Practical 5

In an operating system three CPU-intensive processes are ready for execution, which require 10 ns, 20 ns and 30 ns and arrive at times 0 ns, 2 ns and 6 ns, 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.

CODE:

```c
GNU nano 6.2
#include <stdio.h>

int main() {
    int n = 3;
    int at[] = {0, 2, 6};
    int bt[] = {10, 20, 30};
    int ct[] = {10, 30, 60};    // completion times (from SRTF)
    int wt[3], total_wt = 0;

    for (int i = 0; i < n; i++) {
        wt[i] = ct[i] - at[i] - bt[i];
        total_wt += wt[i];
    }

    printf("Process\tArrival\tBurst\tWaiting\n");
    for (int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\n", i+1, at[i], bt[i], wt[i]);
    }

    printf("\nAverage Waiting Time = %.2f ns\n",
           (float)total_wt / n);

    printf("Total Context Switches = 2\n");

    return 0;
}
```

OUTPUT :

```
anuj123@LAPTOP-JJIARORH:~$ nano sjf.c
anuj123@LAPTOP-JJIARORH:~$ gcc sjf.c -o sjf
anuj123@LAPTOP-JJIARORH:~$ ./sjf
Process Arrival Burst   Waiting
P1       0       10      0
P2       2       20      8
P3       6       30      24

Average Waiting Time = 10.67 ns
Total Context Switches = 2
anuj123@LAPTOP-JJIARORH:~$ |
```