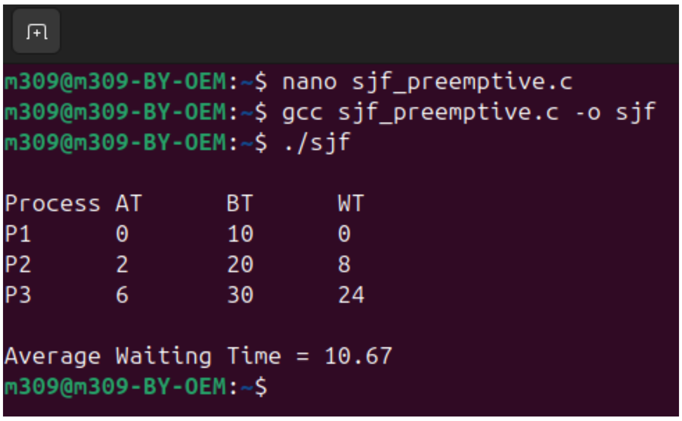
Write a program to implement Shortest Job First (SJF) Preemptive Scheduling for three processes and calculate the total context switches and average waiting time. The processes have burst times 10ns, 20ns, and 30ns, arriving at 0ns, 2ns, and 6ns, respectively.

Code:

```
#include <stdio.h>

int main() {
    int n = 3;
    int at[3] = {0, 2, 6};     // Arrival Time
    int bt[3] = {10, 20, 30}; // Burst Time
    int rt[3];               // Remaining Time
    int ct[3], wt[3];

    int time = 0, complete = 0;
    int min, shortest;
    int check = 0;
    float avg_wt = 0;

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

    while (complete != n) {
        min = 9999;
        check = 0;

        for (int i = 0; i < n; i++) {
            if (at[i] <= time && rt[i] < min && rt[i] > 0) {
                min = rt[i];
                shortest = i;
                check = 1;
            }
        }

        if (check == 0) {
            time++;
            continue;
        }

        rt[shortest]--;
        time++;

        if (rt[shortest] == 0) {
            complete++;
            ct[shortest] = time;
            wt[shortest] = ct[shortest] - at[shortest] - bt[shortest];
            if (wt[shortest] < 0)
                wt[shortest] = 0;
        }
    }

    printf("\nProcess\tAT\tBT\tWT\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]);
        avg_wt += wt[i];
    }

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

    return 0;
}
```

Output:



```
m309@m309-BY-OEM:~$ nano sjf_preemptive.c
m309@m309-BY-OEM:~$ gcc sjf_preemptive.c -o sjf
m309@m309-BY-OEM:~$ ./sjf

Process AT      BT      WT
P1      0       10      0
P2      2       20      8
P3      6       30      24

Average Waiting Time = 10.67
m309@m309-BY-OEM:~$
```