

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 processes have to wait before getting the CPU.

INPUT:

```
krupa_bawane@LAPTOP-AH4V: srtf.c
GNU nano 7.2
#include <stdio.h>

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

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

    while(complete != n)
    {
        min = 9999;
        shortest = -1;

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

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

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

        rt[shortest]--;
        t++;

        if(rt[shortest] == 0)
        {
            complete++;
            finish_time = t;
            wt[shortest] = finish_time - bt[shortest] - at[shortest];

            if(wt[shortest] < 0)
                wt[shortest] = 0;
        }

        context_switch--;
    }

    float avg = 0;

    for(int i = 0; i < n; i++)
    {
        printf("P%d waiting time = %d\n", i+1, wt[i]);
        avg += wt[i];
    }

    printf("Average waiting time = %f\n", avg/3);
}
```

```
krupa_bawane@LAPTOP-AH4V: srtf.c
GNU nano 7.2

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

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

rt[shortest]--;
t++;

if(rt[shortest] == 0)
{
    complete++;
    finish_time = t;
    wt[shortest] = finish_time - bt[shortest] - at[shortest];

    if(wt[shortest] < 0)
        wt[shortest] = 0;
}

context_switch--;

float avg = 0;

for(int i = 0; i < n; i++)
{
    printf("P%d waiting time = %d\n", i+1, wt[i]);
    avg += wt[i];
}

printf("Average waiting time = %f\n", avg/3);
}
```

```
krupa_bawane@LAPTOP-AI4V x  +  v  -  o  x
GNU nano 7.2 srtf.c
{
    context_switch++;
    prev = shortest;
}

rt[shortest]--;
t++;

if(rt[shortest] == 0)
{
    complete++;
    finish_time = t;
    wt[shortest] = finish_time - bt[shortest] - at[shortest];

    if(wt[shortest] < 0)
        wt[shortest] = 0;
}

context_switch--;

float avg = 0;

for(int i = 0; i < n; i++)
{
    printf("P%d waiting time = %d\n", i+1, wt[i]);
    avg += wt[i];
}

avg = avg / n;

printf("Average waiting time = %.2f\n", avg);
printf("Context switches = %d\n", context_switch);

return 0;
}

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo       ^A Set Mark   ^] To Bracket
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^_ Justify    ^/ Go To Line ^E Redo       ^G Copy       ^_ Where Was
```

OUTPUT:

```
krupa_bawane@LAPTOP-AI4VE3LU:~$ nano srtf.c
krupa_bawane@LAPTOP-AI4VE3LU:~$ gcc srtf.c -o srtf
krupa_bawane@LAPTOP-AI4VE3LU:~$ ./srtf
P1 waiting time = 0
P2 waiting time = 8
P3 waiting time = 24
Average waiting time = 10.67
Context switches = 2
krupa_bawane@LAPTOP-AI4VE3LU:~$
```