

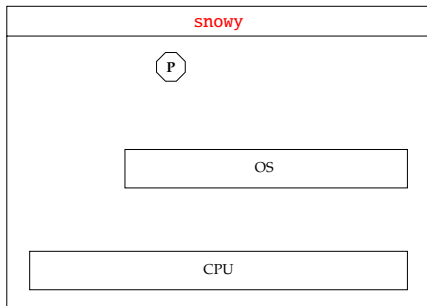
COMS20001

Concurrent Computing

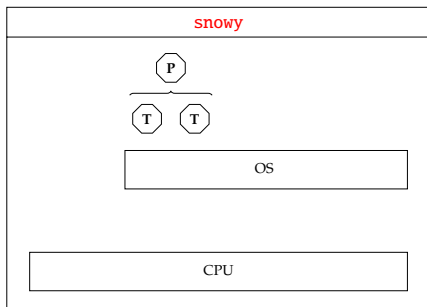
- ▶ The concept of concurrent computing exists at *various* scales



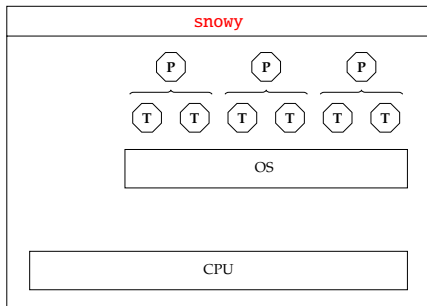
- ▶ The concept of concurrent computing exists at *various* scales



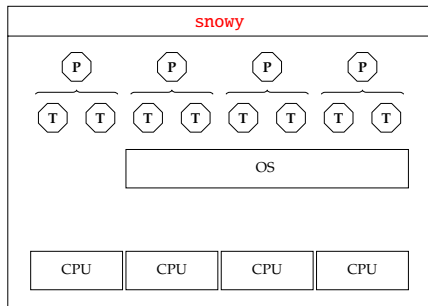
- ▶ The concept of concurrent computing exists at *various* scales



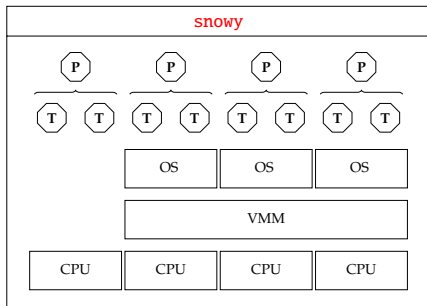
- The concept of concurrent computing exists at *various* scales



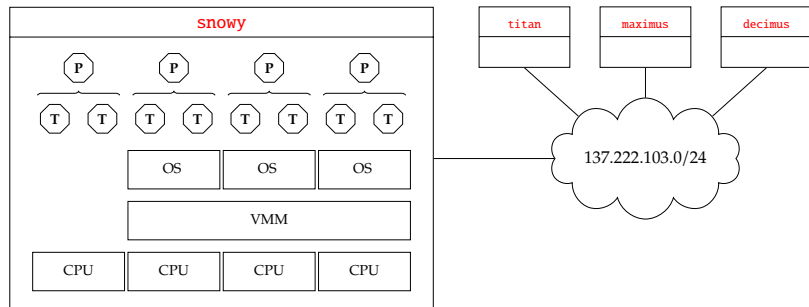
- ▶ The concept of concurrent computing exists at *various* scales



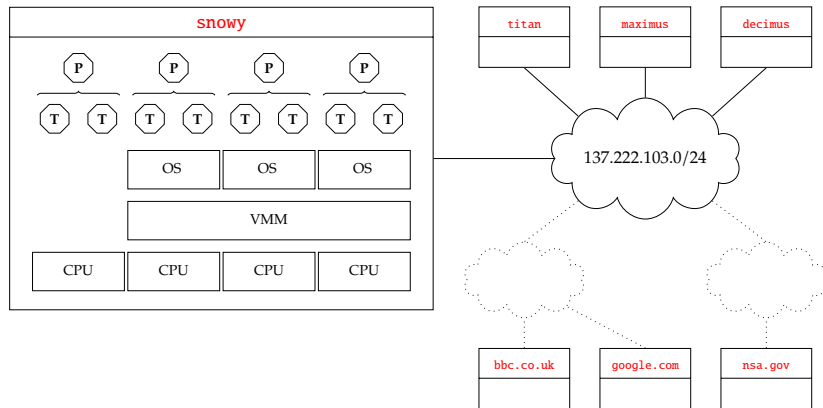
- ▶ The concept of concurrent computing exists at *various* scales



- The concept of concurrent computing exists at *various* scales



- The concept of concurrent computing exists at *various* scales



- ▶ **Content:** the latter half of COMS20001 concerns
 1. **operating systems**, and
 2. **computer networks**.
- ▶ **Delivery:** our approach to each topic is st.
 - ▶ we want to focus on principles to avoid dependency on specific techniques tied to specific (time-limited) technology, *but*
 - ▶ although one could view each topic abstractly, this conflicts with the practical, real-world nature of both topics, *so*
 - ▶ we'll use concrete examples and technical detail to illustrate the underlying principles, *and*
 - ▶ we'll take a bottom-up approach.

► Goal: understand

1. hardware/software interface

- interrupts, port- and memory-mapped I/O, DMA
- ARMv7-A, Cortex-A8

2. device management

- block devices, character devices, network devices device drivers

3. process management

- threads, processes, context switches, cooperative scheduling pre-emptive scheduling
- fork, exec, exit, etc.

4. memory management

- translation, protection, swapping, segmented memory, paged memory, demand paging
- PMSA, VMSA

5. file management (i.e., file systems)

- files, directories, meta-data, contiguous, linked and indexed block allocation
- ext2, ext3

while emphasising **general** concepts as applied in **specific** technologies (using a running example) ...

- ... *or*, in simple terms, understand how Linux works.

► **Goal:** understand

1. physical layer

- transmission, modulation, multiplexing, metrics

2. link layer

- addressing, framing, multiple access, switches
- 802.3 (Ethernet), 802.11 (WiFi)

3. internet layer

- addressing, fragmentation, forwarding, routing
- IP, ICMP, DHCP, ARP

4. transport layer

- connection management, ARQ, flow control
- UDP, TCP,

and then

5. application layer

- sockets API
- NAT, DNS

while emphasising **general** concepts as applied in **specific** technologies (using a running example) ...

- ... *or*, in simple terms, understand how the Internet works.

References

- [1] A. Silberschatz, P.B. Galvin, and G. Gagne.
Operating System Concepts.
Wiley, 9th edition, 2014.
- [2] W. Stallings.
Data and Computer Communications.
Pearson, 9th edition, 2010.
- [3] A.S. Tanenbaum and H. Bos.
Modern Operating Systems.
Pearson, 4th edition, 2015.