Concurrency his out town -lowed atting wood at hone to Thread: Share an address space typeratri monitis-In ways teeks that she so -30 mosks en Multi-threaded program structure: 1 Producer/Consumer : mainimmotob-nold [®] Lipeline * Need mutual-exclusion for enilion sections (m)Danisa masour stack - stones temp var. so individual for every thread. ... moriton, well, condition ;

heap

Code

Acremics maked
(I) Correctness
— mutex
— progress (deadlack - free)
— bounded (stanvotion - free)

-common for all threads of the same process

(n) fainness

	0.000
	Concurrency
	* need to work with kennel-level through.
	* Interrupt >timen interrupt
	Syscall OS-43 Tolk help Florit interrupt
	Non-determinism:
	*Need mutual-exclusion for critical sections
	shared resource
	low-level primitives: load, stone, disable interrupt, test & so handly prove not laubivioni as any part and to vote high-level " monitor, lock, condition var., semapho
Ш	deocks: 103 of to abound 110 not nommos - good
	(1) Connectness
	-mutex
	— progress (deadlock-free)
	-bounded (stanuation-free)
(DFainness
(1	m Performance

Implementing lock: Whinteroupt a sprantone simoth : MININ

Disaduantage:

] (lovern Ini subbox Ini) prinx Ani (1) निष्ठा other process रियाता वाष्ट्रित छन्। interrupt call कदाक भारति

ati

: lovasn = abbox

(11) only works on uniprocessors.

;610 navter

m process can keep control of CPU for arbitrary length.

Implementing tock: Whad + stone water ton lie ralubordos x.

lock = 0;

simoto

acquire ()

ODoesn't ensure mutex. -> lock check & set not atomic.

while (bck = = 1)

Lock implementation with xcha

You'd init (bock t * lock) {

bock -> Flag = 0;

lock=1;

void acquire (nock t *Lock) }

white (xehg (Lock-flog, 1) ==1)

void release (bok_t *lock) { 10 = pal 7 - toal

```
XCHG: Atomic exchange on test & set w: stool gritnems/gml
                                                 : agotravboard
   int xchg (int *addr, int newval) {
                            White Will somed watto lale 1)
          int old = *addr;
          *addn = newval;
                              ecessonation no extense who D.
          return old;
      }, Algar prontidios not uso to lontinos gost nos azisona (1)
    *scheduler will not interrupt until xong is executed completely
                                                     10ck = 0;
ODDOESN't ensure mutex. - Local che
                                        atomic
                                                     acquire()
             . simofa for toa &
    Lock implementation with xchg
                                            while (work == 1)
    void init (lock_t *lock) {
                                                bock=1;
   void acquire (lock-t *lock) {
      while (xchg (lock-)flag, 1) ==1)
  void release (lock_t *lock) {
       lock->Plag=0;
```

xchg (** lock → flag, 1) ⇒ returns prev lock value, and assigns

Flag=1.

(AWATER 1 pass area, artail lock acquire area

Flag.

Disadvantange:

Clock();

: (INDAL A

B unlack(): 2

1 Unfair

lock in lock

A B

O 20 40

B while (2)

→ B while (1) hop-4 ल्यांच्याय याकावा (5 pinlock)
CPU access कन्द्राइ किन्सू कार्यन्त्र कार्य कियुष्टे स्वाह ती।

A time slice

gots ram lock acquire

बहुवा कार्य दिवादिय beocess त्यकाप

lock only at

* uniprocessor zom while (1) check rops mo nto, multiprocessor uz

		And the state of t		de dool &
3/1	Fairness: Ticket 1	tocky marker	to Bord	, J.N. 1/ Com 45,
	Trokeb	TunA	- LIHOD	
alli	A lock(); DOO	E LOVE SOUND	. Etj	
	Block();	1	1214	
	C'lock();	2		: sprustr
	A whock();	L		Anal .
	A lock();	3		
	B unlock();	2		8 4
	(Hooking a) is partice igna	THE R-good (t)	olidu 8	20
6 g	Egal for Fight of	०८५८ यन्त्रहाक विक्	0.000	ime slice
			equipe	a wow me
	ng 100000 e		प्रकार ३६०० वर्ष	that em
				its alle
2 90	scassongithum of the	oth Job with	b (c) winder	mys 90
_		% 1+		

D. A : 9/donnul .

- *Ticket lock is a kind of spinlock. Starvation free but wastes CPU * princess states: Running, Runnable, Waiting.
- * Slide-4 context switching time = 0 Sat apply i Now, suppose context switching time = 2 sec. Calculate scheduling time.
- * Forth Running process yield to state change 21 Ready/ Runnable state-4 2771

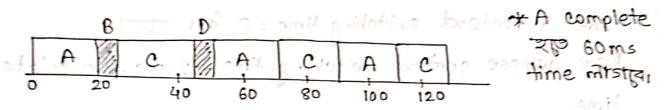
Spinlock Performance:

Without yield: 10 (thread * time_slice) With yield: 0 (thread * context switch)

Apple alles Among housing A house and (the show

B, O 43 off Guene to append the & A * दिखे process इति wock पत्र उत्ता नियं लाष्ट्र एक एक एक एक पिनरे ता, (Minimizing content switch cost) नवून कोंग state assign করব "Waiting" instead of "Runnable".

A, B, D contend for lock, C is not. * * process states: Running, Runnable, Waiting.



IFTER E-state gldnanite

Spinlack Performance:

* 4 of process ज्याकेत बाह्न Run बाह्मीत अन्न — innu

Waiting: B,D

Runnable: A, C

njezo okota nie nga (taon dotiwa beathor pritioni agai "pritioni" agai

Lock implementation: Block when waiting typedef struct {
 bool lock = false;

bool guard = false; Il queue access atomic assts only missign queue — t quara; Il stated lock as Ens & process

} LockT depended, other queue

* queue access needs to be atomic.

* yield controls the damage. Can't avoid rithetnemelomi Hood!

Block state = sleeping state : salof :

bool book - folse;

1) Why guard needs to be atomic;? - to ensure that there's no roace condition for the queque

® why okay to spin on quard? → Interest time and of spin and time waste all, processes will be in waiting state.

else {
 l → lock = trove;
 l → guard = false;

Dondering important

Ordering important

Outst guard set rooff CPU

THEF THEM reace cond. arroive

ANDER THEM REACE

ANDER THEM REACE

ANDER THEM REACE

ANDER THEM REACE

ANDER THE REACE

Why not set lock=false when unpack? → oterm white short process 43 only be acquire lock() call not mister

- ® Is there a nace condition? → parkl) 43 cutst unparkl)
- * set park() ensures that 1-guard = false up my park marg

Condition Variables

Concurrency objective >7 Mutex
Ordering

greep "Name" | we -l 1/Doesn't ensure outst greep nun

* thread 43 join == process 43 waits tot gray of this

Condition Variable: 5-demoits : acitationary and airt

queue of waiting throeads

D wait (cv, ...) ⇒ wait (cond-t *cv, mutex-t *lock)

@ signal (cv; ...) ⇒ signal (cond_t * cv)

* child execute par only exit() call only zombie state a my, povent-43 rotor com zomes (wait() call) child process terminate श्रवा > last func. call

Join implementation: Attempt-1

- *Parent-az wait func. az tooz lock input tha 2100 parent bock निद्ध तो खुडागूर शामा wait func. पत्र आद्धी 05 unlock कहा विद्या
- * Child unlack उन्हार जारिस श्वेणण एपि क्रिके के कि कि कि कि कि lock already acquired -> GONTA powert wont JUSTAI MENTS CHILD unlock rolls exit rolls, then OS what a powent too lock into powert पर्वार याकि काक कर्मा

*Need to keep state in addition to CV's him to bound & Condition Variable:

Join Implementation: Attempt-2

child void thread_exit() { => Swap TOTTE deadlock - 4 FGM

would done to by vox f-base) time the original signal and and and CEU fally barrent also

Cond-signal (&c); langing done == 0, from vary

etote sidmos go llos (1tino wait riaga fang child want

2230arg blids (los () tion) Engdone of Total exit sorger 21181

los only tool Ale sta

L - Jamest A: noito framalami niol

Powert - 27 wait func A7 tows work input the 216 por Just Apollow 20 हिंगड़ हर David How महिर मुंबाइ के मुक्ती dow way by hurung graff find larger type first would bling to port opened acquired - sour powers work waster sales unlack as a exit apply then OS related powert for lack of Die with all the sold trung

	Pandunga I American
	Producer/Consumer Problem:
	हिं सिंगितर रहे। व्या
	C1. C2 C3
	P. P
	P, P2 Py P5 Pb P, P2 P3 Javo an atala gass D
	blod sow sting 13 27 fact agout o
l	
	1 producer, 2 consumer: mont endow broadt revened to
l	, = 60.000.000
	P:
	P1 P2 P4 P5 P6 P1 P2 P3
	C1: C1 C2 C3
	c_2 : c_1 c_2 c_3
	C2.
l	
	* BER wait, signal same -> consumer onthe consumer for
	व्याभा प्रिक आर्थ।
	The All a
	Two CVs:
	$C1: C_1 C_2 C_3$
	C2:
	P: P2 P4 P5 P6
-	* checking it if cond. At Taby while () Tabo 2501
	V

* How to ensure fainness in this produce consumer problem?

O Keep state in OV's

1 Always do wait signal with lock held

1 Whenever thread wakes from waiting, necheck state

Pi P2 P4 P5 P6 P1 P2 P3 Semaphones

* CER wait, signal same -> consumer

ा हात का शा खा है।

01: 0,02

Two City.

ور و وهم و و

9, 9, 9, 9, 9

+ checking it it cond. in the while is the prisoned

```
: sanodynmas ar VO dtie not.
                Semaphones
 sem_init()
                                  3 Unioj hosmit blow
 Wait on Test -> decrements value (22) tipes _ 1933
              if $>,0 priocess not sleeping
                 17 <0 " sleepingtice-bound blow
                                sem-post (ks);
      $ 5em-init(2);
        P, wait (); 111
         P2 wait (); 110
                                      ; (0.21) tini_mes
         Ps wait () ; condlin 1 process sleepingbrisged in secret
Signal on Post -> increments value maldon? anuono ) rocubina?
Birary Semaphone (Lock):
                              * Showed resource/outfor 1st
  init() { sem = 212} slider.
                                               agoubar?
acquire() { semi-wait (& lock: t); } dyfamoss) tion-max
                                               ? (1) Slinker
release() { sem_post (&lock_t); } (suffered) 1117
                              sem-post (sefullbuffer);
```

```
Join with CV vs Semaphones:
      vold thread-join () {
             sem_wait (&s); submissionersober toot as tick
                 grippola for sasoning 0 < 3 7:
       void thread-exit()) file " 0> 91
              sem-post (&s);
                                ; (c) fini-mas 4
                                ETT : (1 How A
     sem_init(&5,0);
                               011 : () flow &
   * TO STAT Dependency GOSTAT Semaphones.
Som Producer/Consume Problem: @ Capacine
                                Charge Semaphone (Lock):
   * Shored resource/buffer 1st
                                Consumer
                                while (1) {= mai } (1 tini.
    Producero
     while (1) {
      sem-wait (& empty buffer); sem-wait (& full buffer);
       Fill (&buffer); {: (+-x) old) de (&buffer);
                                  sem-post (&emptybuffor);
       sem-post (&fullbuffer);
```

emptybuffer initialize = 1 monumen pool solitions elithium to

* semaphone पत्र अयोक्षेत्र atomic शता : Insmaring?

1-8m , 1-8m

Frich consumer must grab unique filled element

* Shoved buffer with N elements

emptybuffen initialize = N

fill buffer "

N=5

	E(5)	F(0)
P()	4	1
P()	3	2
c()	4	1
<u>e()</u>	5	0
C(1)	5	-1

	1 2					
6	em 3		1 0	r = asiloi	tini rulluc	enptys
	*MWtip	le producer	, consumer	3	00	117.57
	* Shave	d buffer wi	th N eler	nents	wiff	wd Ilwit
				to garage	FR GOOD	* semay
	1 Each	consumer mi	ust groab	unique fill	ed element	ŧ
	_	producer	••	" em	pty "	
			ments	ele U Alfu	nother l	* Share
	× my-i,	my-j		iolize = N		
		(20 1)		U = 32 KIRI	31111 3137120	2 1167
			(0 = "	J197	7 100 100
						7 = M
					,	
			·	(0)3	(E)	
				1	N	()9
				S	3	()9
						())
				1	P	1
				0	2	(19

lock-acquired. => Deadlock

sem

emotypuffer initialize = * Multiple producer, consumer * Shoved buffer with N elements Requirement: IFE simoth gates Fr anodysmes * 1 Each consumer must great unique filled element * Shoved buffer with N elements \emptyset producer n u * my-i, my-j shared value. U = osilaitiai astfudytama r977 Not 1117 *producer/consumer-uz same lock. Consumer #1 2=11 sem_wait (&mutex); E(5) | F(0) sem-wait (&fullbuffer); > country context switch 25m consumer book अन् भूकाम यात्र। my-j = findfull (& buffer); buognose बार्बबार करित सिल्ला

Deadlock us livelock Important for exam

Deadlock out of syllabus.

- * Livelock: Both troying to access same shared resource but failing. (Dining philosopher)
- * Deadlock: One waiting for another to finish using the resource.
- * producer/consumer outmint lock friends react 210 ATGI.
- *अर्थियात्र my-i, my-j access बाह्मीय कत्य Lock निया -

9909qomas —

- Traden worker

READER/WRITER LOCK:

- *multiple produces can work parallely in producer/consumer problem Here, only one writer can write at a time.
- * multiple consumer cannot work parallely in p/c problm. Here, multiple reader can read at a time.
- * What if we need to proionitize readen/writen?