA & SEG_MASK) >>SEG_SHIFT
    3. Offset = VA & OFF_Mask
    4. Segmentation cons:
        1. Requires ____ block of memory for each segment

Revision

1. Segmentation
    1. Registers containing:
        1. Start VA
        2. Bounds
        3. …. (think Stack)
        4. … (save memory using identical code segment)
    2. Segment = (VA & SEG_MASK) >>SEG_SHIFT
    3. Offset = VA & OFF_Mask
    4. Segmentation cons:
        1. Requires ____ block of memory for each segment
            1. Can lead to _____ and _____

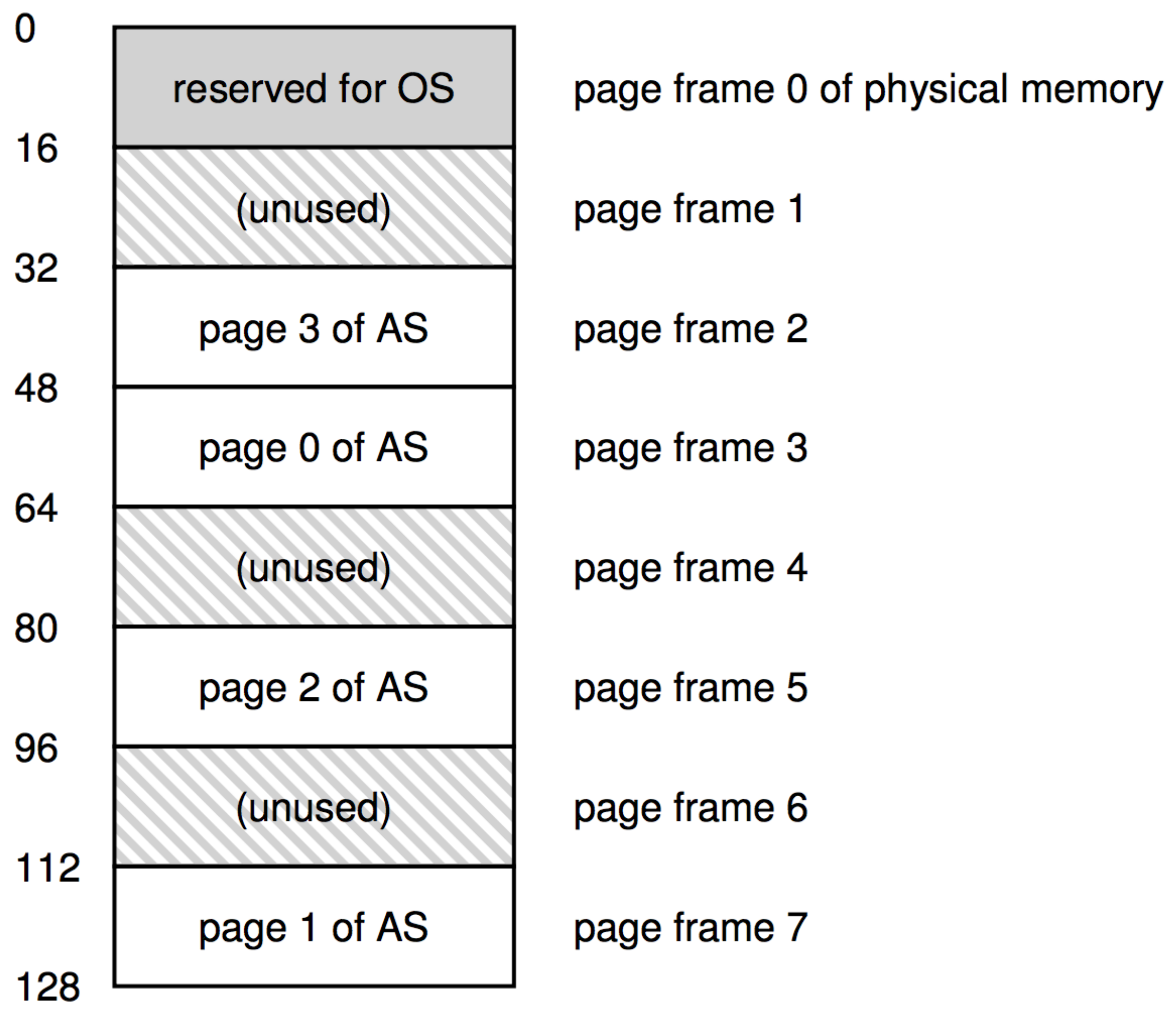
Revision

Revision

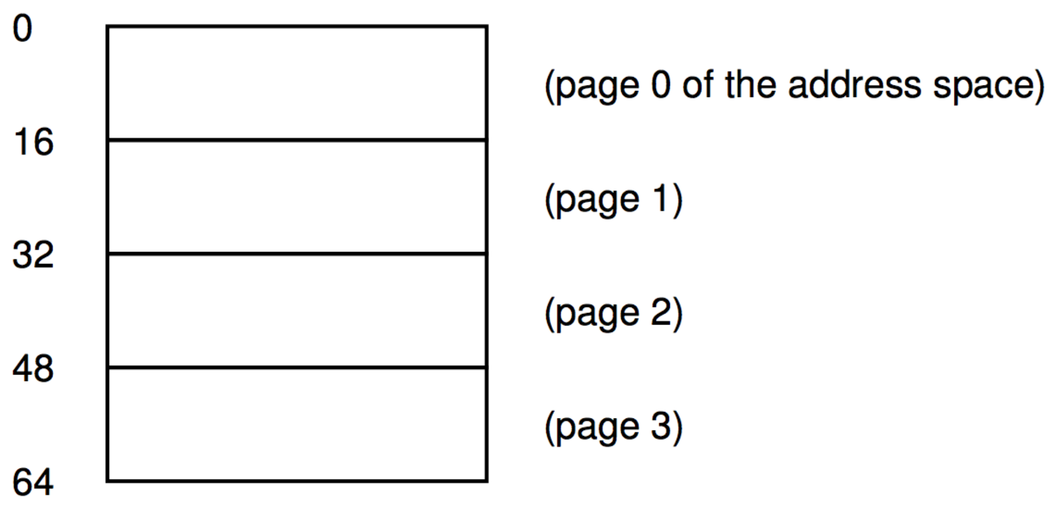
1. Large contiguous memory causes problems
    1. What happens if we map every byte of VA to a byte of PA?
        1. Reduces fragmentation?
            1. External?
            2. Internal?
        2. How much space needed per-process to store mapping?
2. Middle ground?

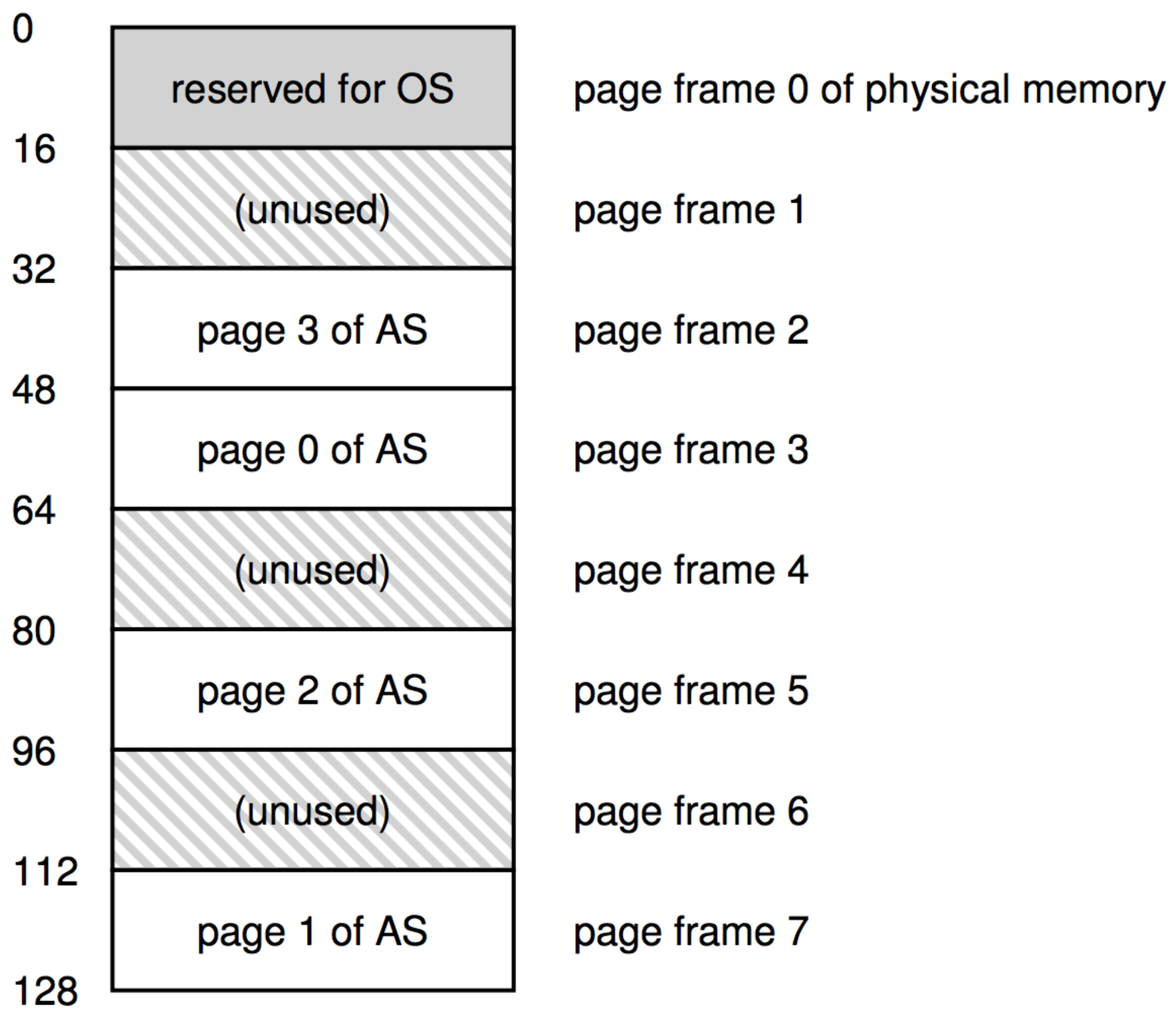
Revision : Paging

Revision : Paging

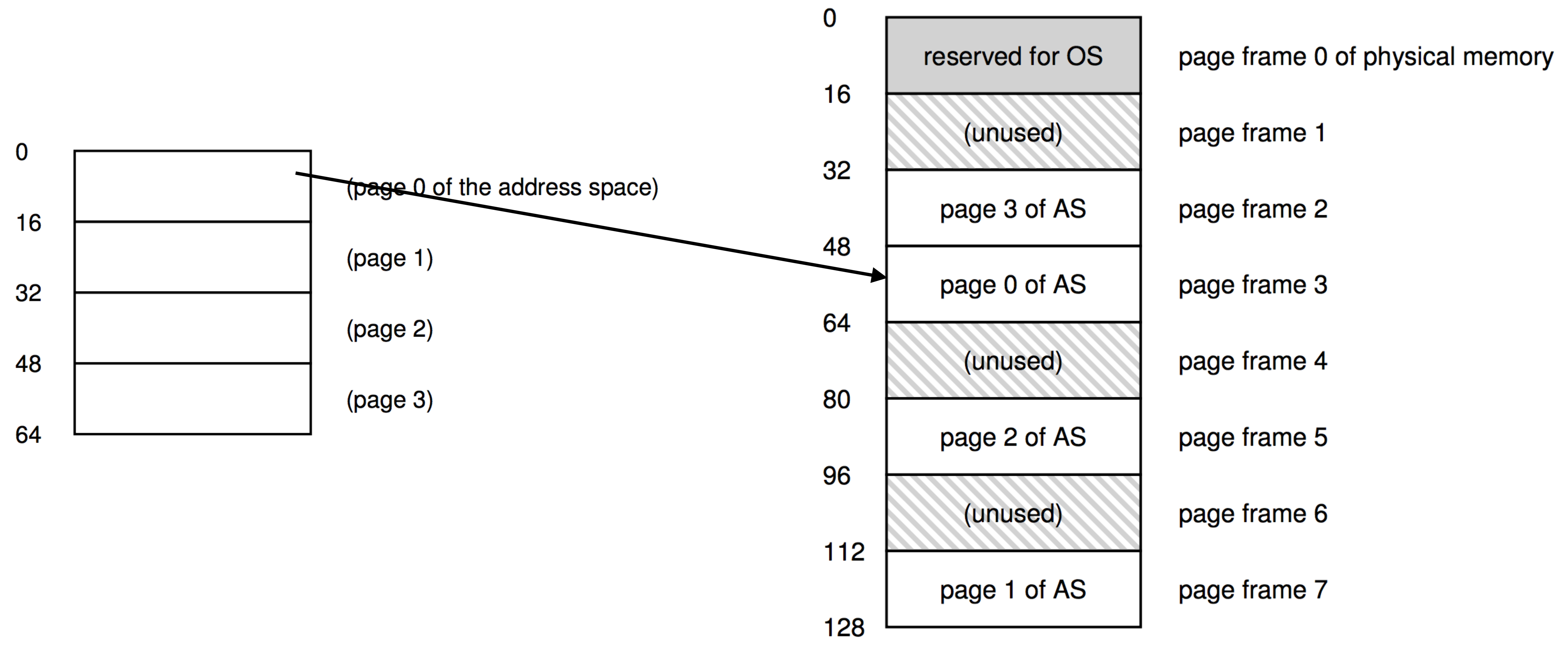


Revision : Paging

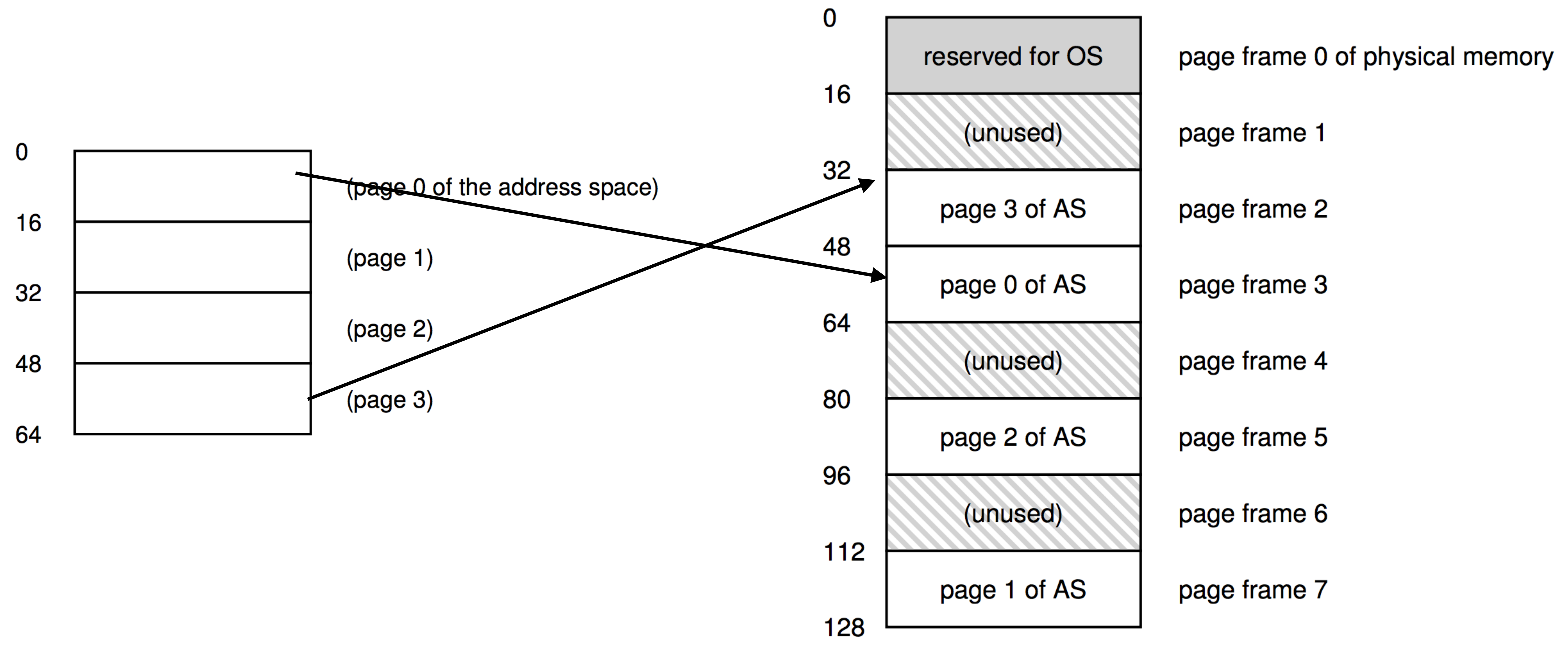




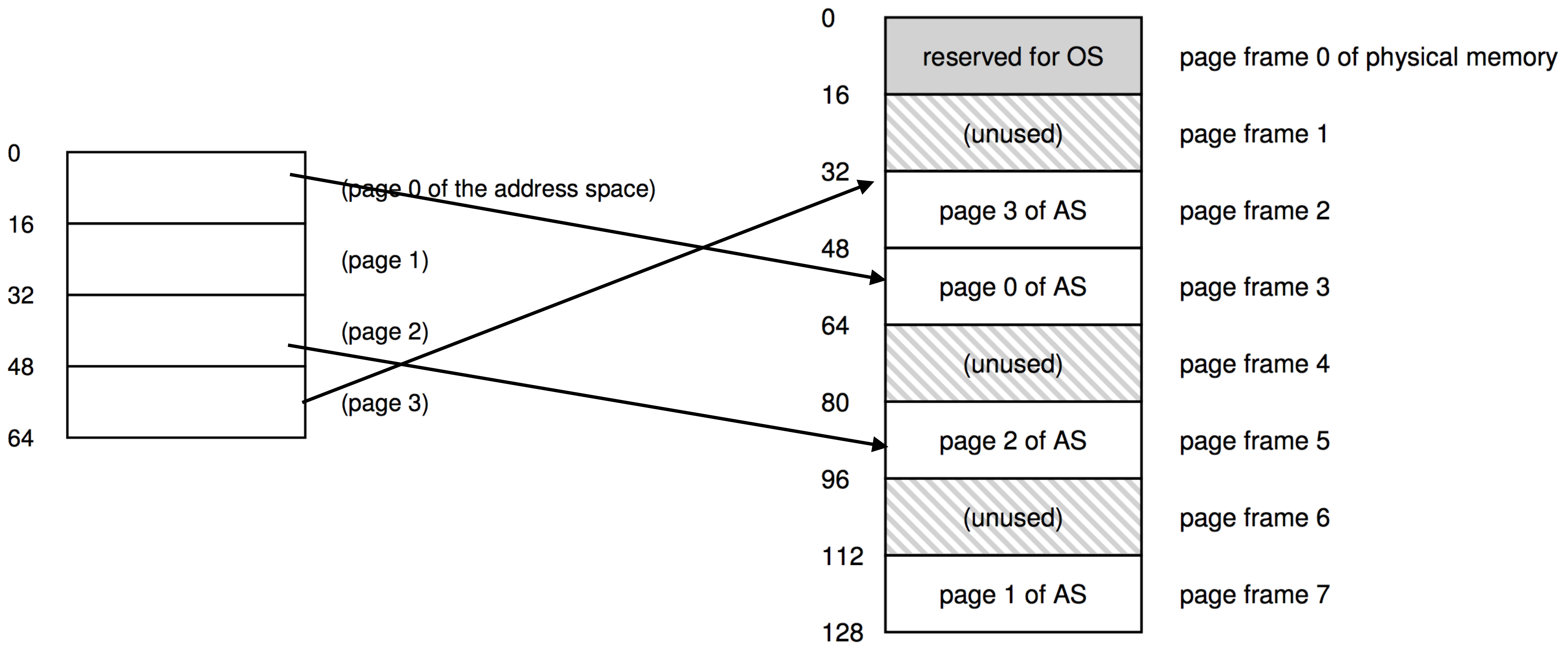
Revision : Paging



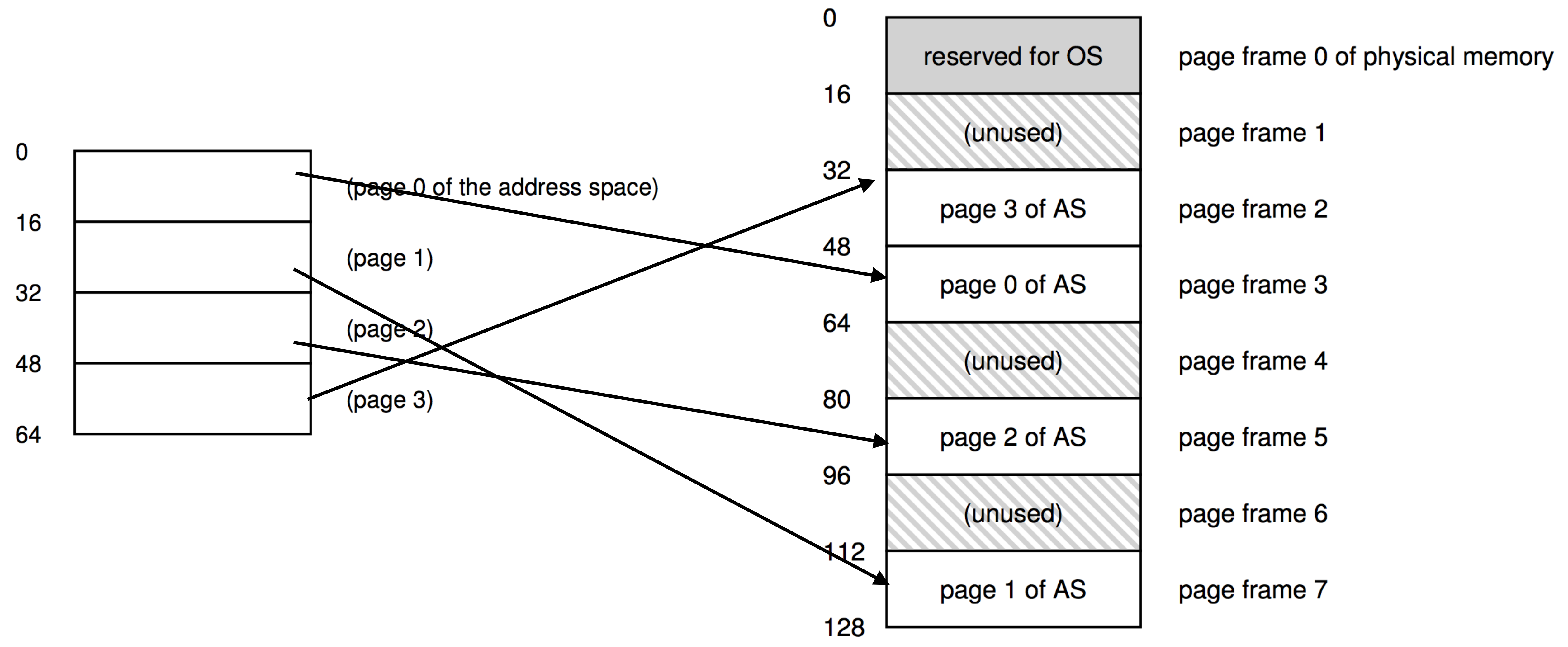
Revision : Paging



Revision : Paging



Revision : Paging



Example

```
movl 21, %eax
```

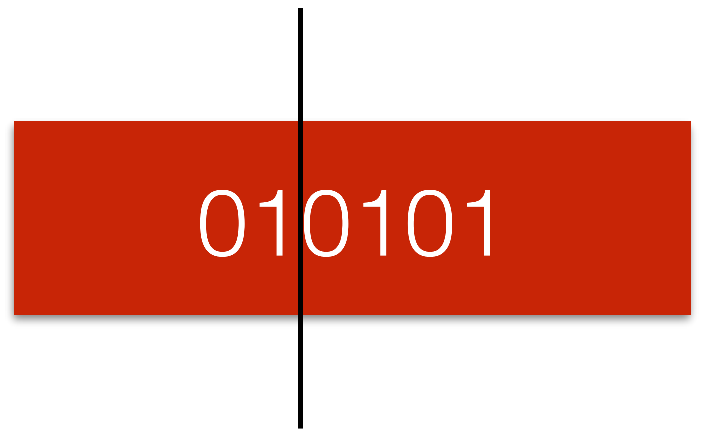
Example

```
movl 21, %eax
```

010101

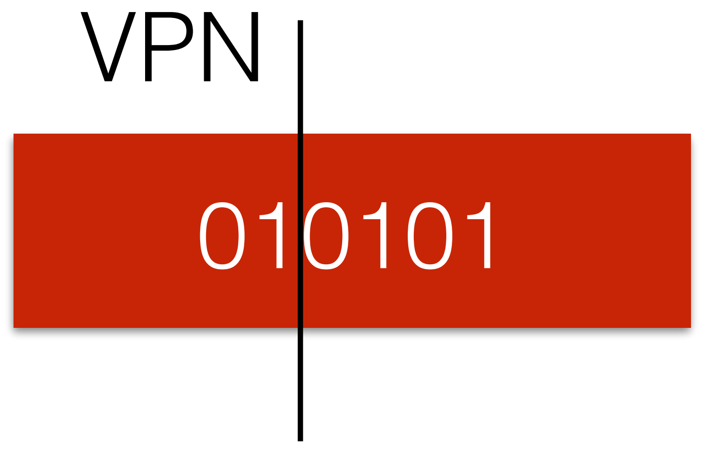
Example

```
movl 21, %eax
```



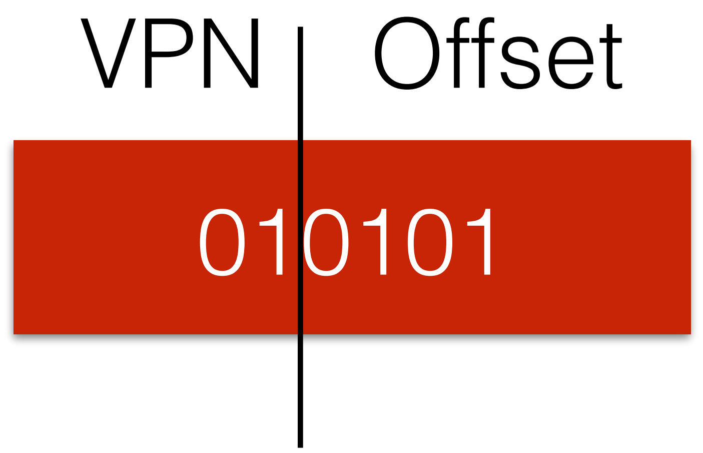
Example

movl 21, %eax



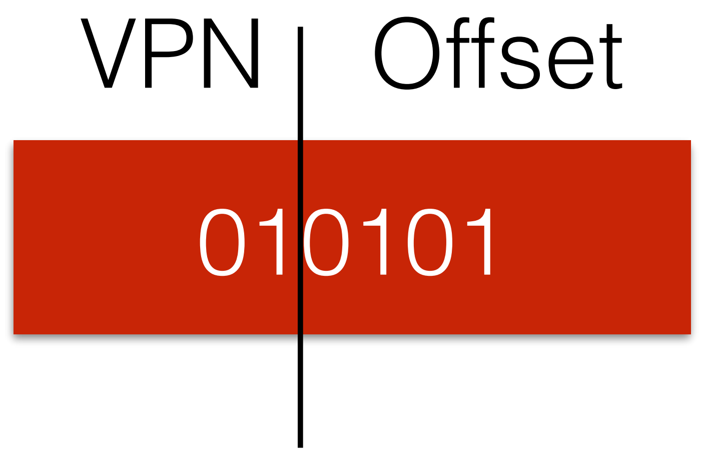
Example

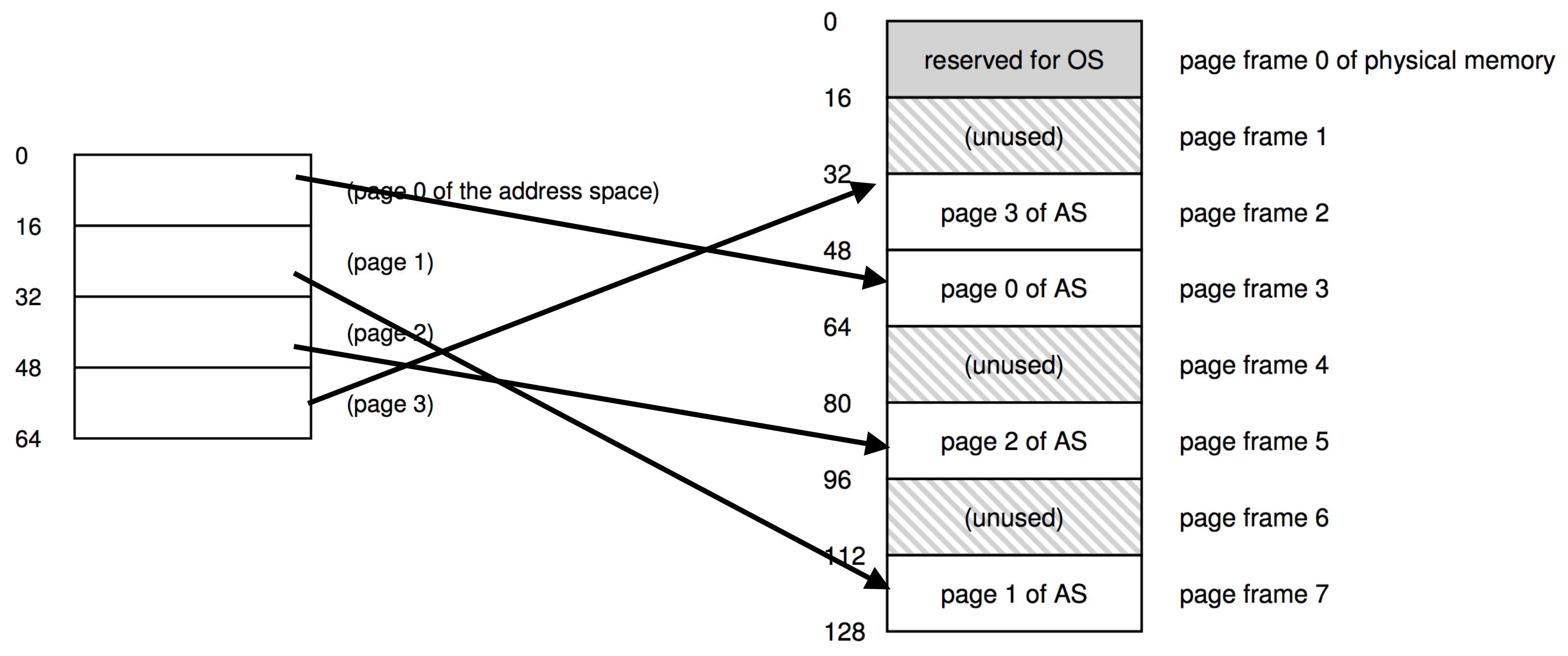
movl 21, %eax



Example

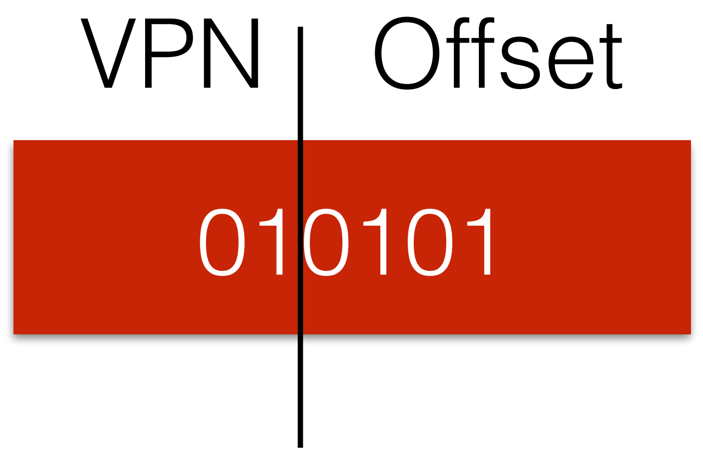
movl 21, %eax

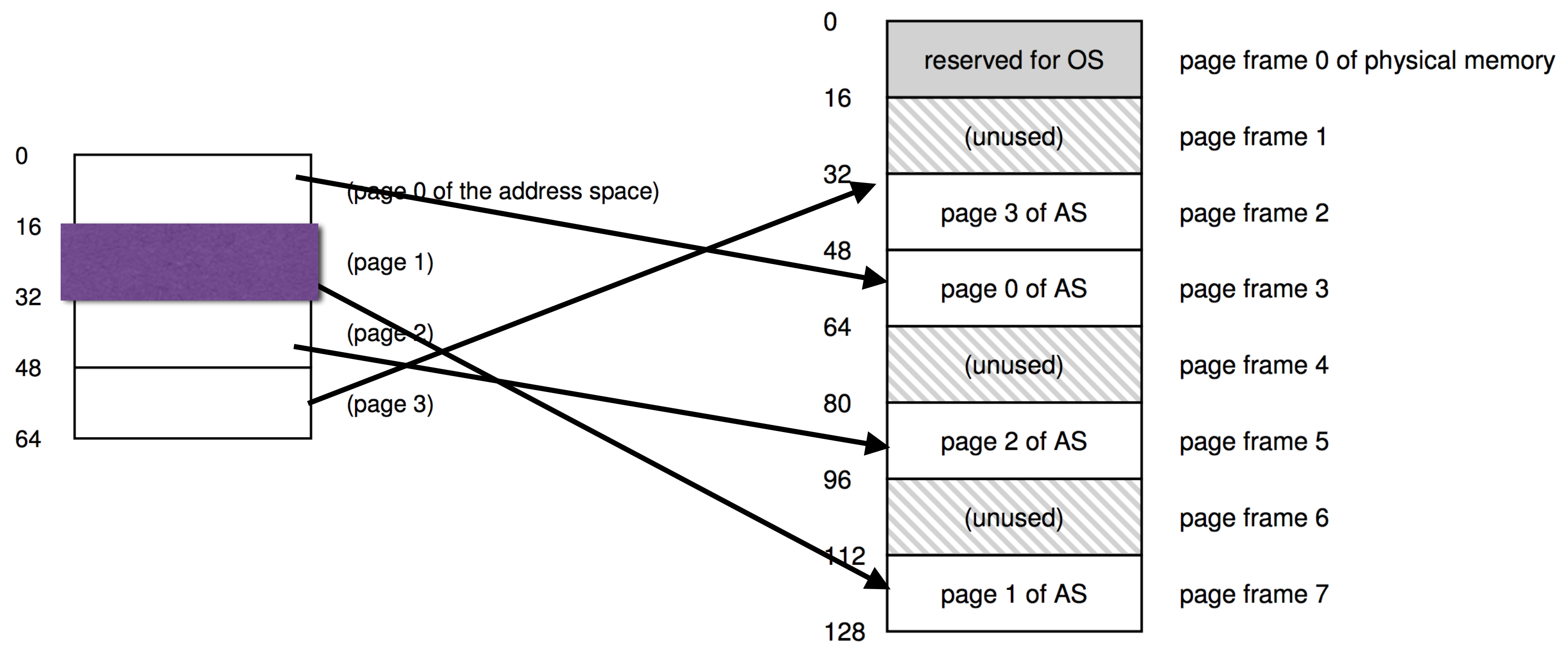




Example

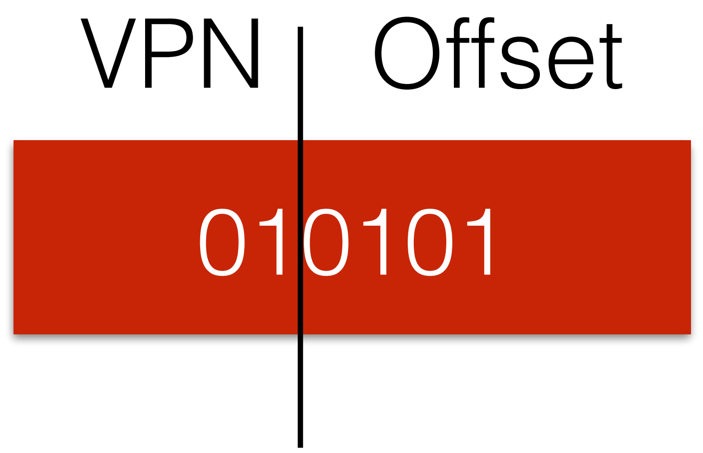
movl 21, %eax

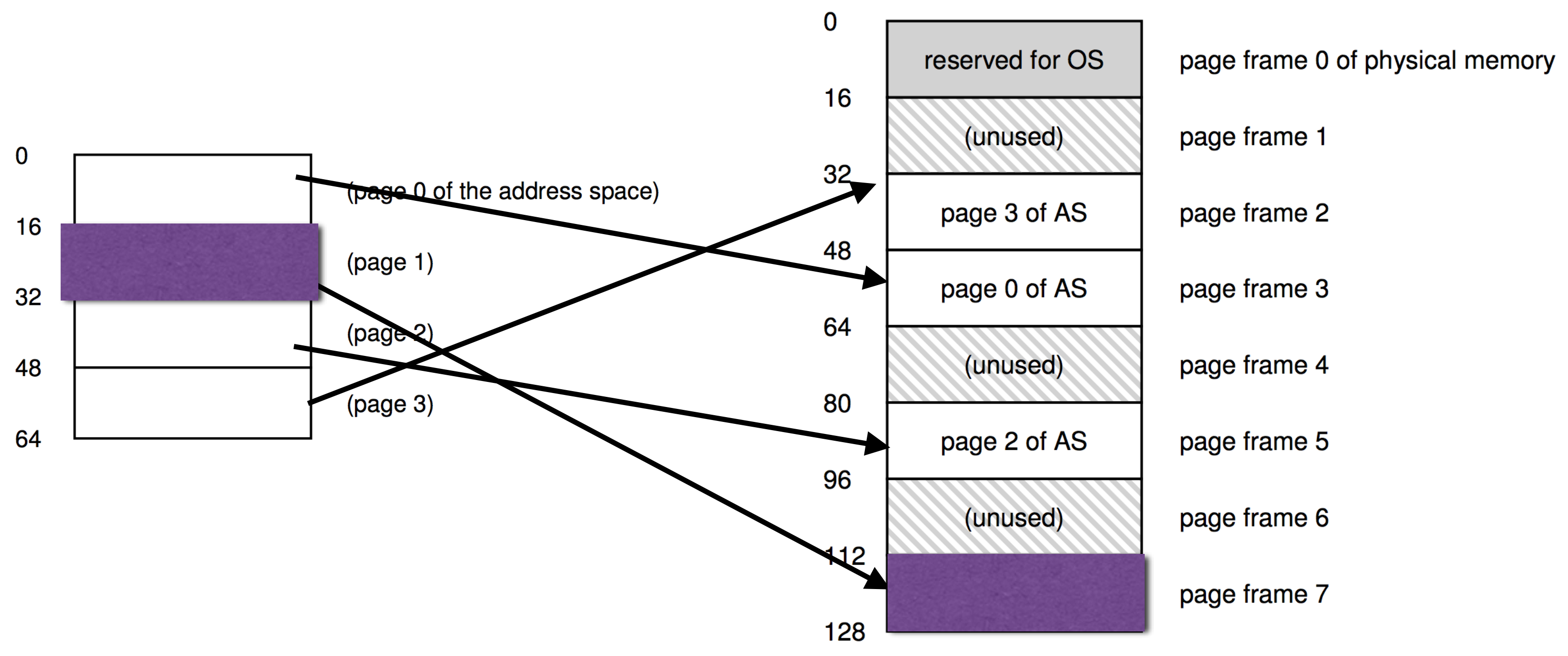




Example

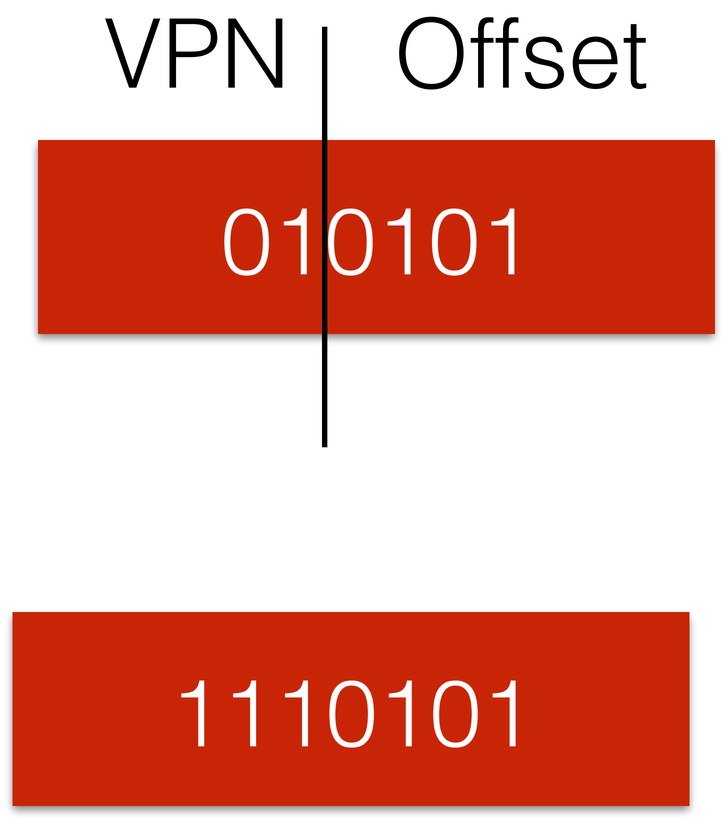
movl 21, %eax

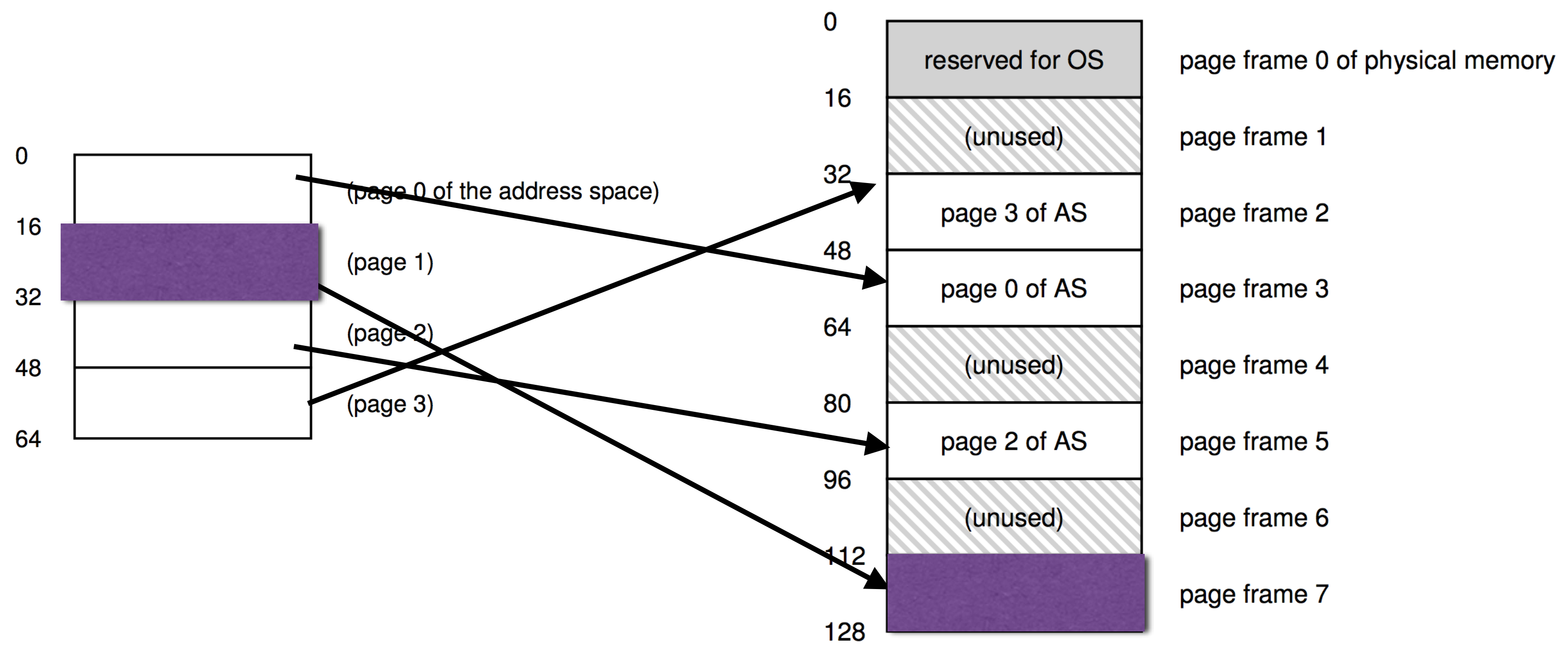




Example

movl 21, %eax





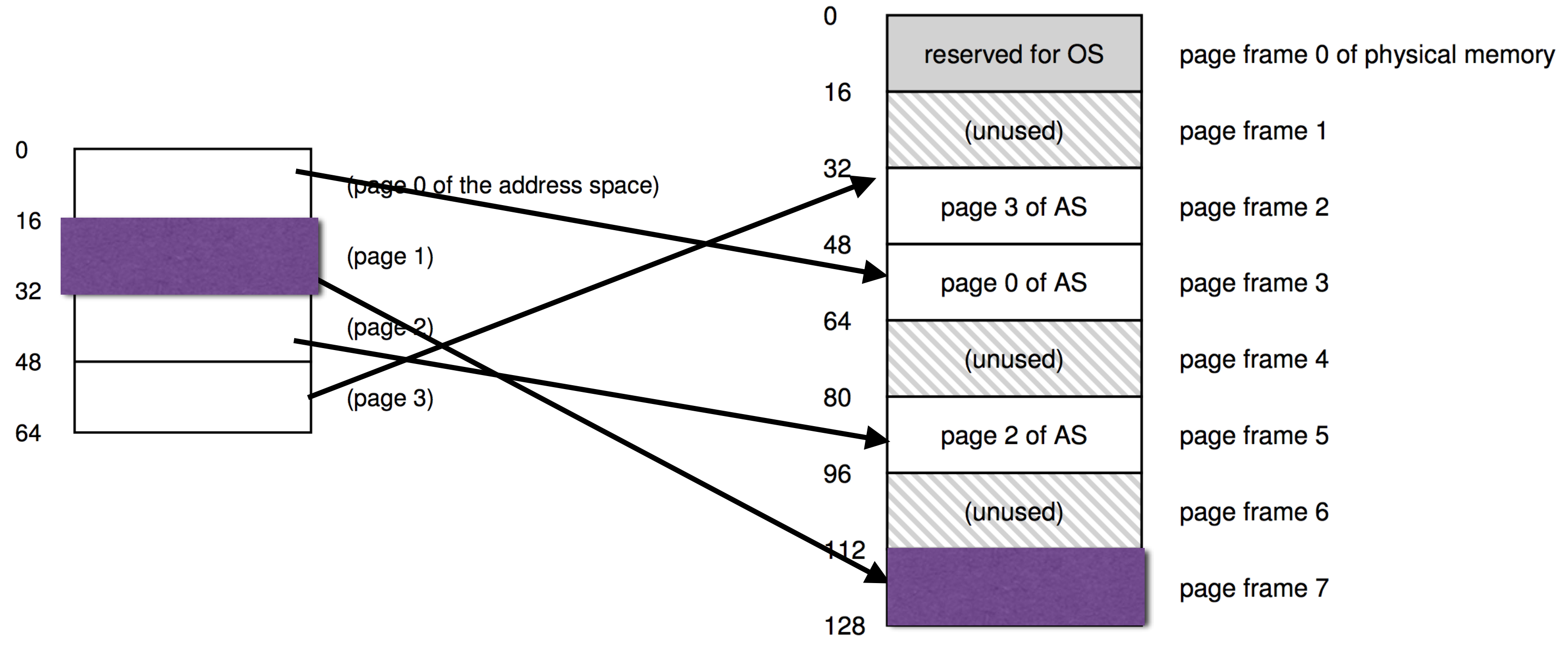
Example

movl 21, %eax

VPN | Offset

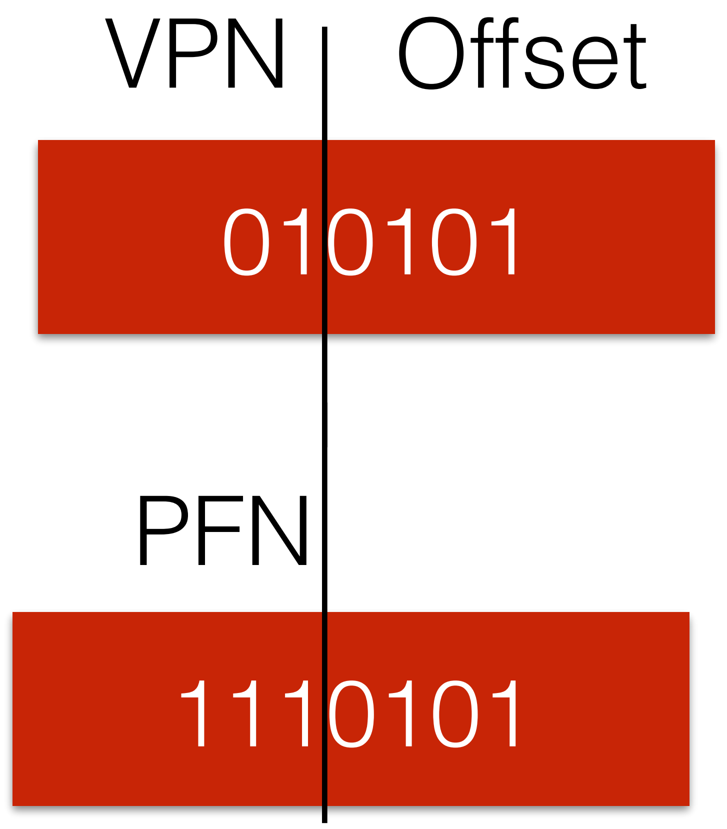
01 | 0101

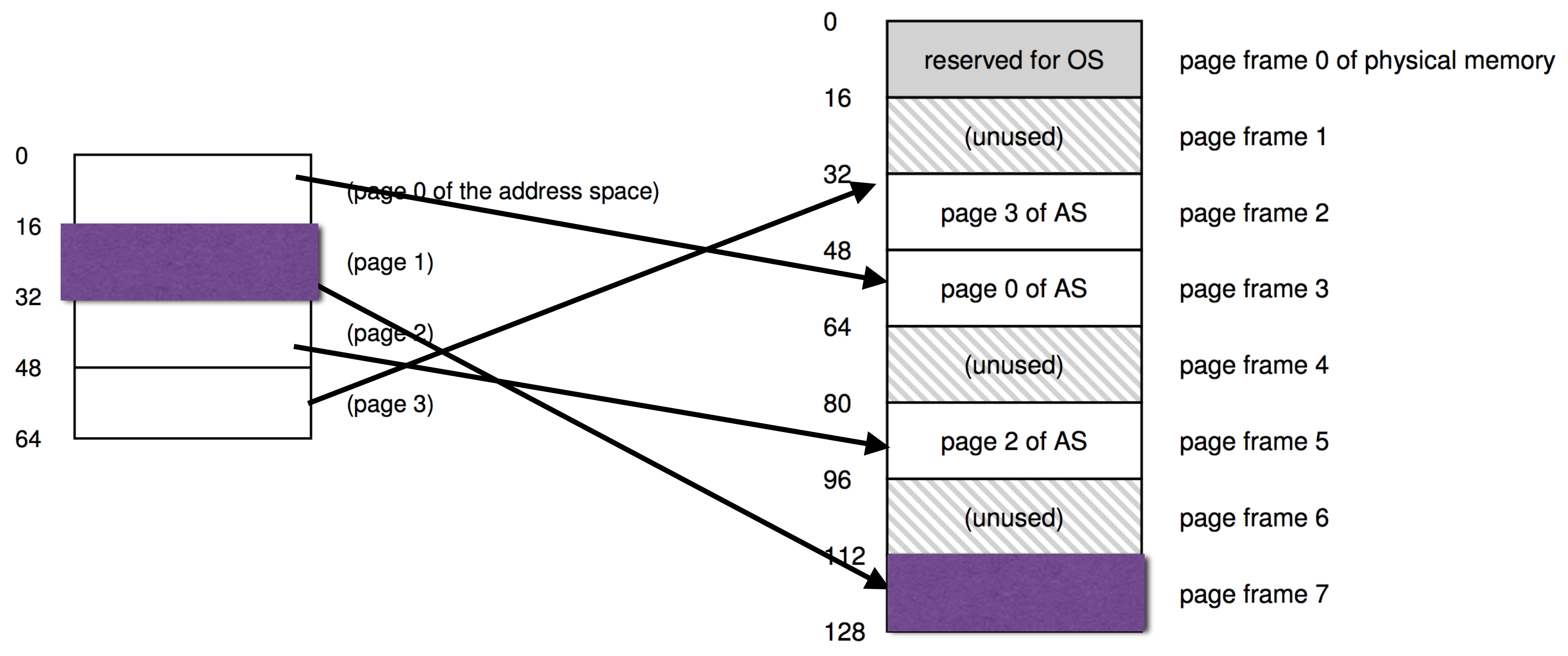
111 | 0101



Example

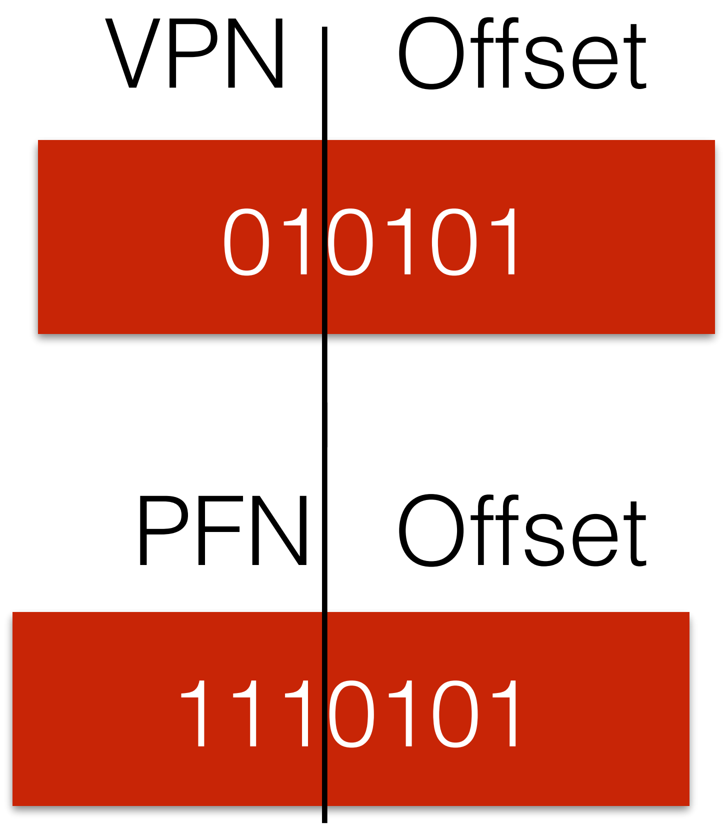
movl 21, %eax

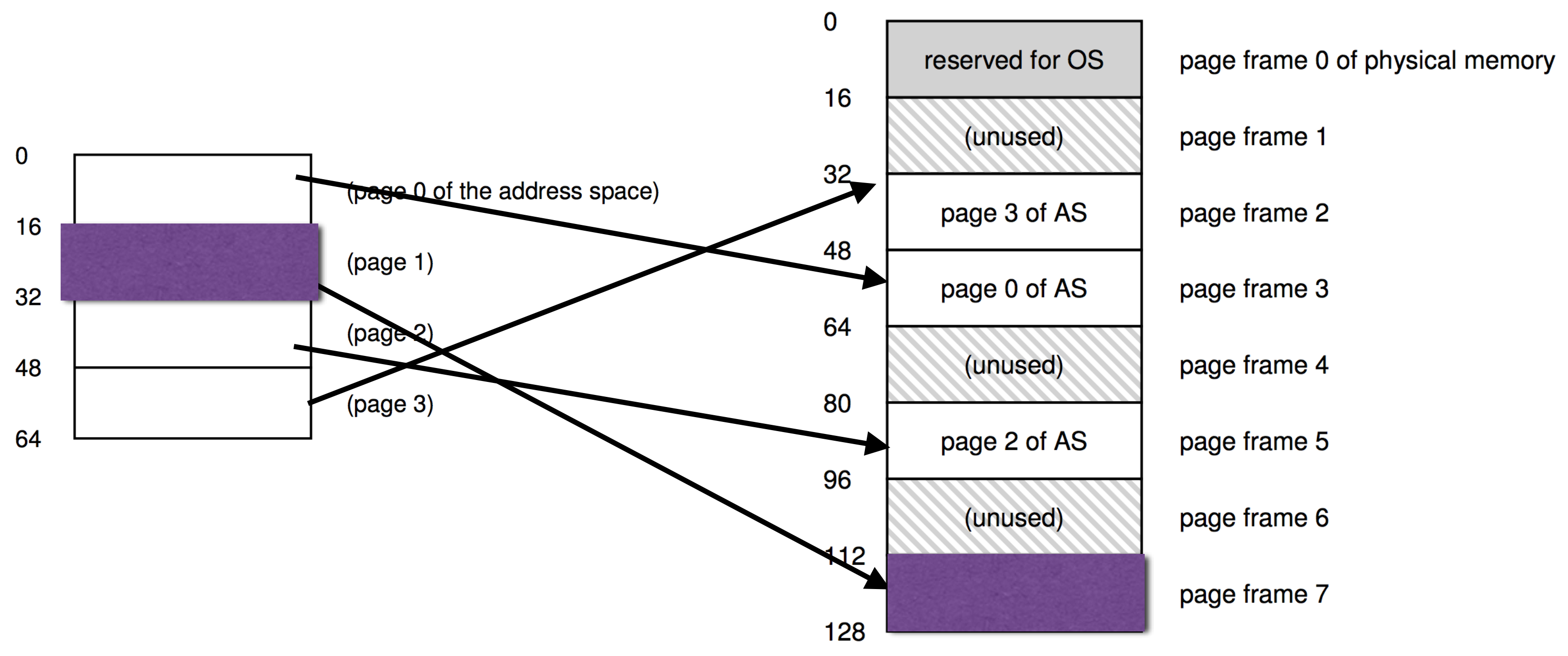




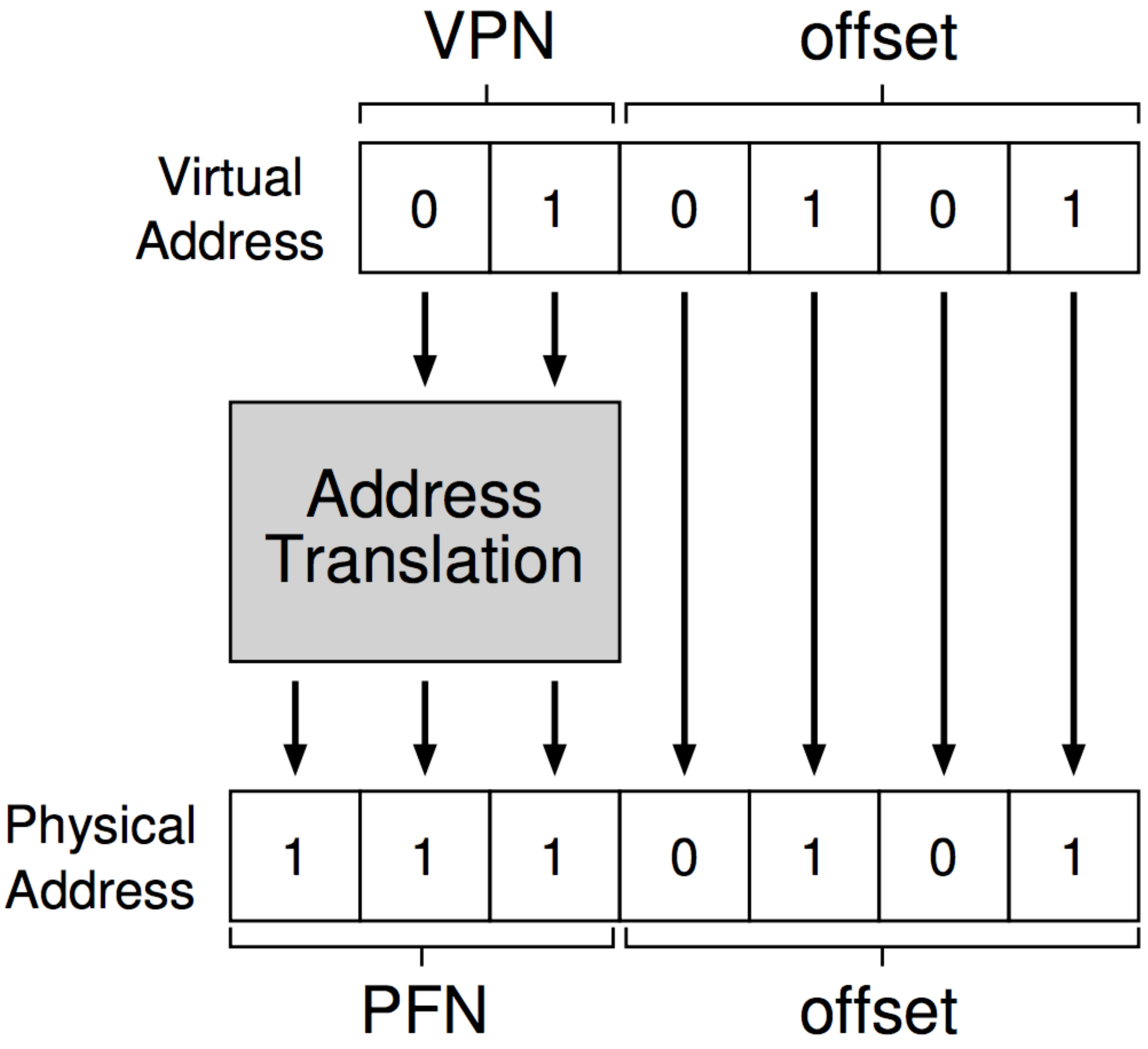
Example

movl 21, %eax





Address Translation Summary



Page Table Storage

Page Table Storage

- Let’s consider 32 bit address space

Page Table Storage

-
- 32 bit address space with 4 KB pages

Page Table Storage

-
-
- 4 KB pages -> ____ bits?

Page Table Storage

-
-
-
    - 12 bits Offset

Page Table Storage

- Remaining bits = 32 - 12 = 20

Page Table Storage

- 20 bit VPN

Page Table Storage

- # pages = 2^20

Page Table Storage

- # translations required = ____

Page Table Storage

- 2^20

Page Table Storage

- 4 bytes per translation -> 4 * 2^20 MB = 4 MB/ process

Page Size Tradeoffs?

Page Size Tradeoffs?

- Small size

Page Size Tradeoffs?

-
    - More # of translations

Page Size Tradeoffs?

-
    -
        - More memory overhead/process

Page Size Tradeoffs?

-
    -
        -
        - Less chances of fragmentation

Page Size Tradeoffs?

- Large size

Page Size Tradeoffs?

-
    -
        -
        -
-
    - Less # of translations

Page Size Tradeoffs?

-
    -
        -
        -
-
    -
        - Less memory overhead/process

Page Size Tradeoffs?

-
    -
        -
        -
-
    -
        -
        - More chances of fragmentation

Page Table Storage

Page Table Storage

Not really stored on MMU

Page Table Storage

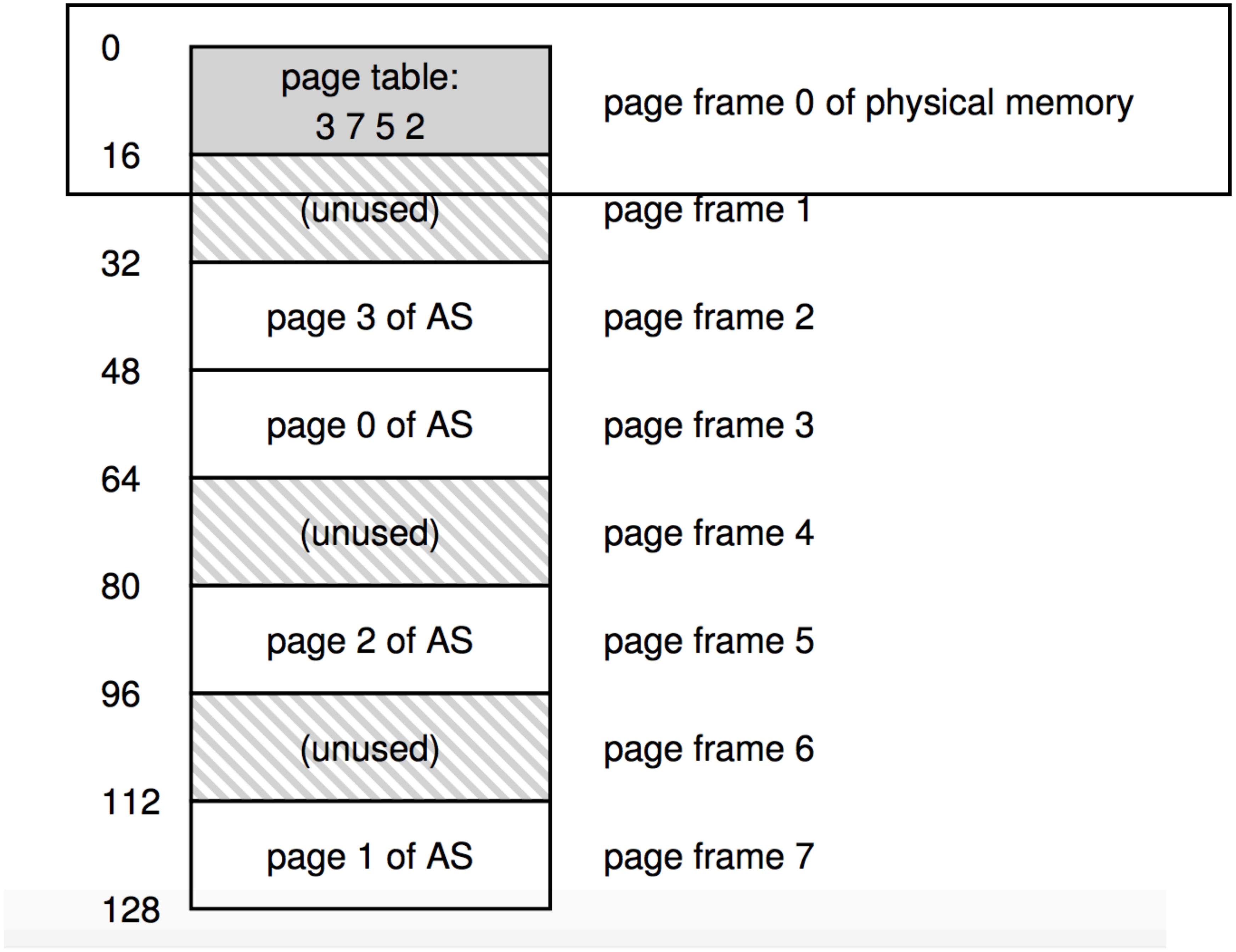
Not really stored on MMU

- In memory

Page Table Storage

Not really stored on MMU

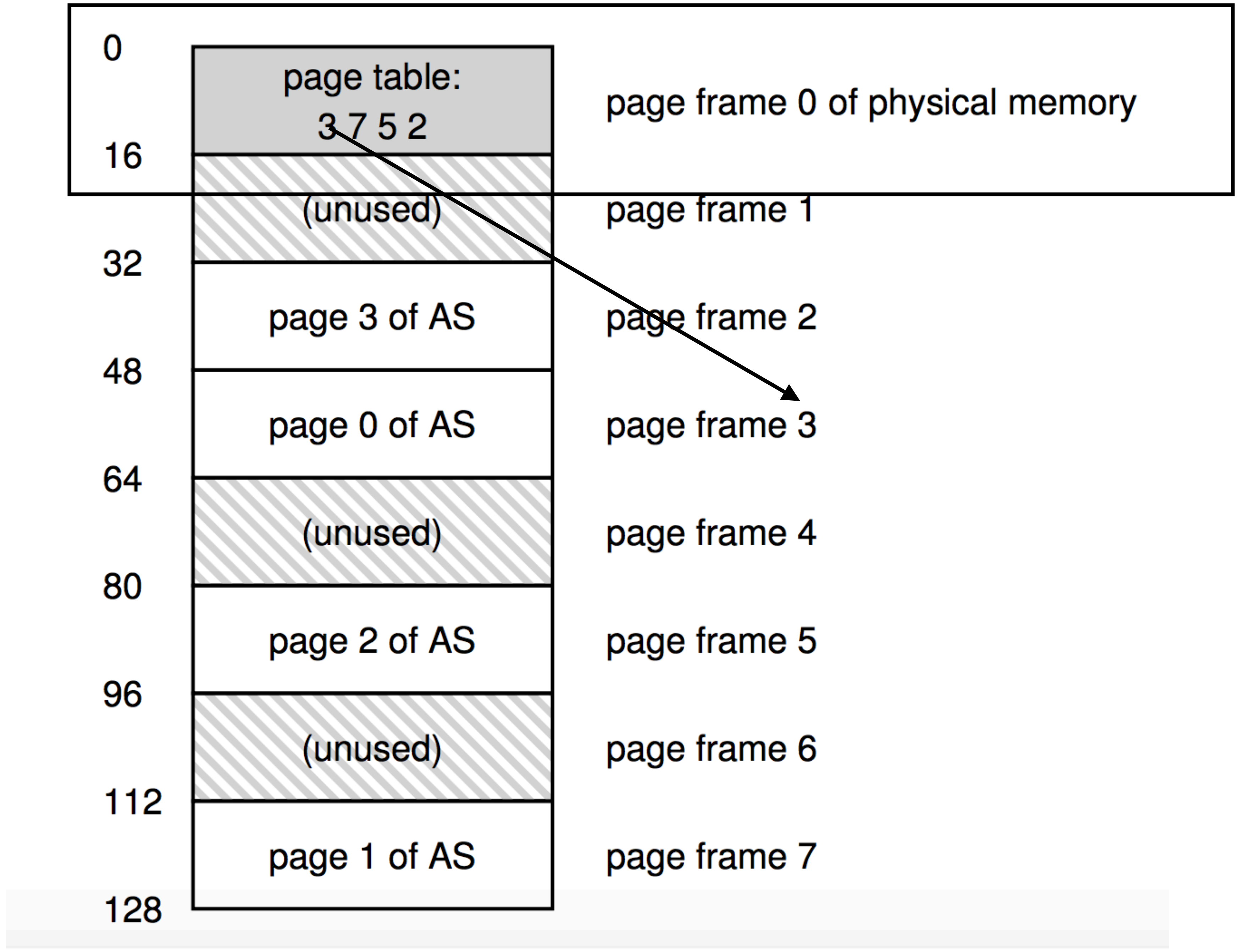
- In memory



Page Table Storage

Not really stored on MMU

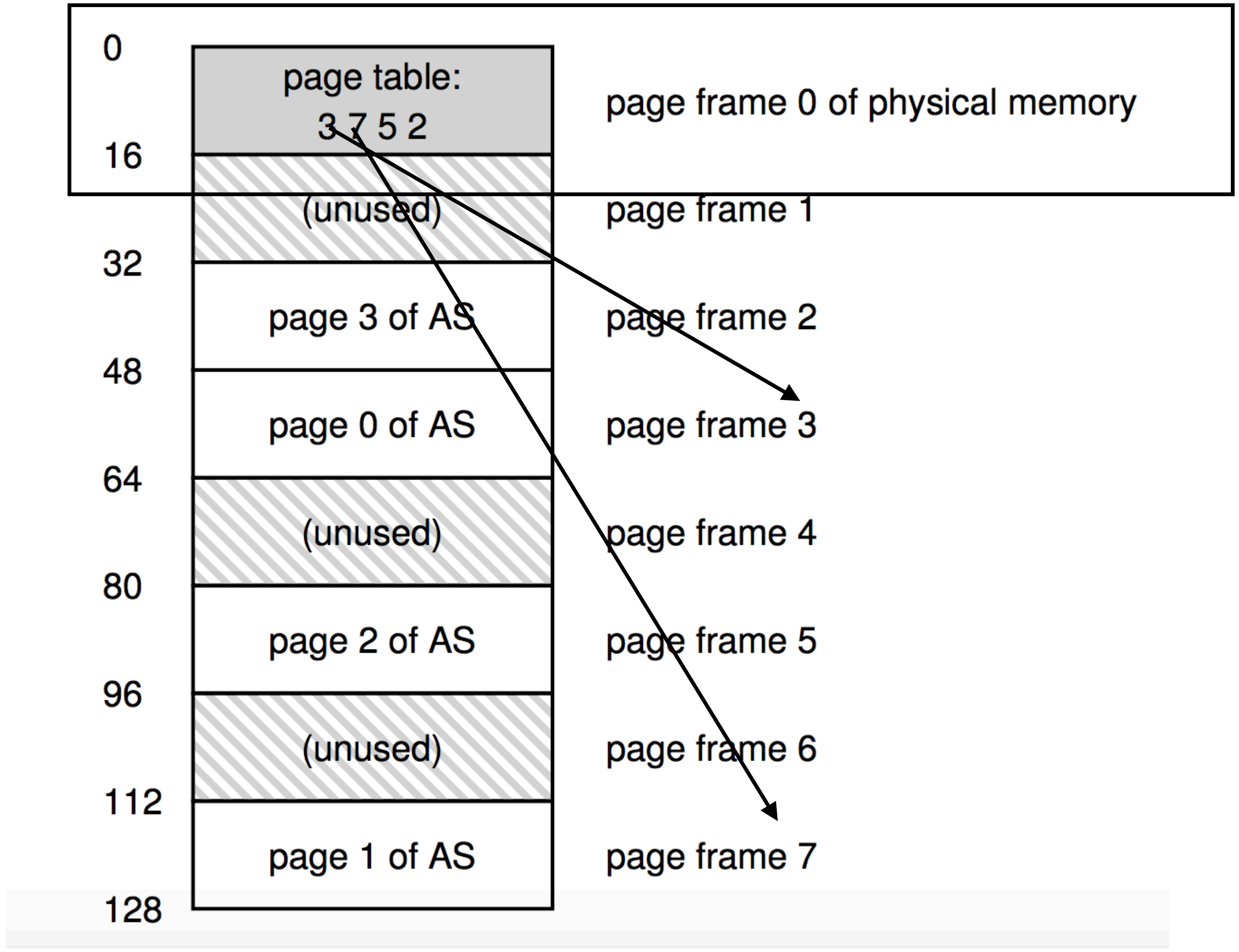
- In memory



Page Table Storage

Not really stored on MMU

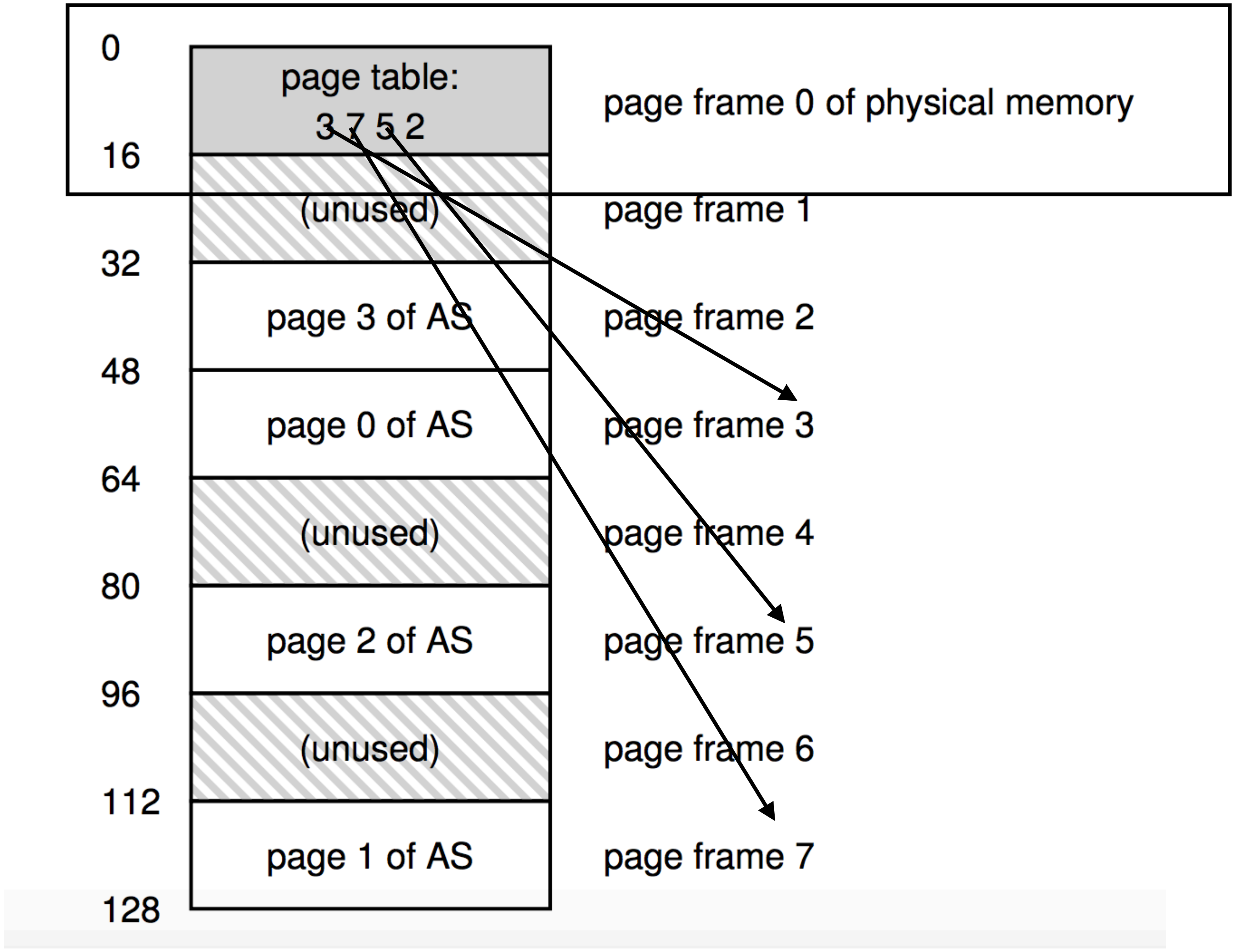
- In memory



Page Table Storage

Not really stored on MMU

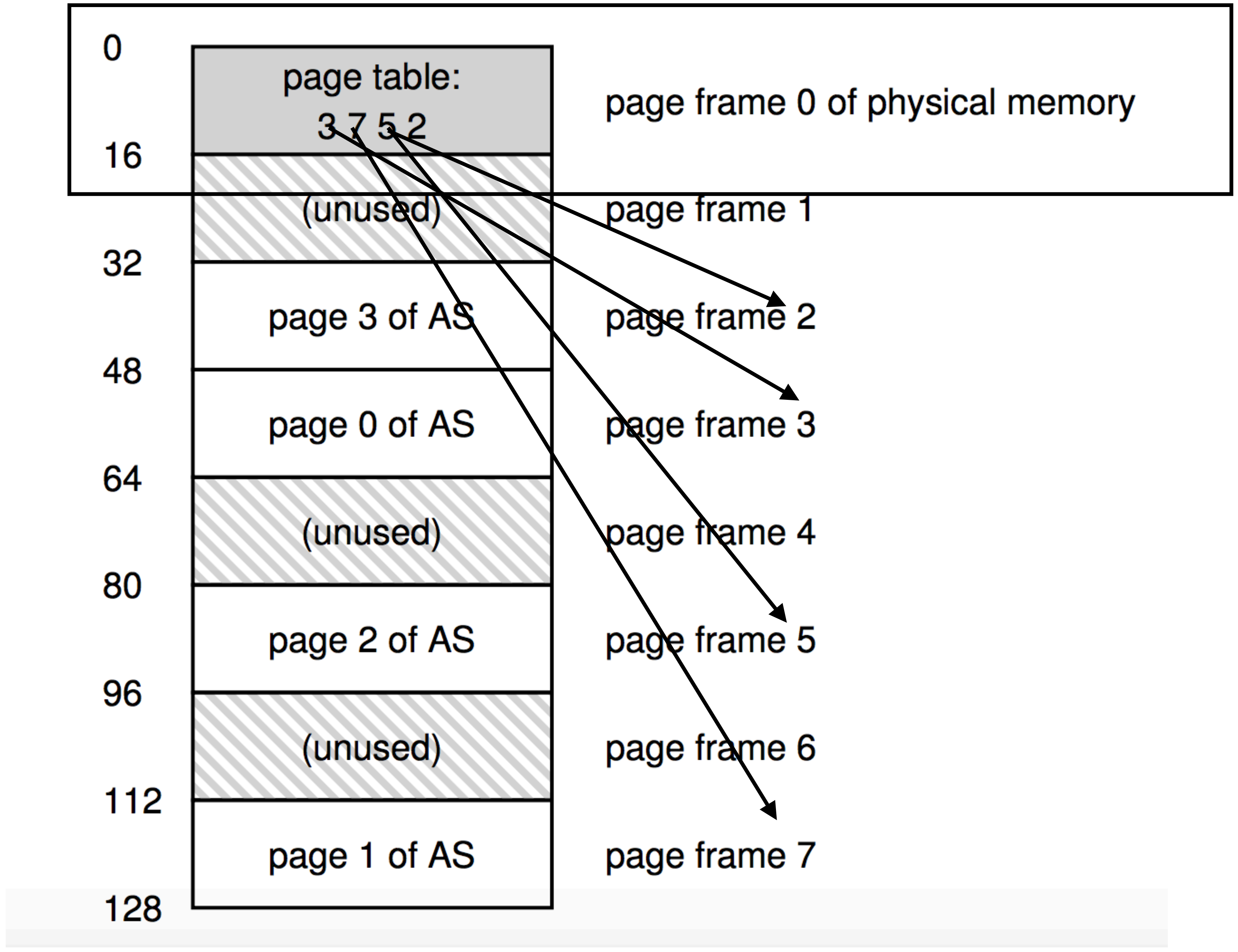
- In memory



Page Table Storage

Not really stored on MMU

- In memory

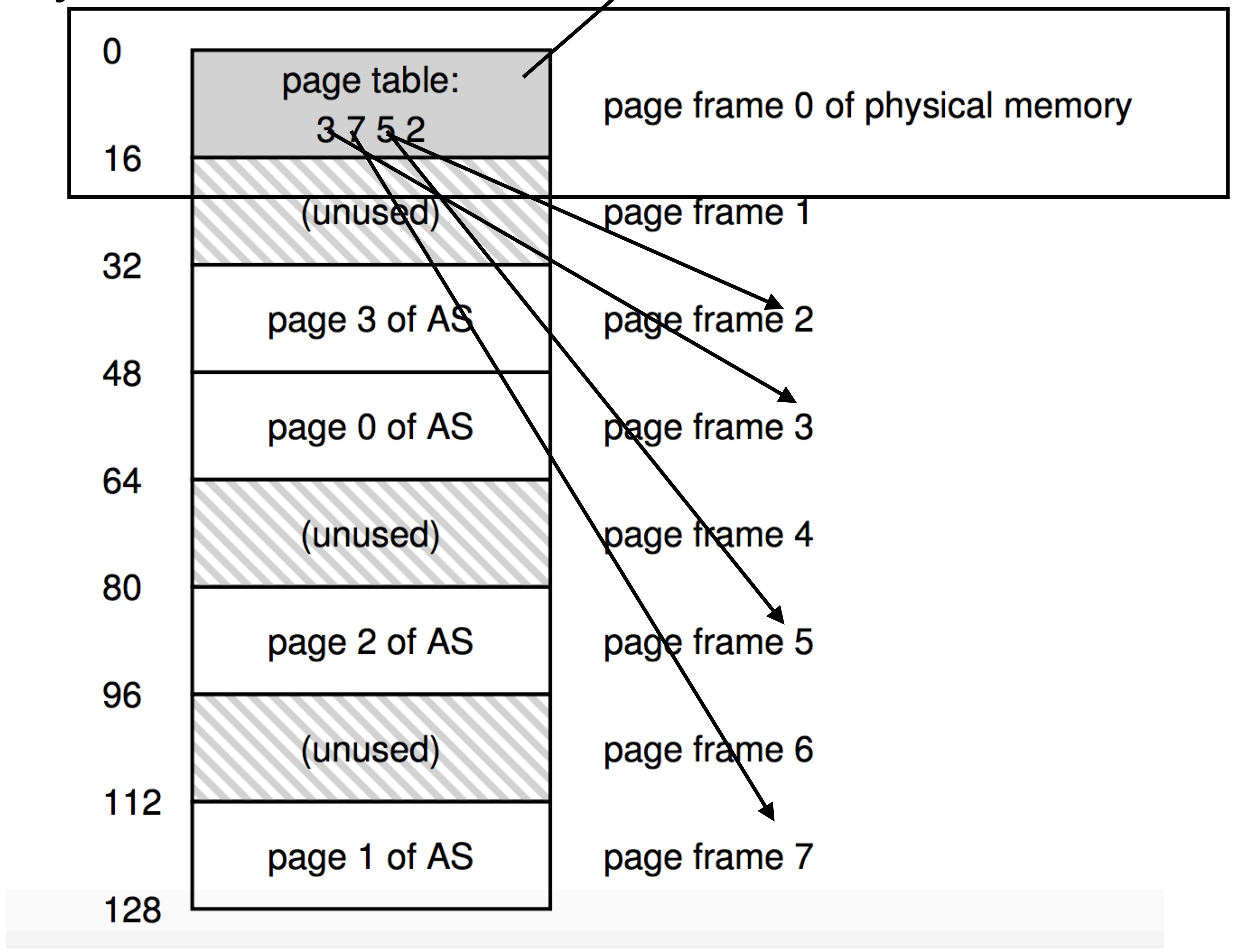


Page Table Storage

Not really stored on MMU

- In memory

Linear page table



What else is in the Page Table?

What else is in the Page Table?

What else is in the Page Table?

- Protection bit : Read/Write/Execute?

What else is in the Page Table?

-
- Present bit: On Memory or HDD/SSD?

What else is in the Page Table?

-
-
- Reference bit: Is the page popular/being referenced?

What else is in the Page Table?

-
-
-
    - Else?

What else is in the Page Table?

-
-
-
  -
- Valid bit: Is translation valid?

What else is in the Page Table?

- Dirty bit: Modified since brought to memory?

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

$\leftarrow$ *(EDI + 4*EAX) = 0

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Address of array[0]

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

*(EDI + 4*EAX) = 0

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Index into array (i)

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

*(EDI + 4*EAX) = 0

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Index into array (i)

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

*(EDI + 4*EAX) = 0

I = I + 1

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Index into array (i)

```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

*(EDI + 4*EAX) = 0

I = I + 1

Is I == 1000

Worked Out Example

```
int array[1000];
...
for (i = 0; i < 1000; i++)
    array[i] = 0;
```

Index into array (i)

```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

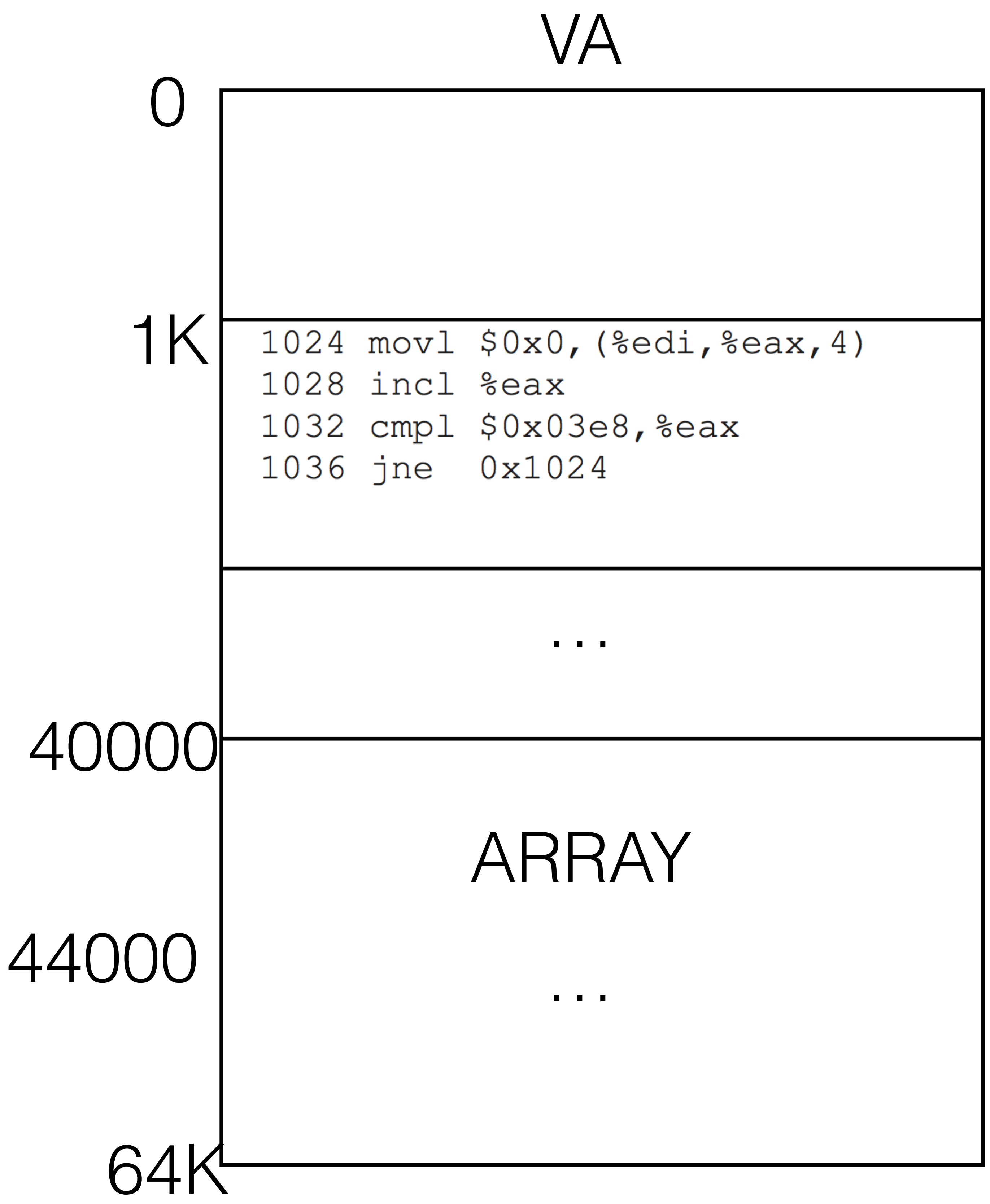
*(EDI + 4*EAX) = 0

I = I + 1

Is I == 1000

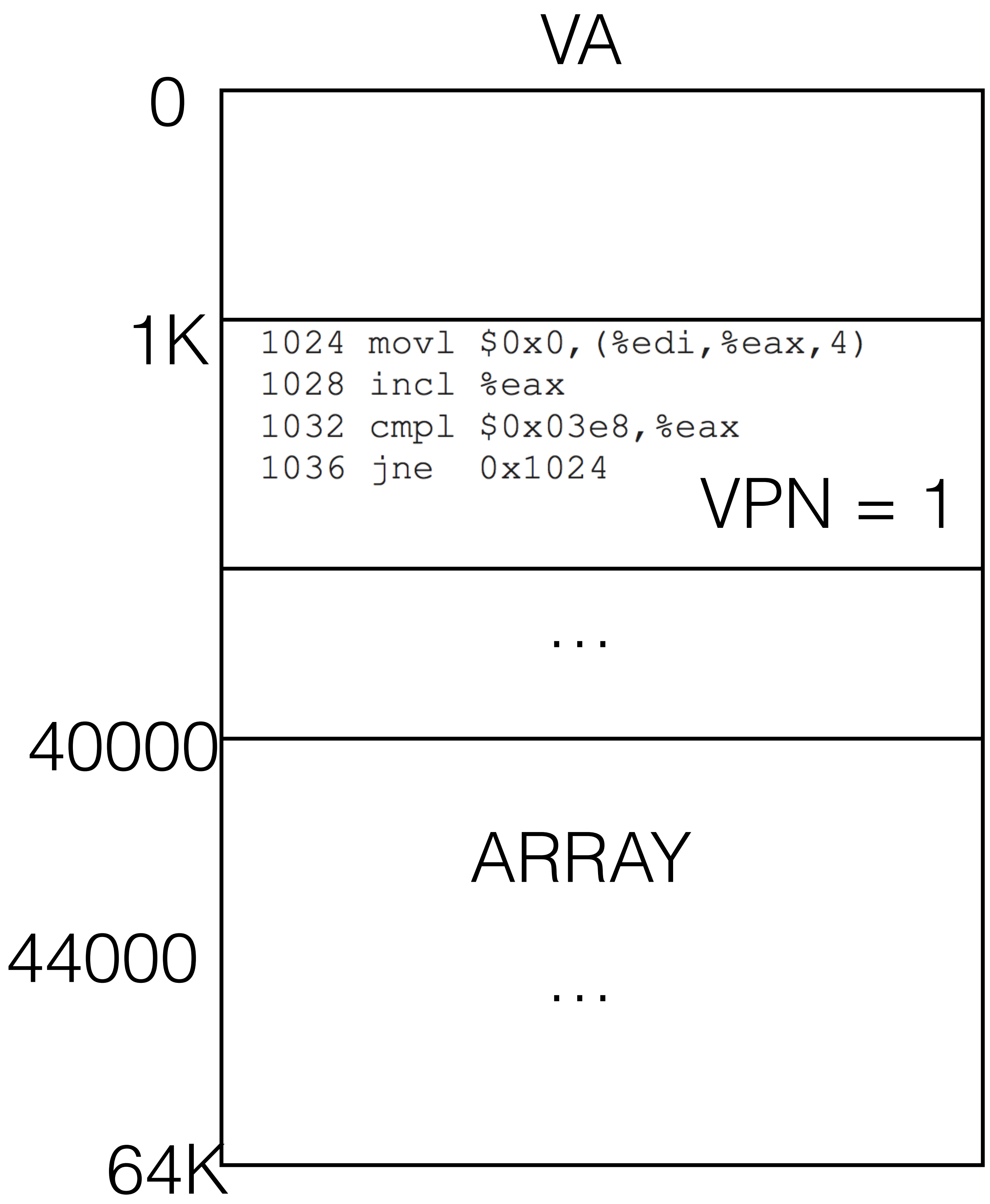
If Above is False

Worked Out Example



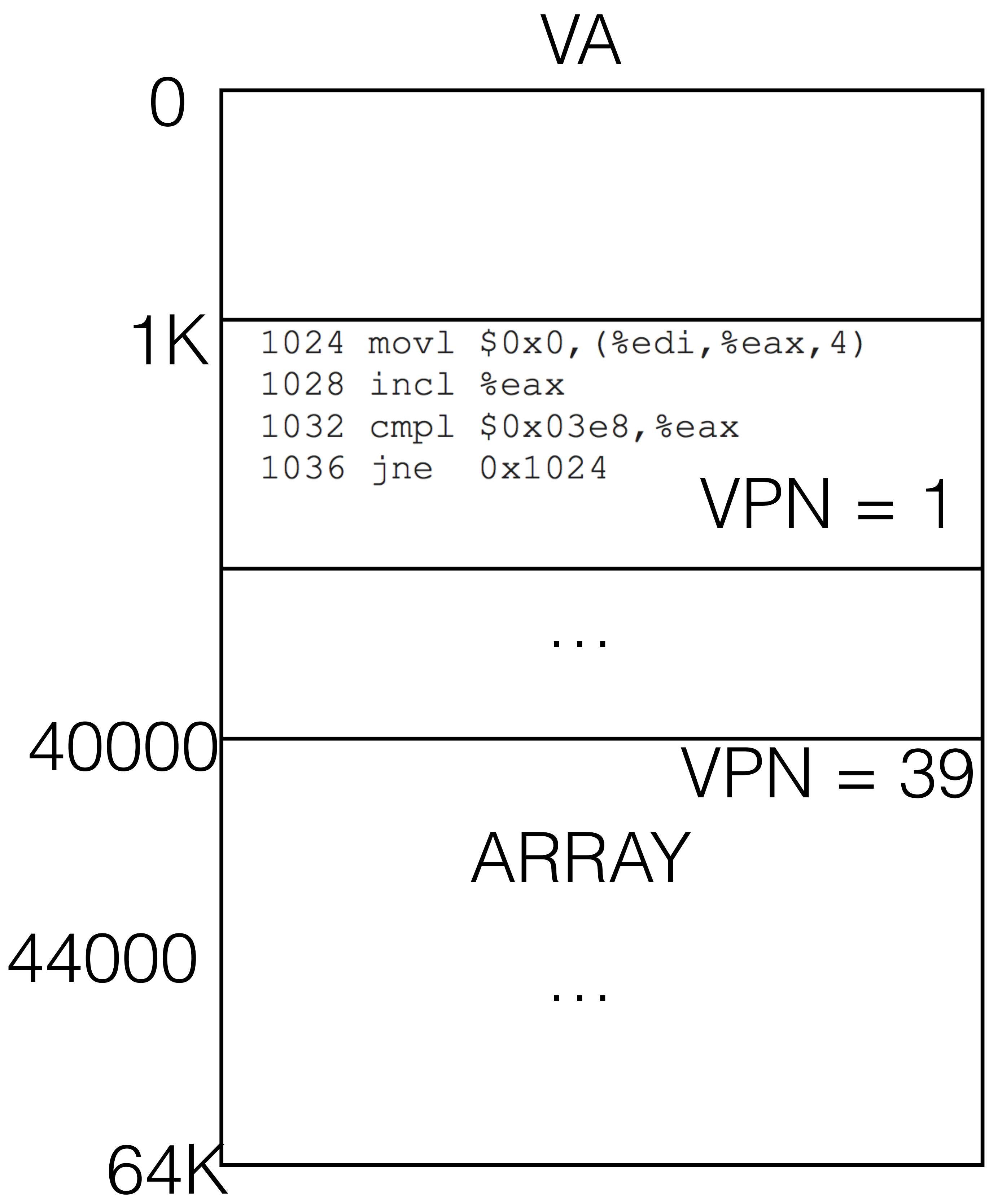
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



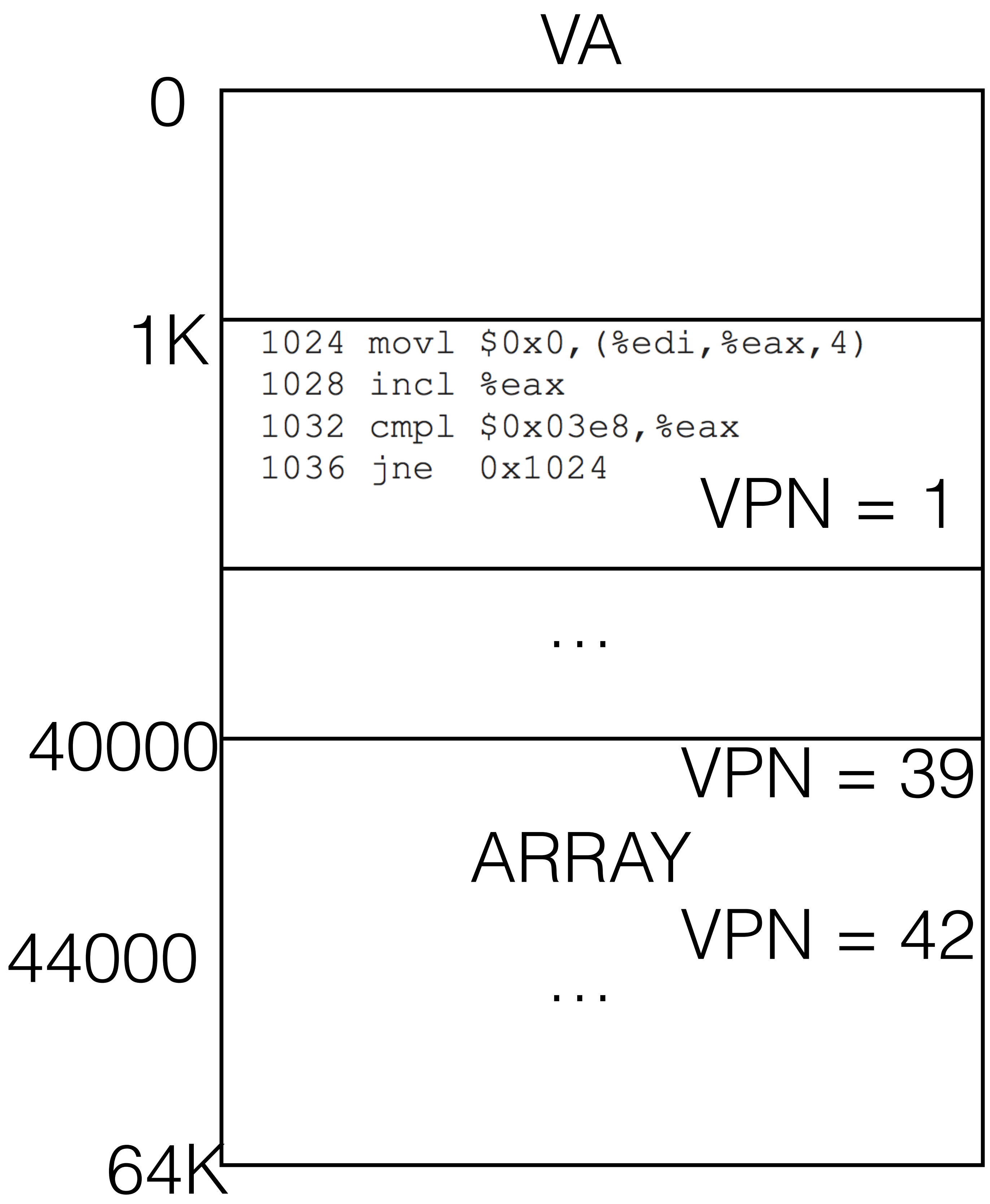
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



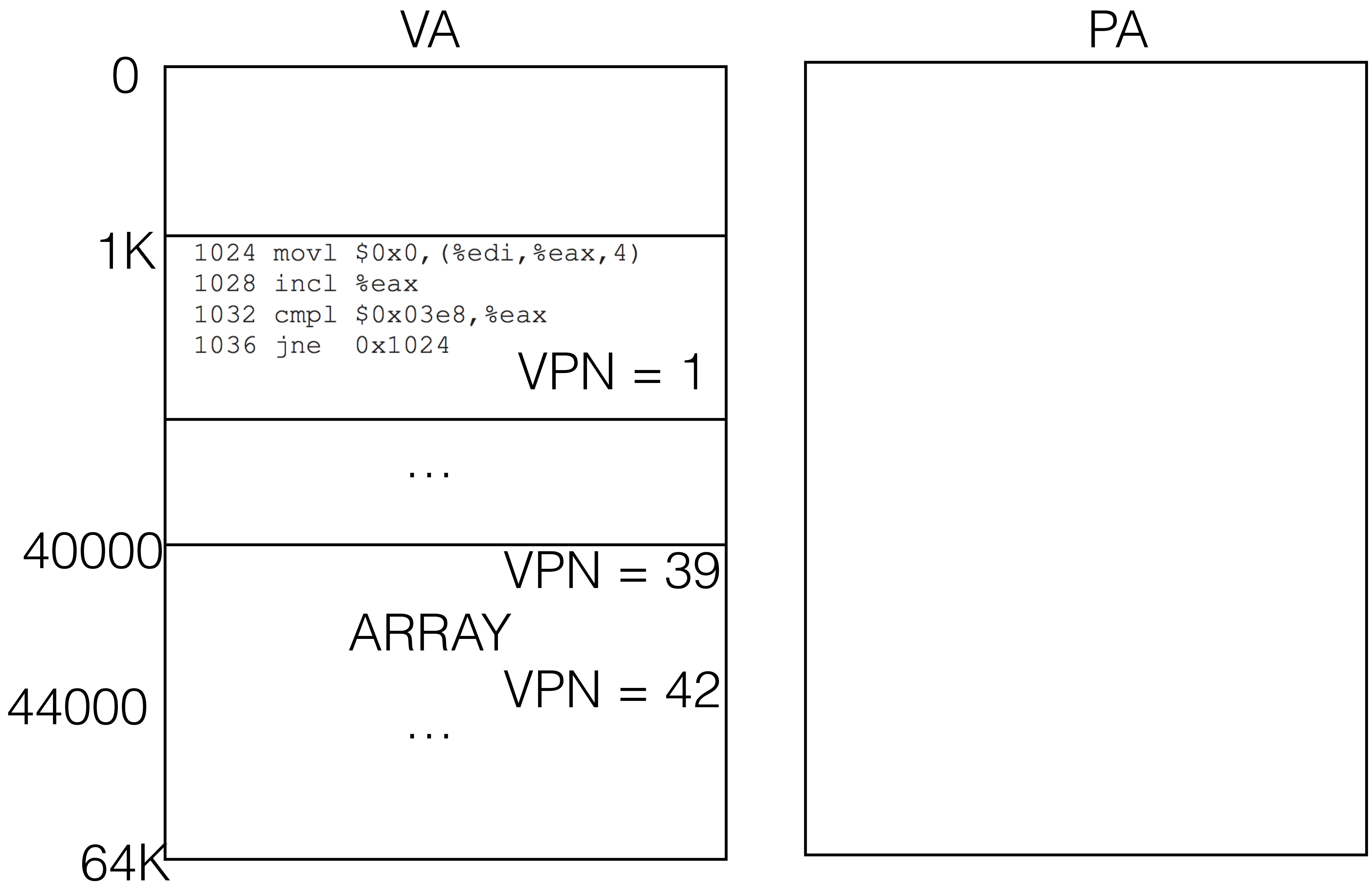
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



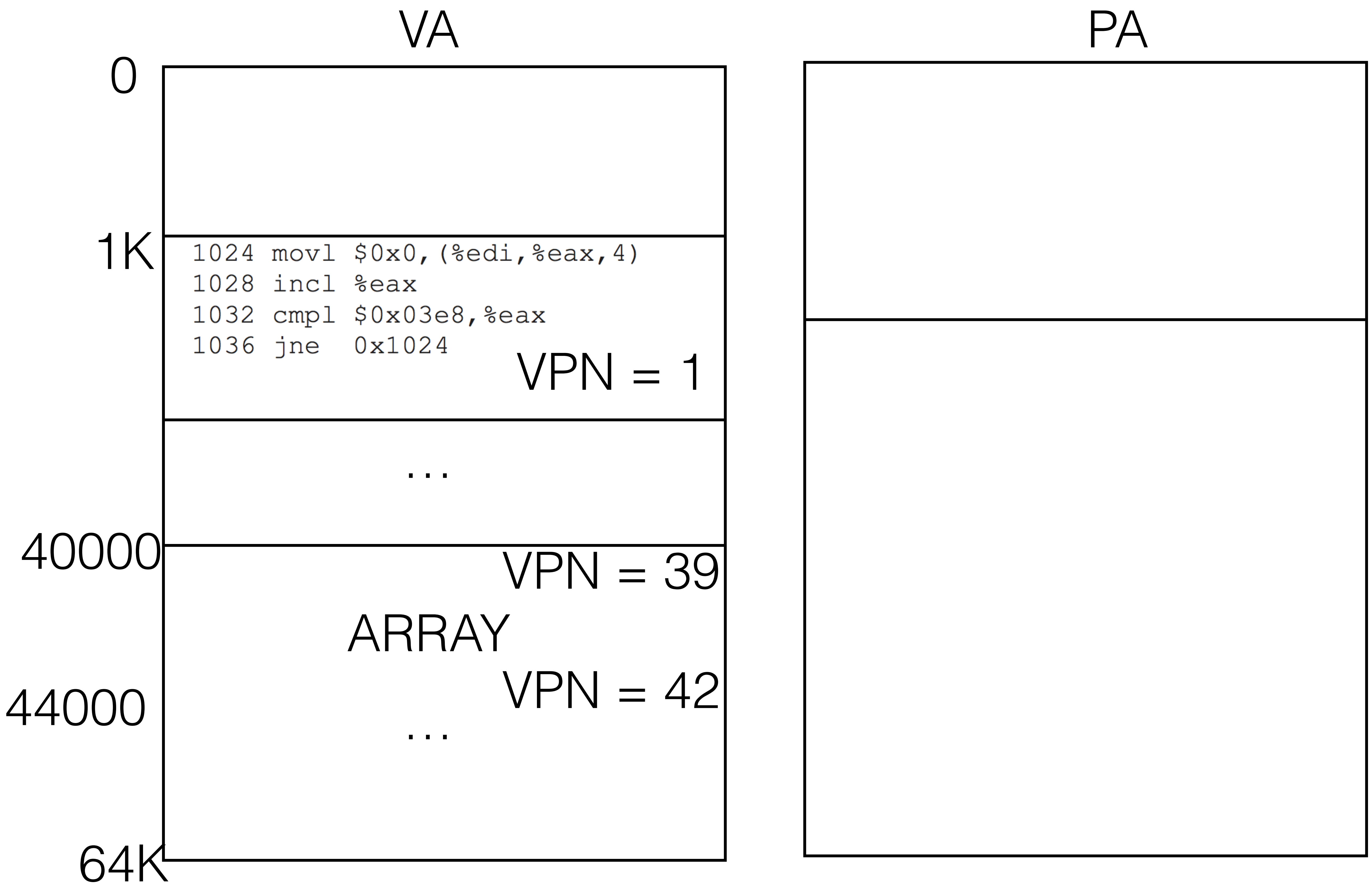
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



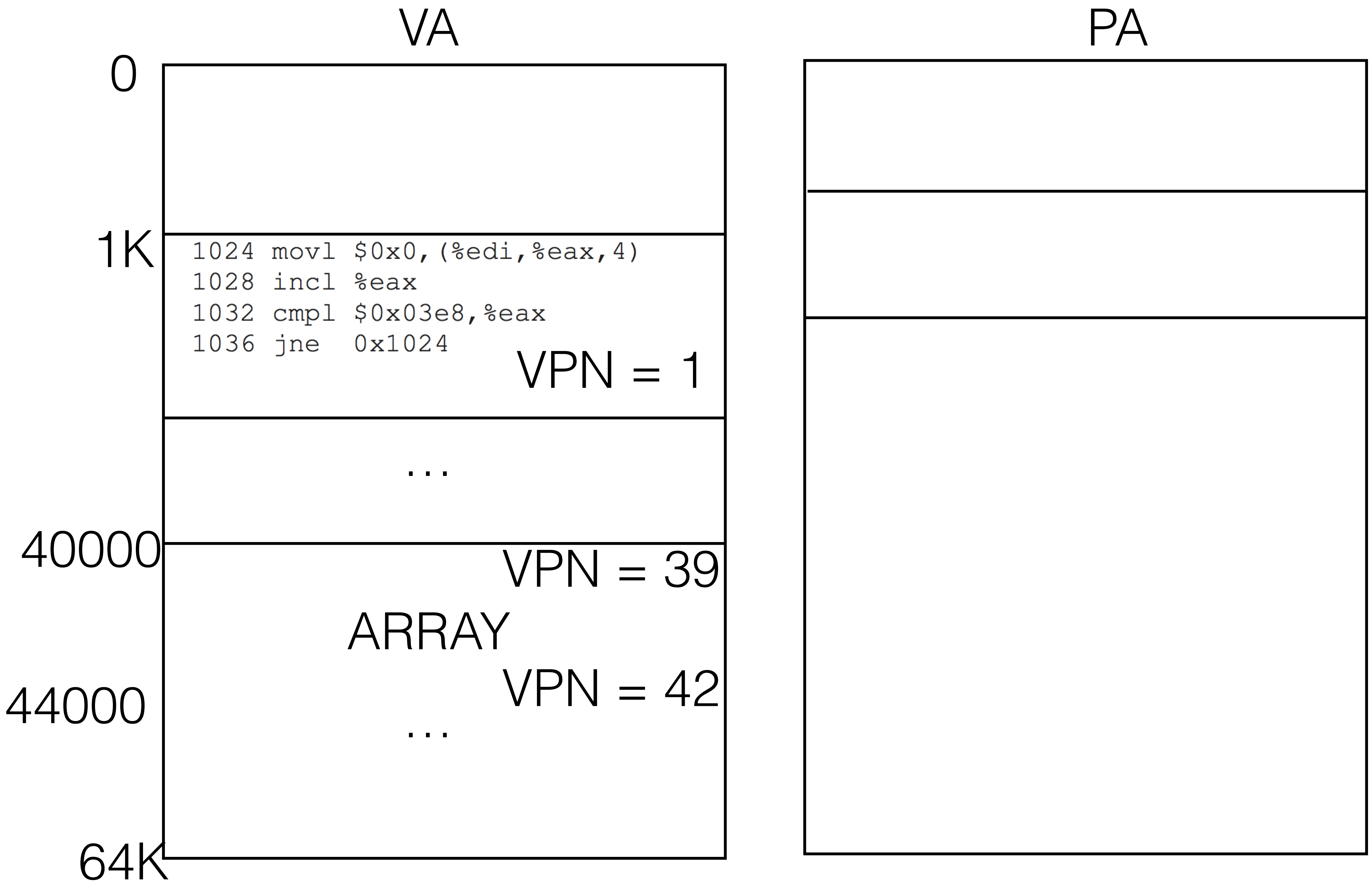
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



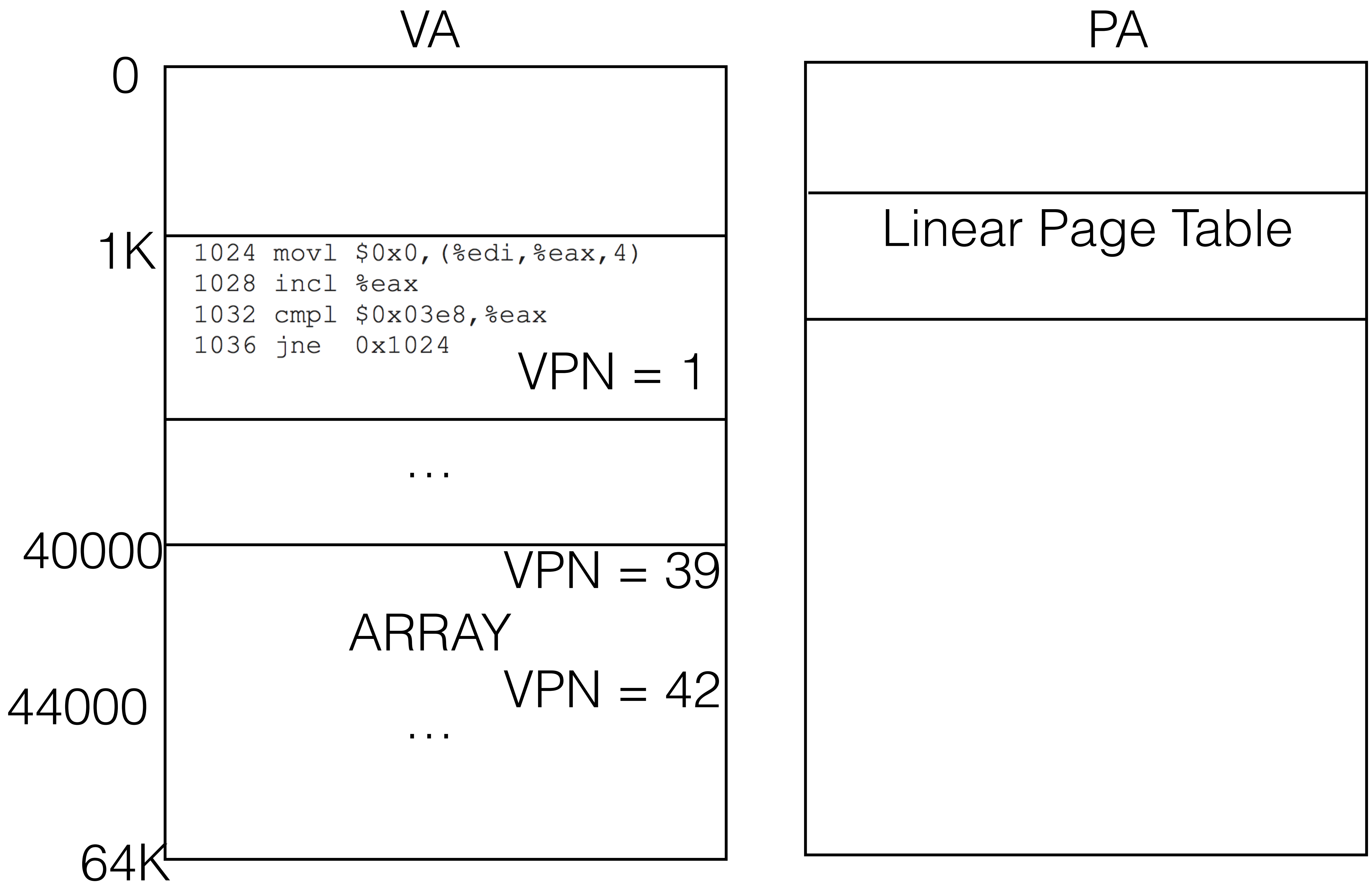
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



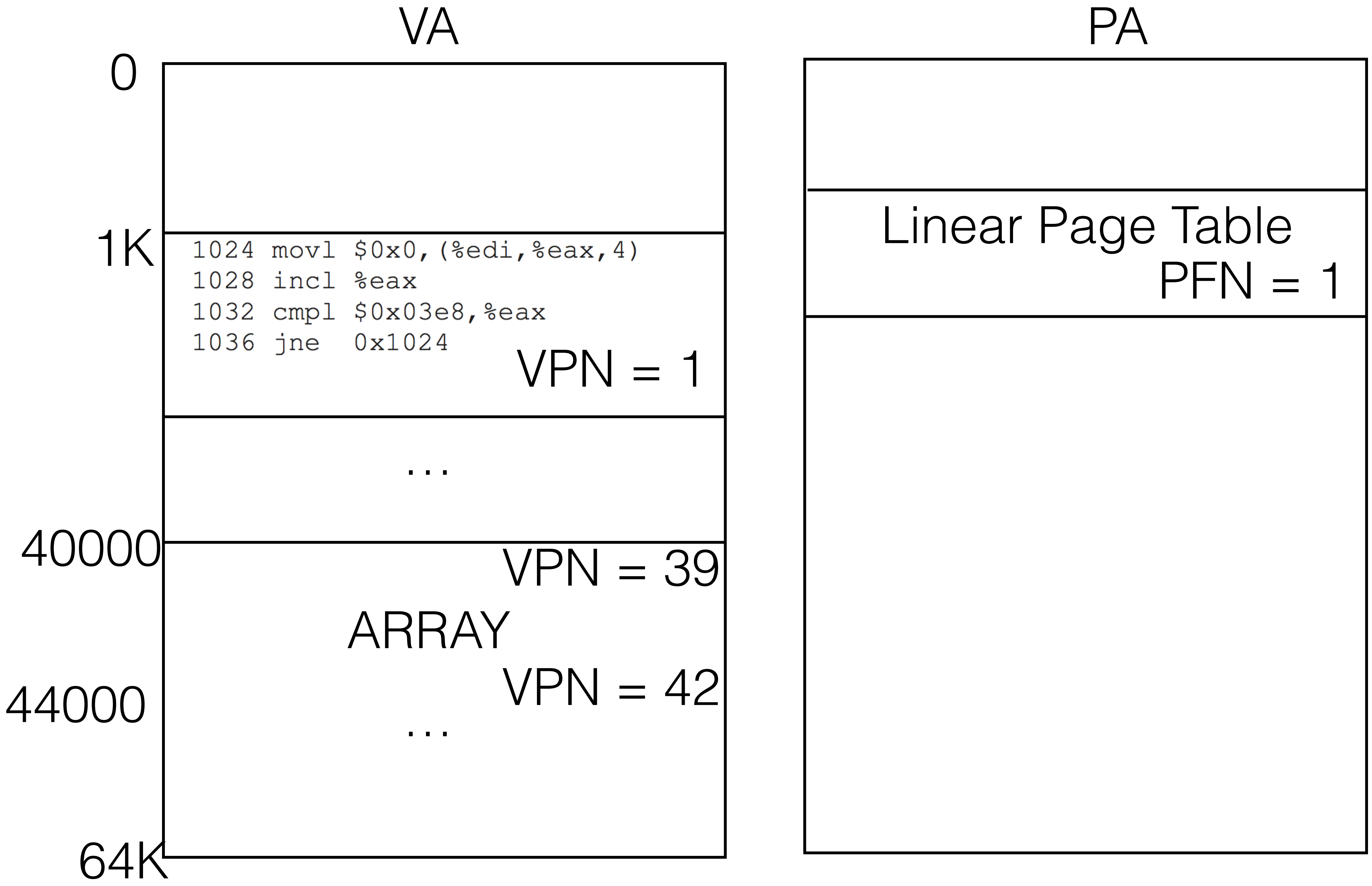
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



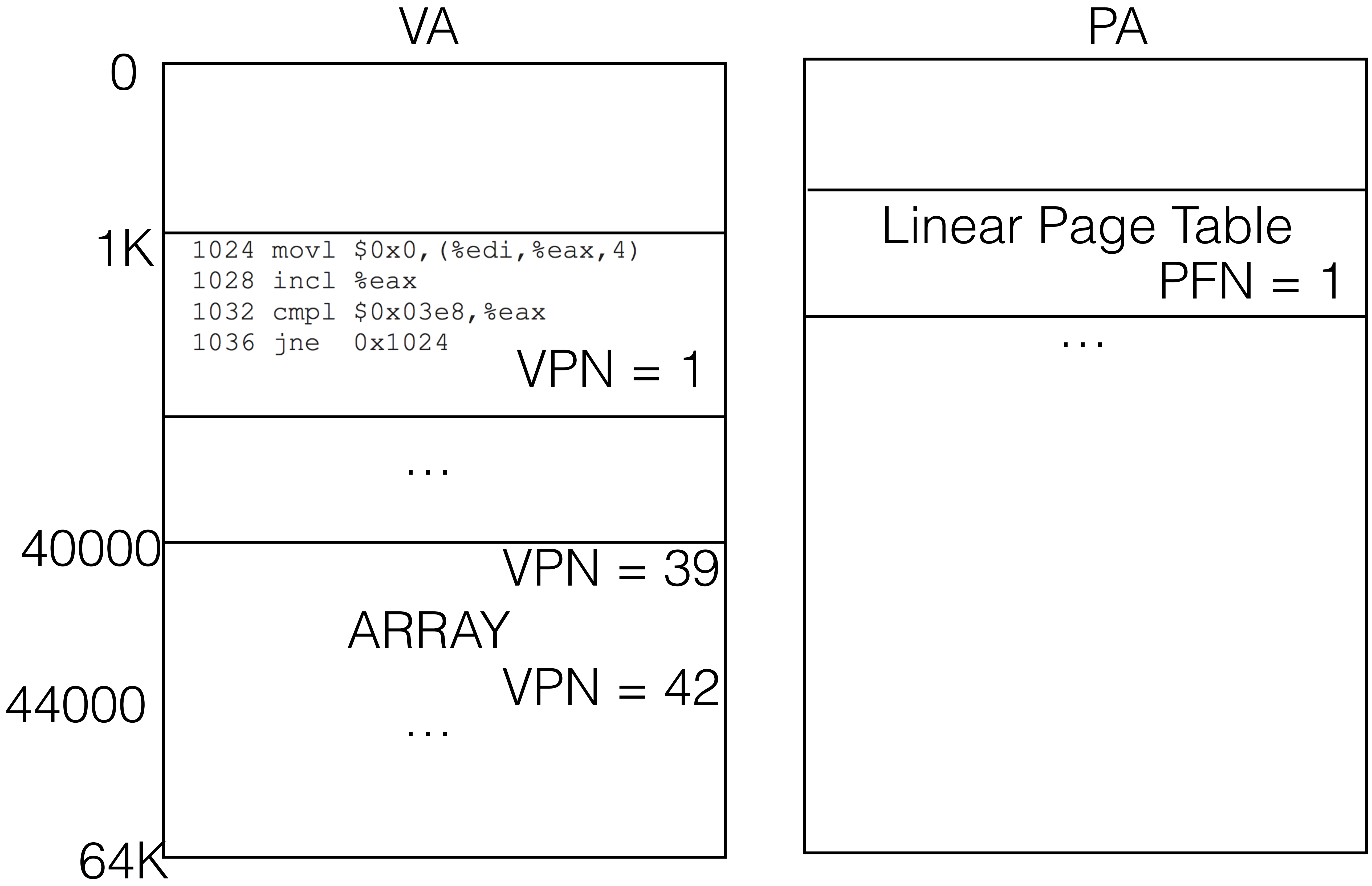
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



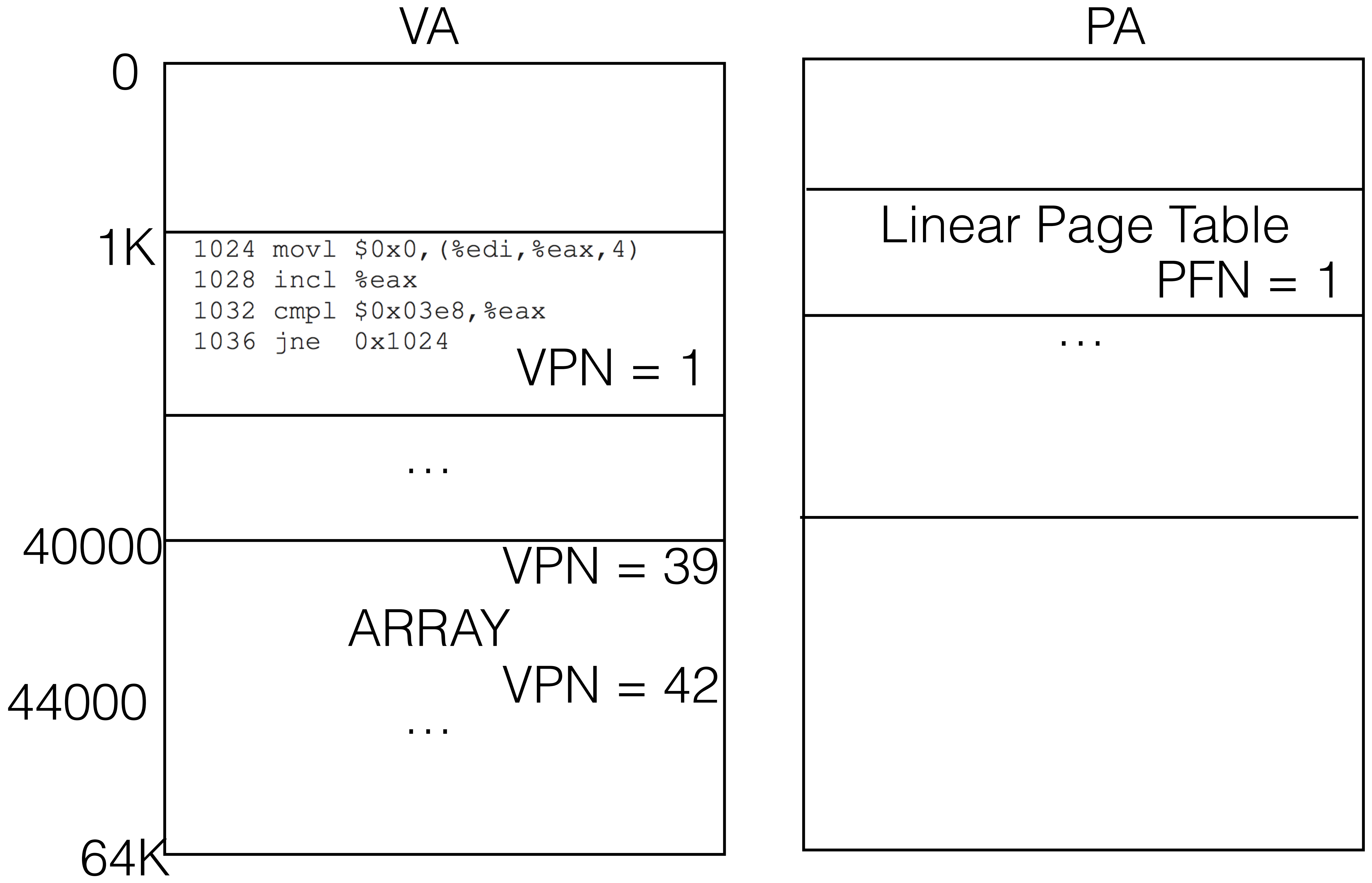
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



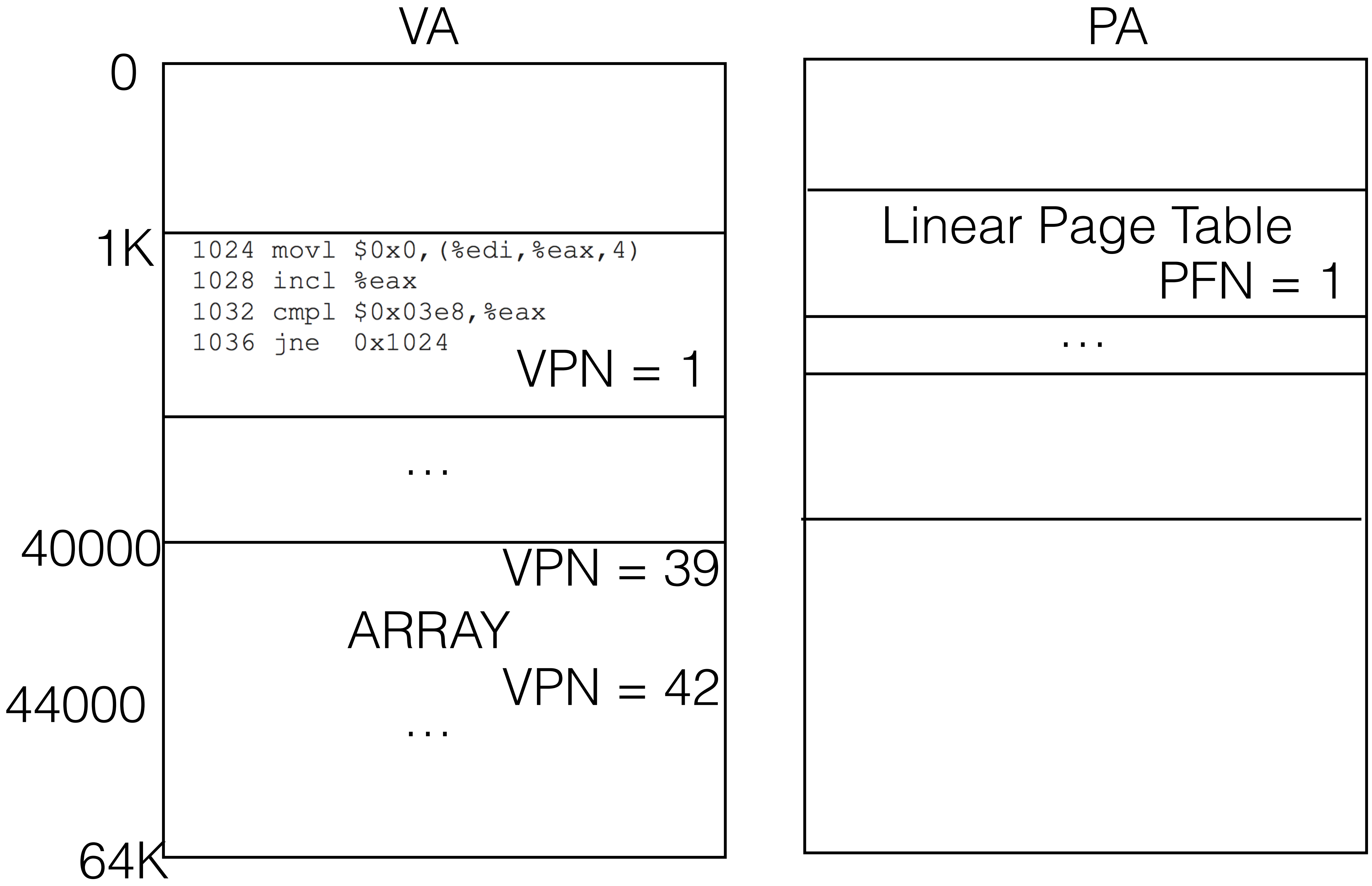
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



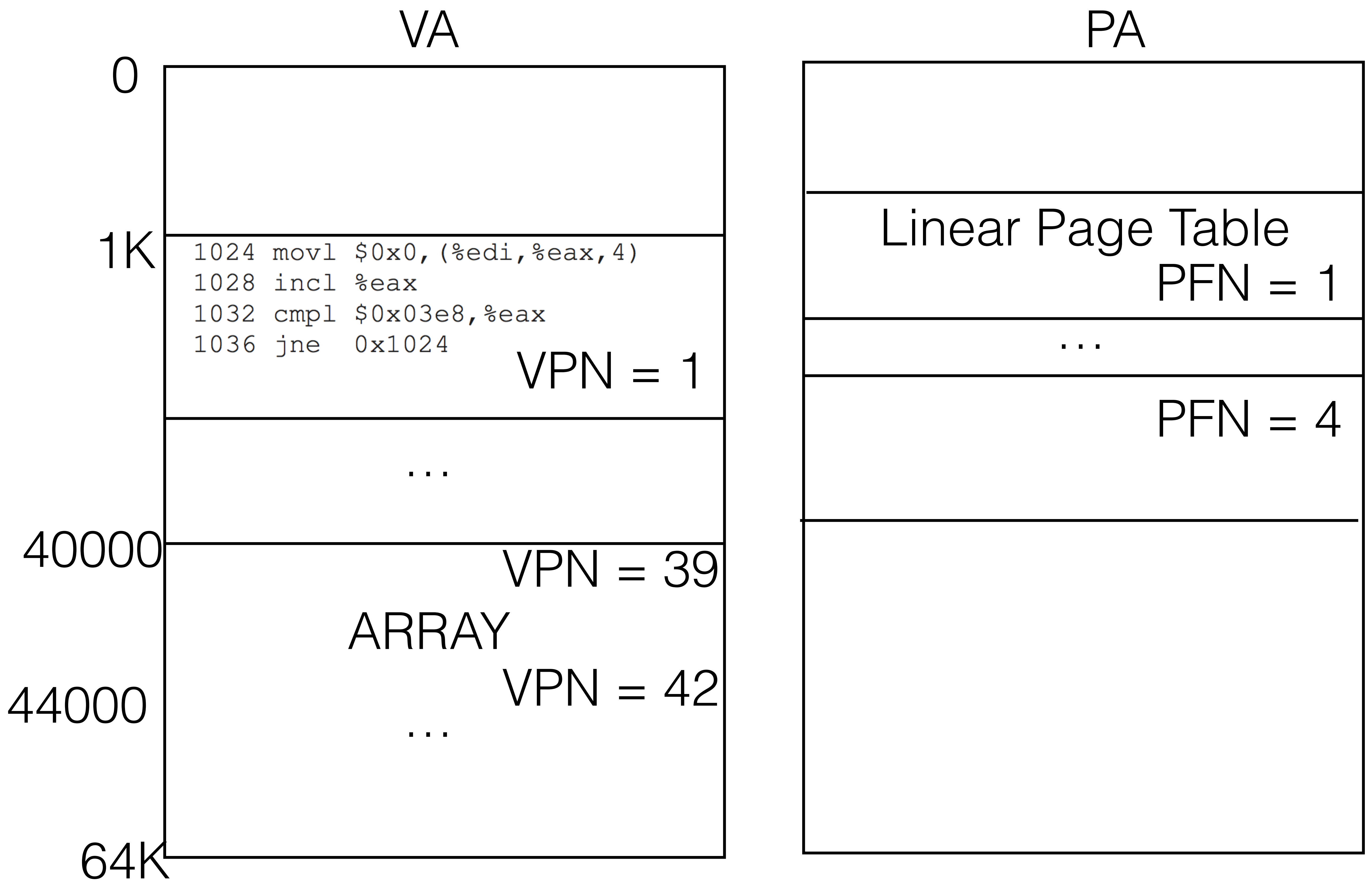
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



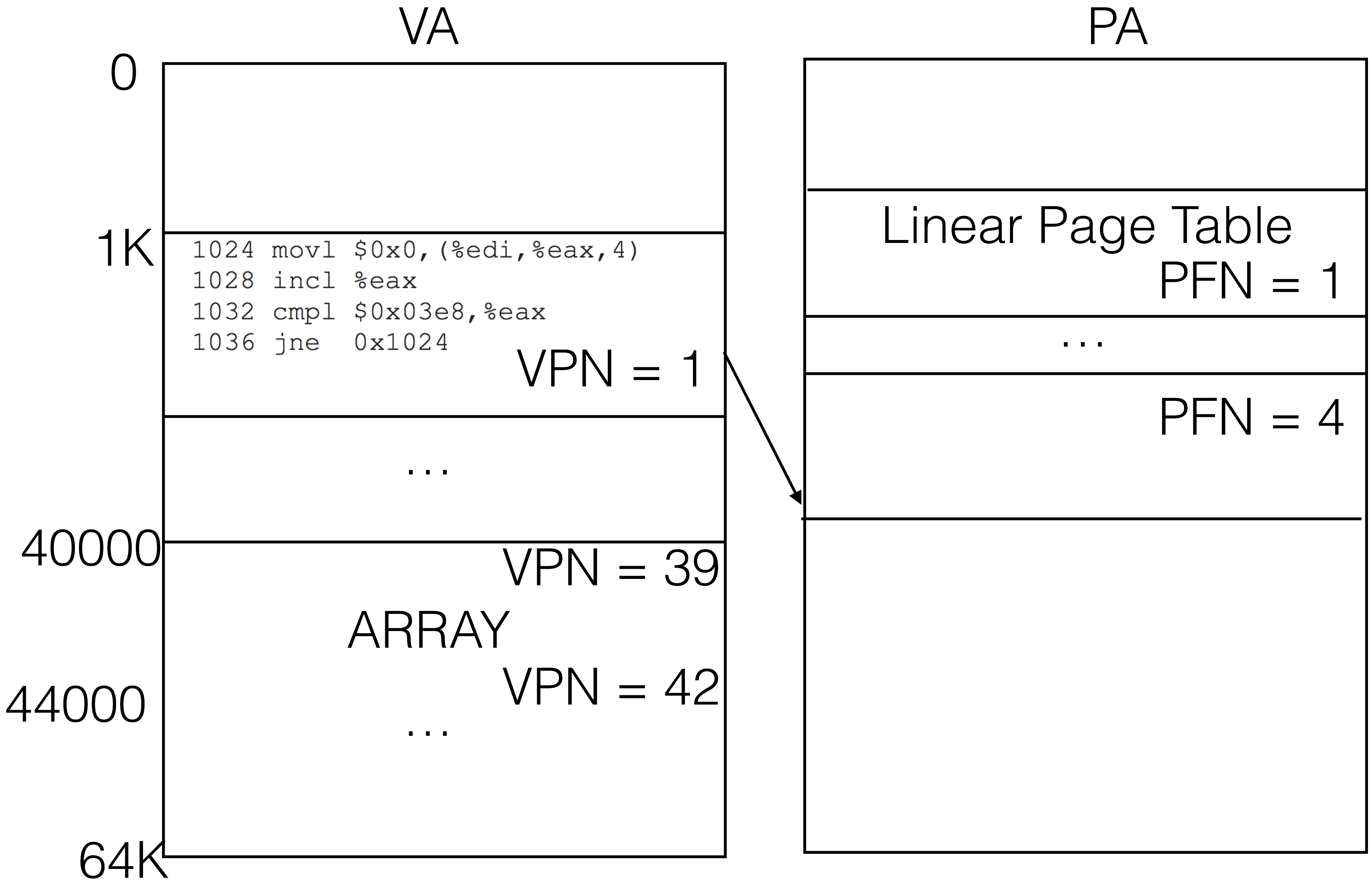
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



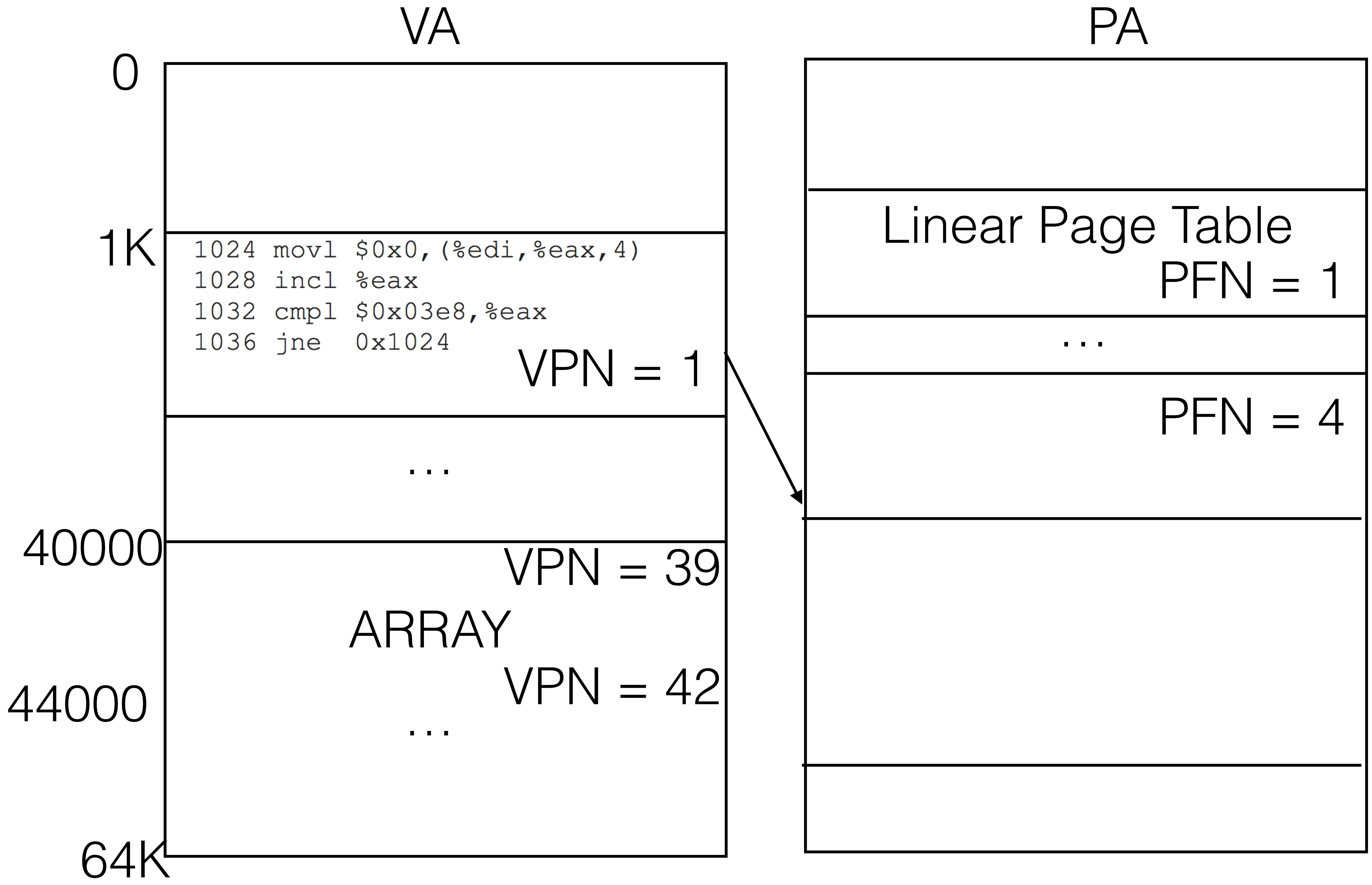
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



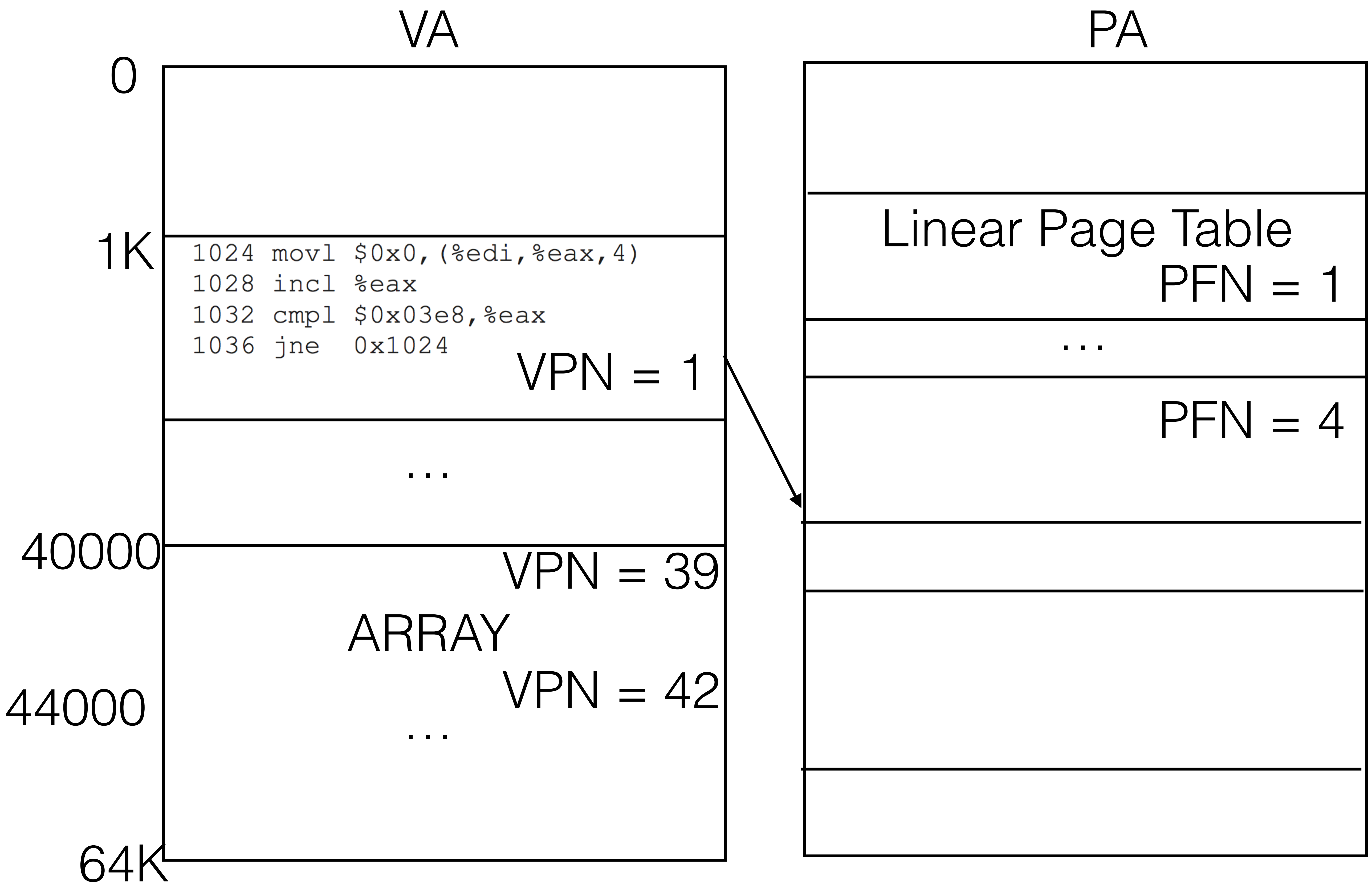
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



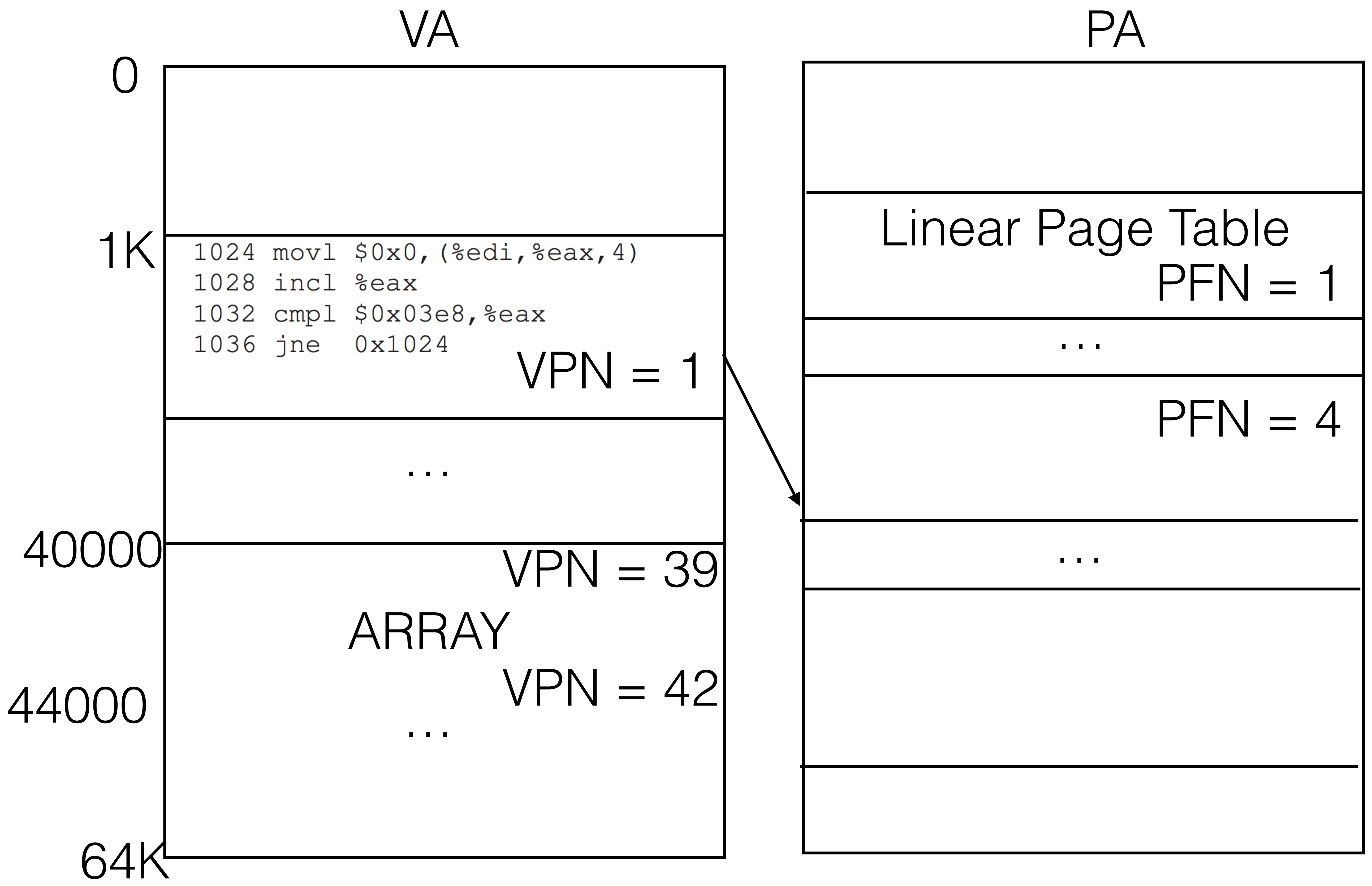
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



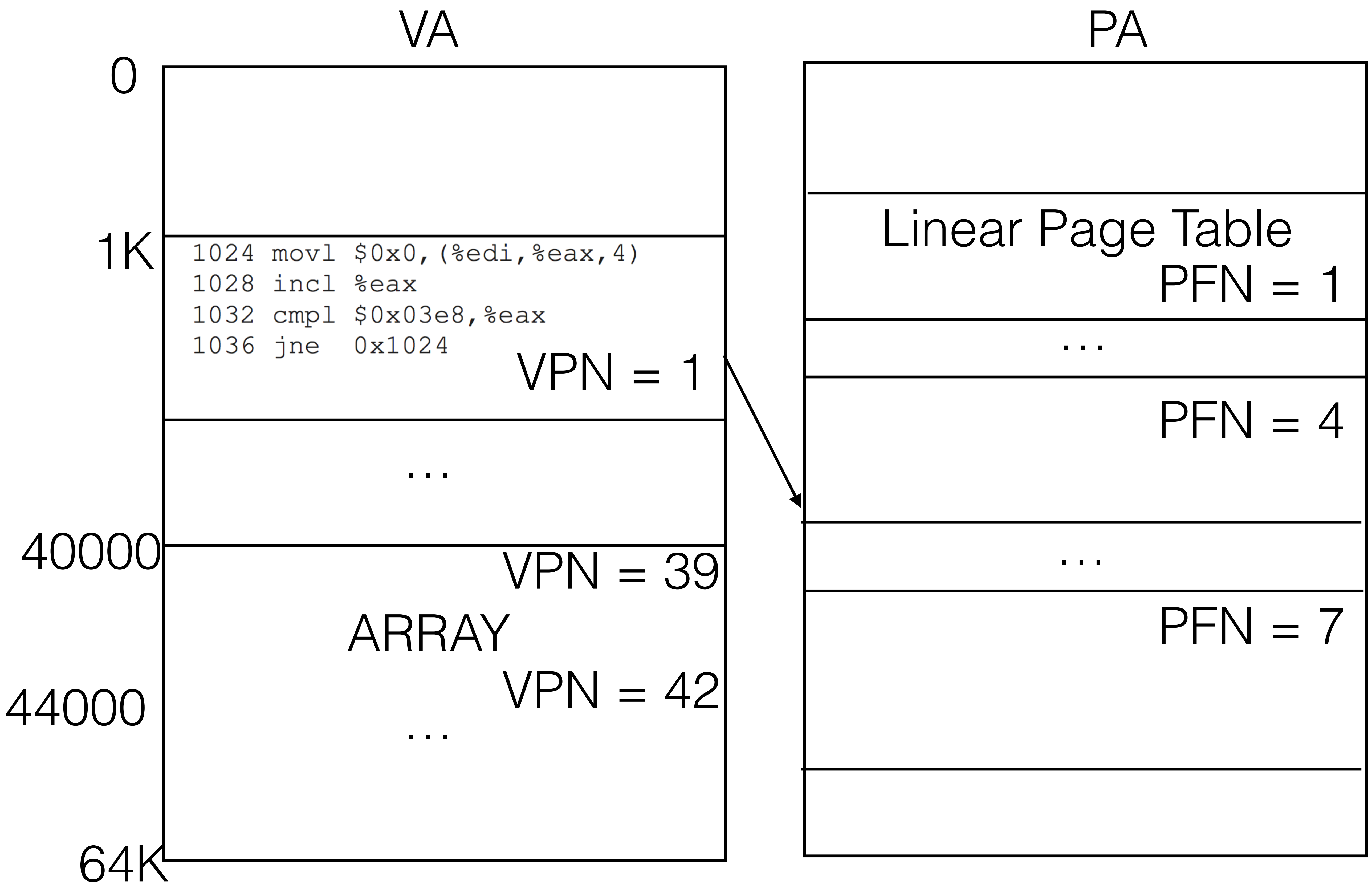
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



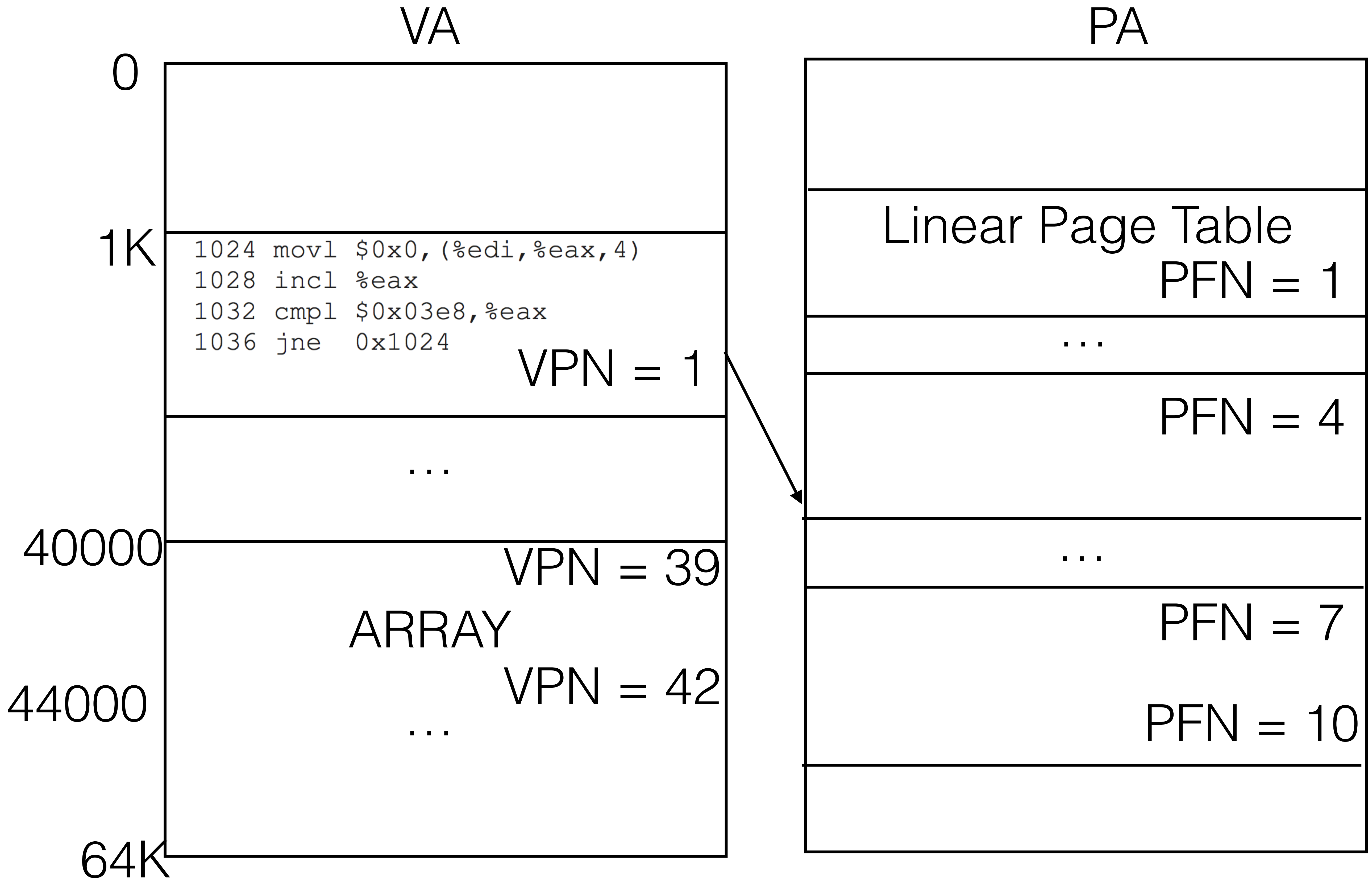
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



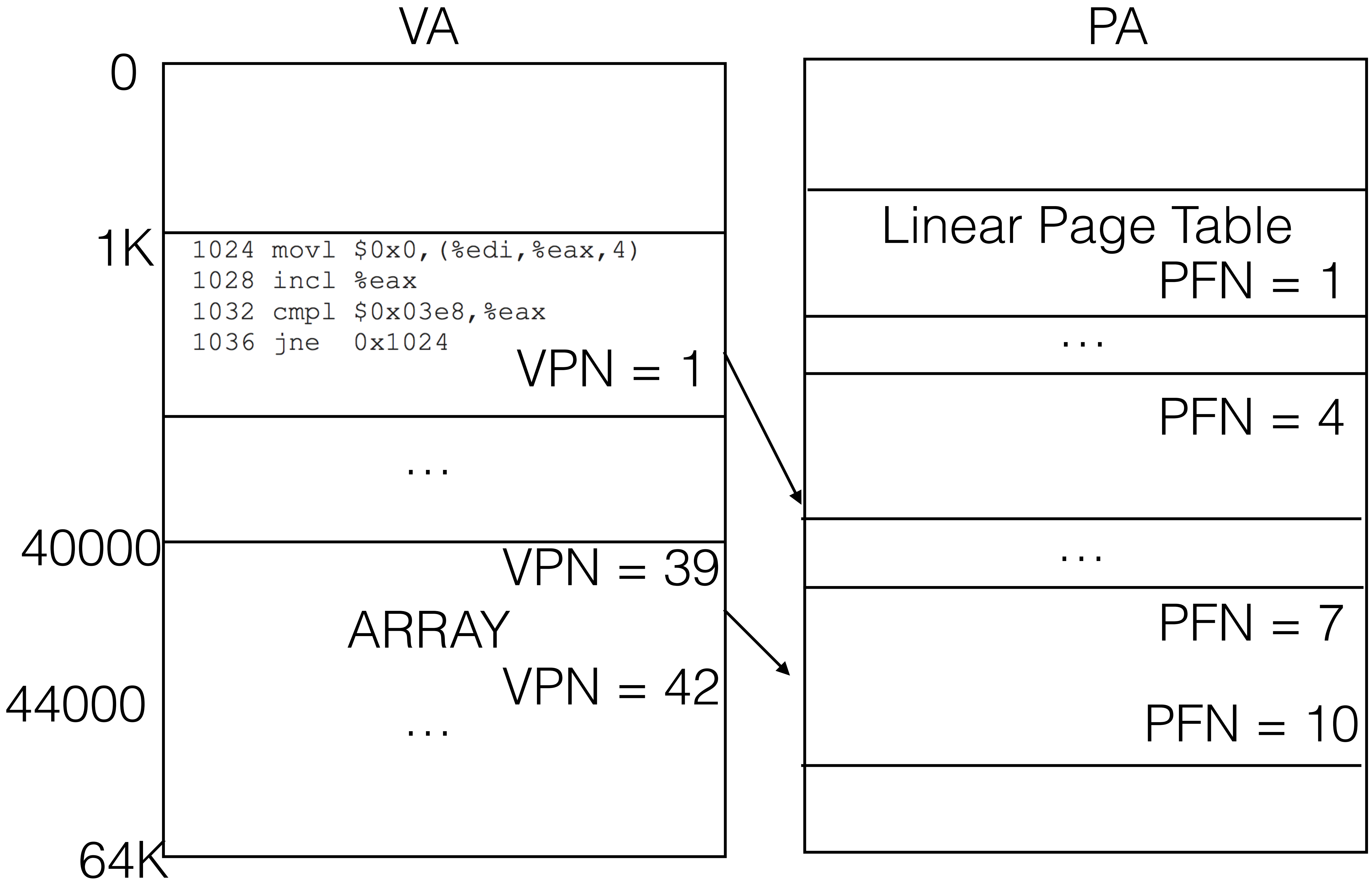
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



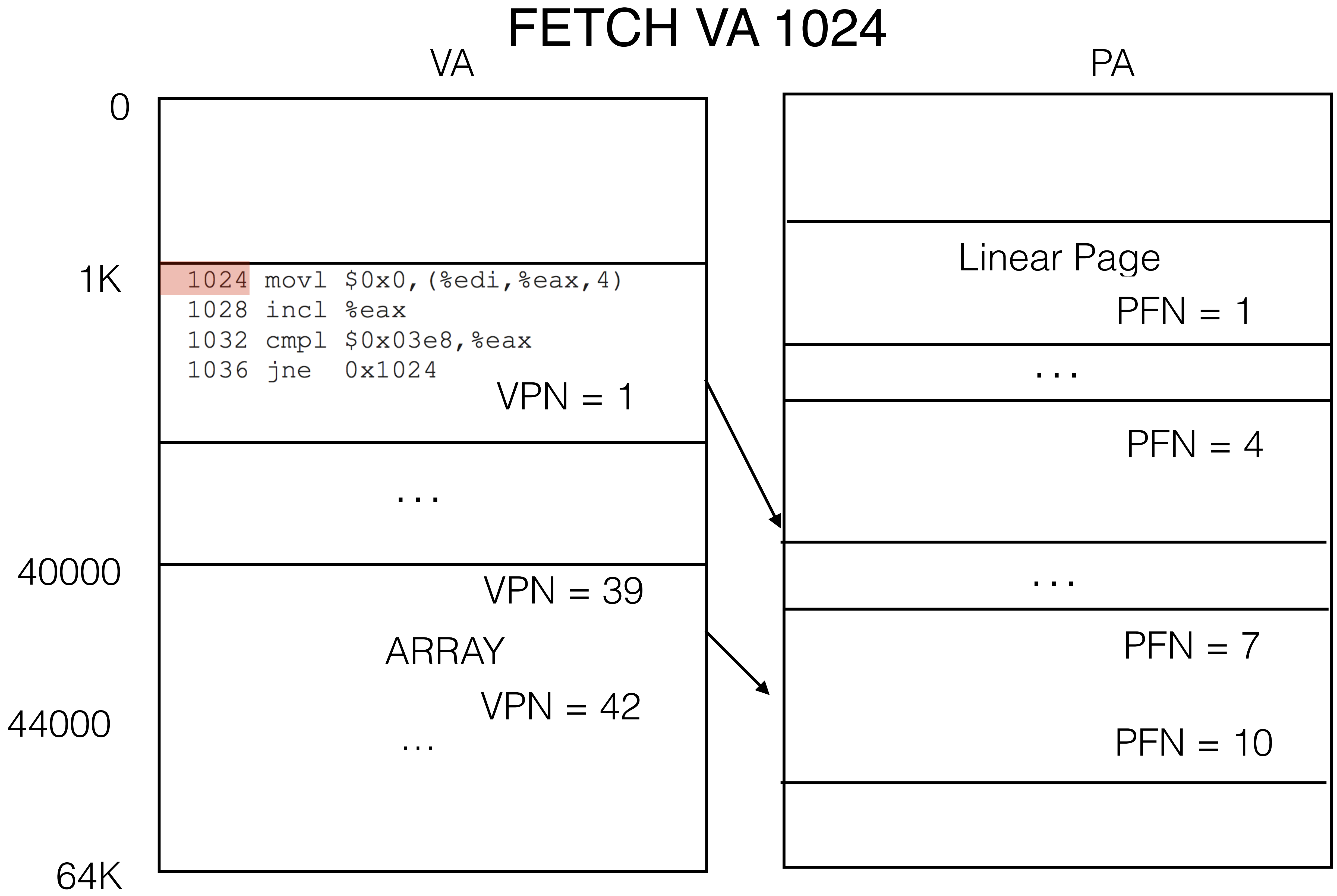
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

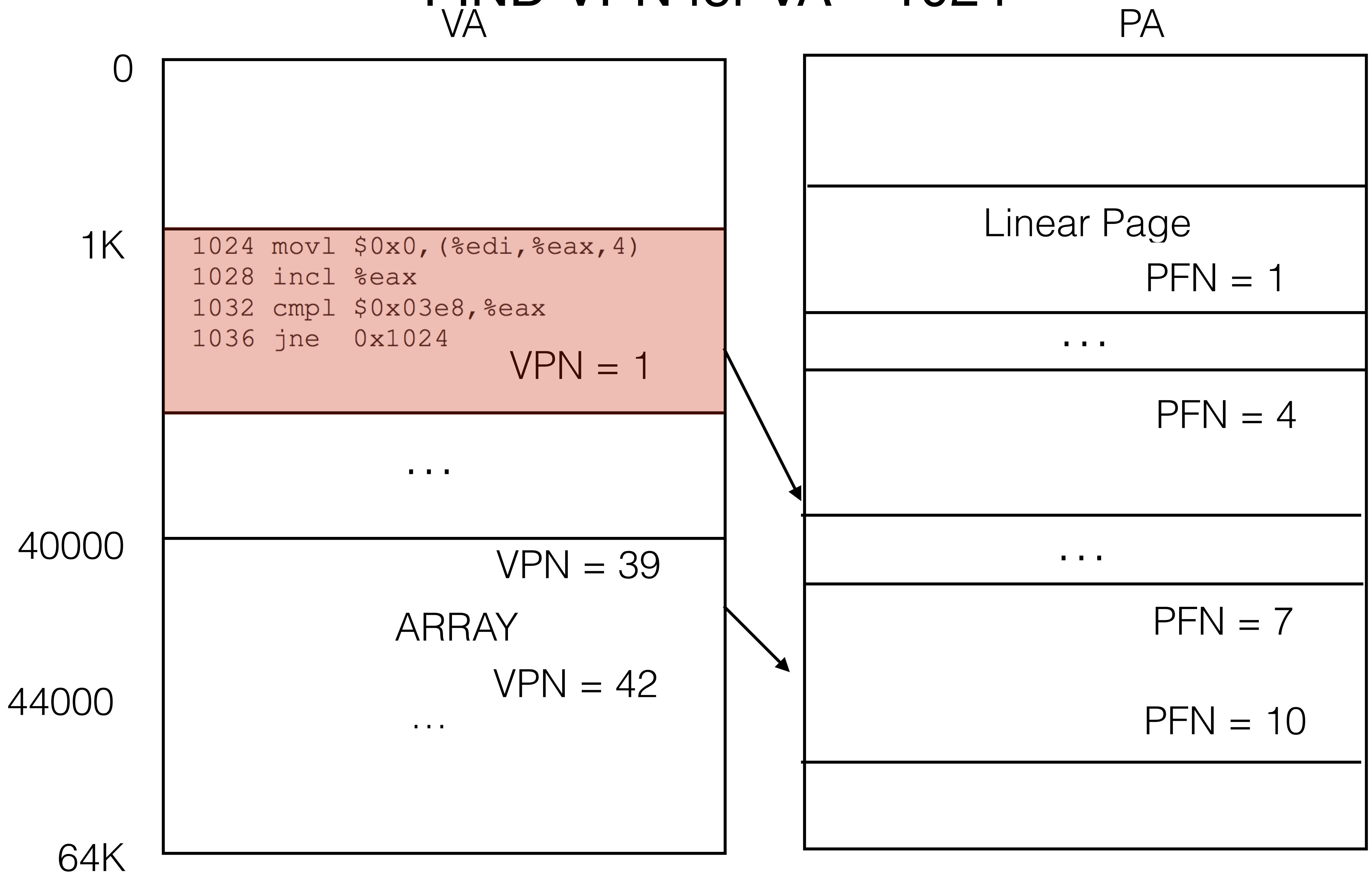
Worked Out Example



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

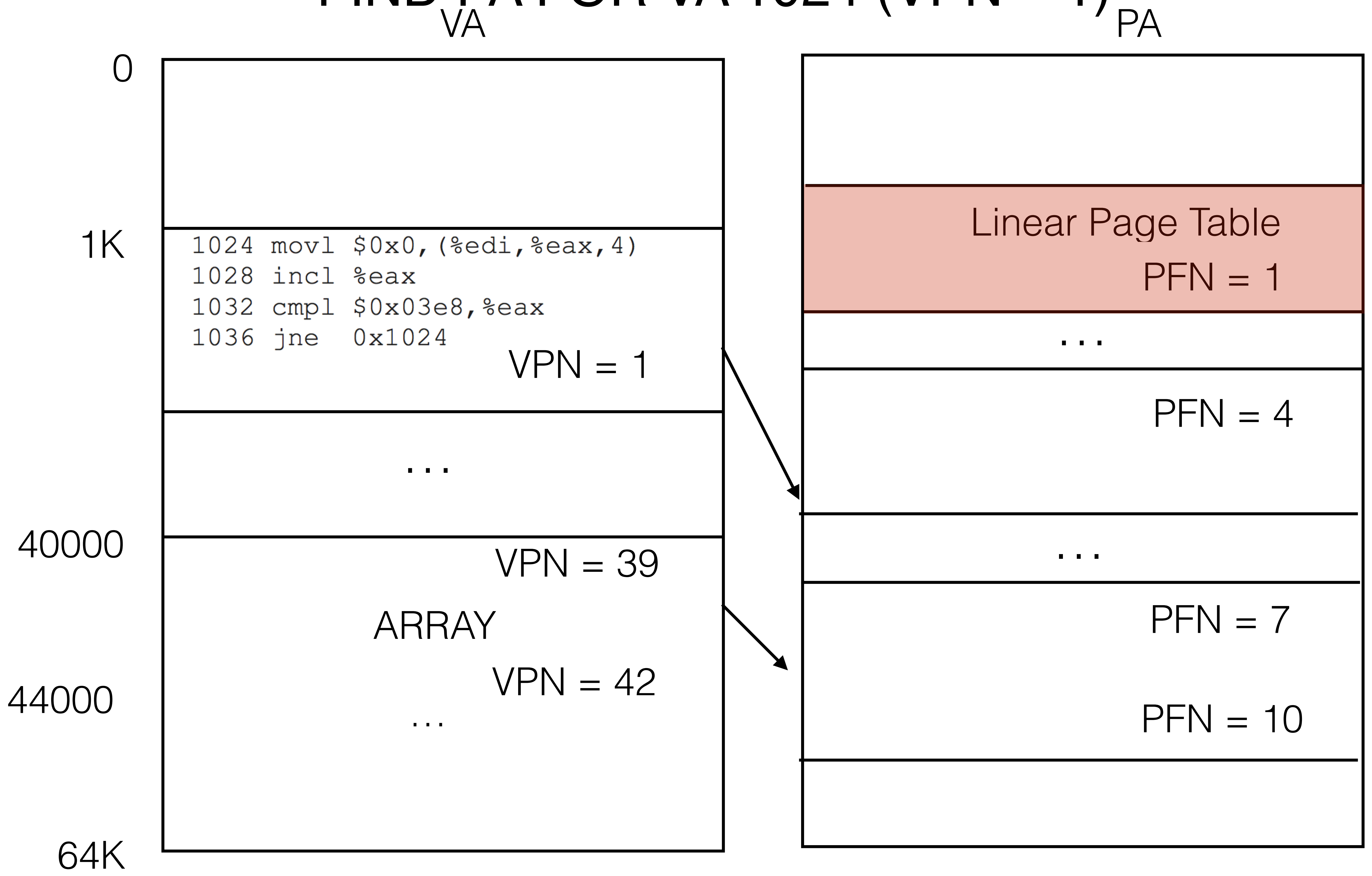
FIND VPN for VA = 1024



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

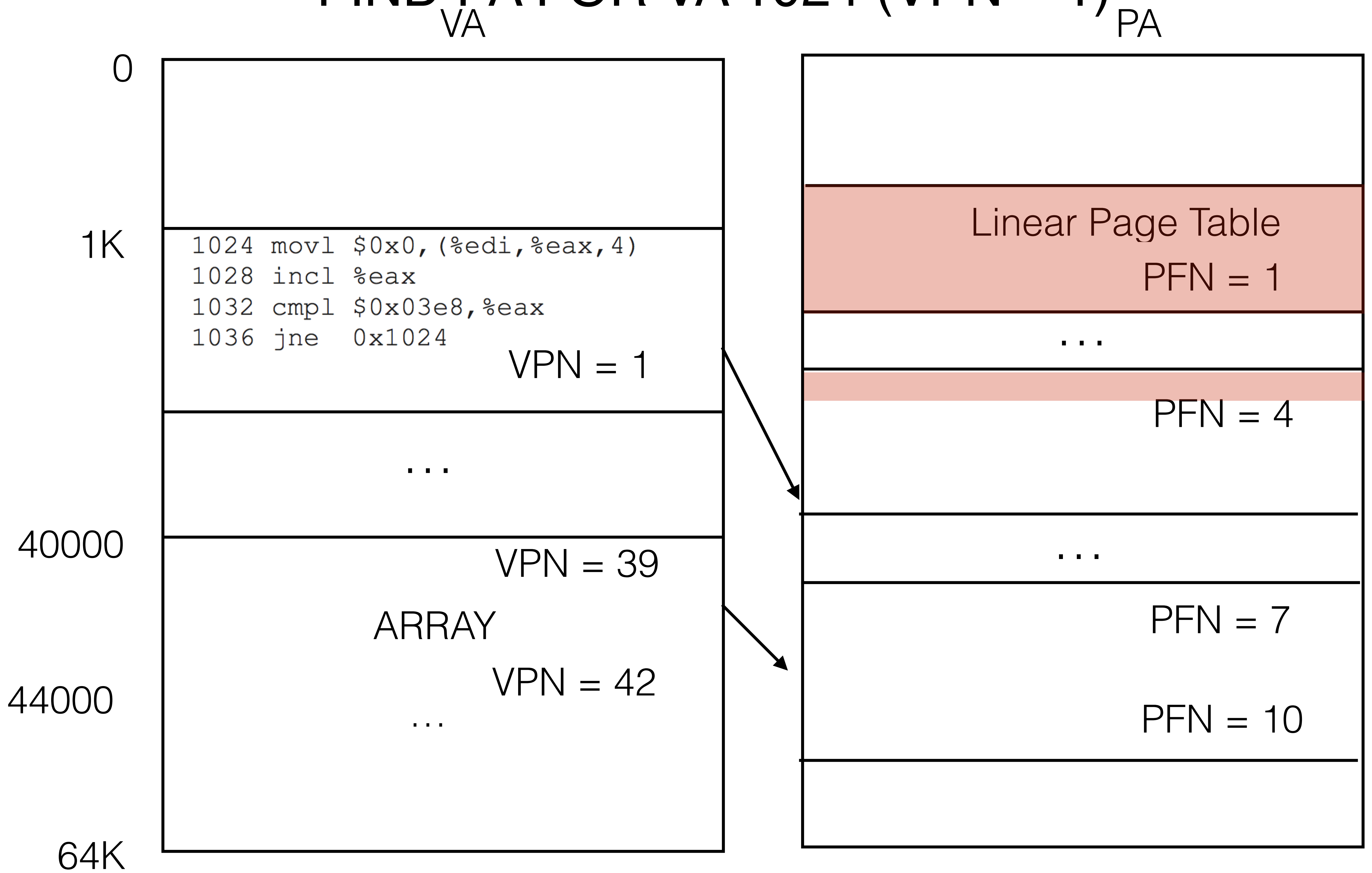
FIND PA FOR VA 1024 (VPN = 1)



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

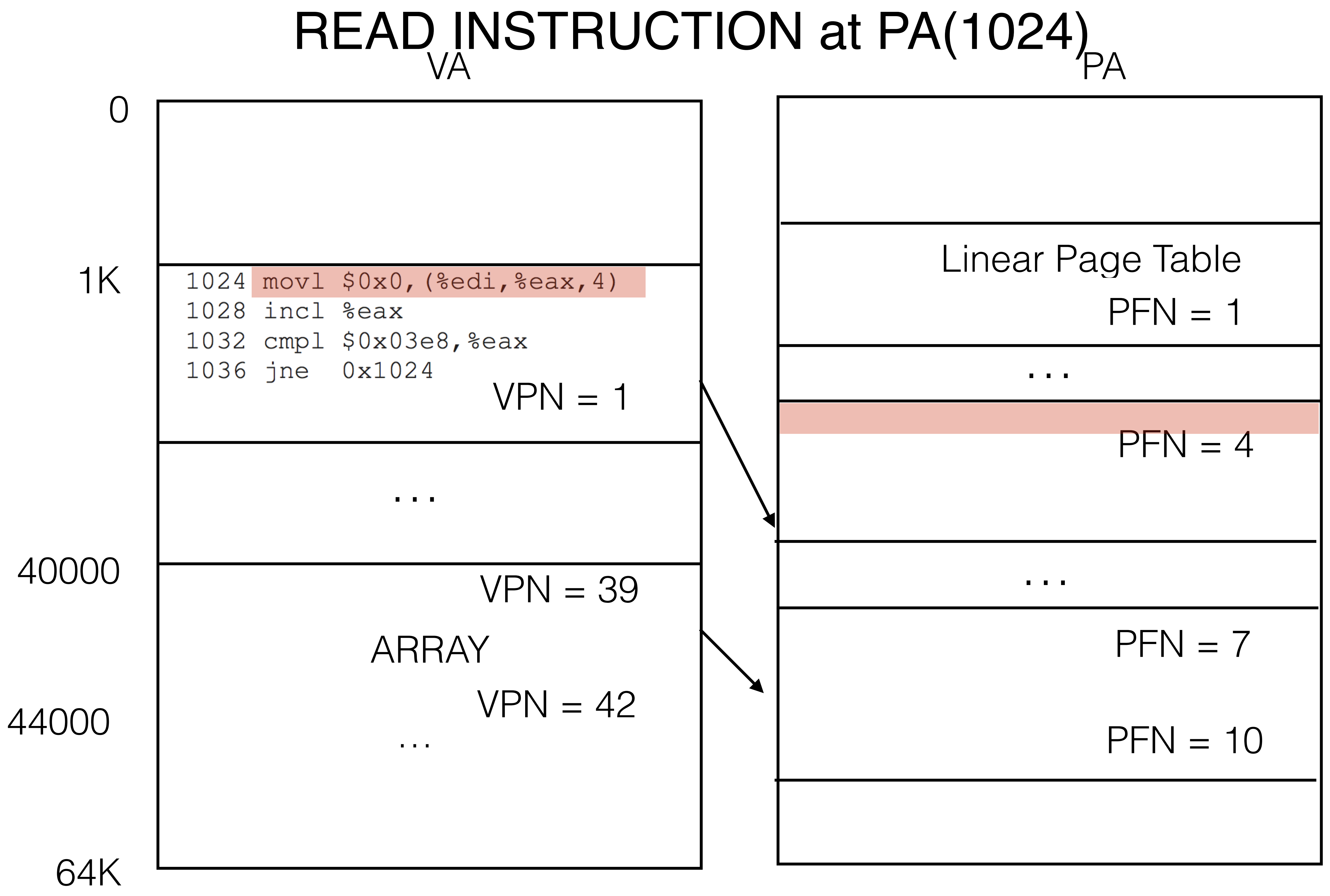
Worked Out Example

FIND PA FOR VA 1024 (VPN = 1)



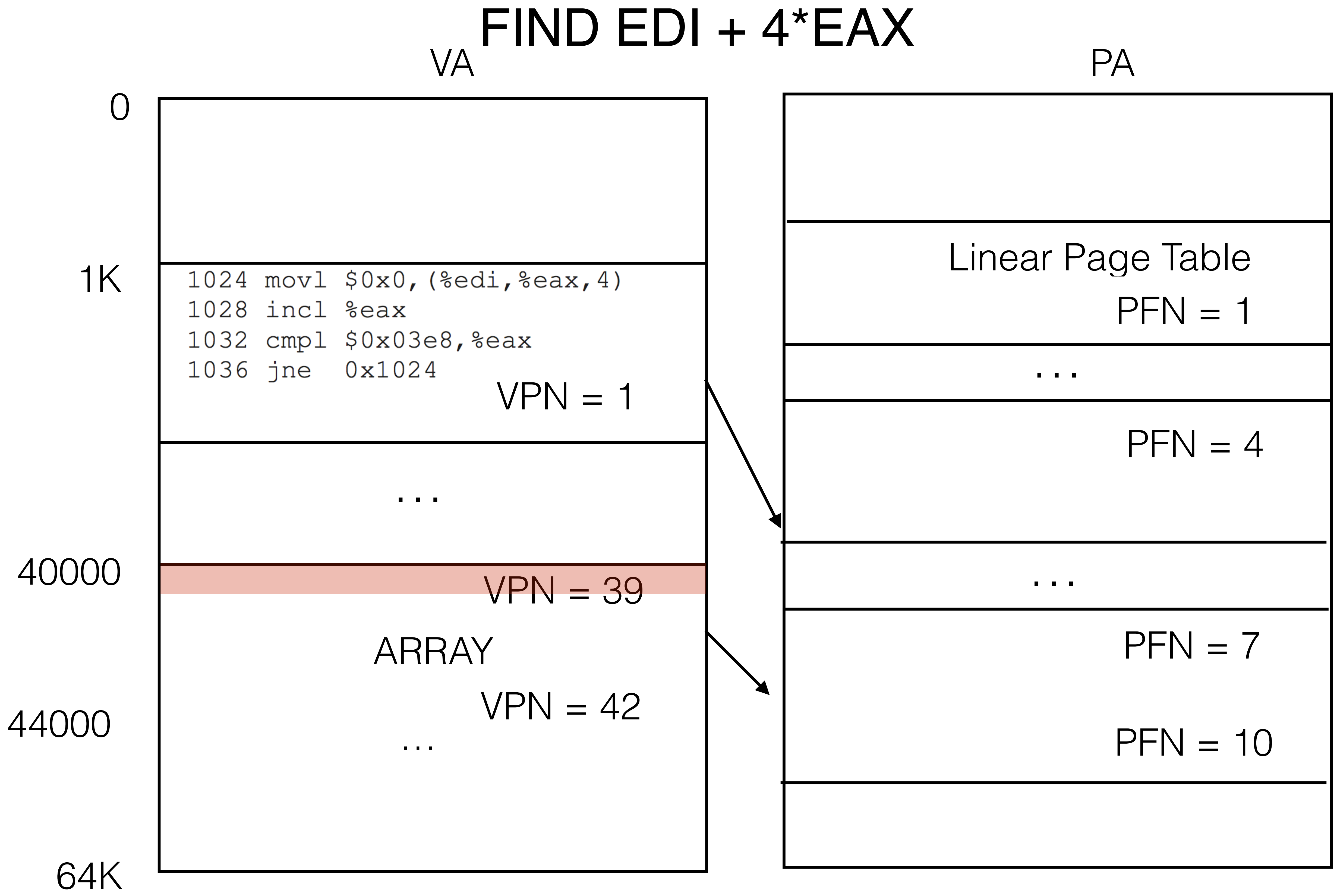
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

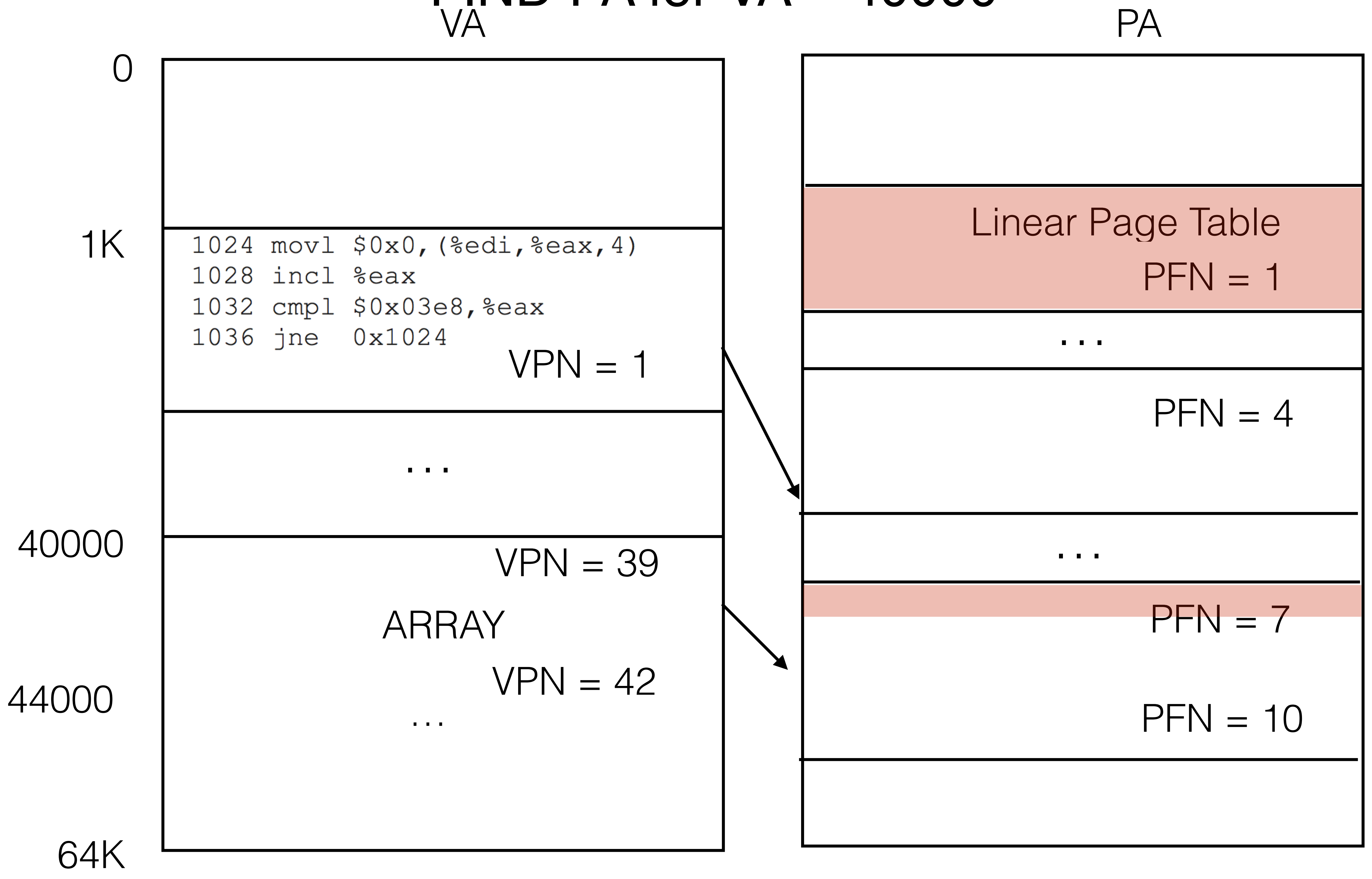
Worked Out Example



```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

Worked Out Example

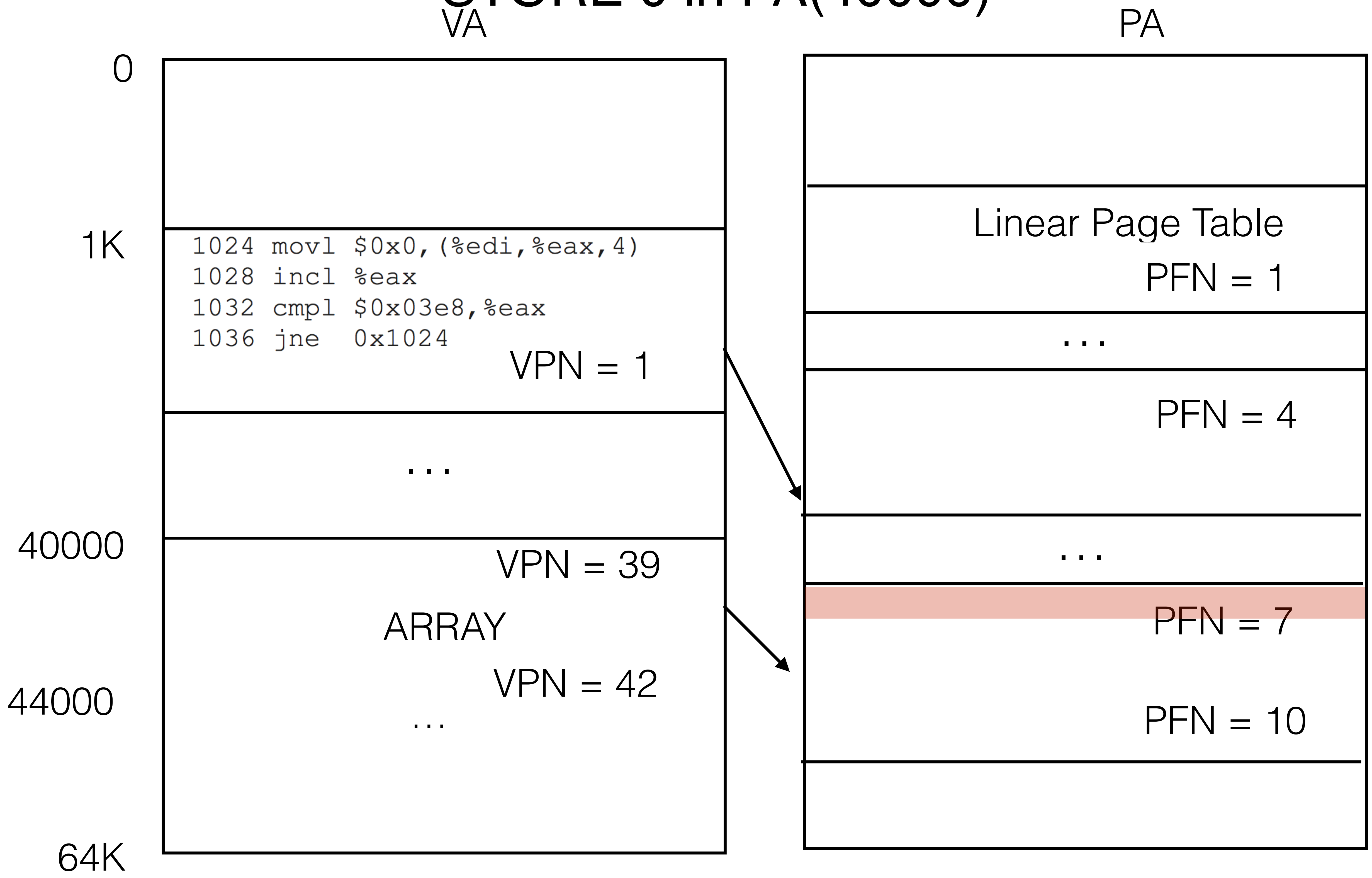
FIND PA for VA = 40000



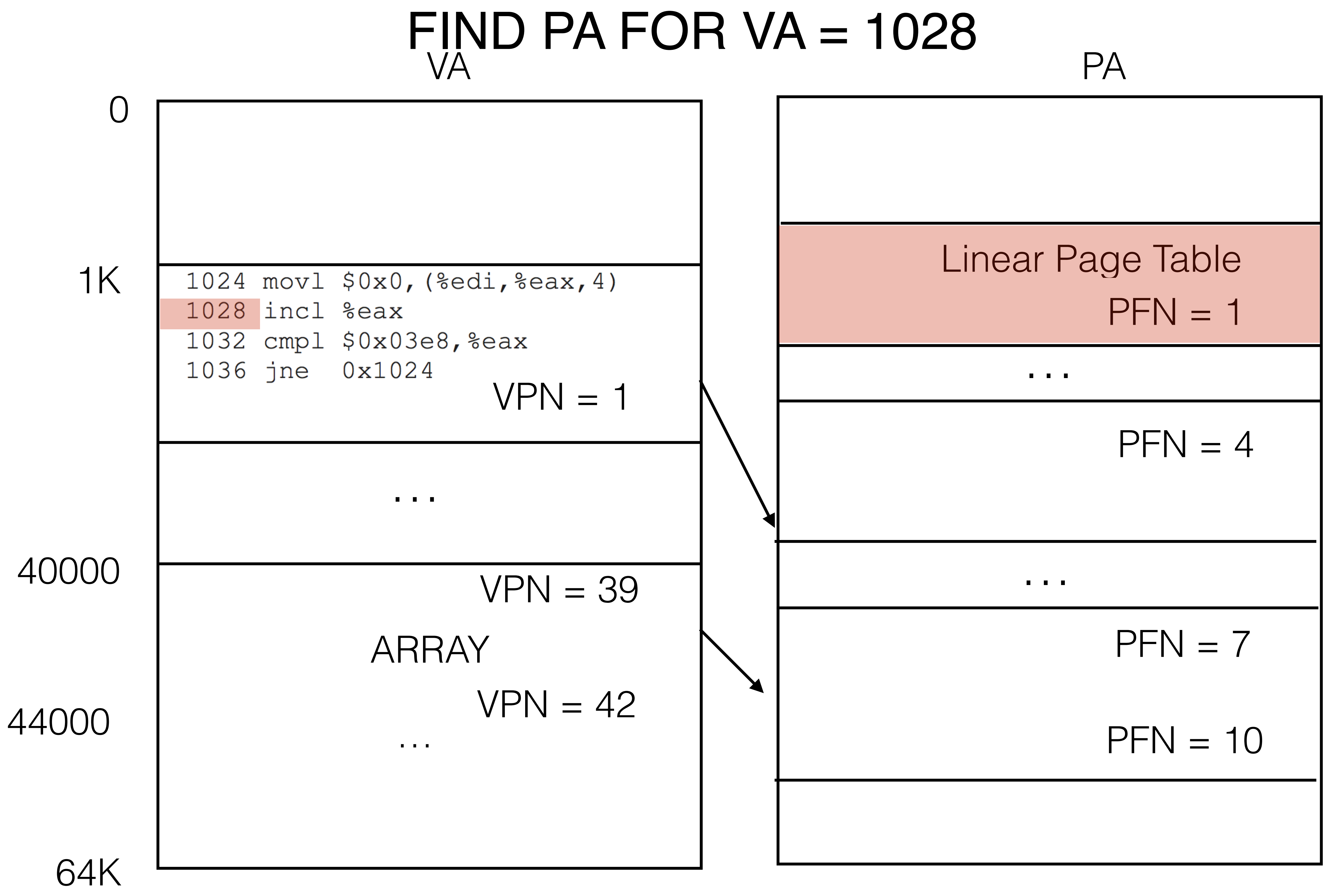
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

STORE 0 in PA(40000)



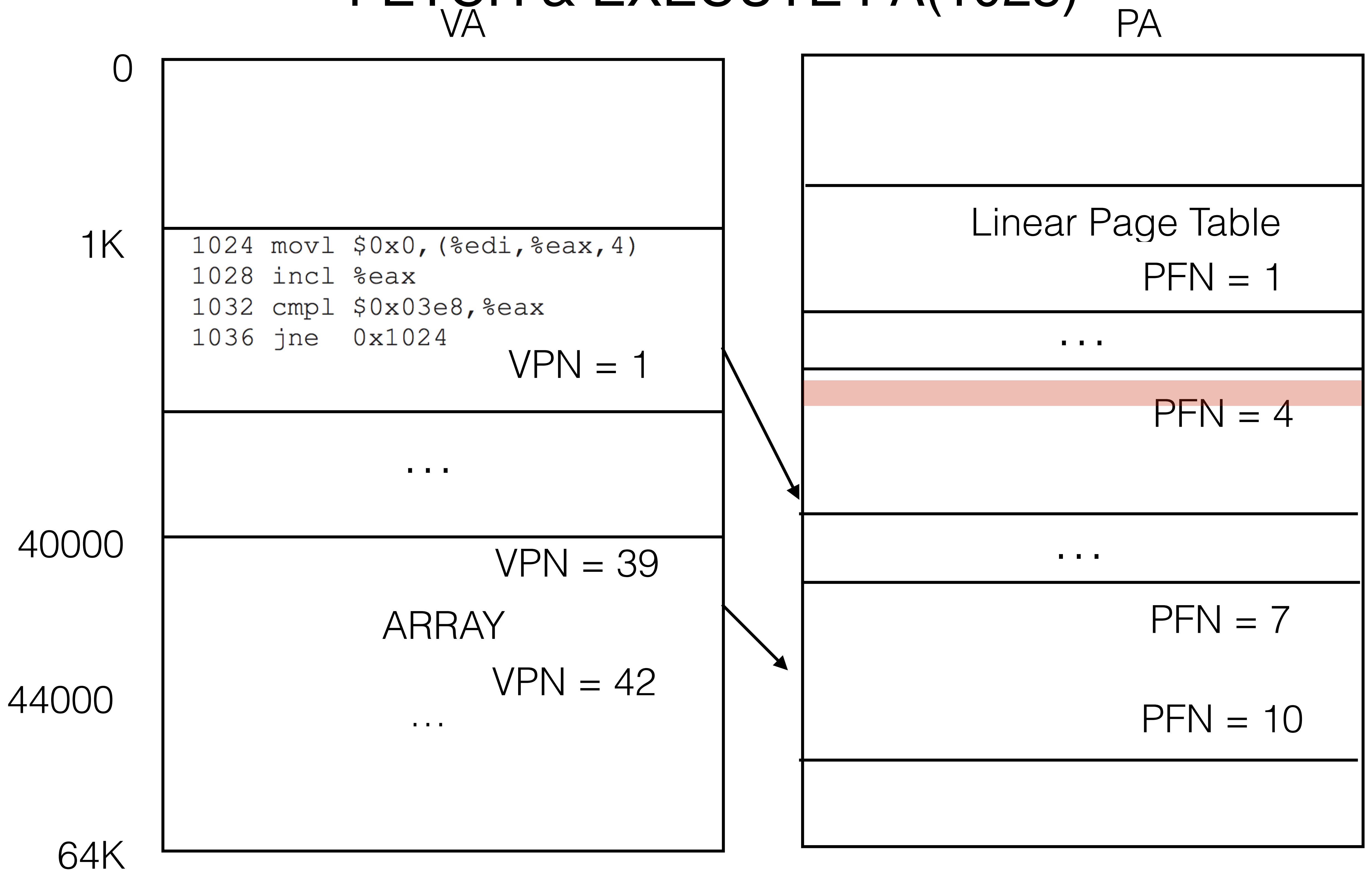
Worked Out Example



```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

Worked Out Example

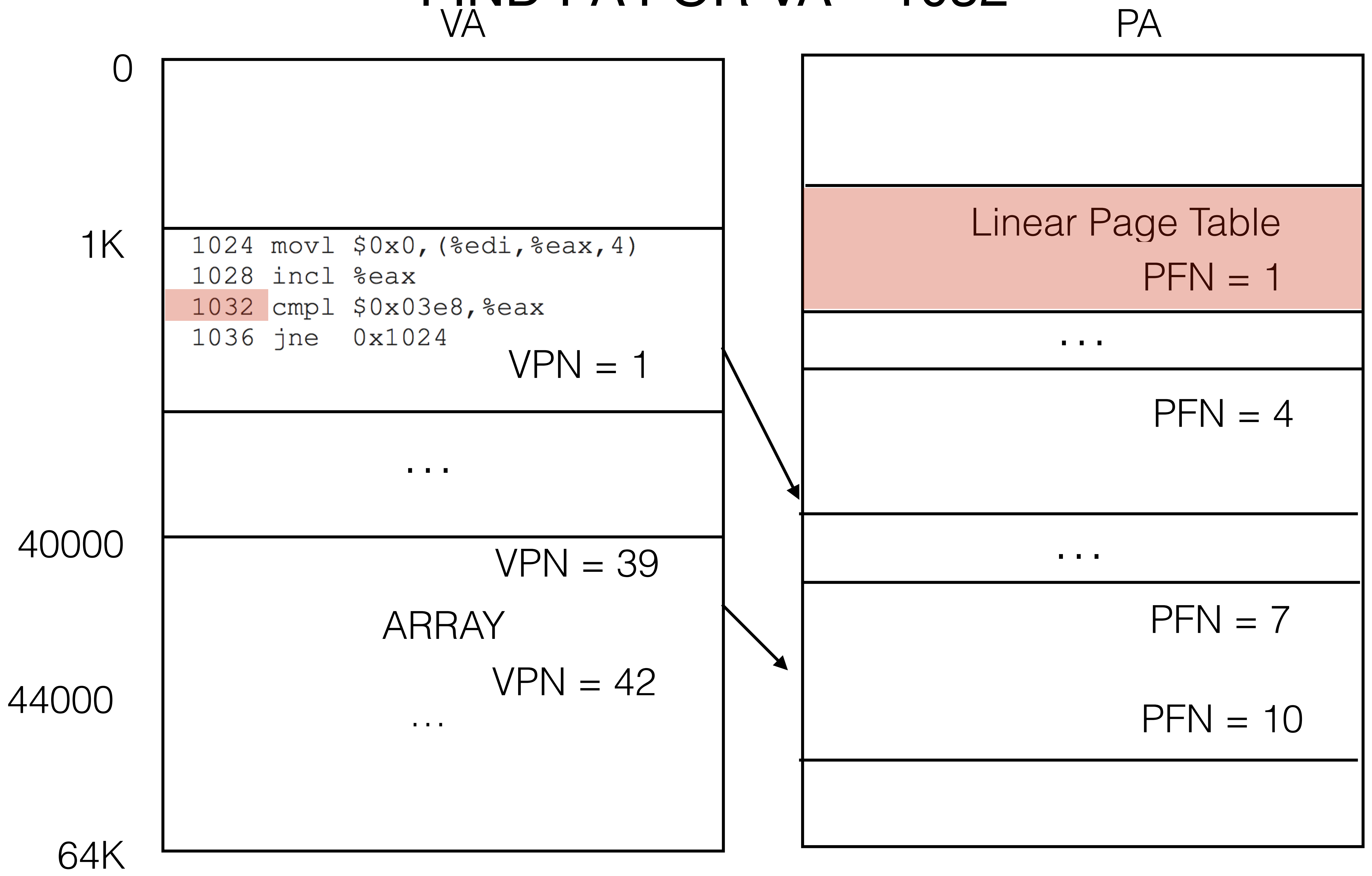
FETCH & EXECUTE PA(1028)



```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

Worked Out Example

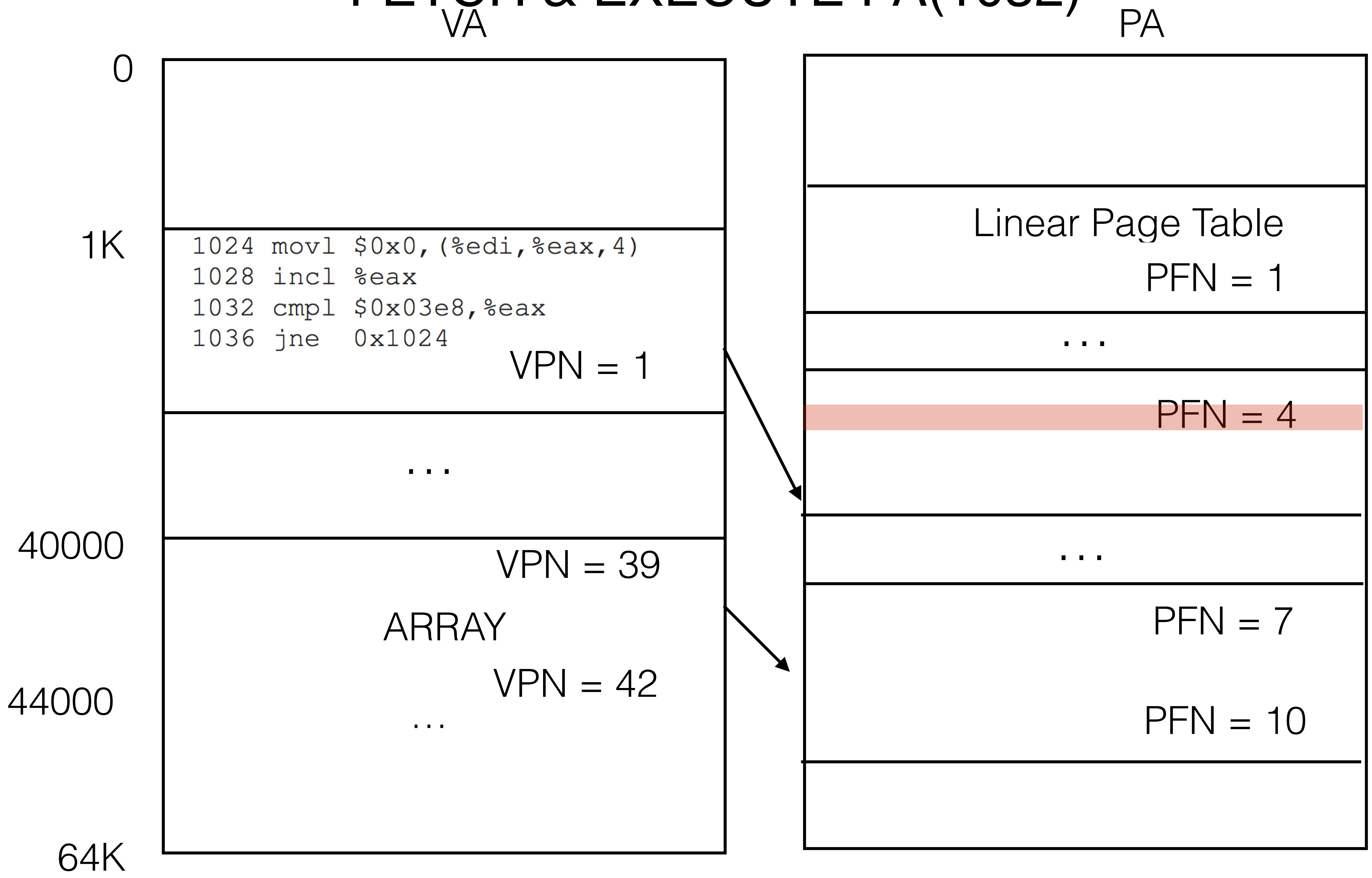
FIND PA FOR VA = 1032



```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

Worked Out Example

FETCH & EXECUTE PA(1032)



```
1024 movl  $0x0,(%edi,%eax,4)
1028 incl  %eax
1032 cmpl  $0x03e8,%eax
1036 jne   0x1024
```

Example Summary

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)
3. Read PTE from memory

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)
3. Read PTE from memory
4. Extract PFN (page frame num)

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)
3. Read PTE from memory
4. Extract PFN (page frame num)
5. Build PA (phys addr)

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)
3. Read PTE from memory
4. Extract PFN (page frame num)
5. Build PA (phys addr)
6. Read contents of PA from memory into register

Example Summary

1. Extract VPN (virt page num) from VA (virt addr)
2. Calculate addr of PTE (page table entry)
3. Read PTE from memory
4. Extract PFN (page frame num)
5. Build PA (phys addr)
6. Read contents of PA from memory into register

SLOW!

Caching Makes Sense!

Factorial with and without memoization

Caching - Translation Lookaside Buffer (TLB)

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
    - Extract the PFN from TLB (PFN = TLB[VPN])

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
    - Extract the PFN from TLB (PFN = TLB[VPN])
    - Generate PA from PFN (Add offset)

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
    - Extract the PFN from TLB (PFN = TLB[VPN])
    - Generate PA from PFN (Add offset)
    - Access memory assuming protection checks work

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
    - Extract the PFN from TLB (PFN = TLB[VPN])
    - Generate PA from PFN (Add offset)
    - Access memory assuming protection checks work
4. If not found, it is a TLB Miss. :(

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
   - Extract the PFN from TLB (PFN = TLB[VPN])
   - Generate PA from PFN (Add offset)
   - Access memory assuming protection checks work
4. If not found, it is a TLB Miss. :(
   - Access Page table to find the translation

Caching - Translation Lookaside Buffer (TLB)

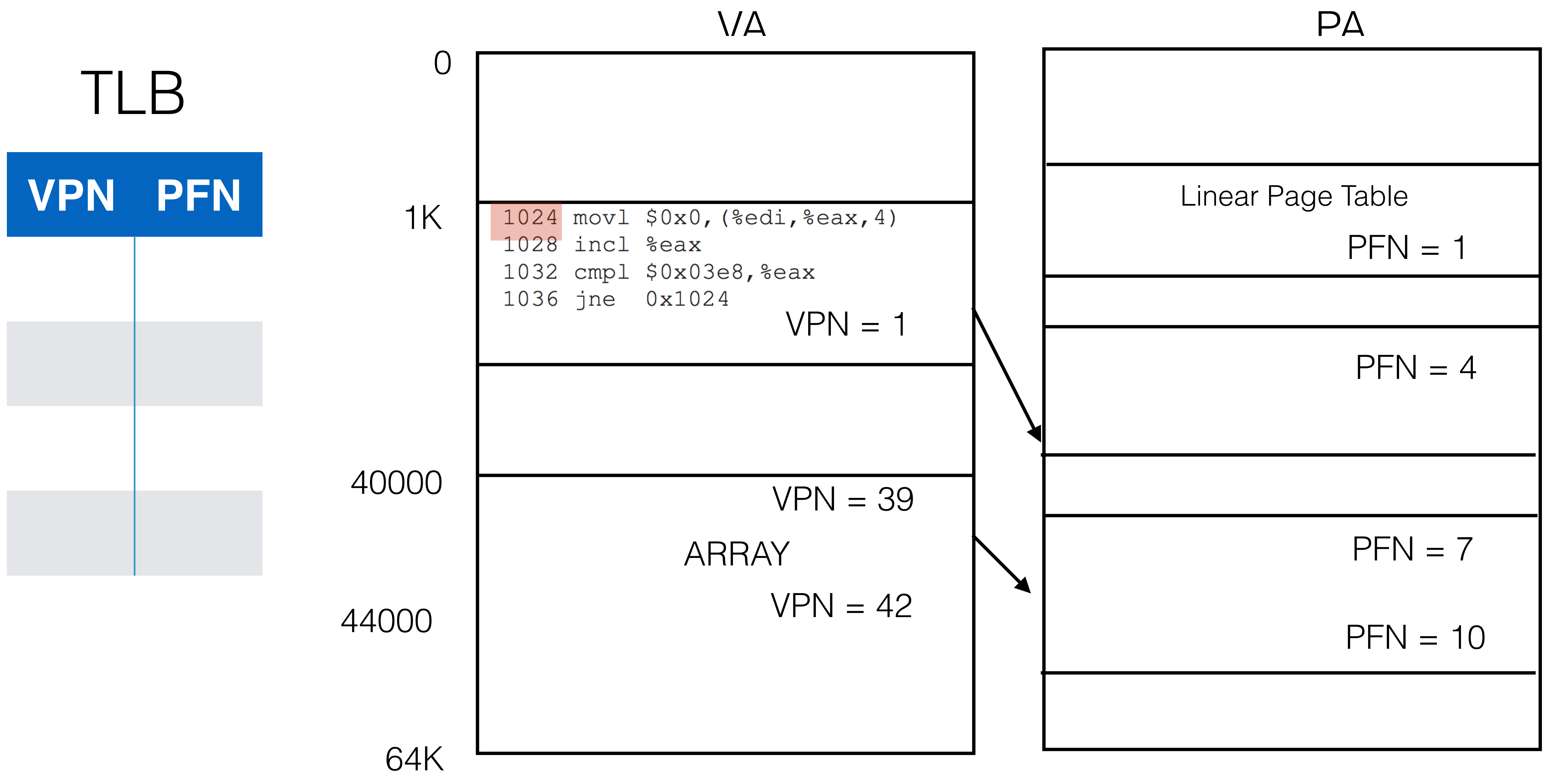
1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
   - Extract the PFN from TLB (PFN = TLB[VPN])
   - Generate PA from PFN (Add offset)
   - Access memory assuming protection checks work
4. If not found, it is a TLB Miss. :(
   - Access Page table to find the translation
   - Add translation to TLB (TLB[VPN] = PFN)

Caching - Translation Lookaside Buffer (TLB)

1. Get the VPN from VA
2. Check if TLB has VA
3. If found, it is a TLB Hit. Yay!
    - Extract the PFN from TLB (PFN = TLB[VPN])
    - Generate PA from PFN (Add offset)
    - Access memory assuming protection checks work
4. If not found, it is a TLB Miss. :(
    - Access Page table to find the translation
    - Add translation to TLB (TLB[VPN] = PFN)
    - Goto Step 2

Worked Out Example

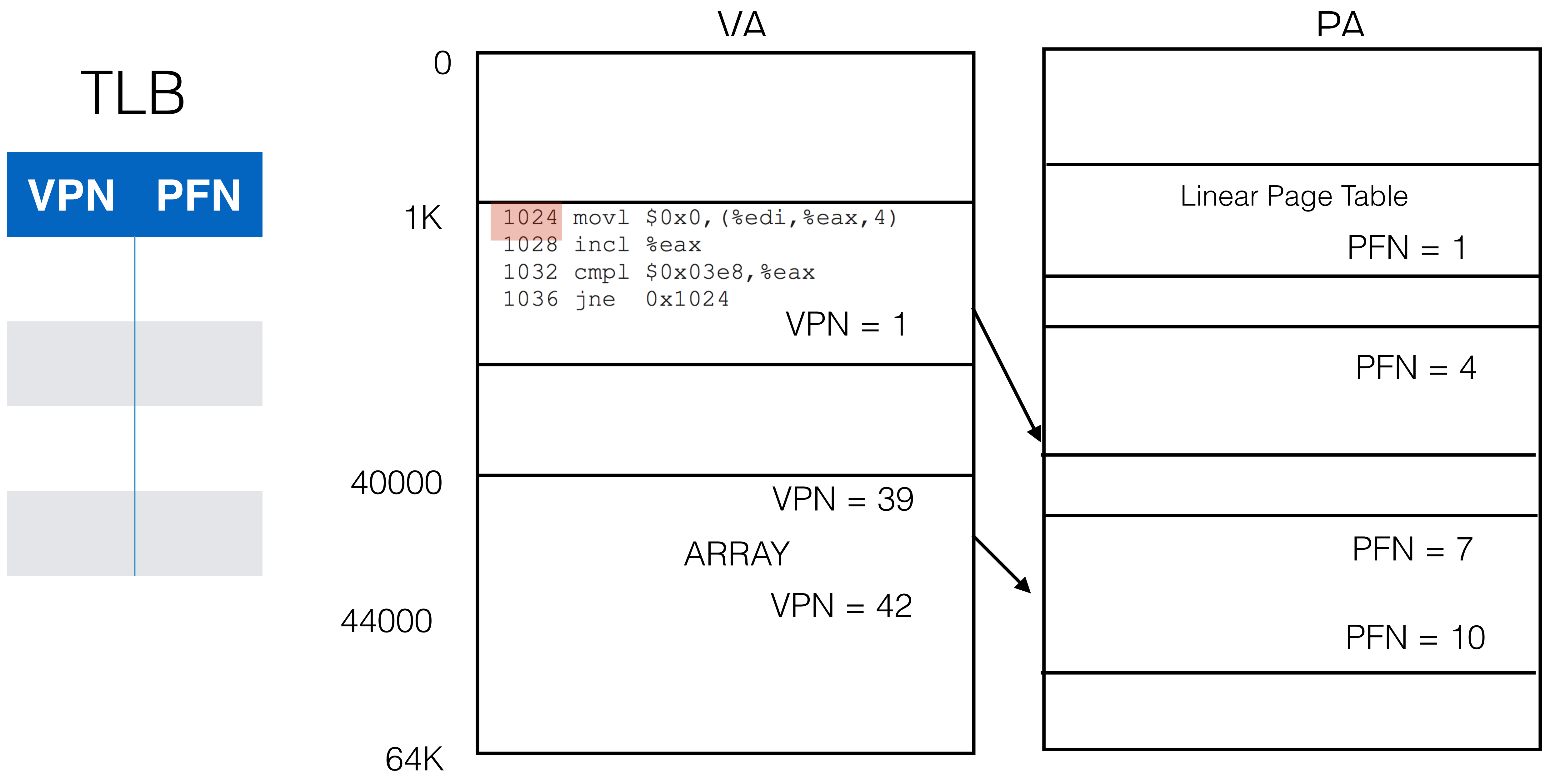
FETCH VA 1024



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

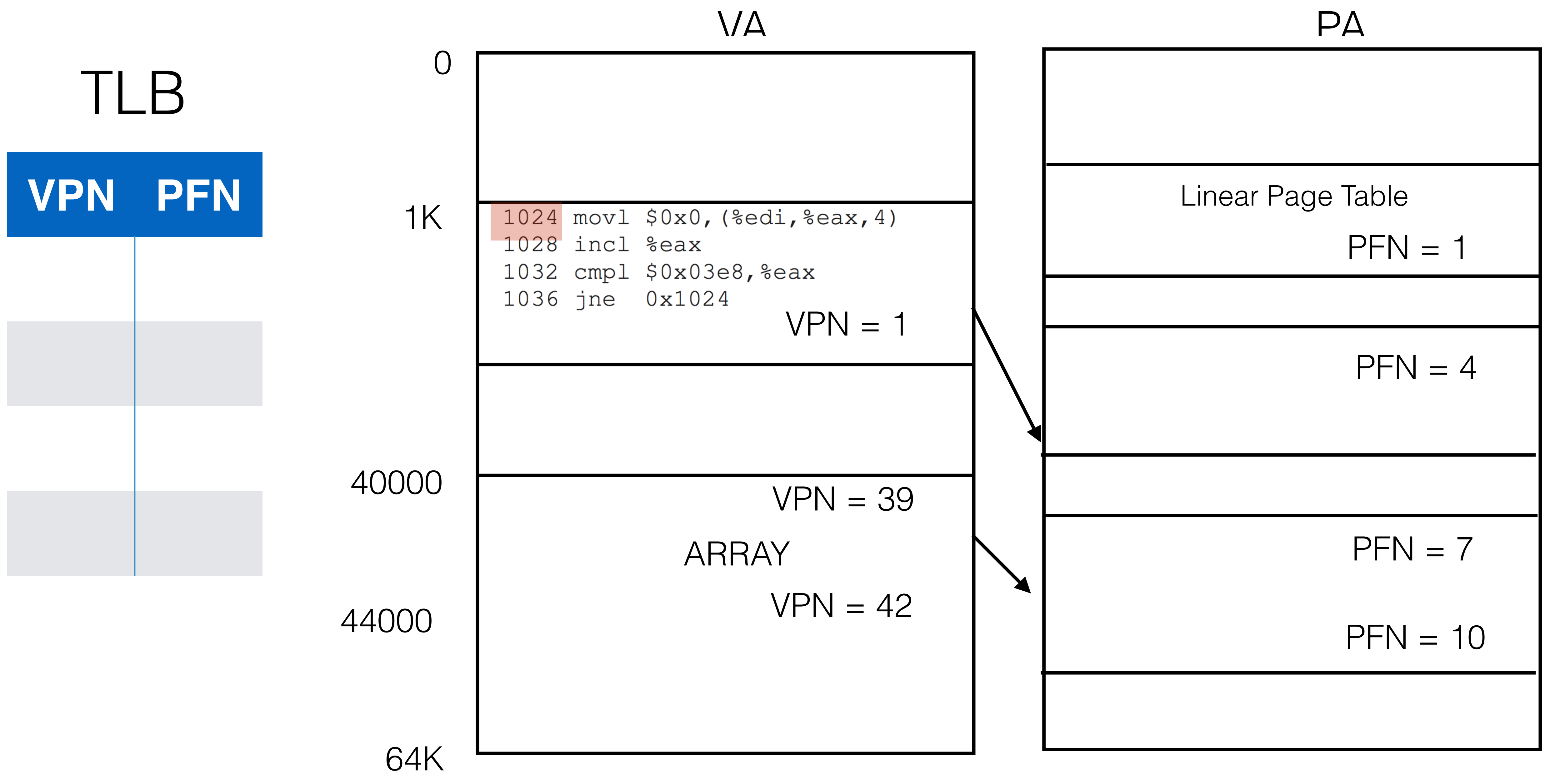
Get VPN for VA 1024. VPN = 1



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

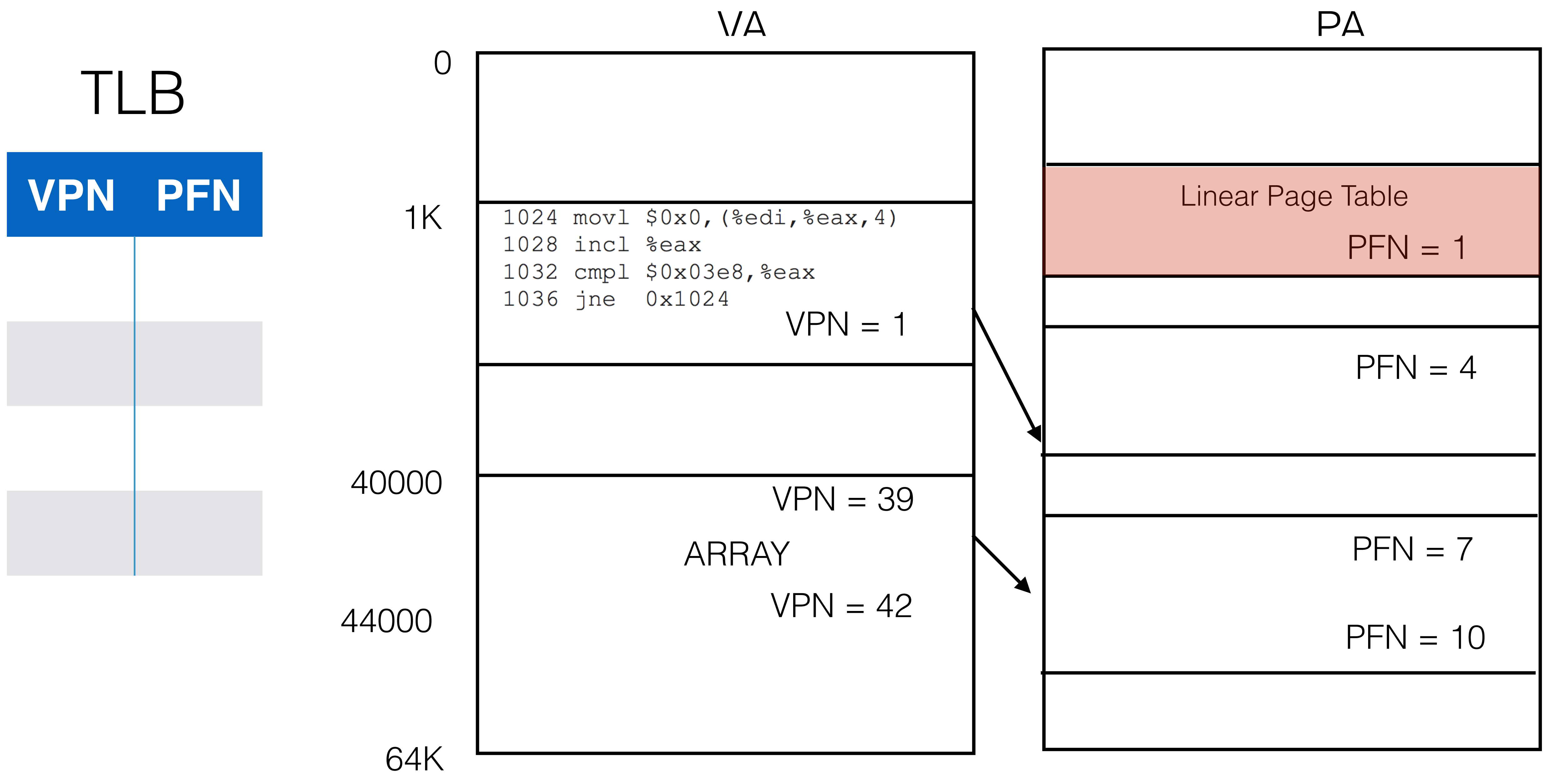
LOOK IN TLB for VPN = 1. Not found. TLB Miss!



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Find PFN for VPN = 1 by accessing Page Table



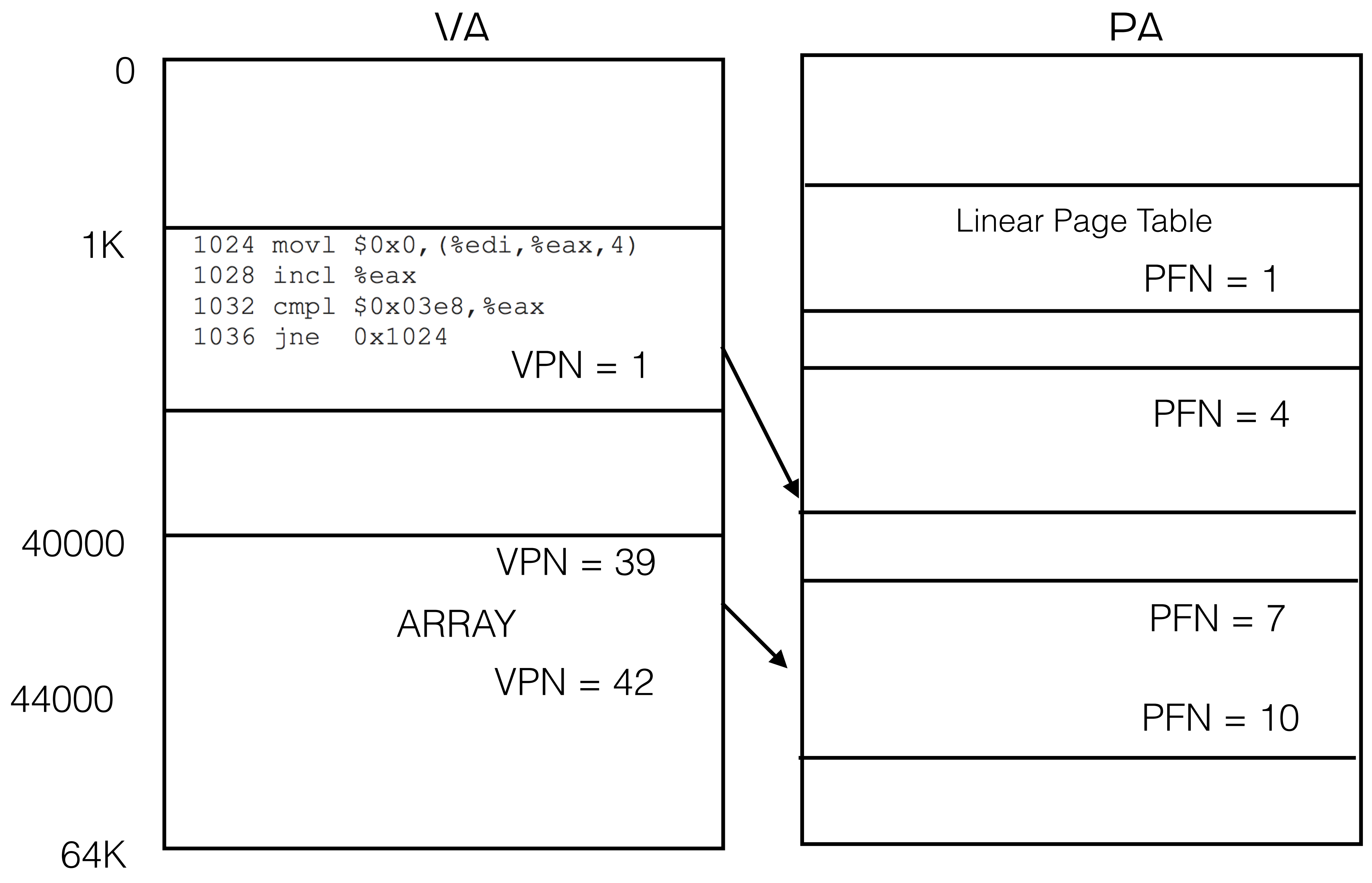
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Add entry to TLB

TLB

VPN	PFN
1	4



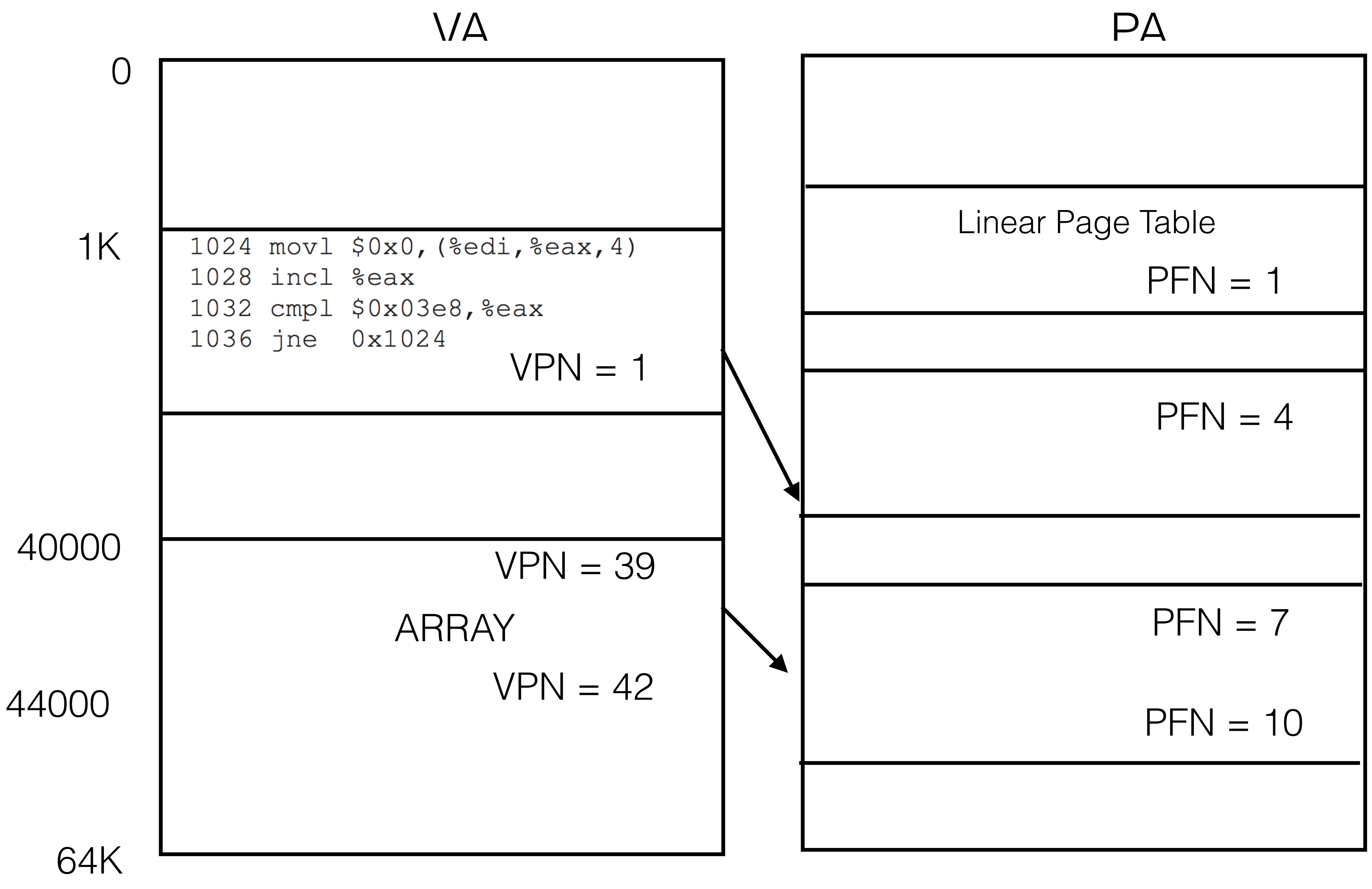
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Search for translation of VPN = 1 on TLB

TLB

VPN	PFN
1	4



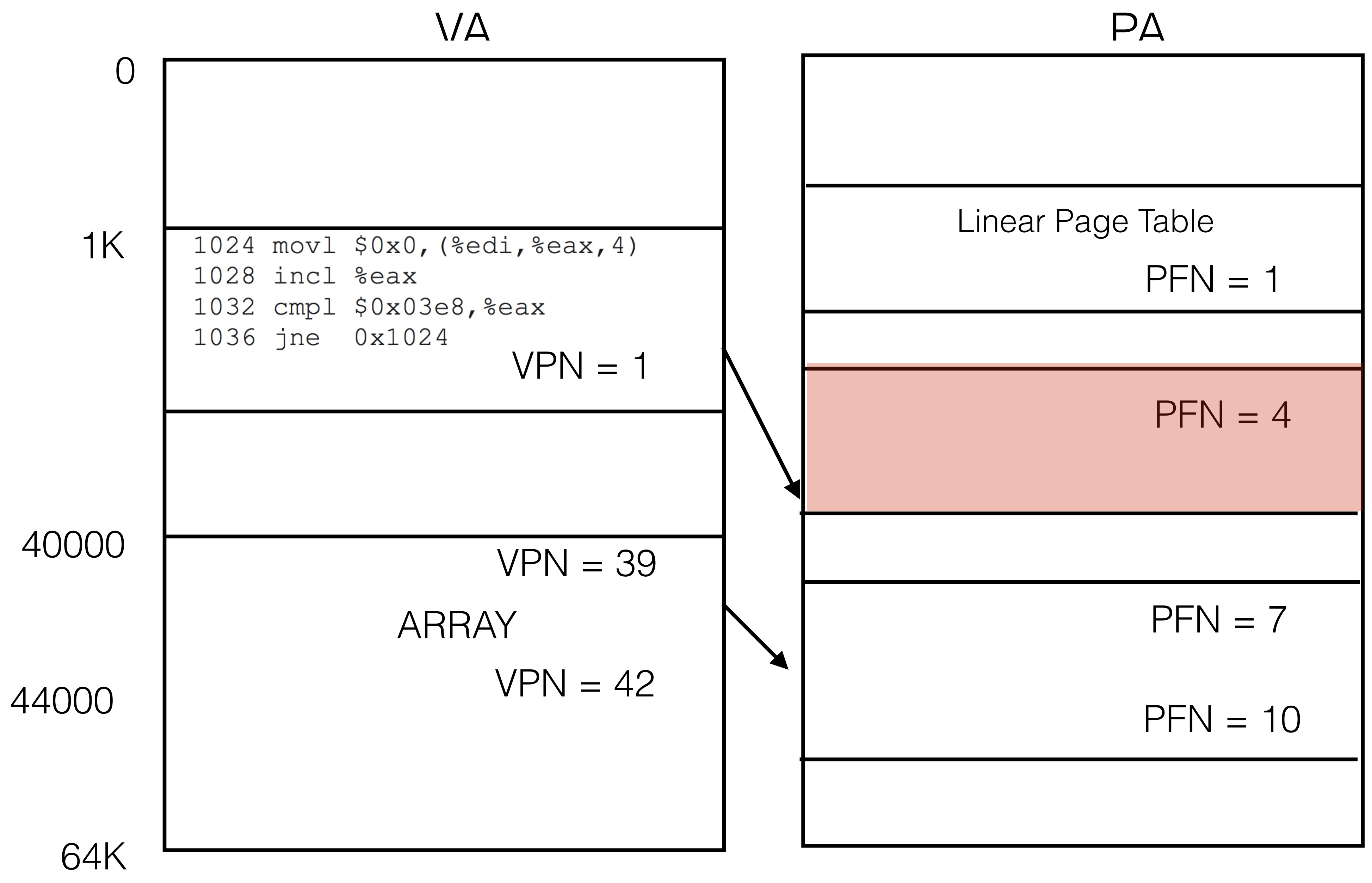
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Goto PFN 4 and create PA by adding offset

TLB

VPN	PFN
1	4



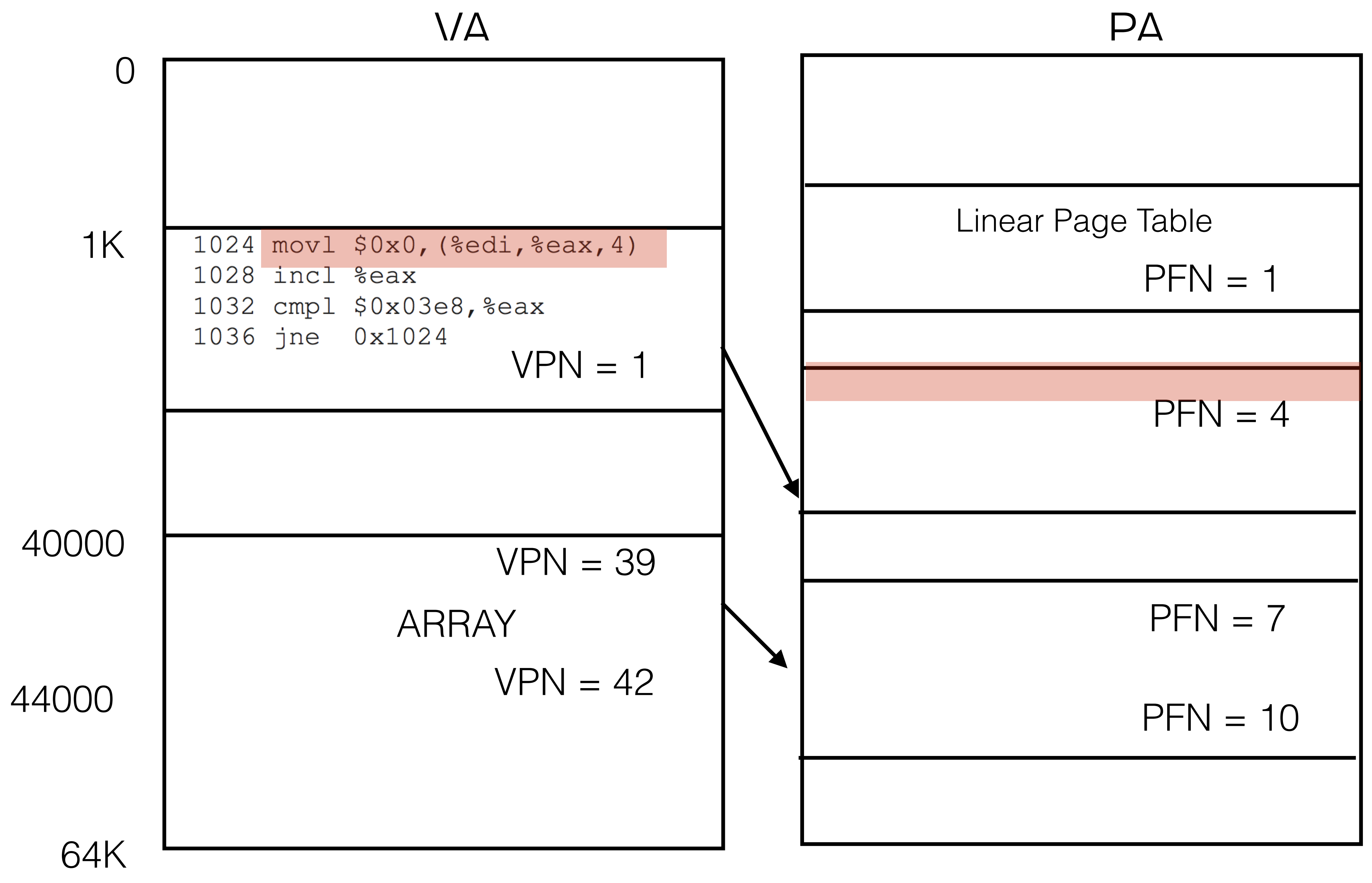
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

READ INSTRUCTION at PA(1024)

TLB

VPN	PFN
1	4



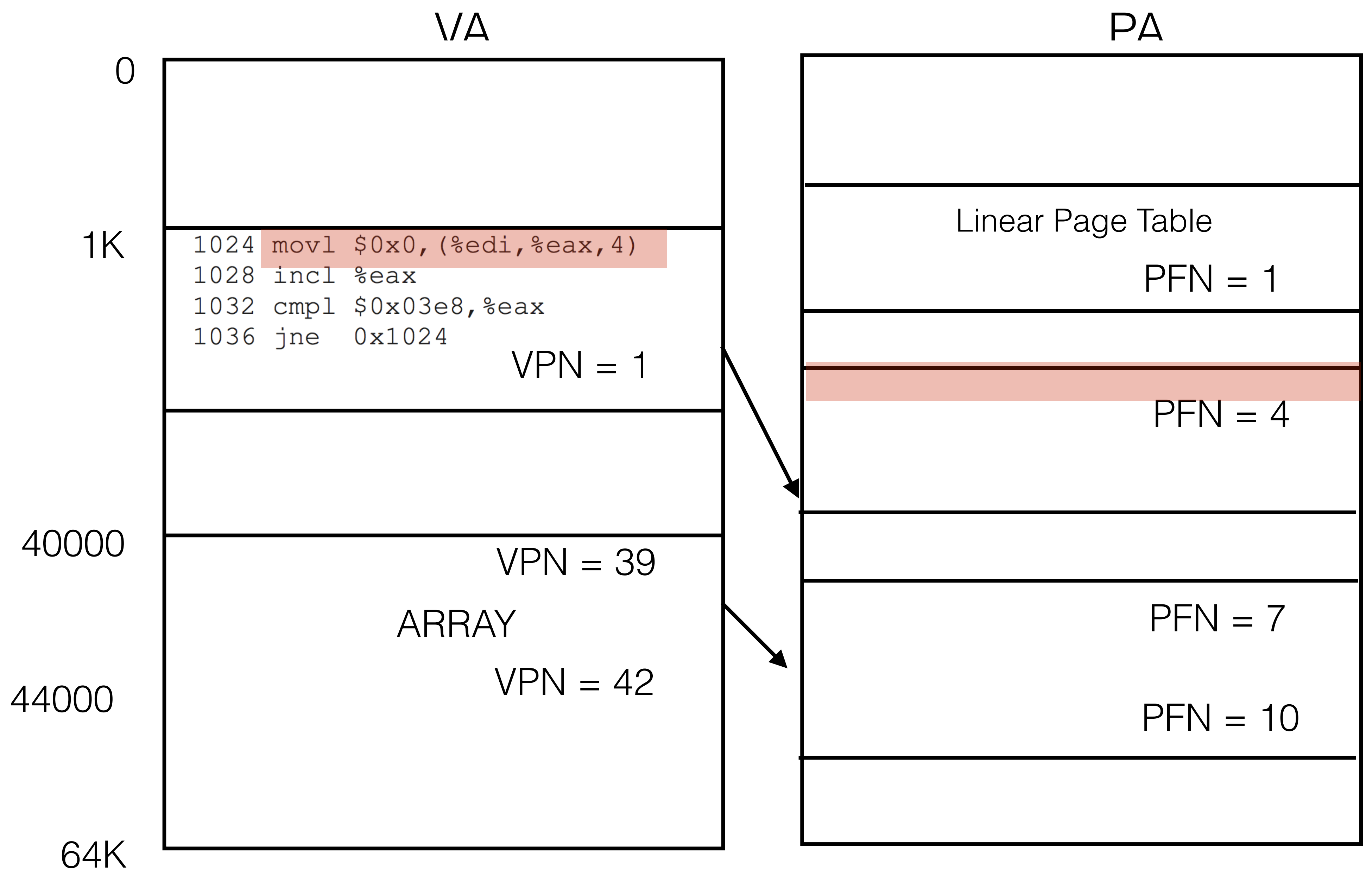
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

READ INSTRUCTION at PA(1024)

TLB

VPN	PFN
1	4



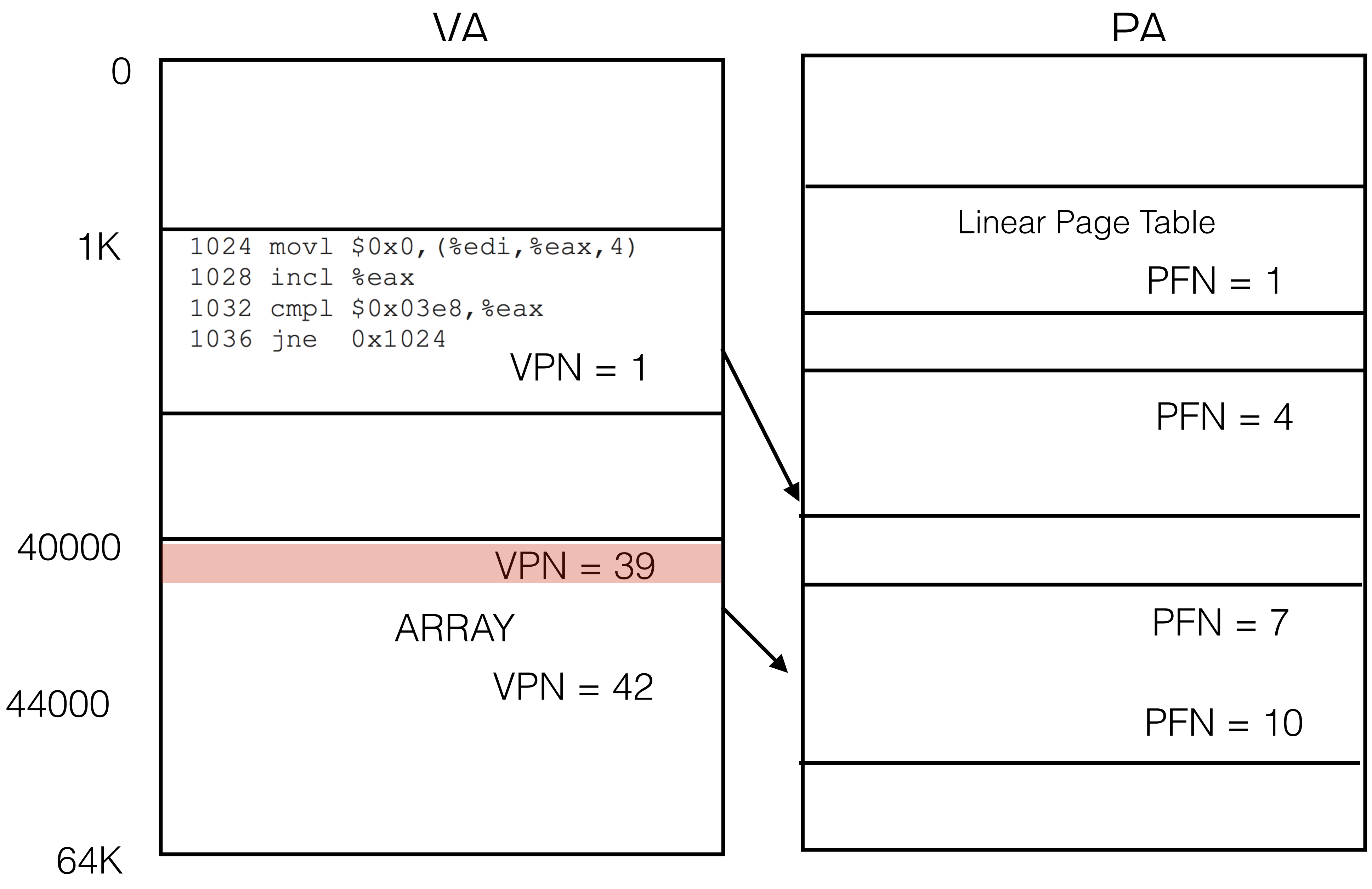
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Find EDI + 4*EAX -> VA = 40000

TLB

VPN	PFN
1	4



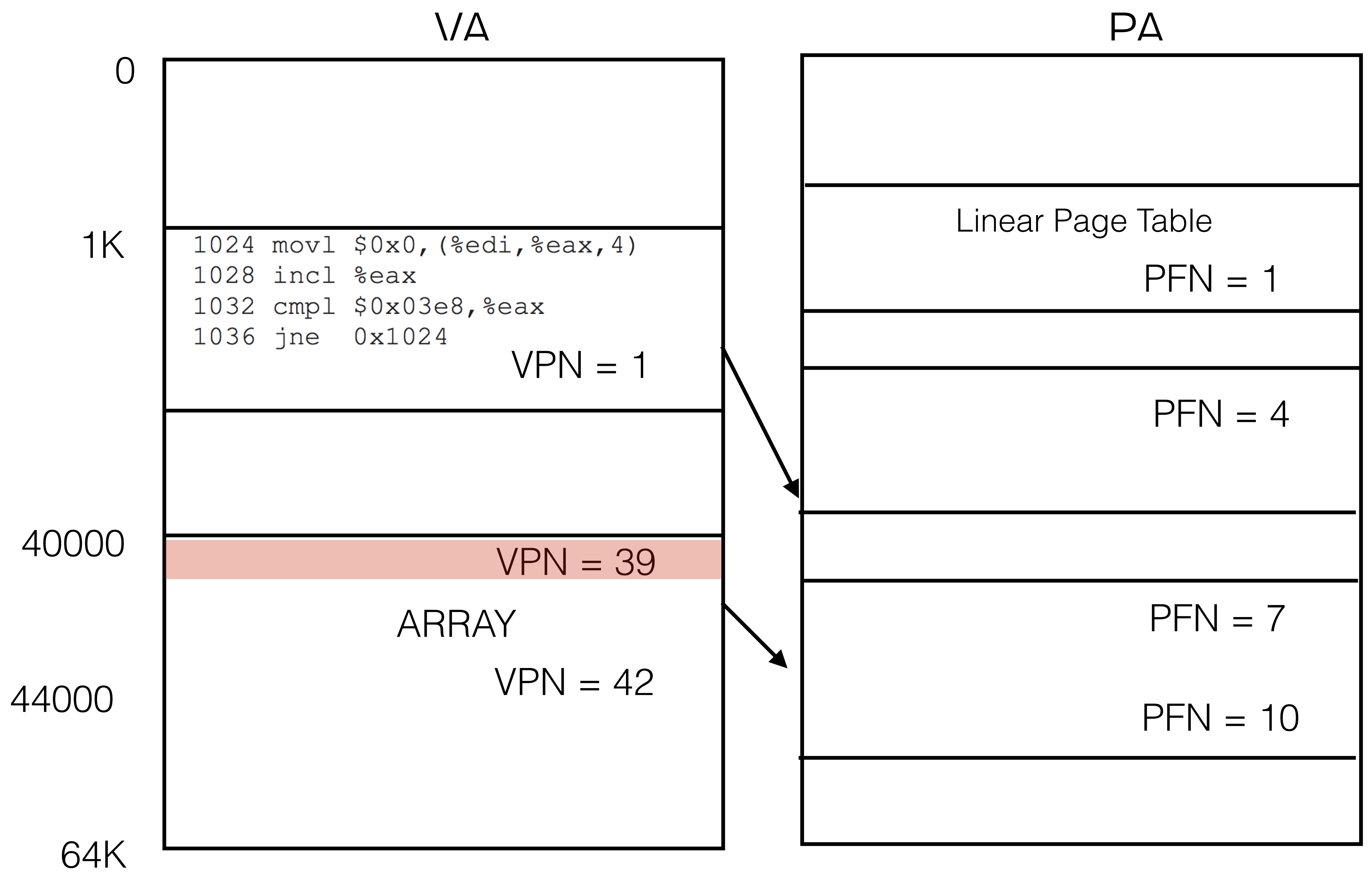
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Find VPN for VA 40000. VPN = 39

TLB

VPN	PFN
1	4



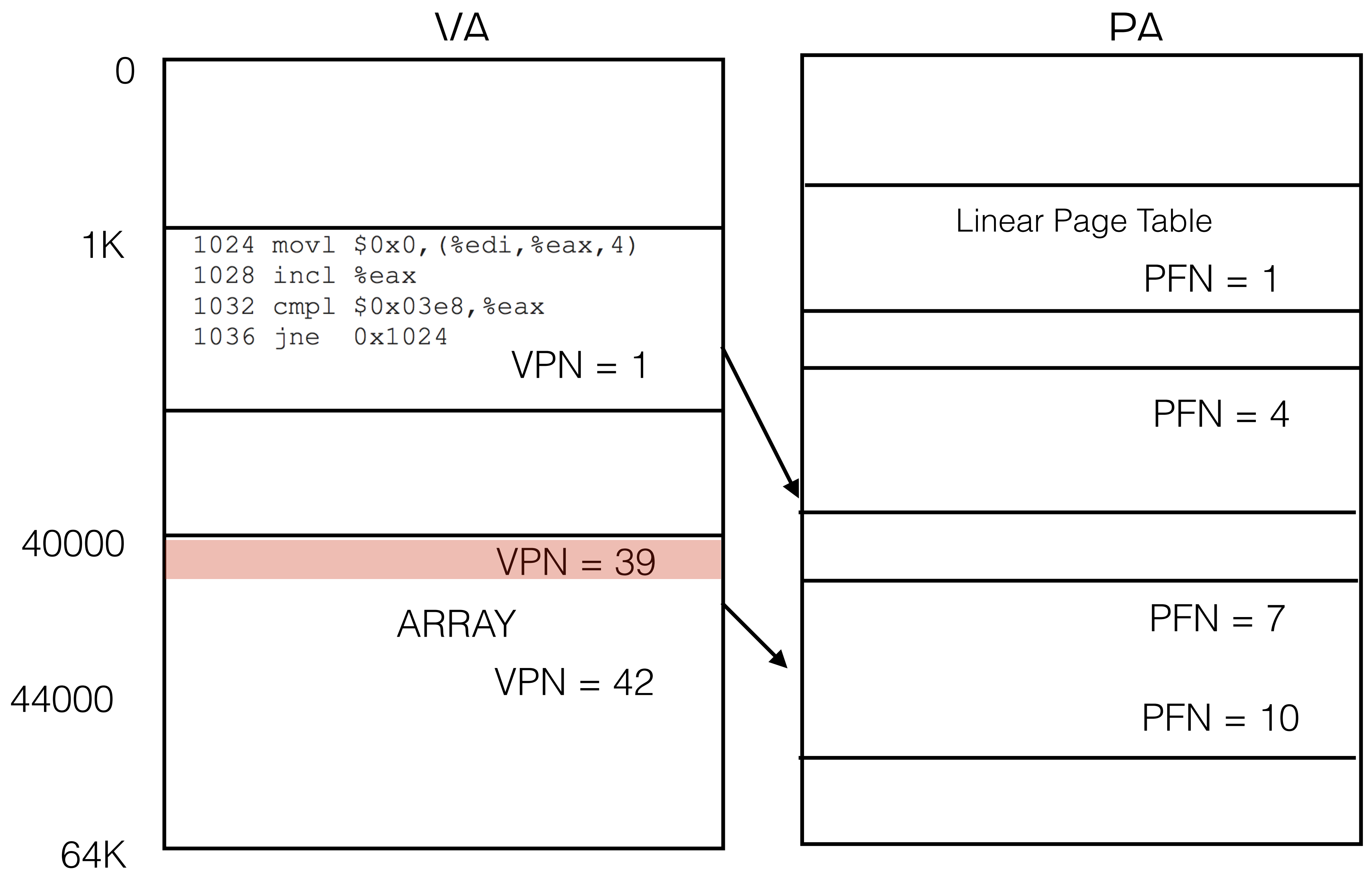
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Check TLB for VPN = 39. Miss!

TLB

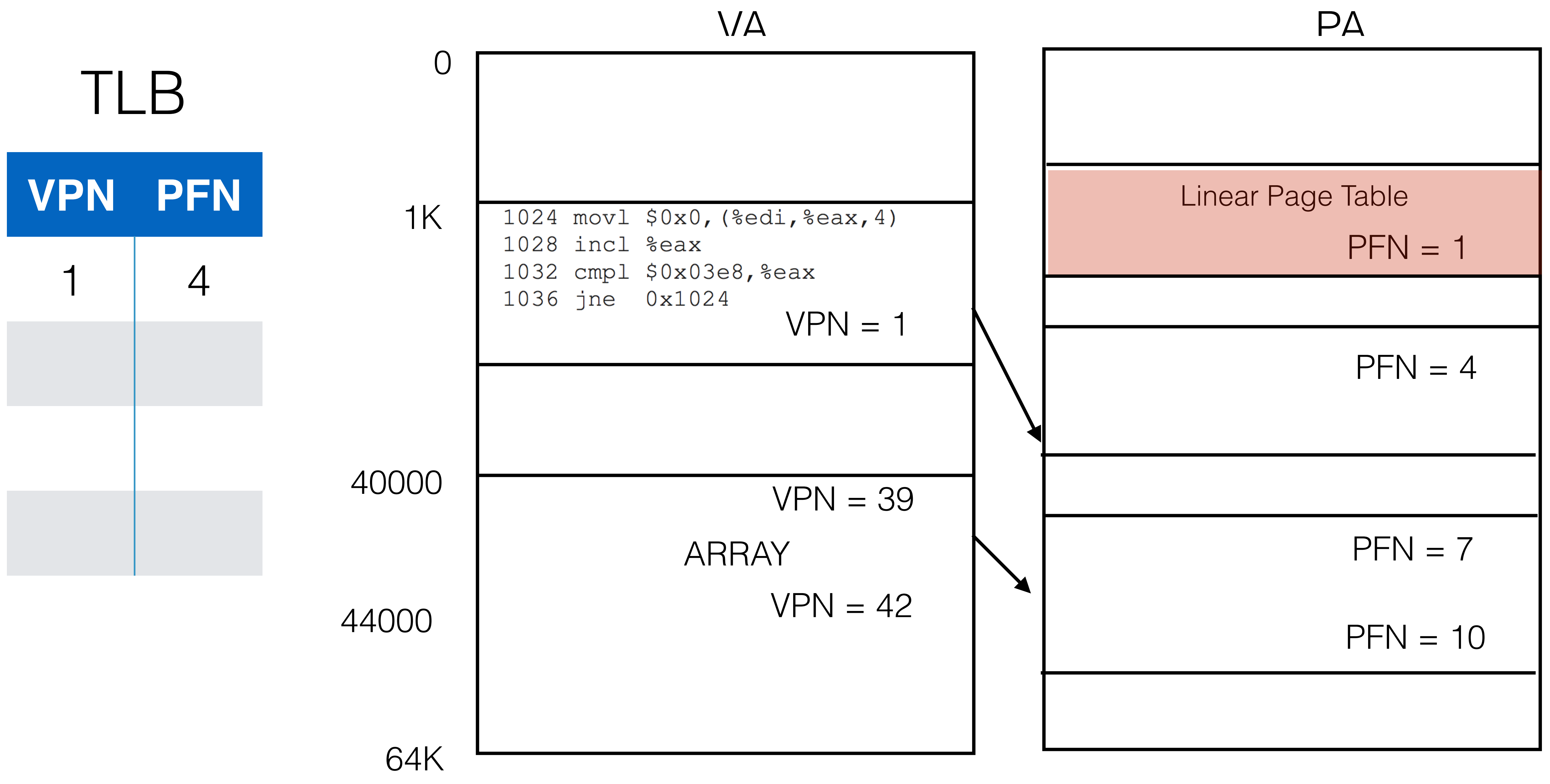
VPN	PFN
1	4



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Get PFN for VPN = 39 from Page Table



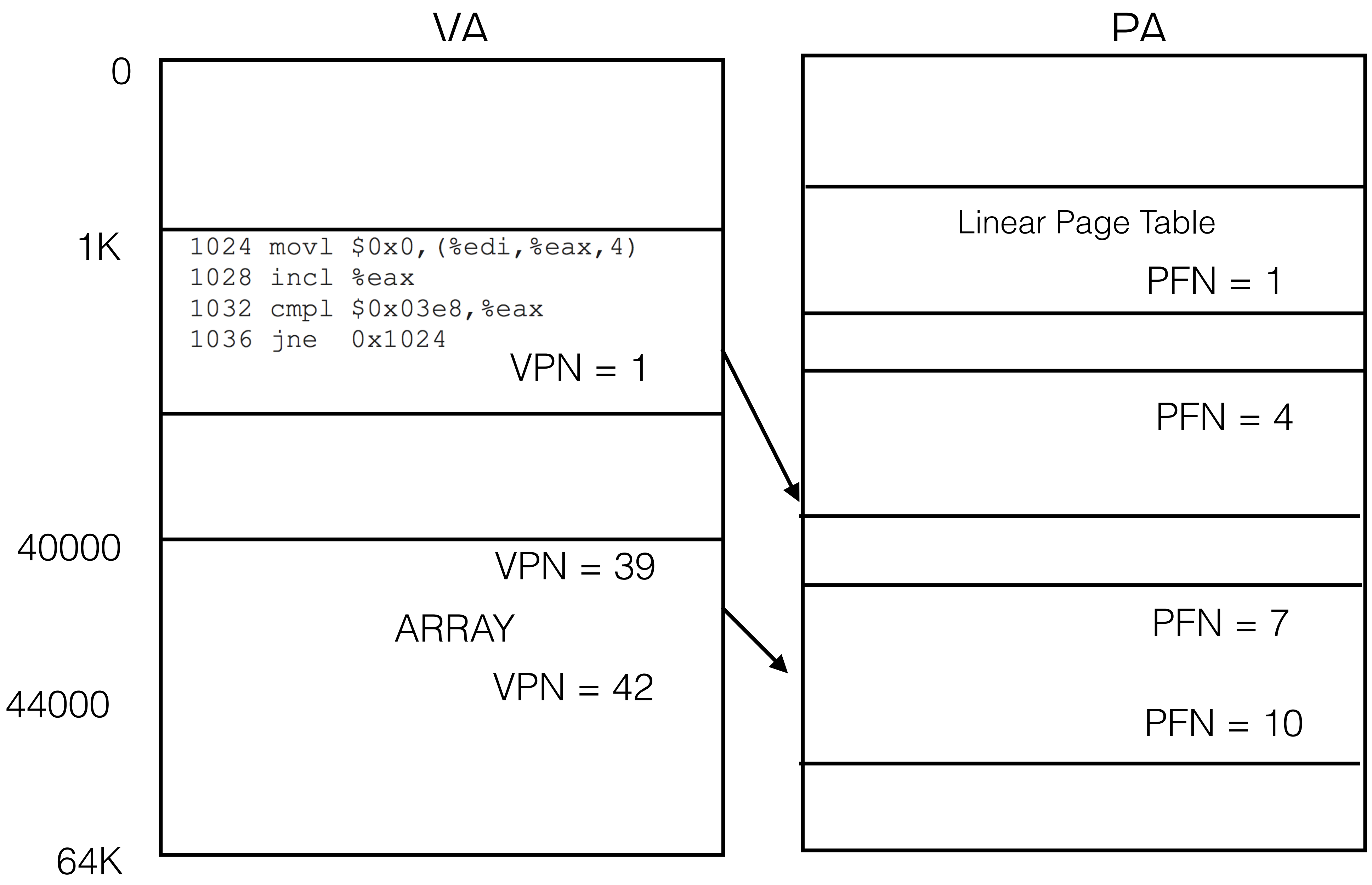
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Store translation in TLB

TLB

VPN	PFN
1	4
39	7



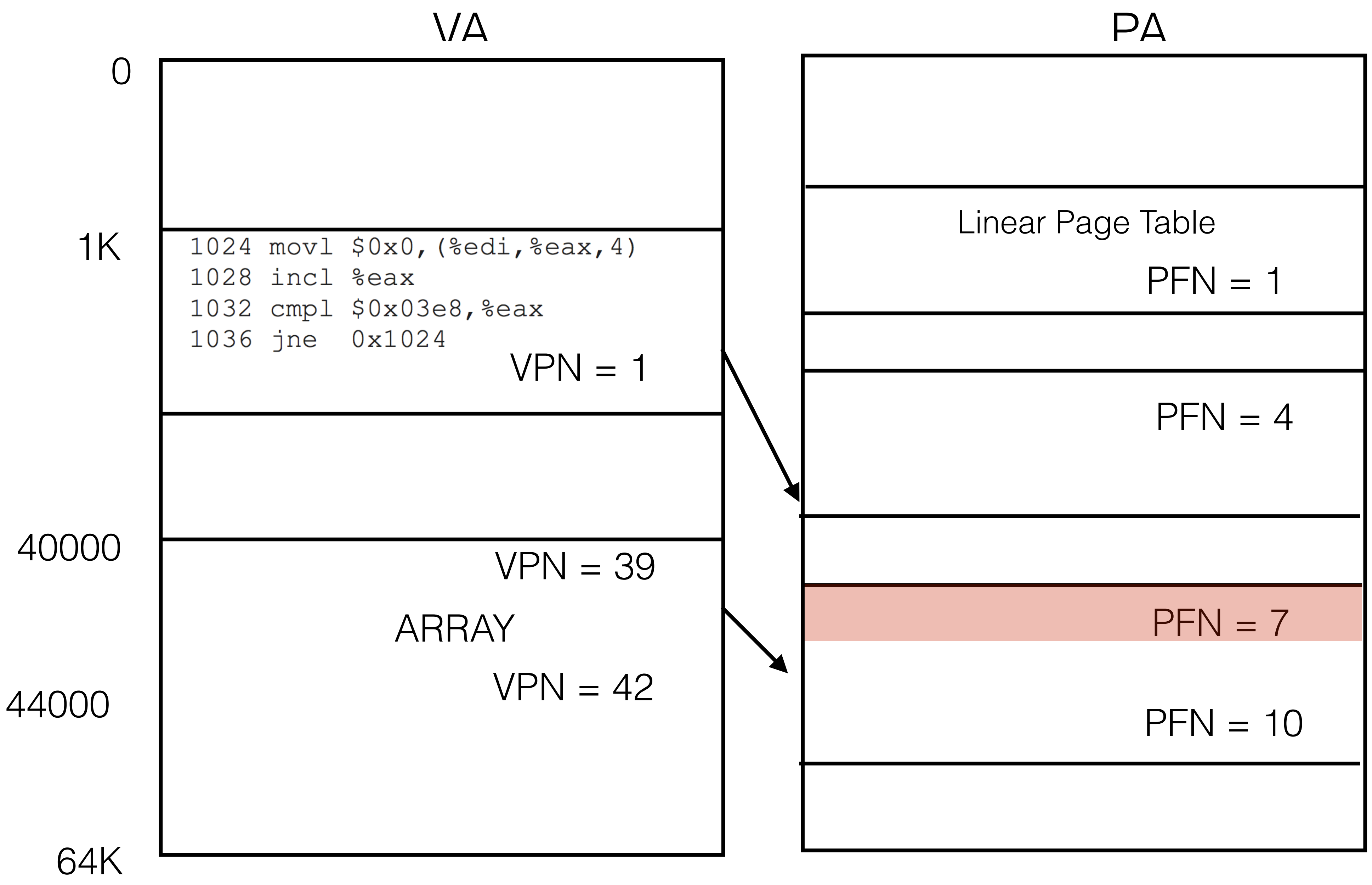
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Find PFN of VPN = 39 from TLB. Add offset to get PA.

TLB

VPN	PFN
1	4
39	7



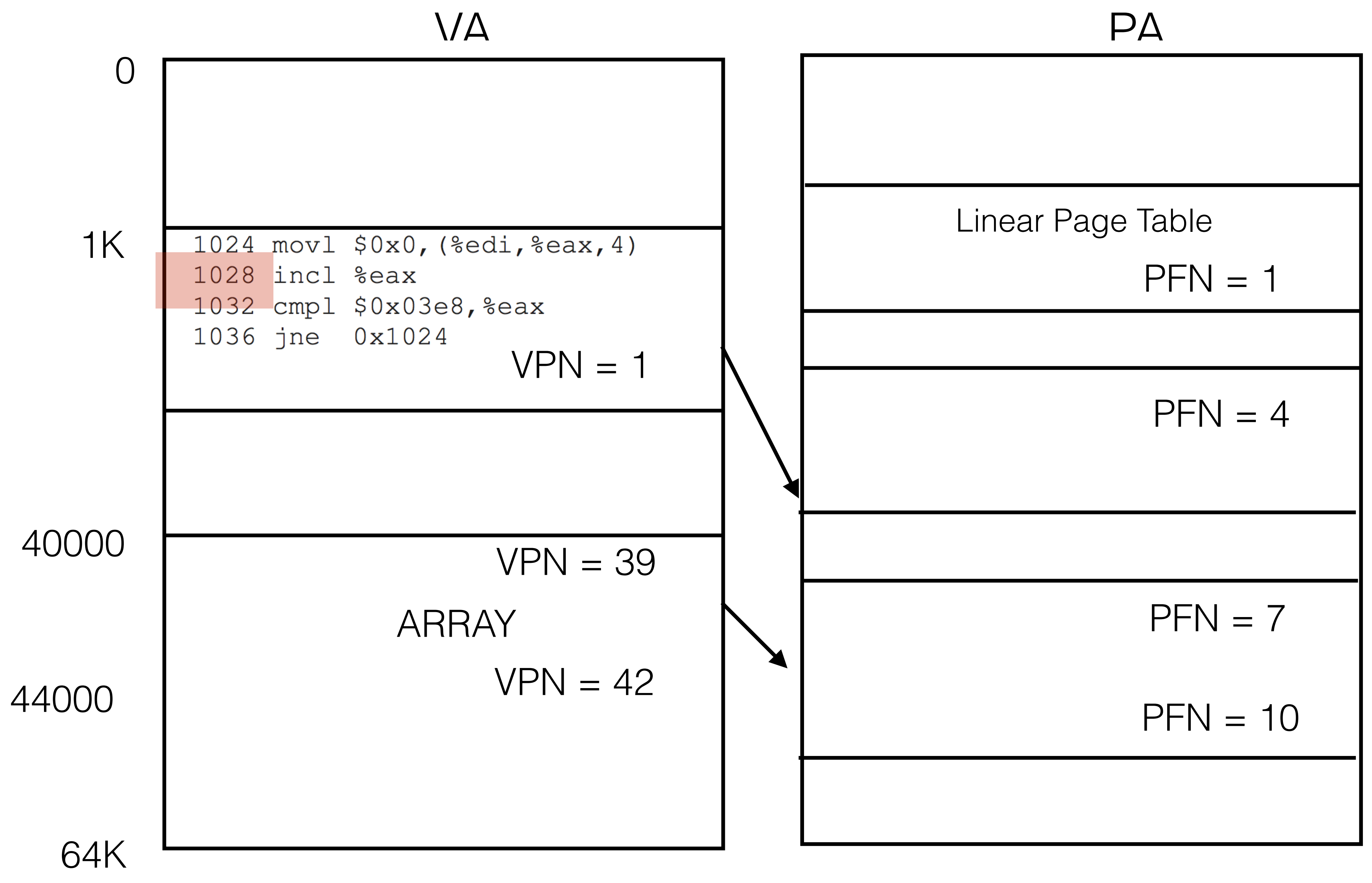
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

FIND PA FOR VA = 1028

TLB

VPN	PFN
1	4
39	7



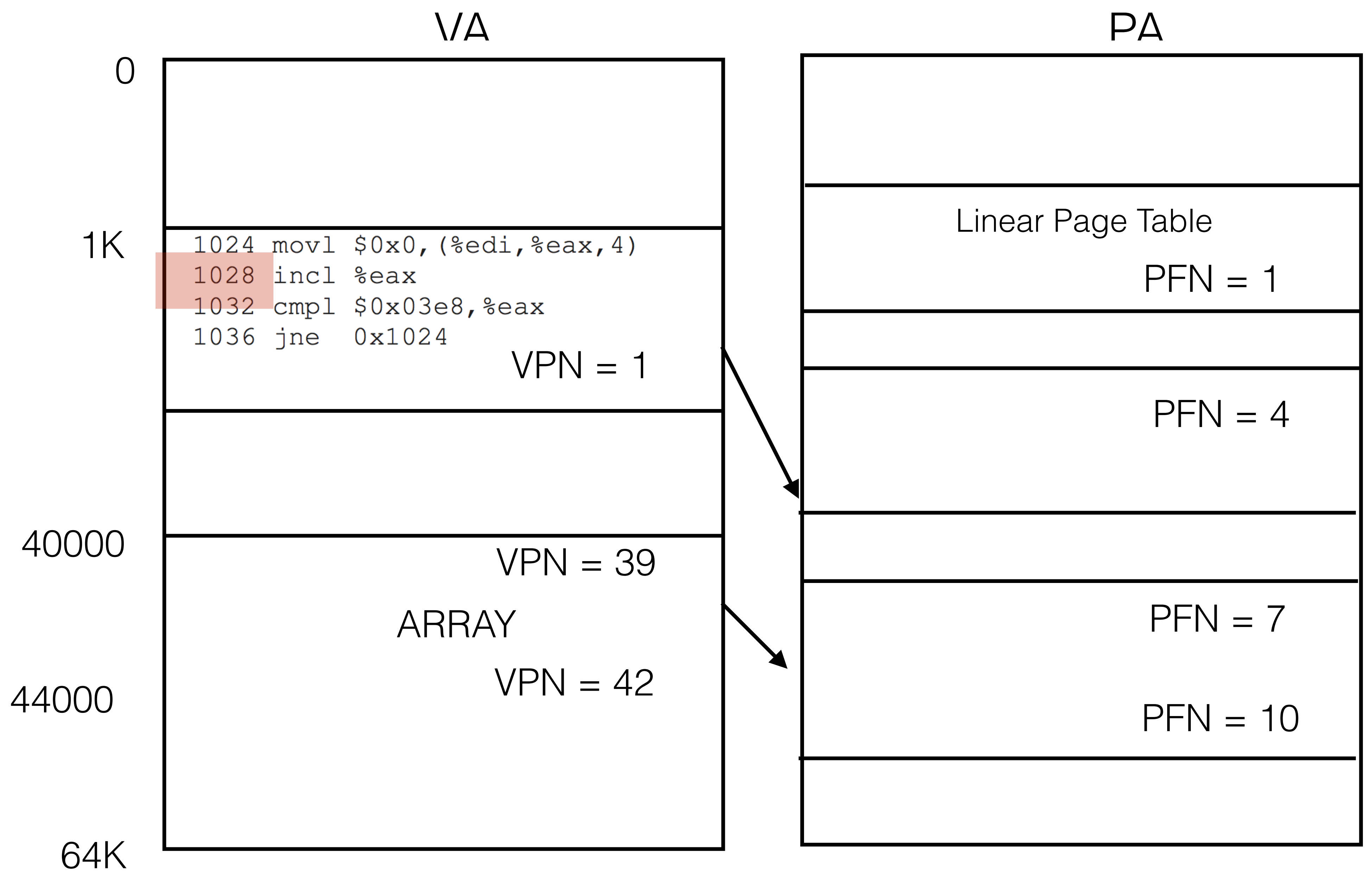
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

VPN = 1. Find Translation in TLB for VPN = 1. Found!

TLB

VPN	PFN
1	4
39	7



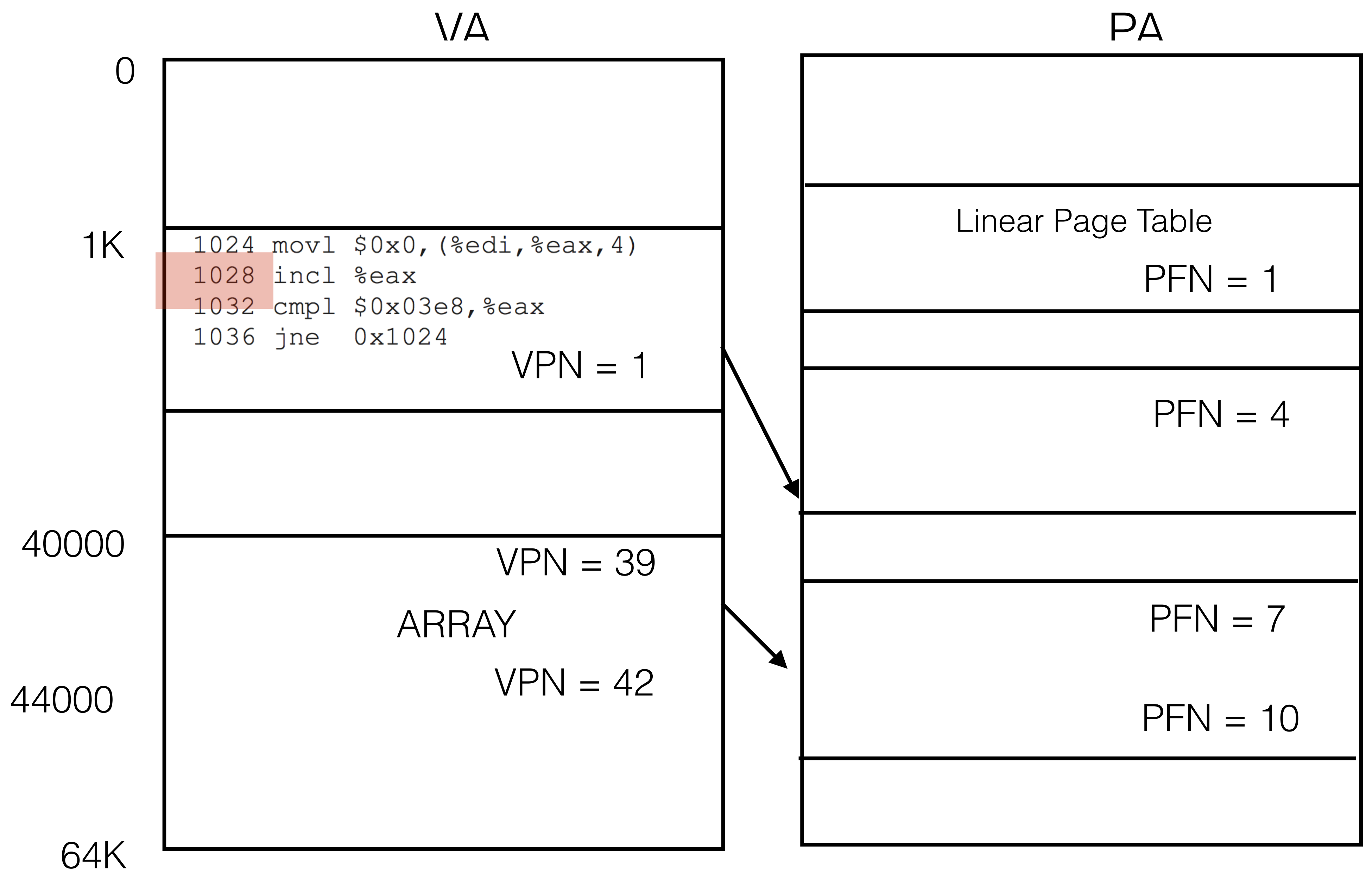
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

PFN = TLB[1] = 4

TLB

VPN	PFN
1	4
39	7



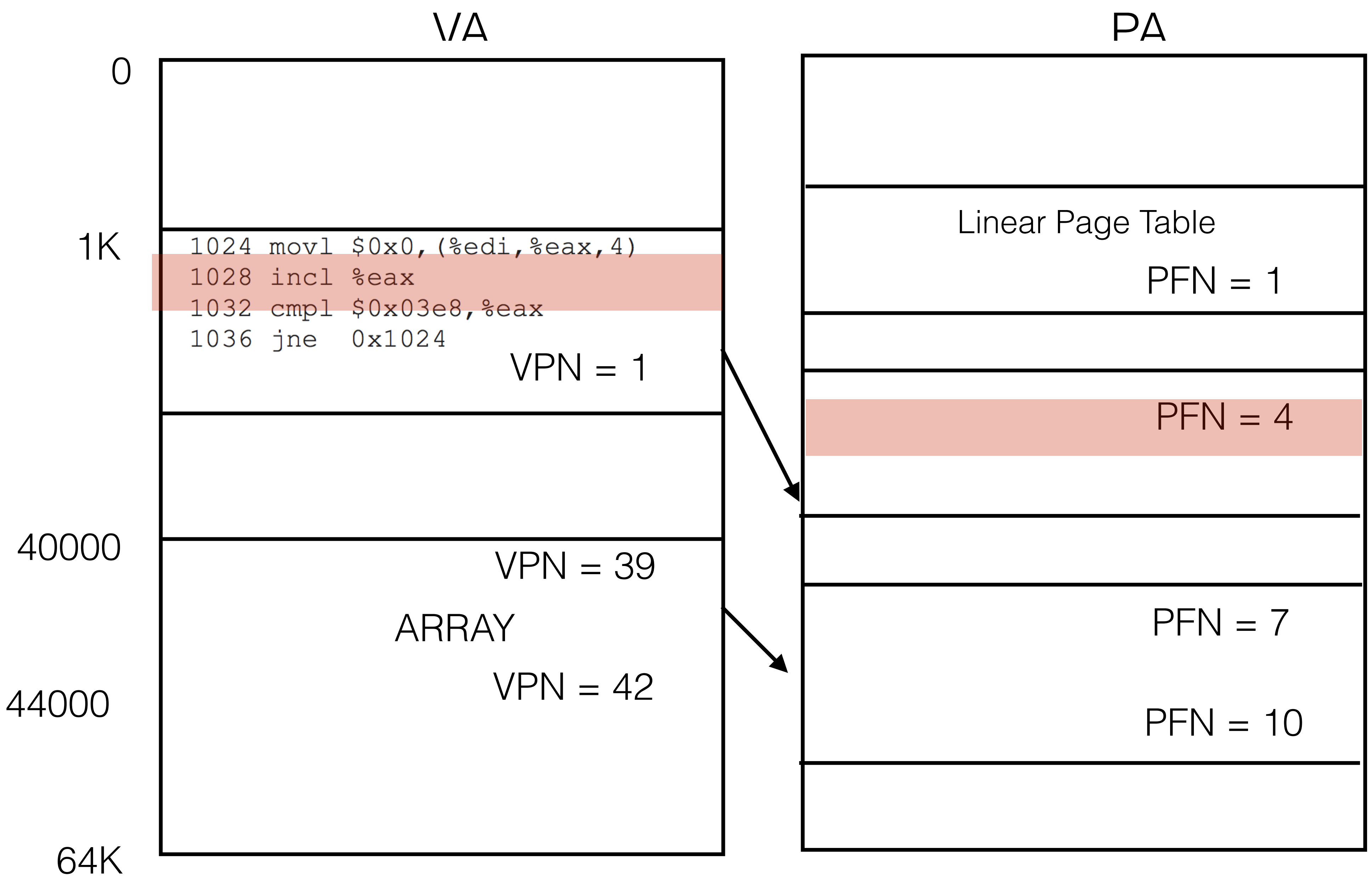
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

Get PA by adding offset to PFN = 4 and execute

TLB

VPN	PFN
1	4
39	7



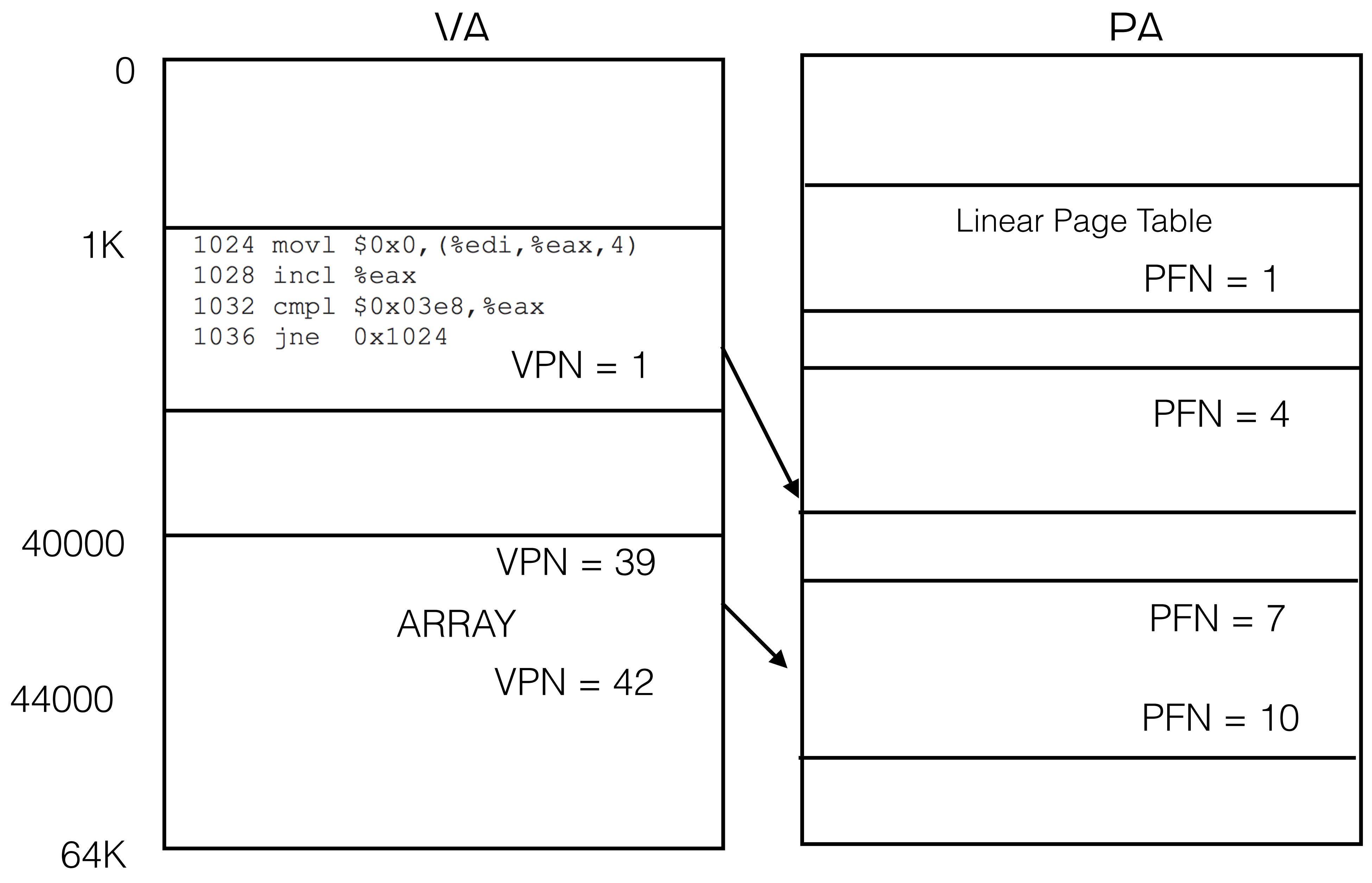
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

1032, 1036, 1024, 1028,…….

TLB

VPN	PFN
1	4
39	7



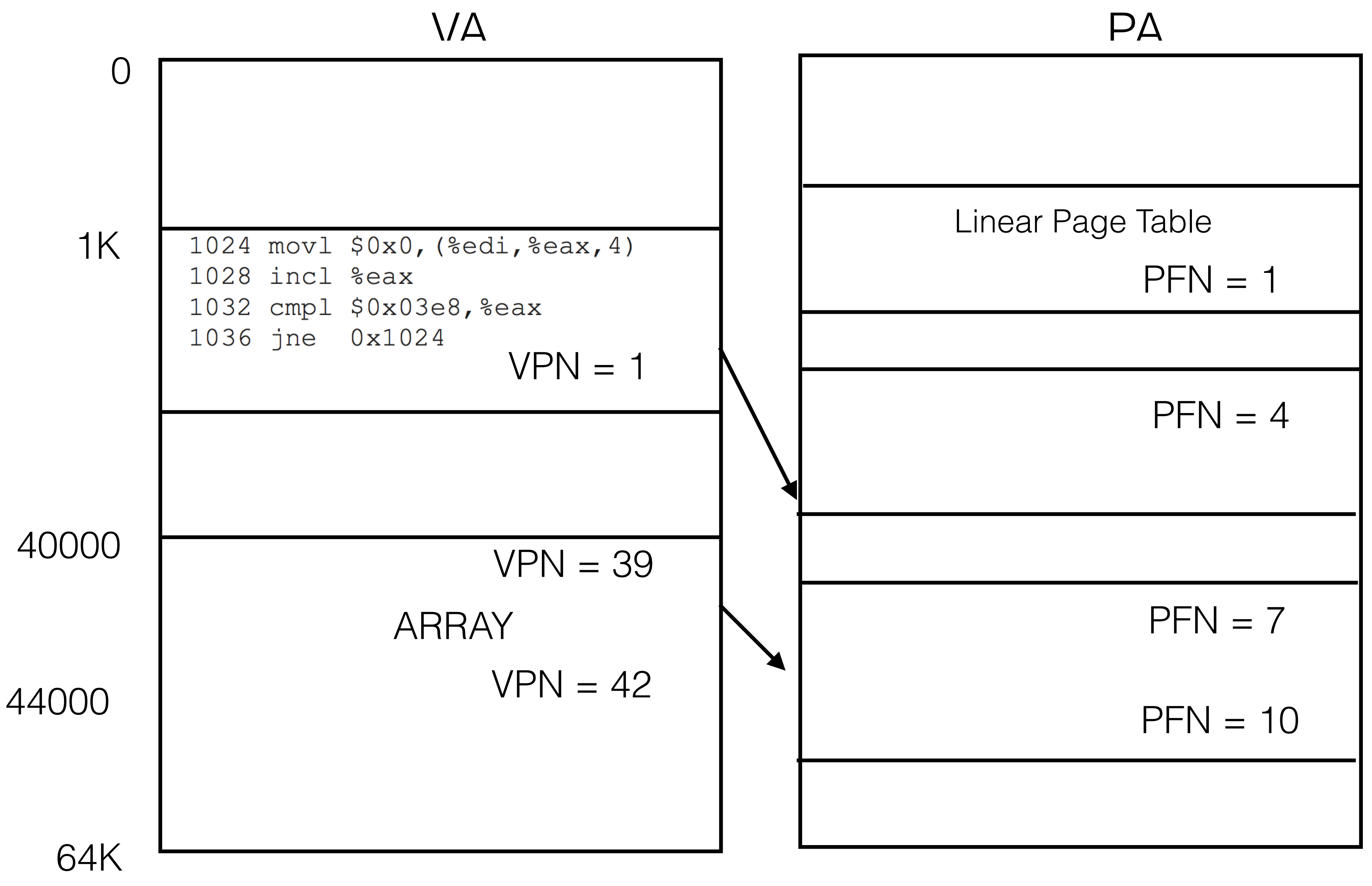
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

EDI + 4*EAX = 40000 + 4*(1024/4) = 40000 + 1K -> VPN =40

TLB

VPN	PFN
1	4
39	7



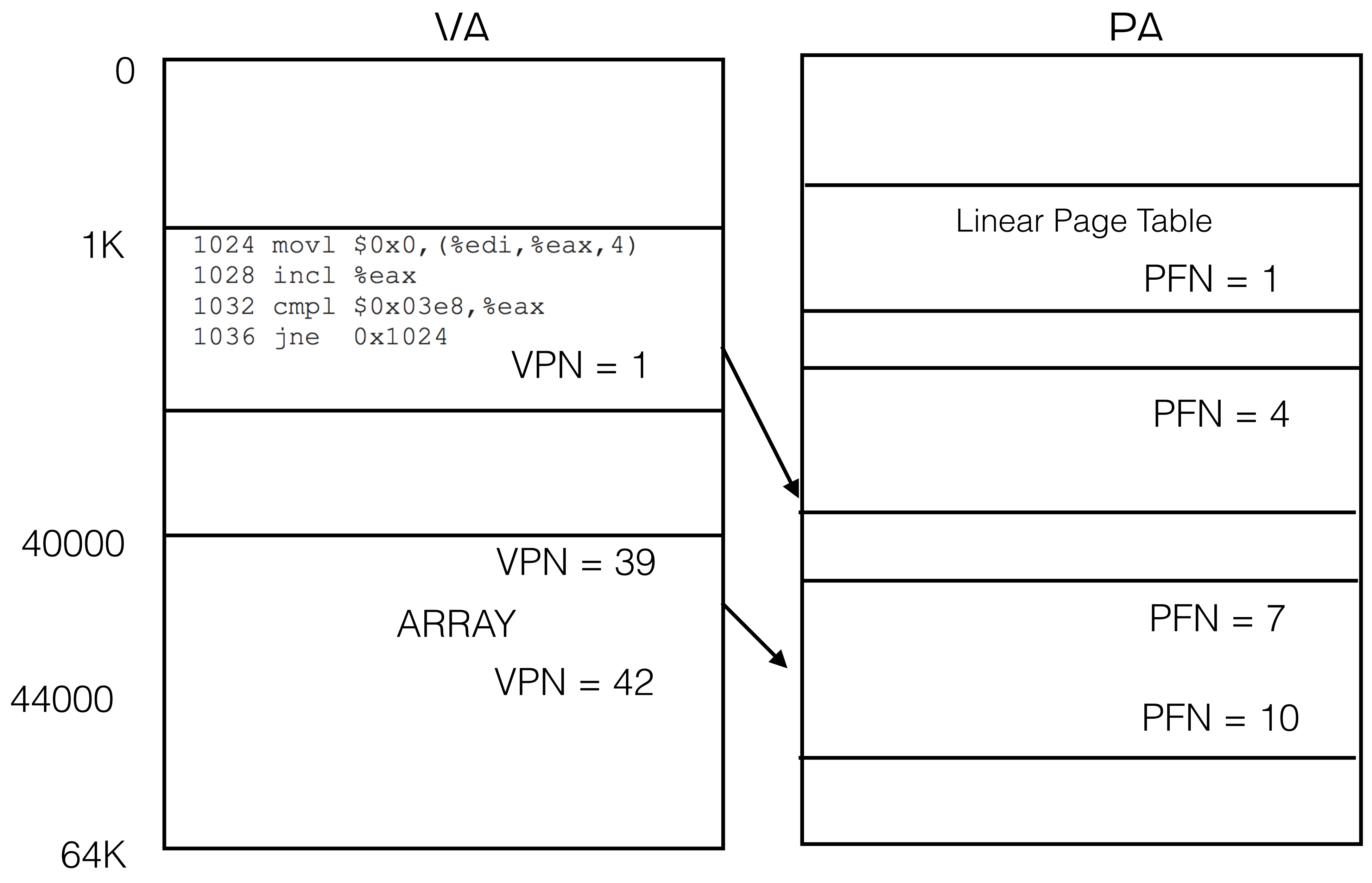
```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Worked Out Example

TLB miss for VPN = 40…

TLB

VPN	PFN
1	4
39	7



```
1024 movl $0x0,(%edi,%eax,4)
1028 incl %eax
1032 cmpl $0x03e8,%eax
1036 jne  0x1024
```

Spatial and Temporal Locality

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)
2. Spatial locality -> TLB has good hit rate

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)
2. Spatial locality -> TLB has good hit rate
    1. Arrays elements are spatially close (EDX + 4*EAX)

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)
2. Spatial locality -> TLB has good hit rate
    1. Arrays elements are spatially close (EDX + 4*EAX)
    2. Instructions are spatially close (1024, …)

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)
2. Spatial locality -> TLB has good hit rate
    1. Arrays elements are spatially close (EDX + 4*EAX)
    2. Instructions are spatially close (1024, …)
3. Temporal locality -> TLB has a good hit rate

Spatial and Temporal Locality

1. Hit rate = TLB Hit/(TLB Hit + TLB Miss)
2. Spatial locality -> TLB has good hit rate
    1. Arrays elements are spatially close (EDX + 4*EAX)
    2. Instructions are spatially close (1024, …)
3. Temporal locality -> TLB has a good hit rate
    1. Loop. Re-using same instructions which exist in TLB

Memory Cycle Rate Example

Memory Cycle Rate Example

Memory Cycle Rate Example

1. Hit = 1 clock cycle

Memory Cycle Rate Example

1. Hit = 1 clock cycle
2. Miss = 30 clock cycles

Memory Cycle Rate Example

1. Hit = 1 clock cycle
2. Miss = 30 clock cycles
3. Miss rate = 1%

Memory Cycle Rate Example

1. Hit = 1 clock cycle
2. Miss = 30 clock cycles
3. Miss rate = 1%
4. Cycle rate = .99*1 + .01*(30 + 1) =1.3 cycles

Context Switch

TLB

VPN	PFN
1	4
39	7

Context Switch

P1 running

TLB

VPN	PFN
1	4
39	7

Context Switch

P1 running

TLB

VPN	PFN
1	4
39	7

Context Switch

P1 running

P2 running

TLB

VPN	PFN
1	4
39	7

Context Switch

TLB

P1 running

P2 running

VPN	PFN
1	4
39	7
…	…
1	30

Context Switch

P1 running

P2 running

TLB

VPN	PFN
1	4
39	7
…	…
1	30

What will VPN 1 be mapped to?