

Name: Aravinda Kumar Reddy Thippareddy

Roll No.: CS20BTECH11053

Course: OS-II

Assignment-4

REPORT

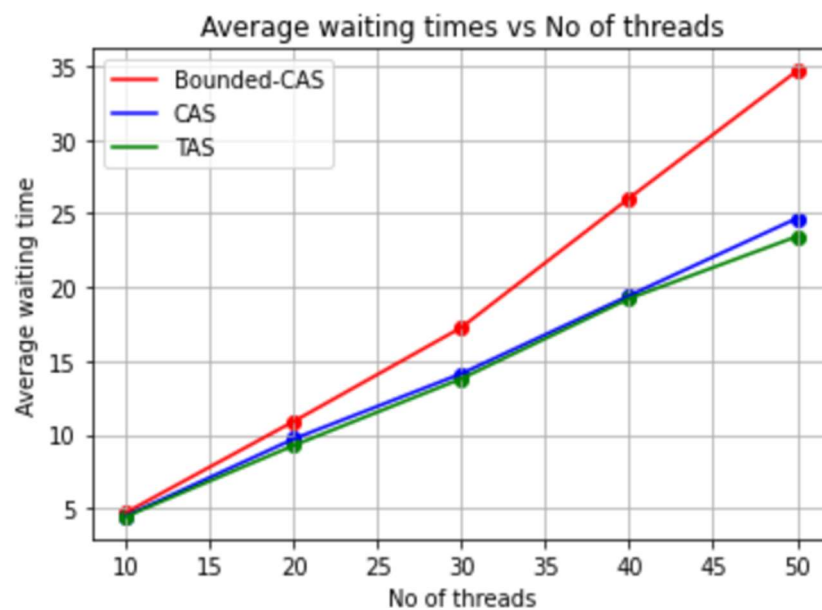
Analysis of my output:

- My output satisfies the mutual exclusion condition.
- In my output file, at some places it is printed as 'NULL'. This is because two parallelly executing threads are writing to the output file at the same time.
- But exit and enter statements of different parallelly executing threads are not written to the output file at the same time (since we are restricting the count of threads that are executing CS to one).
- Possible parallel printings to output file are,
 1. Request, Exit
 2. Request, Enter
 3. Request, Request
- You can observe that total time taken by the program to execute is less than the sum of waiting times, because one or more threads will be waiting parallelly.

Note:

- For plotting the graphs, I have taken values of lamda1 and lamda2 as 2 and 1 respectively.

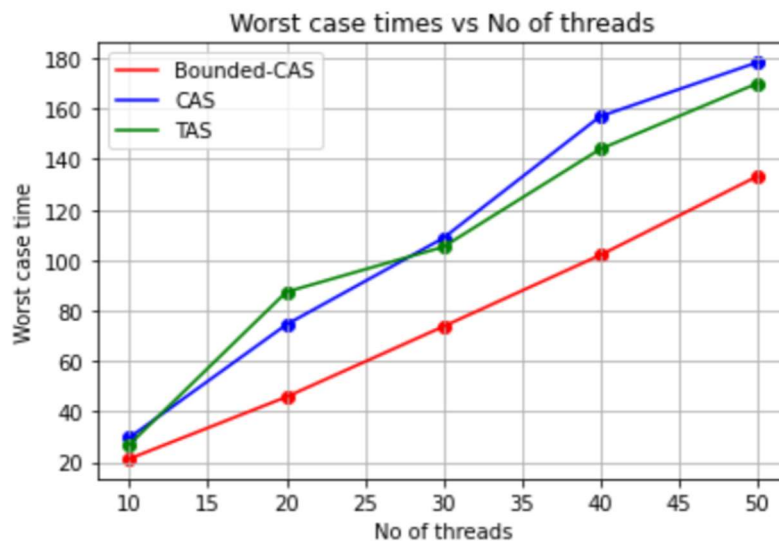
Graph (average waiting times v/s number of threads):



Observations:

1. As the number of threads increase 'Average waiting times' of Bounded CAS, CAS and TAS increases.
2. 'Average waiting times' of CAS and TAS are nearly same (can be seen from the graph). And 'Average waiting times' of CAS and TAS increases nearly at the same rate(w.r.to number of threads).
3. As the number of threads increase, the 'Average waiting times' of Bounded CAS increases faster than the 'Average waiting times' of both CAS and TAS.

Graph (worst waiting times v/s number of threads):



Observations:

1. As the number of threads increase 'Worst case times' of Bounded CAS, CAS, and TAS increases.
2. 'Worst case times' of CAS and TAS are nearly same (can be seen from the graph). And 'Worst case times' of CAS and TAS increases nearly at the same rate(w.r.to number of threads).
3. As the number of threads increase, the 'Worst case times' of Bounded CAS increases slower than the 'Worst case times' of both CAS and TAS. (Because of the fact that bounded waiting is implemented in Bounded CAS)