Part I

Lecture 5

Keywords:
- Synchronization

Diagram:
- Students
- Keep waiting
- Use monster
- Prof(s)
- Each student and prof is thread
- Buffer = 2
- Wait line
- 1-10 min.
- 5 min
- Arrive
- Leave

Code:
```c
if line
    wait
else if not 2 yet
    wait
wait
```

Example:
```
condition C
```

1. if I am first
2. I amfirst = false
3. wait(c)
4. else
5. signal(c)

Diagram:
- Wait
- Wait(t)
- Signal
- Wait
Memory management
- allocation
- fragmentation.

1. Link all the free memory together into a free list.

   - Free list
   - 0 alloc(n)
   - First fit: search the free list to find the first one that has enough room.
   - Best fit: find the next suitable one.

   => So there will be external fragmentation.

2. malloc(m) will alloc enogh space + a header (store length).

3. Buddy allocation
   - Internal fragmentation

4. Paging
   - Fixed size pages

Virtual memory
- Page fault options
- When write try to a ro page
  - OS
  - Do not write
- When write try to a ro page
  - Terninate

- Allocate memory - Unix memory map
- Lazy page tables
- Demand page
- Copy on write
- Unix
- Fork
- Father copy one (child)
  - Cannot affect each other
Part II

3. address space

stack

BSS

code

D-E: demand allocate
B-C: demand allocate
A-B: map /a/file

PC = A fetch \rightarrow\text{fault}

Add space

[AB R/o]
[BC cow]
[DE cow]

Share memory (between processes)

Two processes sets of mappings to same space

So we need shared library (DLL)

So memory shortage will cause dead lock

Thrashing

4. Working set (Code反复: never used, it is not WS)

of the process

Active data ( " footprint"

is with respect time, so this is only for process

that can be defined

5. What is the data that need to be kept in memory?

Access frequency

Page \rightarrow 5 accesses/sec

less frequently

6. How is DLL work?

The memory map a library

Shared

to space address

Printed

When a process call a function, bompile it, not know where

Print() point is, but dynamic will reduce the add of print()
1. Virtual machines

VM providing this

VM

 emulator

regs

let it own for one instruction

update

usual

track user or supervisor mode.

how to implement it

Emulate all the supervisor code.

in time

hardware trap the S but stay in VM

VM

paravirtualization, efficient

TRAP: LOAD_CR3

OS

trap + emulated devices