Lab16
Paging

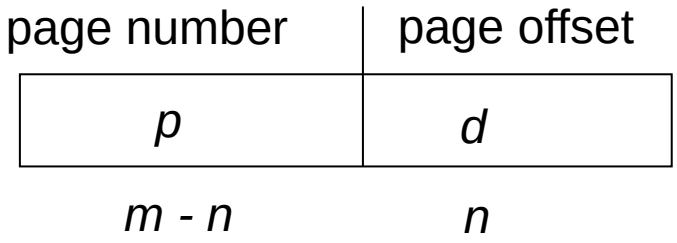
TA: Bo-Hua Xu

Professor: Hsung-Pin Chang

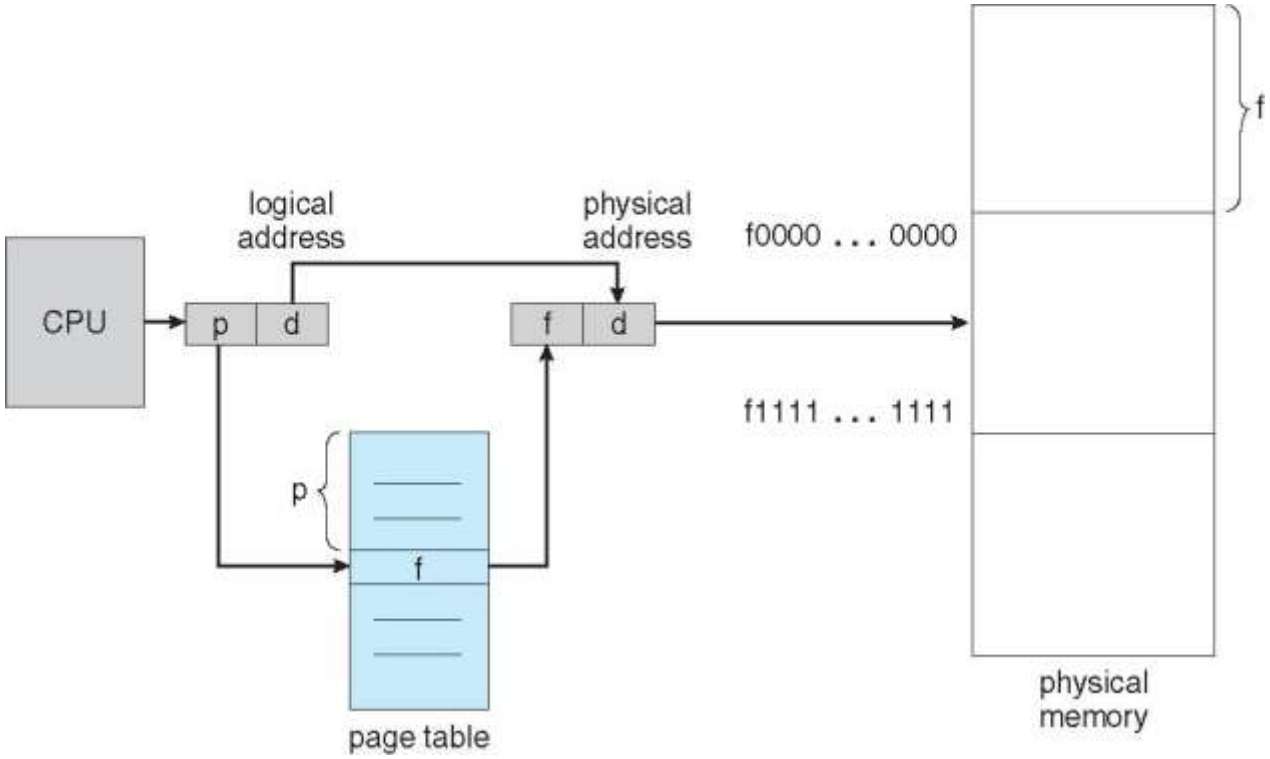
Operating System Lab

Address Translation Scheme

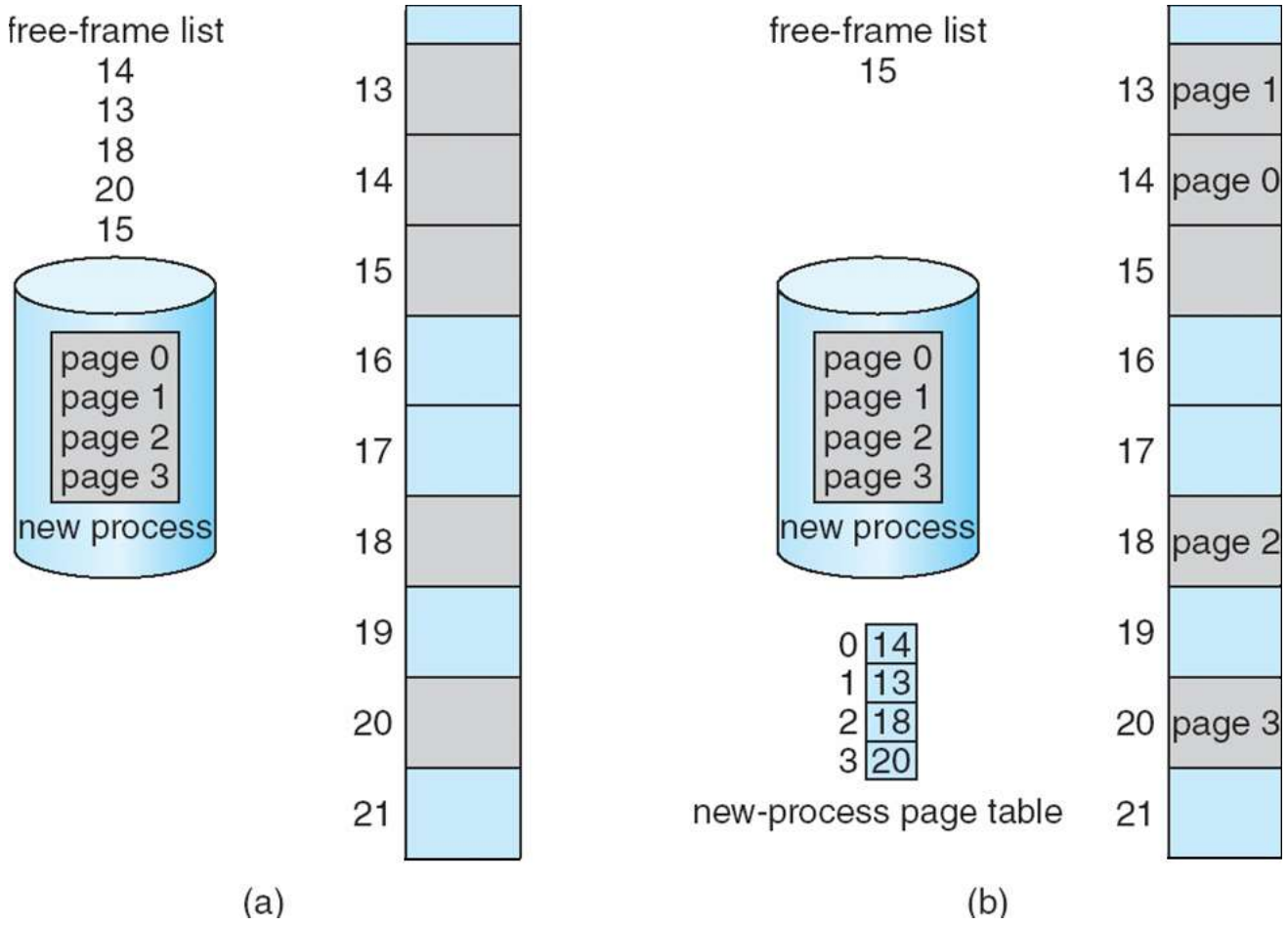
- Address generated by CPU is divided into:
  - **Page number (*p*)** – used as an index into a *page table* which contains base address of each page in physical memory
  - **Page offset (d)** – combined with base address to define the physical memory address that is sent to the memory unit



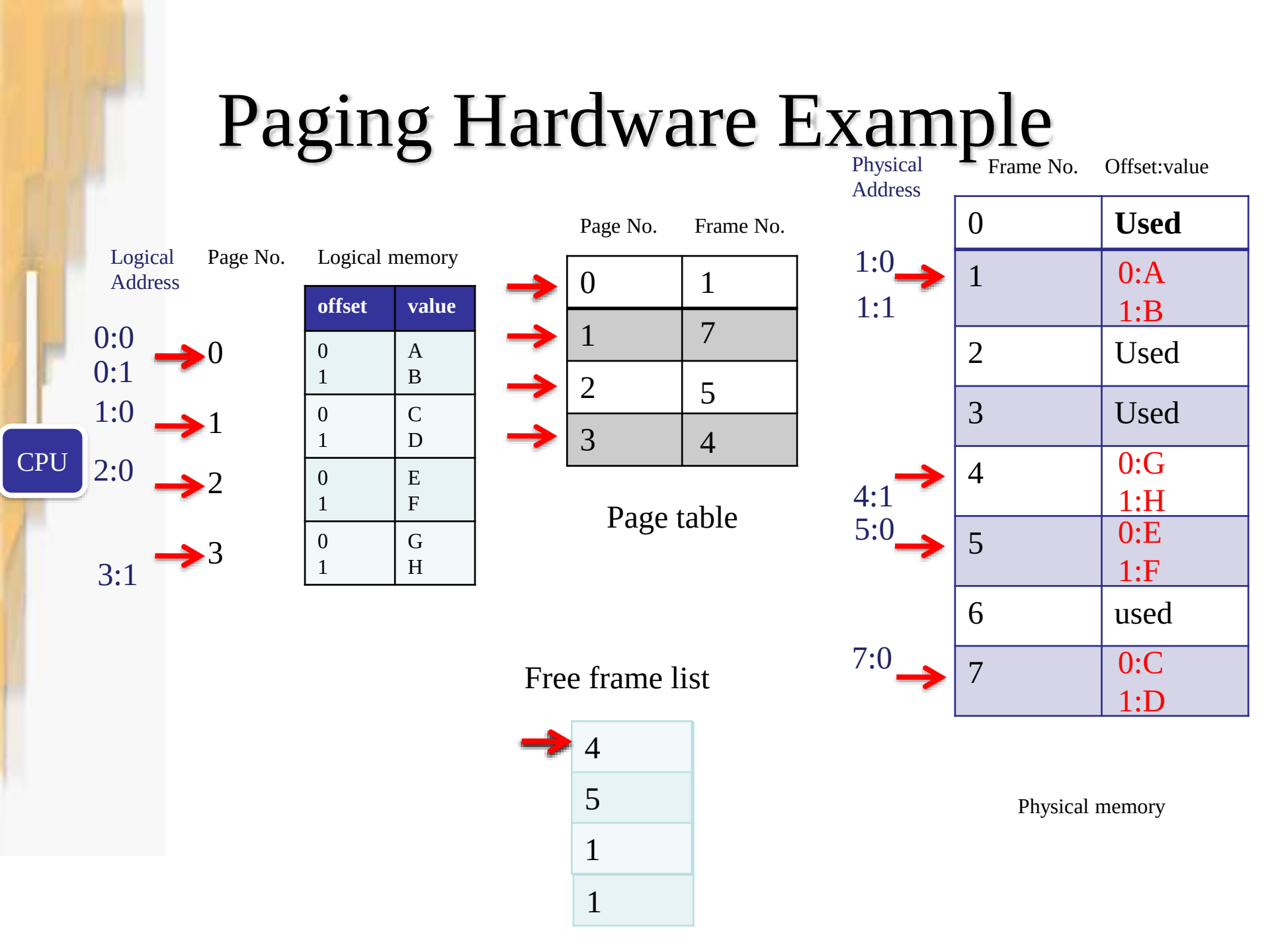
Paging Hardware



Free-frame list



Paging Hardware Example



Page in & page out

- When we want to run a process , we need to page in(place the page into physical memory).
- When the physical memory has no free-frame to page in pages, we need to do ”page out”.
- Page out means that you swap out the pages which are not needed anymore in physical memory.

Page in & page out(cont.)

- Example:
  - Look at Fig.pf, the physical Memory is out of free-frames.
  - Now, a new process need to page in, what should we do?

Fig.pf

0	ee
1	sa
2	zz
3	ww
4	xz
5	aa
6	gg
7	bb
8	cc

Physical memory

Page in & page out(cont.)

Solution:

We just page out the pages

which are not needed anymore

(use algorithm that likes LRU or Others)

Then we have free-frames to page in.

Fig.pf

0	ee(page out)
1	sa
2	zz(page out)
3	ww
4	xz(page out)
5	aa
6	gg
7	bb(page out)
8	cc

Physical memory

Exercise(90%)

Write a simulate paging program

- There are 2 processes(see next page).
- Paging them into physical memory by using free-frame list and page table.
- Please show
  - logical memory list + page table (60%)
  - physical memory list + free frame list (30%)
- Non-contiguous memory allocation(use random)

Random

- #include <stdlib.h>
- #include <time.h>

- srand(time(NULL));
- a = (rand%5)+1
  - a will get values in range 1~5。

Exercise(cont.)

A's Logical memory

0	ab
1	ef
2	cd

0	3
1	0
2	9

Page table A

B's Logical memory

0	rx
1	yy
2	zz

0	6
1	12
2	5

Page table B

Free frame list

4
7
8
13
1
2
11
10
14
15

Physical memory

0	ef
1	
2	
3	ab
4	
5	zz
6	rx
7	
8	
9	cd
10	
11	
12	yy
13	
14	
15	

Exercise(cont.)

```
oslab@oslab-VirtualBox:~/paging$ ./paging
process1's page_table is:
page    frame
0       9
1       8
2       0
process1's logical memory is:
page    data
0       rx
1       yy
2       zz
process0's page_table is:
page    frame
0       4
1       7
2       2
process0's logical memory is:
page    data
0       ab
1       ef
2       cd
```

Exercise(cont.)

```
physical memory is:
frame  offset data
0      0      z
0      1      z
1      0
1      1
2      0      c
2      1      d
3      0
3      1
4      0      a
4      1      b
5      0
5      1
6      0
6      1
7      0      e
7      1      f
8      0      y
8      1      y
9      0      r
9      1      x
10     0
10     1
11     0
11     1
12     0
12     1
13     0
13     1
14     0
14     1
15     0
15     1
free frame is:
1  3  5  6  10 11 12 13 14 15
```

Reference

- **Operating System Concepts *Tenth Edition***
  - ***Ch9 main memory***
    - https://www.os-book.com/OS10/slide-dir/PPTX-dir/ch9.pptx