Digital Assignment

Name: G. N: Hish
Rog No: 198080012
Course: Operating
System
Date: 17/2/2020

flag[2]; flag [i]=1; while (flag [i]) of (from == i) ¿ flag [i]=0; while (twn= =j); Hog[i] = = 1; J two =j; flag[i]=0;

flag [2]; int twon= 3 flog [v]=1 while (flag [i]) & if (twon == i) flag[j]=0; while(twon==i); flog [j]=1; trun = ) ; Hag [i] J=0;

solution des critical section:

- \*) Mutual exclusion, \*) progress,
- \*) Boundad waiting

if Mutual Exclusion:

(2)

In the above process Pogip, if the process of 'Po' is executing in its critical section then no other process' p, I can be execution C.s. Like wise, if P, is executing In its critical section, then po can't be executing in C.s.

iit Progress.

Case ::-

If 'Po'is showing intrest it is executed in kines when 'P.' is not showing interest to execute

Po	P,
, ,	×
1	· ×
/	×
5	
	1.

Case ii:-

Executed n fines when or rest it is not showing intrest to is not showing intrest to

行行

Bounded waiting

no. of times that 'Po' is allowed to enter its critical section after Pi'has

made a request to enter its Critical section and before that request is granted:

If the 3 condition satisfies the solution for exitical section.

of fet.

Po

while (true)

Estate ij=interested;

turn=i;

while (State ij=interested & turn==i);

zeco: Eical Section>

State I ij= not interested;

ze code coulside critical section>

J

\* In the above Goods it will enter into while loop. Then state [i] will be essigned as interested and turn asi.

\*) After that using catile loopit will check the condition.

While (state [j] = : interested the town = )

det bothe one kon the oppisteres ofterwise false.

\*) The condition fails and enter late Withical section

Depteration so bargices si, net interested

the above code satisfies all the conditions for critical section, as follows

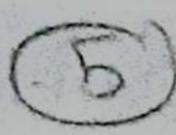
Mutual Exclusion)

The process Po is executing in its Exitical section, then no other process can be execute in its critical section.

Progress

to excule then the other process will not show interest to excul, and Vice Verso.

Bounded Maiting:



times that other process is allowed to enter its critical soction after a Process has made to request to enter its critical section and before that request is granted

3) Process Pi:

-do

\$flagsij=TRUE;

teun=i;

tohile (flagsjj 88 teun==j);

< Critical Section>

flagfi]= FAISe;

Fremainder sections
3 while (True);

Northing is wrong in the following variation of peterson's algorithm

Since it satisfies all the 3' condition for Gritical section, as follows.

If the Phoose pi is oxecuting in its raitical section. Then no other Proless can be executing in their Critical section.

## De progress:

If no process is under execution its Coitical section and these allow the process that evish to enter to its critical section, then the selection of the processes that will enter into the critical section and rest count to Post poned indefinitely.

Bounded waiting:

times that other processes are allowed to other their exitical section after a process has mad a reserved to enter its Critical Section and before that request is granted.