CIS 452 - Operating Systems Concepts

Nathan Bowman

Images taken from Silberschatz book

Paging

Multiprogramming setups discussed so far:

- contiguous memory allocation
- segmentation

Paging is another way of managing memory that does not require contiguous allocation for a process

Paging

Physical memory and logical memory are split into fixed-size blocks

Blocks of physical memory are **frames**

Blocks of logical memory are **pages**

Pages and frames are the same size

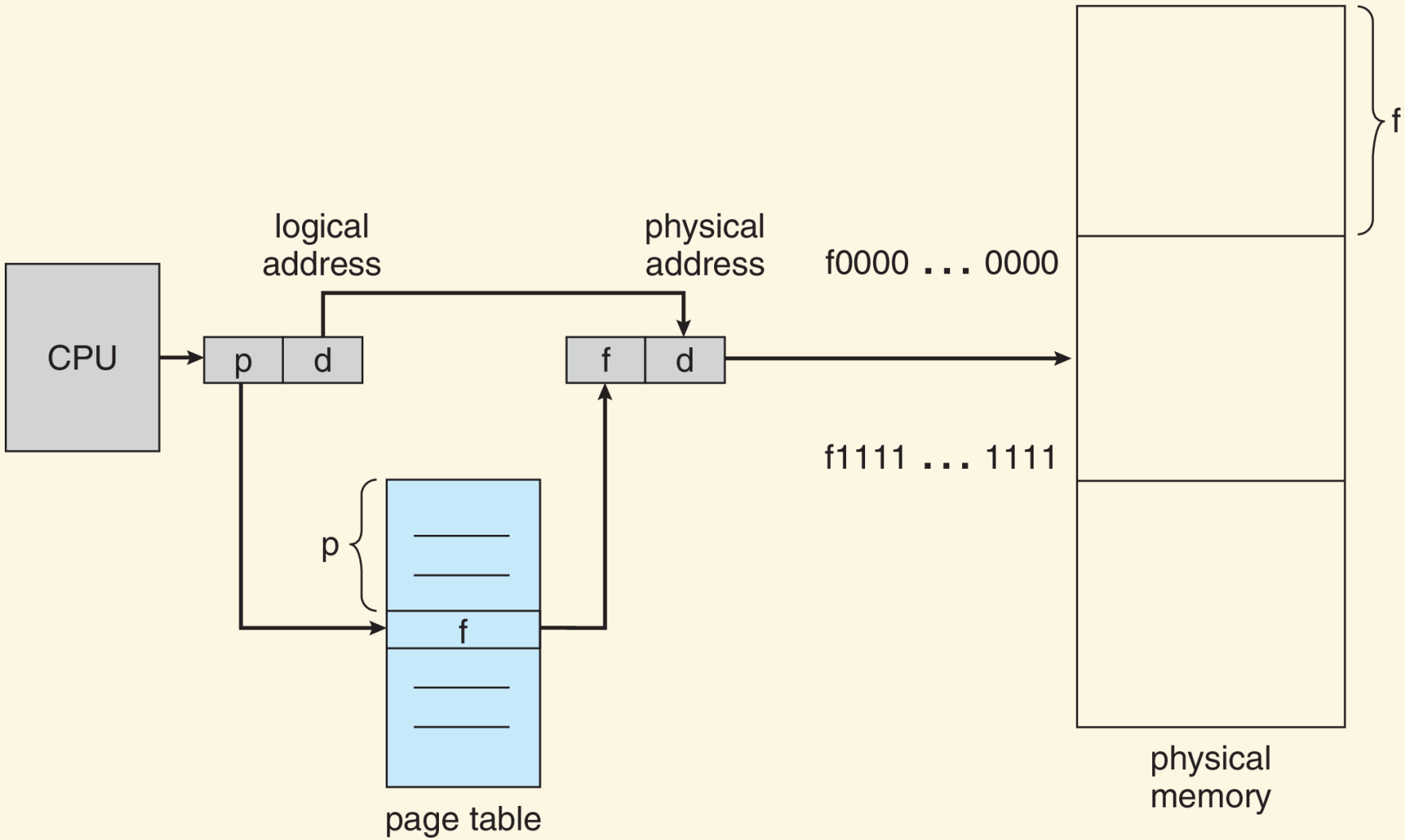
Any page can be mapped to any available frame

How does this memory translation happen?

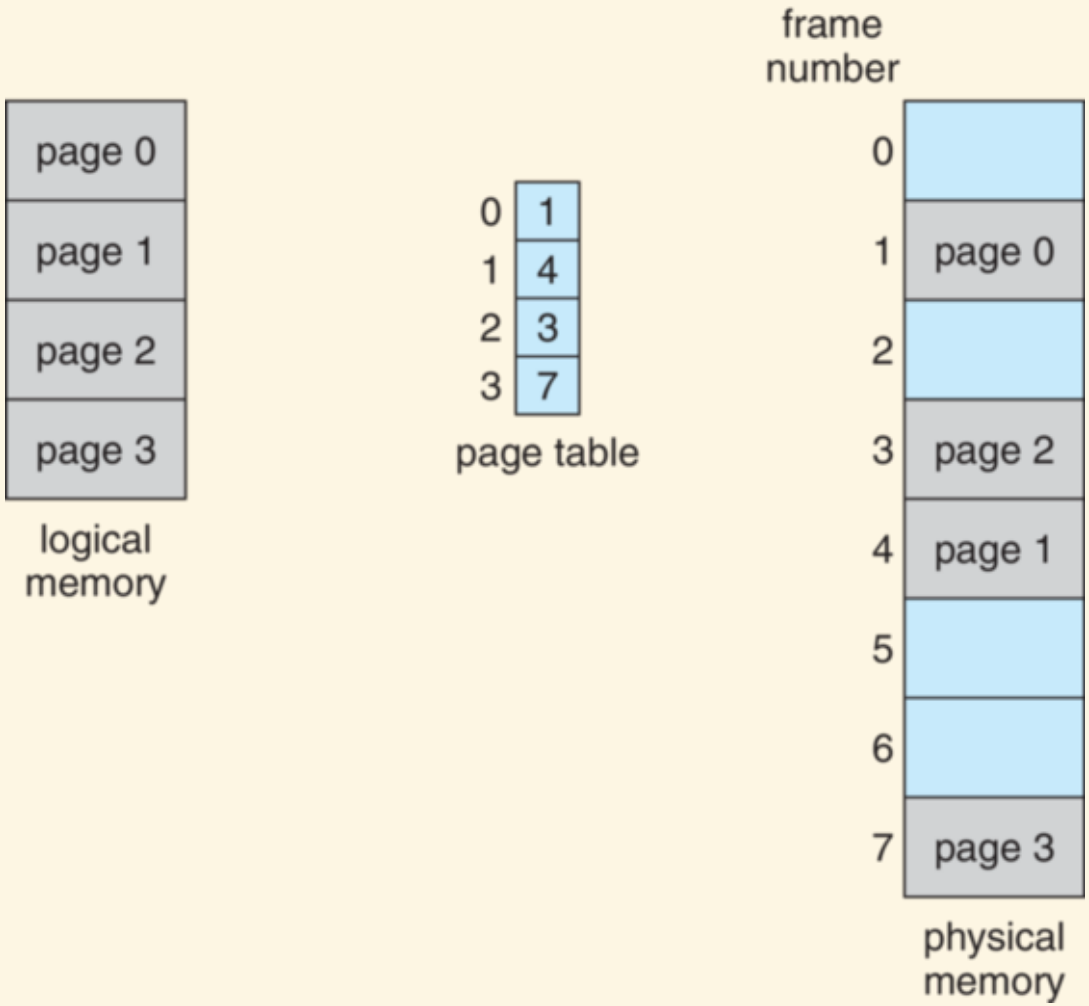
Every address is split into

- **page number** (p)
- **page offset** (d)

OS keeps **page table** mapping page numbers to frame numbers


CPU
logical address
p
d
physical address
f
d
f0000 . . . 0000
f1111 . . . 1111
f
p
f
page table
physical memory

What do logical and physical memory look like?


page 0
page 1
page 2
page 3
logical memory
0 1
1 4
2 3
3 7
page table
frame number
0
1 page 0
2
3 page 2
4 page 1
5
6
7 page 3
physical memory

Notice that (binary) address is broken up into two sets of bits

For a memory of 128 (= 2^7) bytes, e.g.,

```
101 0000
--- ----
  p    d
```

So, both number of pages and size of page will be powers of two

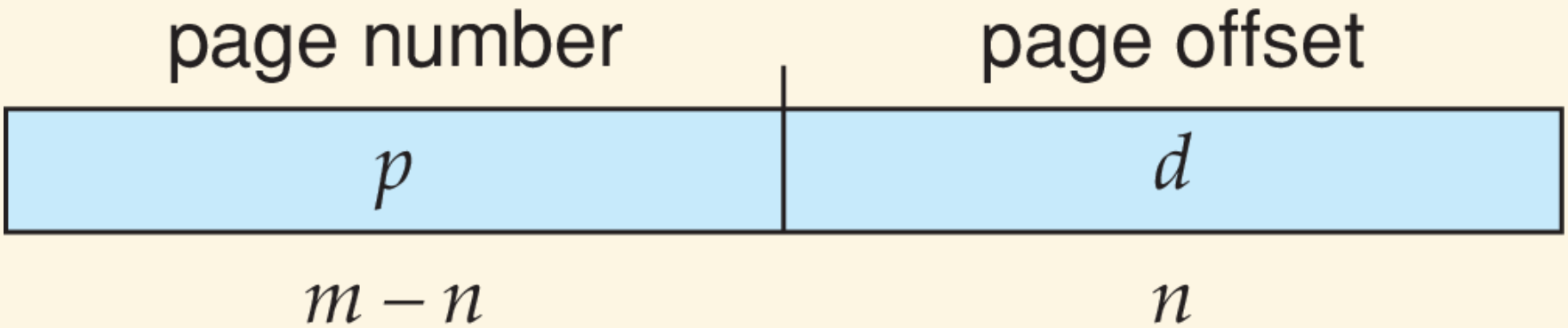
(This splitting of addresses may remind you of cache mapping from 351)

Some general rules:

If logical memory holds 2^m bytes, logical address will have m bits

If page size is 2^n bytes, page offset requires n bits

Remaining (m - n) bits specify page number, and logical memory has 2^(m - n) pages


page number
page offset
p
d
m − n
n

Assume 128 bytes of memory again

Page size is 4 bytes

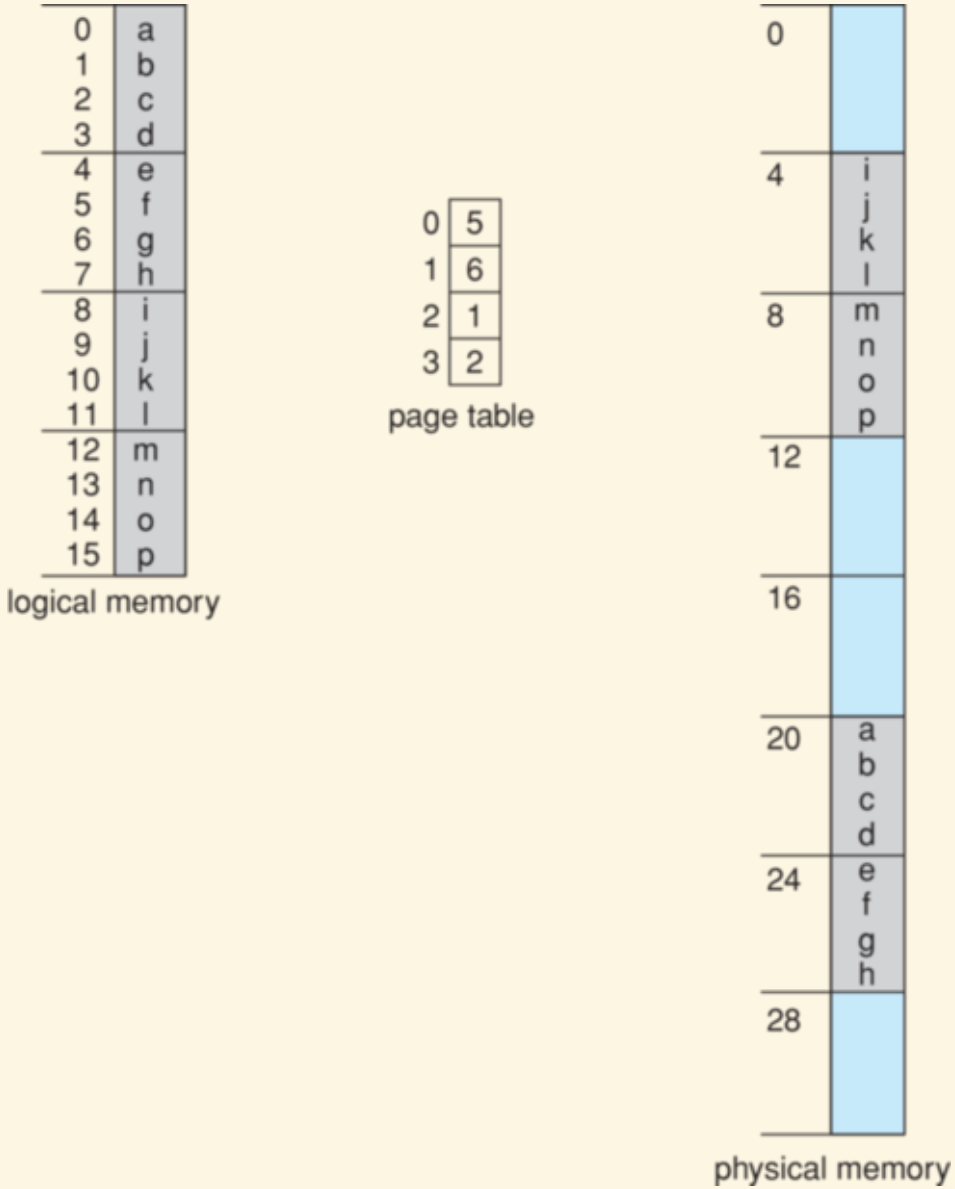
Split up the address:

```
1010000
```

```
10100 00
----- --
    p  d
```

Page 20, offset 0

Next is an example showing how every byte in another (toy) system could be addressed


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
logical memory
0 5
1 6
2 1
3 2
page table
0
4 i j k l
8 m n o p
12
16
20 a b c d
24 e f g h
28
physical memory

Note that logical memory and physical memory can have *different sizes*

Translation is completely transparent to process

Process once again believes it has memory from `[0, max]` and has no idea that pages are actually scattered throughout memory