

synchronization internals① Oct 25th Friday

Lab exam 2

8.30am - 11am

- synchronization
& network programming

② Endsem

- 21st Nov.

- time: ??

③ Project

- 24/25/26th Nov.

- demo + report

④ 4a } today
4b }4c - 30th Oct.

① # lock / spinlock

- atomic instructions

- enable/disable

getlock: mov R0, 1

preemption
interrupts

TSL LOCK, R0

cmp R0, 0

jnz getlock.

lock variable

temp = new;
old = new;
return temp.

② mutex

lock 0/1
yield cpu or lock

③ condition variable

wakeup on
lock available
& check
status

④ semaphores

generalization
of condition for sync.

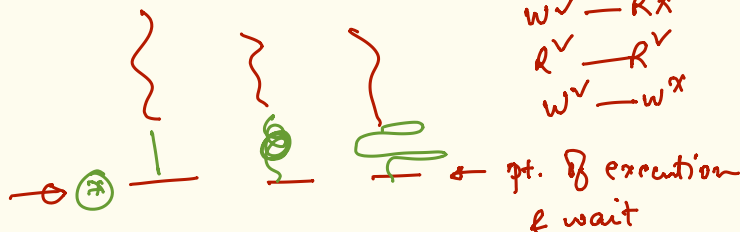
↳ lock - 0/1

⑤ reader-writer lock

similar to mutex

counting
semaphore

⑥ barrier

R^v - W^x
W^v - R^x
R^v - R^v
W^v - W^xpt. of execution
& wait
based on state of
execution
of other threads.⑧ # how a mutex is implemented?

struct mutex {

spinlock s;

int lock; - 0/1

int id;

}

struct mutex m;

```

mutex_lock (mutex *m) {
    spinlock (m->s);
    while if (m->lock == 1) {
        sleep (m->id, m->s);
    }
    m->lock = 1;
    spinunlock (m->s);
}

```

- enters w/
lock taken

- returns w/
lock taken

```

mutex_unlock (mutex *m) {
    spinlock (m->s);
    m->lock = 0;
    wakeup (m->id);
    spinunlock (m->s);
}

```

id spinlock

```

① sleep (m->id, m->s) {
    p = current proc();
    p->status = WAITING;
    (BLOCKED);
    p->lockid = m->id;
    spinunlock (m->s);
}

```

// get PCB
of
calling
thread

```

wakeup (m->id) {
    for all process p
        if (p->lockid == m->id)
            p->status = READY;
}

```

wakeup
next

yield / sched — invokes
schedules

```

    spinlock (m->s);
}

```

networked applications

- application layer protocols.
 - (specifications to enable communication semantics.)
- set of rules
 - Keywords
 - ordering
 - interpretation
- smtp, ftp, pop, imap, http, https, dns, ssh, scp, nfs, telnet

① http: hyper text transfer protocol

- web browsers (client) to communicate w/ web servers
- data: html, audio, video, images...
- (usually) via TCP

Q what happens when an URL is requested?
www.cse.iitb.ac.in

- ① get IP of www.cse.iitb.ac.in → DNS
- ② connect to IP:80 — http port
443 — https
- ③ send HTTP Request — format
- ④ server receives request — HTTP interpretation
- ⑤ server generate HTTP response — format
- ⑥ client receives & decodes response
- ⑦ renders / displays page — html page

HTTP Request

GET

↓
fetch
a resource

POST

↓
send data
to a
resource

HTTP Response

- status code 404
200
- data/content

Q) http 1  ~~one~~ one ~~set~~ TCP connection per object.

http 1.1

http 2

http 3