Memory Management

- Physical Memory System

Page Table

- Word address

- N

- Page

- Memory

- Process

- Table

Date: 3/15
Fragmenation

Cause: Larger and cause the problem cannot support

If mean too small

Address: a smaller memory

- Stack/heap
- Dynamic memory (call)
- Program code/stack of stack
Masked memory

not used

- abnormal (occurred in brain but
  - external (hence there small)

- fragmentation
Sweeping: moving in a direction to divide work

1. impossible in absolute addressing code
2. bit slow in indirect addressing code

compilation → more or branch processes
Code, data, stack, heap

Let have to be contiguous
CPU

Ram

use of logical vs. physical address

Possible

1. Virtual memory

Address

Applications

Programming systems

Physical memory systems
< 1/2 base per process

minimal informed premedication

< no external premedication

Passing sheets
A process is allocated a page table when it is created by the OS. The page table is stored in the page frame. If a page fault occurs, the page is brought in from the disk.
Address translation
- done by mmu - mem mapping

- L2 cache - data path

- L1 cache - data path
Architecture
- word addressible
- 1000 words/page
20th century

\[
\frac{4x}{\text{area}} \quad \frac{\text{perimeter}}{100} = 4 \text{ kg/m}
\]

Park size

L size of public table

\(x\) width of public table

\(D\) long park table on concrete

\(\text{exceeds of public}\)
fetch addr

CPU fetch

memory

Get the main method from the page table in memory