

# Practical 5 : Process management Based practical

## 5. Process management Based practical

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.

CODE:

```
M ~
GNU nano 8.7                                     hello.c
#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("%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:

```
ASUS@LAPTOP-5DB0ADS1 MSYS ~
$ nano hello.c

ASUS@LAPTOP-5DB0ADS1 MSYS ~
$ gcc hello.c -o hello

ASUS@LAPTOP-5DB0ADS1 MSYS ~
$ ./hello
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
```