Lecture 30 - Paging

CprE 308

March 27, 2015

Paging

Review: Scenario

Ideal World (for the programmer)

- I'm the only process in the world
- I have more memory than I need at my disposal

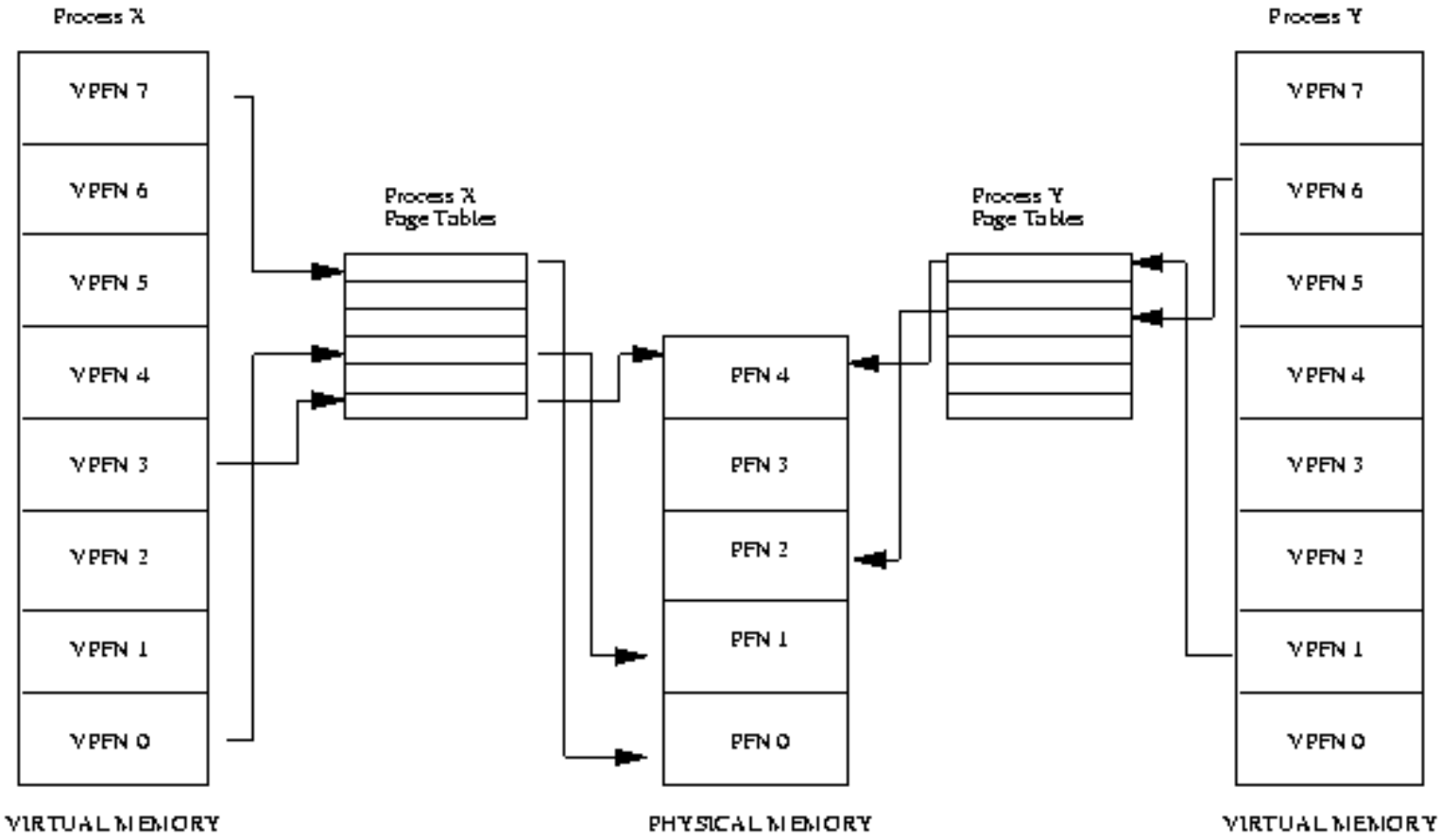
Real World

- Many processes in the system
- Not enough memory for them all
- Not all processes play nicely

Review: Goal of Memory Management

- Present the ideal world view to the programmer, yet implement it on a real system
- Add memory protections without getting in the way of the programmer

Review: Virtual Memory



Structuring Virtual Memory

- Paging
  - Divides the address space into fixed-sized pages
  - Reduces fragmentation, increases efficiency
- Segmentation
  - Divides the address space into variable-sized segments
  - Enables memory protections (Example: data, code, uninitialized, shared memory, etc.)
- Modern OS's use a mixture of both schemes (paged segmentation)

Typical Page Table Entry

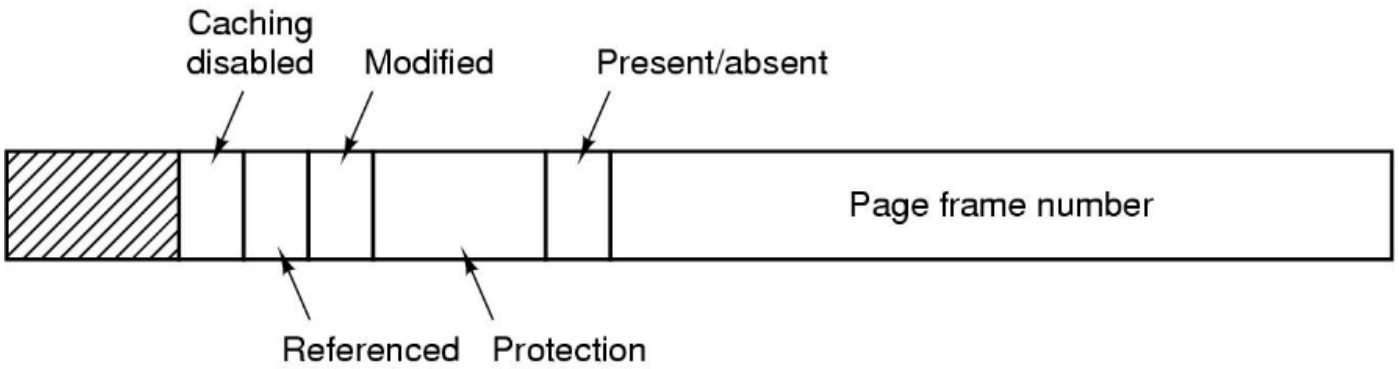


Figure 2: Page Table Entry Fields

Paging Example

- Consider a virtual memory system with two processes
  - Let the physical memory consist of 24 words and the page frame size of four words
  - Process 1 consists of 16 words (a through p)
  - Process 2 consists of 12 words (A through L)

Paging Example (Process 1 Virtual Memory)

Process 1 Virtual Memory

Virtual Address	Memory Contents
0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Paging Example (Process 1 Virtual Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Paging Example (Process 2 Virtual Memory)

Process 2 Virtual Memory

Virtual Address	Memory Contents
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	J
10	K
11	L

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Figure 5: Process 2 Virtual Memory

Paging Example (Process 2 Virtual Memory)

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Figure 6: Process 2 Virtual Memory

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	
	1	
	2	
	3	
1	4	
	5	
	6	
	7	
2	8	
	9	
	10	
	11	
3	12	
	13	
	14	
	15	
4	16	
	17	
	18	
	19	
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	
	5	
	6	
	7	
2	8	
	9	
	10	
	11	
3	12	
	13	
	14	
	15	
4	16	
	17	
	18	
	19	
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	
	9	
	10	
	11	
3	12	
	13	
	14	
	15	
4	16	
	17	
	18	
	19	
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	
	13	
	14	
	15	
4	16	
	17	
	18	
	19	
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	
	17	
	18	
	19	
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	m
	17	n
	18	o
	19	p
5	20	
	21	
	22	
	23	

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	m
	17	n
	18	o
	19	p
5	20	I
	21	J
	22	K
	23	L

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Physical Memory)

- Suppose process 1 is running and it tries to access the contents of the virtual address **15**, what is the result?

Paging Example (Physical Memory)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	m
	17	n
	18	o
	19	p
5	20	I
	21	J
	22	K
	23	L

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Page Faults)

- Suppose process 1 is running and it tries to access the contents of the virtual address **15**, what is the result?
  - Virtual address **15** is in process 1's virtual page **3**. According to the page table for process 1, the virtual page **3** is paged in physical memory as page **4**, which means the value **p** will be immediately fetched from memory.

Paging Example (Page Faults)

- Suppose process 1 is running and it tries to access the contents of the virtual address **9**, what is the result?

Paging Example (Page Faults)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	m
	17	n
	18	o
	19	p
5	20	I
	21	J
	22	K
	23	L

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Page Faults)

- Suppose process 1 is running and it tries to access the contents of the virtual address **9**, what is the result?
  - Virtual address **9** is in process 1's virtual page **2**. According to the page table for process 1, virtual page **2** is not paged in physical memory (flagged as invalid in the page table). A **page fault** occurs, and physical memory will need to be swapped before the value **j** can be fetched from memory.

Paging Example (Address Translation)

Process 1

- Virtual Address **2** to Physical Address
- Physical Address **5** to Virtual Address

Process 2

- Virtual Address **2** to Physical Address
- Physical Address **22** to Virtual Address

Paging Example (Address Translation)

Process 1 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	a
	1	b
	2	c
	3	d
1	4	e
	5	f
	6	g
	7	h
2	8	i
	9	j
	10	k
	11	l
3	12	m
	13	n
	14	o
	15	p

Process 2 Virtual Memory

Virtual Page	Virtual Address	Memory Contents
0	0	A
	1	B
	2	C
	3	D
1	4	E
	5	F
	6	G
	7	H
2	8	I
	9	J
	10	K
	11	L

Physical Memory

Physical Page	Physical Address	Memory Contents
0	0	E
	1	F
	2	G
	3	H
1	4	e
	5	f
	6	g
	7	h
2	8	a
	9	b
	10	c
	11	d
3	12	A
	13	B
	14	C
	15	D
4	16	m
	17	n
	18	o
	19	p
5	20	I
	21	J
	22	K
	23	L

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Paging Example (Address Translation)

Process 1

- Virtual Address **2** to Physical Address
  - **10**
- Physical Address **5** to Virtual Address
  - **5**

Process 2

- Virtual Address **2** to Physical Address
  - **14**
- Physical Address **22** to Virtual Address
  - **10**

Implementation Notes

- Virtual memory is just a concept
  - It's addresses/values are always contiguous
  - It's values only really exist in physical memory
  - Page frames are just logical groupings (that can be calculated on the fly)
- Only need to store page tables

Implementation Notes

Process 1 Page Table

Virtual Page	Physical Page
0	2
1	1
2	invalid
3	4

Process 2 Page Table

Virtual Page	Physical Page
0	3
1	0
2	5

Physical Memory

Physical Address	Memory Contents
0	E
1	F
2	G
3	H
4	e
5	f
6	g
7	h
8	a
9	b
10	c
11	d
12	A
13	B
14	C
15	D
16	m
17	n
18	o
19	p
20	I
21	J
22	K
23	L

Implementation Notes

- Virtual page frames are always in order starting at 0
  - No need to store virtual page numbers in page table (just store physical page numbers in order)
- Techinically we don't "store" addresses either

Implementation Notes

Process 1 Page Table

Physical Page
2
1
invalid
4

Process 2 Page Table

Physical Page
3
0
5

Physical Memory

Memory Contents
E
F
G
H
e
f
g
h
a
b
c
d
A
B
C
D
m
n
o
p
I
J
K
L

Implementation Notes

- If our page table stores 4 virtual pages mappings how many bits do we need to represent each page?
- If our page size is 4 words, how many bits do we need to represent each possible page offset?

Implementation Notes

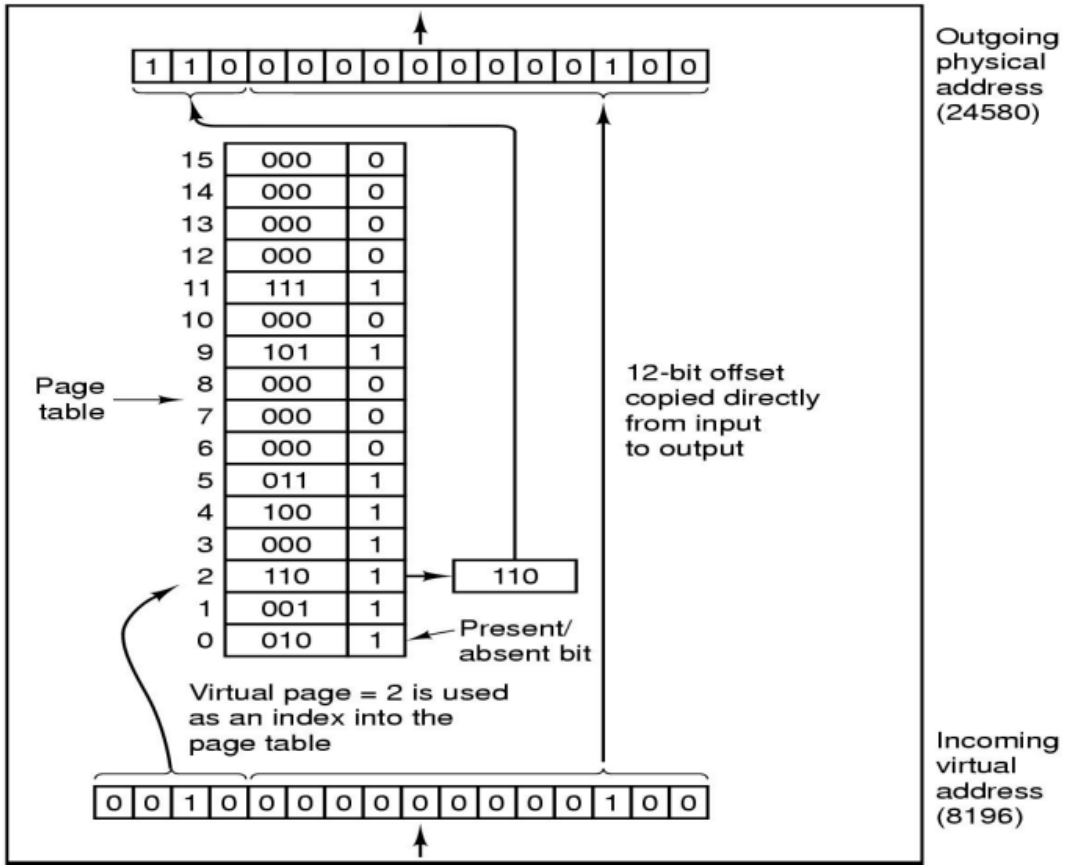


Figure 19: Address Translation

Quiz

- Any questions before the quiz?