Producen-consumers:

int count =0 int count =0 that court =0 void Consumen (void) void producer (void) 5 int Hem C; 9 int Hemp; While (true) ? while (true). > while (count ==0); 2 > Produce - ikm (ikmp) Hem c = Bufferlant; - > while (count = = h); out = (out +1) mad n; Buffen [in] = Hem P; , count = count -1; in=(in+1) mod n; > | process Hem (Henc); count = count +1; load Rp mcaunt load Rc, moount! INCR RP! BECR Rc; Stone m [count], Rp; Store m Count R. ;}

- Water (Table) that continued

Charlet feet forming

The figure to a love !

Crifical Section It is part of the progream where shareel resources are accessed by vanious processes. main Who man or a mile or took non cnitical months are entry section Chitical section exit Section 4 17 0000 000 Synchronization mechanism a section. Principal Mutual Section - Cs single process execute 20) D Progress secondary (3) Bounded wait

(4) No assumption rector related to H/w

Speed.

Maitual Section > Test P. process execute 200110, 018 P2 Process progress: Py monts to execute the critical section fort pe stop the P1. Total progress न्थेल्ड मा Bounded mart: P1 process execute CS. P2 maits. After some den times P2 can process. किर असीम 62 या 6' ए अर्थ माठ 6 था एक किंडण भाल ता, No assumption: Handware 18900 1000 Bounded त्रिकक साइक सा ENOTED TO MILLIAM DE MOINS EL CELTON

Continue of commy show 200 - Nothers low hold (1) for this of the bound (1) for this of the bound (1) for this of the bound (1) for the bo

. 1, 2,90

critical section solution using Lock

do of a comine lock

cs

release lock

entry while (lock == 1); code lock = 1 Chitical Section

[lock = 0]

exit cale I

, so go to to to to to

ins for Compasse to

(a do the son!) a " later

-) multiple process can execute

Lock =
$$0 \rightarrow Vacant$$

 $1 \rightarrow full$

-) execute in usen mode
- -> Multiphrocess solution
- -> No mutual exclusion gnatientere. (Afth Preemption 20 entry section o 01200 mutual exclusion 20

Synchnonization Hav

- Protecting critical regions via locks.
- -) liter provide sep special atomic Handware instruction Atomic = non-intercouptible

Cs soldion using locks I grant without white 100 40 200 acquire lock snight Releage lock |
3 while (TRUE); Jo They Mutex lock -> Protect as by acquine() a lock then releasel) the lock. -) acquire() and release () must be atomic. This solution requires busy waiting. - lock called a spinlock. rcelease () { acquine () { available = True; while (!available); available = False; shops of it will to gione via locks. Harton Director Since of the Strange interior non- "interangiliele

Birary Semaphone 175 W. M. 19520 D. P. for with 18 So 1 Depend on (Hear (Hen p) signal (S) { waits (s) 2 while (\$ <=0); S = S+(1) must : K 6 S = S-1; J was Carparille & 1 it forms (town) . at codo. 20 4 entry section; ental Section: exit section; reminder seation. while (true); sulow 2 DADO IL =: low test 200 , In 42 notify with Did I vay to me . (ett mille).

Counting Senaphone S: -00 to +00 (No Fraction Value) Down (Semaphone S) { S Value = S value # - 1; 4 (5 value < 0){ put process (PCB) in Suspended list Sleep(); else ? return (Catalia) longita 1 3 ING BURYO BOSH (Cartura) tion : -- trous be sol (0 = + true boust) (5 Citorial levision : " stand Longi's

Up (semaphore S) S value = S value + 1; 4 (5 value < 0)2 Select a process form suspended lot Drake up (); (mu) 10 m2

Solution of P-C caring serraphono: (Producen-Consumer) consumen (item c) producere (Hem p) down (Juli) D down (empty); down (s); 2) down (s); Hen c = Duffen but. Buffen [In] = Hem p; out = (aut +1) mod n; In = (In+1) mod n; Town (2); 3) up (to 5) down (empty); 9) Up (Jul) Dut=0 | Jull empty, S N, In out Variables In =3 counting semaphore To full -> no of filled slots.

copend on a empty of it is empty is. Binary semaphone - [5=1] (DATE S value of Copy depend one preemption 27 AT, -OTE Mactual exclution IT).

Readens and writers Problem Continue Self, part o (1) writen - wirthen : clash (2) whiten - Reader : clash (3) readon-treaden: No clash Writ= 1/ mutex = 1, tread count = 0 D(depend on) Readen Whiter wait (mutex); wait (weet); read court ++; Write Operation; of (read count = = 1) { signal (wret); wart (wret); · Signal (mutex); Read Operation wait (mutes); treat court --; if (read count = = 0) {

to signal (to ret);

signal (muter);

Diving - Philosophers Standard System. void philosophers (void) P.) while (thus) Thinking (): Stalle Fataker - font (1), take fork (C+1/6N) EAT (); Philosopens 5 Fonks/ Put Jony(i); Put took (1+1)84) Wait (take- fony (Si)) Wait (take - Honk (S(H)Kaly) Signal (Put-tony (Sitt)).
Signal put-tony (Sitt) modifi Gass 5 Senaphone Si), So S, S2 S3 Sq

Pesudo code: TENTER OCCUP FOR TRIM void philosophen (void) while true file of the said with Thinking (); The thouse the start in entry Wait (take forth (Si)); To modern); TEAT (-) 15 mi promo Vool brown exit [signal (put_tonly (si));
[signal (put_tonly (si)); θ_o ; \leq_o \leq_1 P, : S, S2

 P_0 : S_0 S_1 P_1 : S_1 S_2 P_2 : S_2 S_3 P_3 : S_3 S_4 P_4 : S_4 S_6

Here some time two philosopher can execute. But those have to independent.

@ Deadlock can occoun.

Deadlock occur 20 HUT left. Fork WM 20 the Right tony take more atto at , the way Colat proces preemption sitis Total Deadlock cheate 20 mont controle EATU orion propertions ((:2) shot) they Alth Last Philosophen reverse of M 291 27 Olser Deadlock Prevent of A THE (((12) shop try) longis [
((Longis) shop - try) bright. P2: 52 53 @Hum some there two philosophen an execute. P. H. Hove boye to Independent. (Benglice Can Becometi

Boursed-Batten Problem

Heed three senaphone. m (mutex) empty Jull

Producer (1) slitter

do 2

wait (empty);

wait (mutes);

I add data to buffer signal (mutes); signal (tull);

I While (True)

ô	Pı	ful
12	P2	mu
3	(3	en
		7

11=3tex = 1 opy = 2000 C- audit govern = pol 1

Consumer

Milalas in antog

do { () = [] , s

mait (full); wait (mutes)

1 0 remove data from Butter

signal (mutes);

signal (empty);

I while (True)

Marine Visitory - voust

Two processes -) Pi and Pitan vo mit

While (1)

4

Flag(0)=T)

Turn = 1

while (Tunn == 1 & &

Flag[1] == T);

Critical section;

Flag [0] = F;

Hot Haw Paron) W

while (1) assubant

4 Flag o[1]=T;

Tunn = 0;

while (Tunn = = 0 &&

nother the state (Flagtor= T)

chitical section;

Flag[1] = F

Swile (True)

3 allufo

Tunn = booleau vaniable

Flag = antry, value -> troue, False