Lecture 21

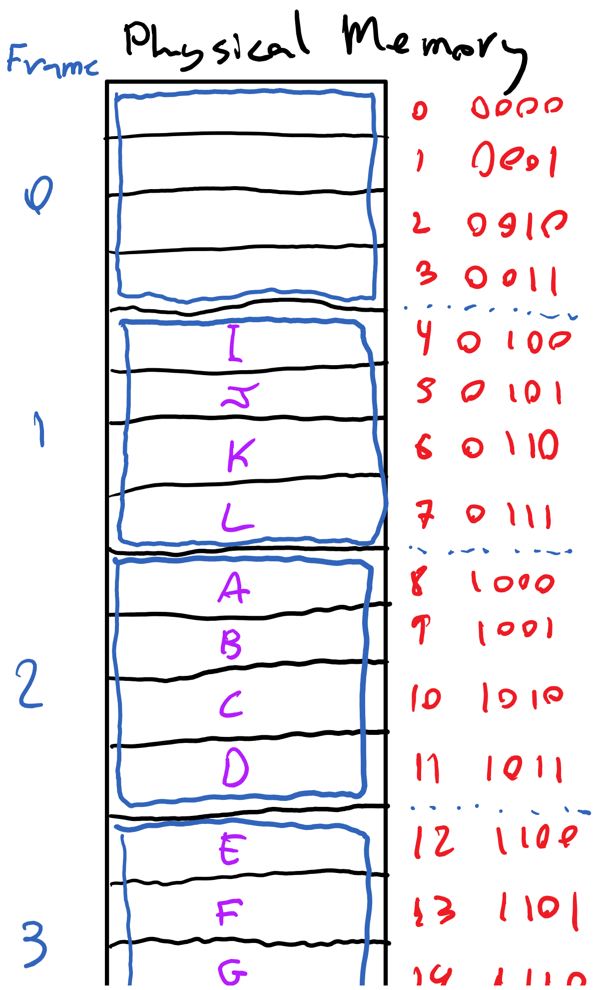
March 29, 2018 9:00 AM

Paging

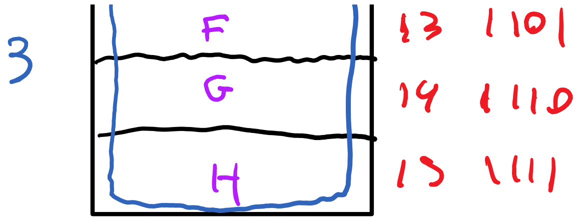
- Physical memory is divided into equal-sized frames
- Divide logical memory into equal size pages

Frame size = Page size

- To run a program w/ a size of n pages, we can do it if there are n free frames
- Load the n pages into the n frames
- Create a page table to translate page numbers to frame numbers



CPU



Page		Frame
0	→	2
2	→	1
1	→	3

Page Table

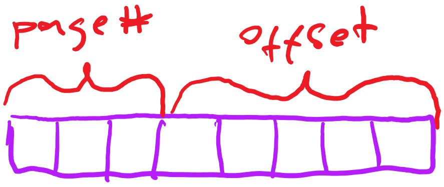
00	0	2	10
01	1	3	11
10	2	1	01

CPU
want

$\underbrace{01}_{\text{page number}}\underbrace{01}_{\text{offset}} \xrightarrow[\text{with frame \#}]{\text{Replace page \#}} 1101$

Consider 8-bit Addresses

$\Rightarrow 2^8 = 256$ byte address



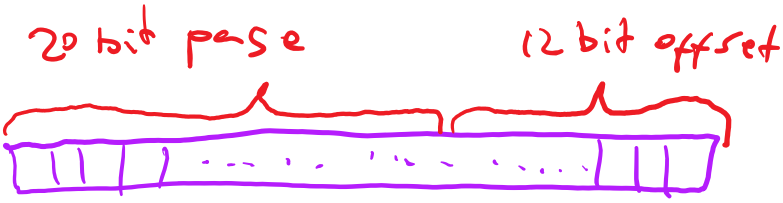
- How big is a page? Ans $2^5 = 32$ bytes
- How many pages are there? Ans $2^3 = 8$ pages

- How many entries are in the page table? Ans $2^3 = 8$
- How big is the page table? 8 * storage needed for a number

Consider a 32-bit Address space

- If the page specifies 10 bits, How many bits are in the offset? Ans 22
- How big is a frame? Ans 2^{32} bytes ~ 4,000,000
- How many pages are there? Ans $2^{10} = 1024$
- How many entries in page table? Ans $2^{10} = 1024$
- How big is the page table? Ans: Need more than 1 byte to count up to 1024. Need 2 bytes for each integer in the page table ⇒ 1024 * 2 bytes = 2 KB

Consider 32-bit Addresses



- How big is a page? Ans 2^{12} bytes = 4 KB

- How many pages are there? Ans $2^{20} \sim 1{,}000{,}000$
- How many entries are in the page table? Ans 2^{20}
- How big is the page table? $2^{20} \times 4$ bytes (size of int)

$= 4$ MB

But, Page table is in memory, which means every memory access requires 2 memory access $\Rightarrow$ System is slower

Partial Solution

- Associative Memory
  (a cache that is electronically near the CPU)

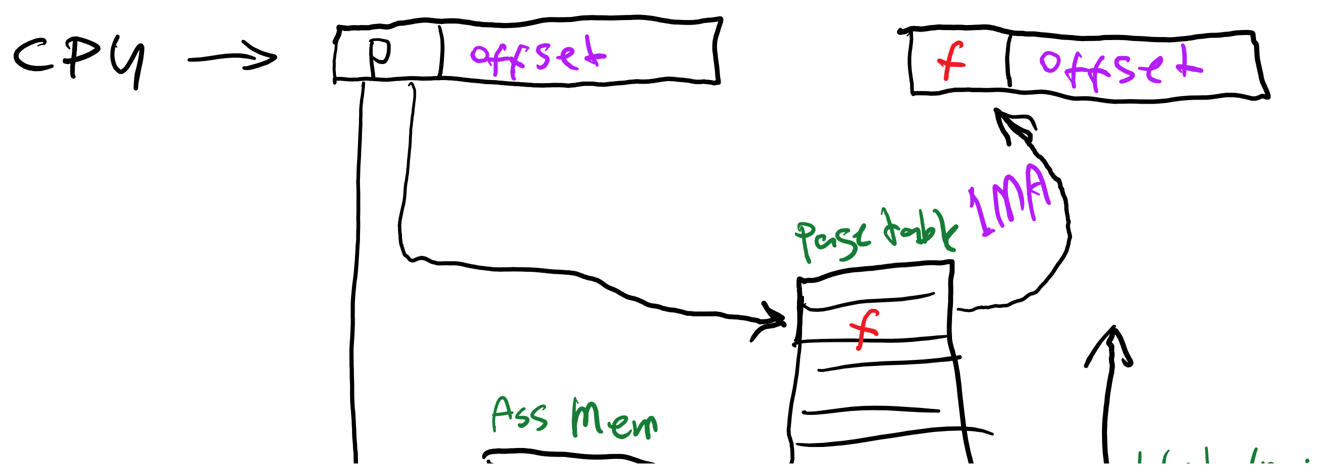
$\Rightarrow$ Effective Access Time (EAT)

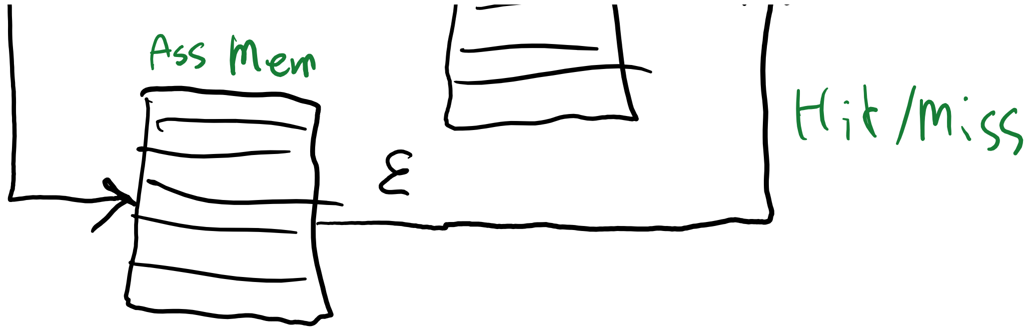
$$EAT = \alpha \overbrace{(1+\varepsilon)}^{\text{Hit}} + (1-\alpha) \overset{\text{miss}}{(2+\varepsilon)}$$

$(0 \le \alpha \le 1)$

$\alpha = 0.99 \Rightarrow$ 99% of the time there is a hit

1% of the time a miss




Ass Mem
ε
Hit/miss