

In an operating system three CPU-intensive processes are ready for execution, which require 10ns, 20ns and 30ns and arrival at times 0ns, 2ns and 6ns, respectively. Write a Program to calculate the total number of context switches needed if the operating system implements a shortest job first (preemptive) scheduling algorithm. Also calculate the average time for which the processes have to wait before getting the CPU.

```

GNU nano 8.7 sjf.c
#include <stdio.h>

int main() {
    int n = 3;
    int bt[3] = {10, 20, 30};
    int at[3] = {0, 2, 6};
    int rt[3], wt[3] = {0};
    int completed = 0, time = 0;
    int shortest = -1, min;
    int done[3] = {0};
    int context_switch = 0;
    int prev = -1;

    for (int i = 0; i < n; i++)
        rt[i] = bt[i];

    while (completed < n) {
        min = 9999;
        shortest = -1;

        for (int i = 0; i < n; i++) {
            if (at[i] <= time && !done[i] && rt[i] < min && rt[i] > 0) {
                min = rt[i];
                shortest = i;
            }
        }

        if (shortest == -1) {
            time++;
            continue;
        }

        if (prev != shortest) {
            context_switch++;
            prev = shortest;
        }

        rt[shortest]--;
        time++;

        if (rt[shortest] == 0) {
            completed++;
            done[shortest] = 1;
            wt[shortest] = time - bt[shortest] - at[shortest];
        }
    }

    float avg_wt = 0;
    for (int i = 0; i < n; i++)
        avg_wt += wt[i];

    printf("Total Context Switches: %d\n", context_switch - 1);
    printf("Average Waiting Time: %.2f ns\n", avg_wt / n);

    return 0;
}

```

```
F0IN@LAPTOP-MC73RDEH MSYS ~  
$ nano sjf.c  
  
F0IN@LAPTOP-MC73RDEH MSYS ~  
$ gcc sjf.c -o sjf  
  
F0IN@LAPTOP-MC73RDEH MSYS ~  
$ ./sjf  
Total Context Switches: 2  
Average Waiting Time: 10.67 ns  
  
F0IN@LAPTOP-MC73RDEH MSYS ~  
$
```

Air: Moderate  
Tomorrow



Search



ENG  
IN

20:58  
01-02-2026