gradeup

Process Management Part- 4

Content :-
1. Process Synchronization
2. Critical Section Definition
3. Requirement of Critical section solution
4. Critical section Algorithms

Process Synchronization :-It is a mechanization to ensure a systematic sharing of resources amongst concurrent processes.

Critical Section :-
It refers to the code segment of a process/job whereby it accesses a shared resource.

Requirement of Critical Section Solution:- An ideal critical section sution meet the following three requirements :-

a. Mutual Exclusion :-At a time **,at most one** of the cooperating processes should be executed in its critical section.

b. Progress :-whenever none of the cooperating processes is executing in critical section and of the some of the cooperating process is waiting to enter their critical section, the none of the waiting processes should immediately enable to enter in the critical section . The decision **must not be delayed indefinitely**. This is the termed as requirement to progress , which must be under all the possible conditions.

c. Bounded waiting :- These must be exist a finite upper bound on the **number of times that other cooperating processes can enter their critical section** , after process P_1 has requested to enter into its critical section and before the request is granted . Normally upper bound is one.

Critical section solutions :-

The general structure of critical section solution will be :-

```
Do {
        Entry

        Critical Section

        Exit Section

        Remainder Section
}while(1);
```

Entry Section :-This refers to the code segment of process that is executed when the process indents to enter in its critical section.
Critical Section :-This is the code segment, wherein the process will access a shared resource. At a time , only one of the cooperating processes should be in its critical section. So, when the process is executing in the critical section, it is enjoying an exclusive access to the shared resources. (Non-Shareable section)

Exit Section:-It segment that is executed by a process , immediately after its exit from the critical section . In this section , a process performs certain operations , indicating its exit from the critical section and thereby enables one of the waiting processes to enter into the critical sections.

Remainder Section :-This is the remaining part of a process's code (excluding entry section, critical section , and exit section). When the process is executing in this section , it implies that it is not waiting to enter its critical section. So the processes executing in their remainder sections are not considered as candidates for entry into their critical section.

Critical Section Algorithms : - An idea of this algorithm is to solve a critical section problem should meet the three requirement viz.
a) Mutual exclusion
b) Progress
c) Bounded Waiting .

Algorithm 1 :-
I finished with it. Now you're it.
This algorithm applies to a set of two processes only, say[P_0,P_1]. Both are cooperating through a shared integer variable "turn". At a time,its value will be either 0 or 1.
Int turn = 0;

It satisfied only Mutual exclusion and Bounded waiting requirements**.** Hence **,**Progress is not meet by this algorithm.So , the main limitation of algorithm is that it doesn't meet the requirement of Progress**.**

Algorithm 2
Let me have it , after you finish with it.
This algorithm also works for the set of two processes only . Unlike algorithm 1,this algorithm takes into consideration the need of process to enter its critical section. Boolean variable **flag** is used to synchronize two cooperating process .

Typedefenum Boolean {false,true};
It satisfied only Mutual exclusion and Bounded waiting requirements**.** Hence **,**Progress is not meet by this algorithm.So , the main limitation of algorithm is that it doesn't meet the requirement of Progress**.**

Algorithm 3 (Peterson's Algorithm)
This algorithm meets all the three requirement of an ideal critical section solution . It is also a generic algorithms that can be used for the synchronization of arbitrary set of N cooperating processes.
It has two variables turn & flag both .

gradeup SUPER

Gradeup UGC NET
Super Superscription

Features:

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

Gradeup Super Subscription, Enroll Now