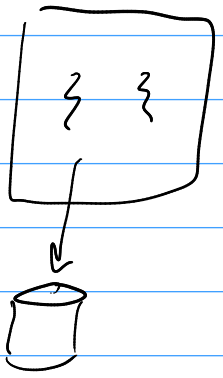
CS 3214 "multithreading"

Dan Williams Joined Fall '21
IBM Research 10 years
OS researcher

multi threading "thread of execution"

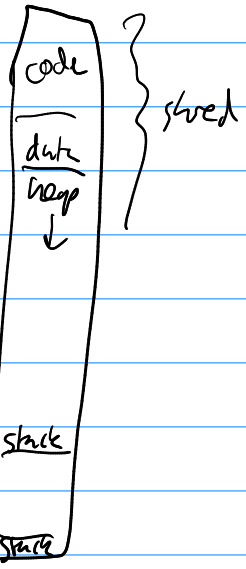


process

- parallel computation
- overlapping I/O/computation

- web server
- UI

addrs



threads share	don't share
- code	- stack
- data	- CPU regs
- heap	
- open fds	

pthread_create
"kernel supported"
SHARE_MM

What can go wrong when you share so much!

```
global_shared++;
```

not atomic

```
mov  mem  %r1
inc  %r1
mov  %r1  mem
```

```
mov m %r1
inc %r1
mov %r1 m
```

race condition

How do we solve it?

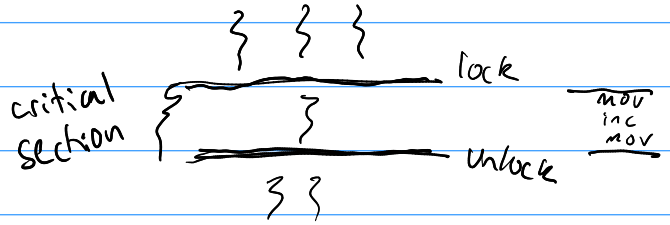
- locking
- avoid sharing → 2 copies of variable update independently

mgmt overhead	realistic?
	- per CPU ready queues
	- region-based malloc

Locking

mutual exclusion "lock"

mutex



- only 1 thread enters critical section at a time
- shared data only updated in critical section

mutexes are for data not code

race condition detection tools: helgrind

valgrind --tool= helgrind

what about library calls

int seed;

srand() { seed=x; }

rand() { ... } involves seed

rand_r

"reentrant"