

Question 1:

In this question, you are required to simulate the starving problem that a process may suffer if all other processes in the system have higher priority than its priority.

It is required to create at least two processes (P1 and P2) with priorities 40 and 35 for example. You should create a third process (Pstarv) with less priority such as 25.

You need to simulate a **context switch** between P1 and P2.

You may use any method to simulate context switch between P1 and P2.

Until now, Pstarv will not be able to run since its priority is the lowest priority. You need to fix that. You are required to change/update/increase the priority of Pstarv each time a context switch occurs. You should increase the Pstarv priority by 1 or 2. You should show in your output that Pstarv's priority is increasing.

So, after a number of context switches, Pstarv will finally be able to run to print its ID and print a celebration message that it is finally running and celebrate that you will get a good grade as well.

Feel free to assume any missing information such as should you have a flag or an argument to decide which process you are interested to avoid starvation, should you use shell commands or not, etc.

P1 and P2 are not supposed to know anything about the Pstarv process. Do not hardcode it inside P1 or P2. So, you should not include the code of increasing the Pstarv priority in P1 nor P2.

Question 2:

You can think of the starvation problem in Question 1 but instead of incrementing the priority of the Pstarv every time a context switch occurs, the priority increment of Pstarv should occur **every two seconds** the Pstarv process is in the ready queue without getting the CPU.

To demonstrate this feature, you are required to just increment and print the "Pstarv priority" every two seconds based on the clock and the clkinit (not the sleep function/system call).

FAQ Q1:

- Can I simulate the context switch using any method and not put both P1 and P2 to sleep alternatively?

Yes.

- Can I create more than three processes?

Yes

- Can I call them anything not necessary P1, p2, Pstarv?

Yes

- What should I do after Pstarv runs and prints its own ID and a celebration message?

Anything you want. You can terminate everything or you can keep working preventing starvation for any other low priority process.

- Can I tell the OS that Pstarv is the one that should not be starved or let the OS finds out?

Anything. Bare min, you tell the OS. A better design is to let the OS finds out.

- Should I show that Pstarv priority increases each context switch?

Yes.

- Sometimes, I see Pstarv running before its time, is it wrong?

Yes.

FAQ Q2:

- Can I use other time interval not 2 seconds?

Yes

- Can I combine the two questions and show both methods in one demo?

Yes