Source Code: For FCFS

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main() {

const int n = 3;

int bt[n] = {24,3,2};

int wt[n], tat[n];

int total\_wt = 0, total\_tat = 0;

auto start = high\_resolution\_clock::now();

wt[0] = 0;

for (int i = 1; i < n; i++) {

wt[i] = bt[i - 1] + wt[i - 1];

}

for (int i = 0; i < n; i++) {

tat[i] = bt[i] + wt[i];

total\_wt += wt[i];

total\_tat += tat[i];

}

auto stop = high\_resolution\_clock::now();

cout << "\n--- FCFS Scheduling ---\n";

cout << "Process\tBurst Time\tWaiting Time\tTurnaround Time\n";

for (int i = 0; i < n; i++) {

cout << "P" << i + 1 << "\t" << bt[i] << "\t\t" << wt[i] << "\t\t" << tat[i] << endl;

}

cout << "\nAverage Waiting Time = " << (float)total\_wt / n;

cout << "\nAverage Turnaround Time = " << (float)total\_tat / n;

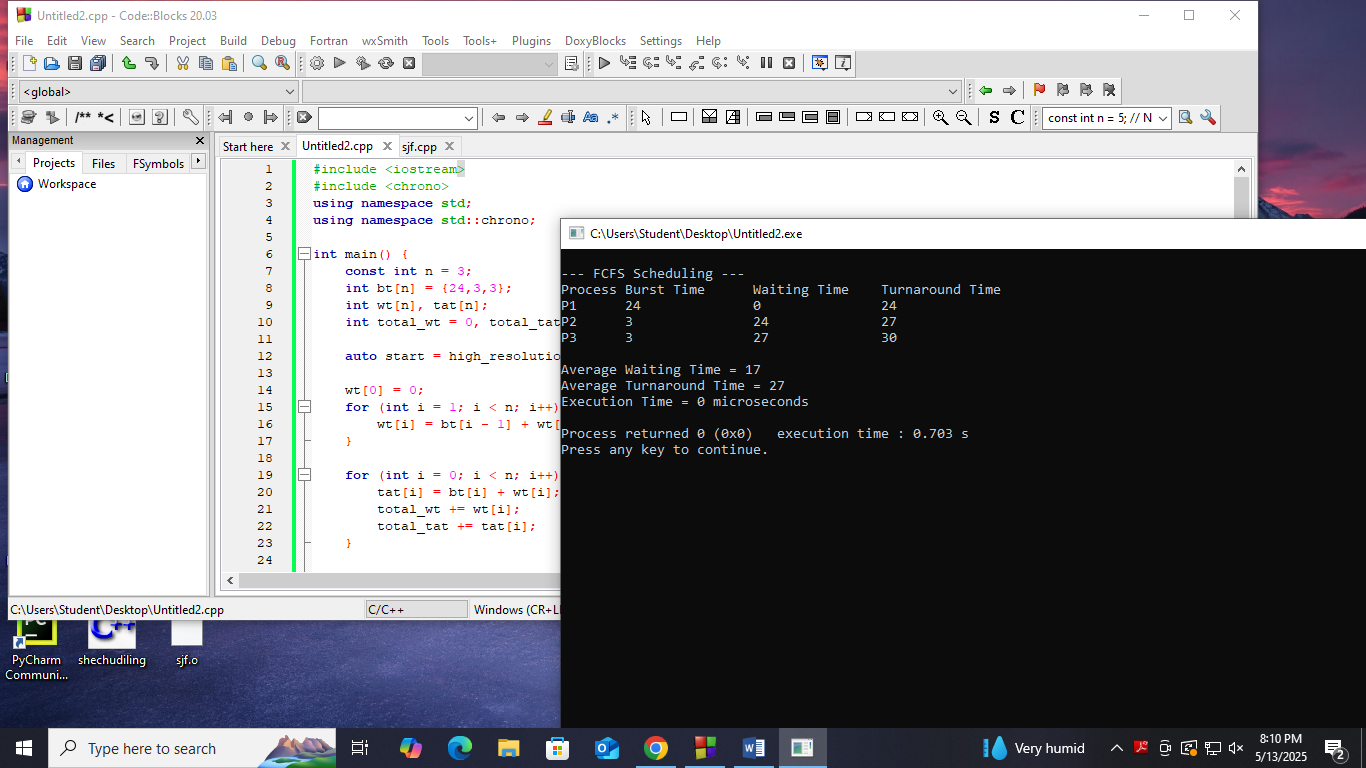
auto duration = duration\_cast<microseconds>(stop - start);

cout << "\nExecution Time = " << duration.count() << " microseconds" << endl;

return 0;

}

**Screenshots of Output:** for FCF



**source code**:for SJF

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main() {

const int n = 3;

int wt[n], tat[n];

int total\_wt = 0, total\_tat = 0;

int proc[n] = {1, 2, 3};

for (int i = 0; i < n - 1; i++) {

for (int j = i + 1; j < n; j++) {

if (bt[i] > bt[j]) {

swap(bt[i], bt[j]);

swap(proc[i], proc[j]);

}

}

}

auto start = high\_resolution\_clock::now();

wt[0] = 0;

for (int i = 1; i < n; i++) {

wt[i] = bt[i - 1] + wt[i - 1];

}

for (int i = 0; i < n; i++) {

tat[i] = bt[i] + wt[i];

total\_wt += wt[i];

total\_tat += tat[i];

}

auto stop = high\_resolution\_clock::now();

cout << "\n--- SJF (Non-preemptive) Scheduling ---\n";

cout << "Process\tBurst Time\tWaiting Time\tTurnaround Time\n";

for (int i = 0; i < n; i++) {

cout << "P" << proc[i] << "\t" << bt[i] << "\t\t" << wt[i] << "\t\t" << tat[i] << endl;

}

cout << "\nAverage Waiting Time = " << (float)total\_wt / n;

cout << "\nAverage Turnaround Time = " << (float)total\_tat / n;

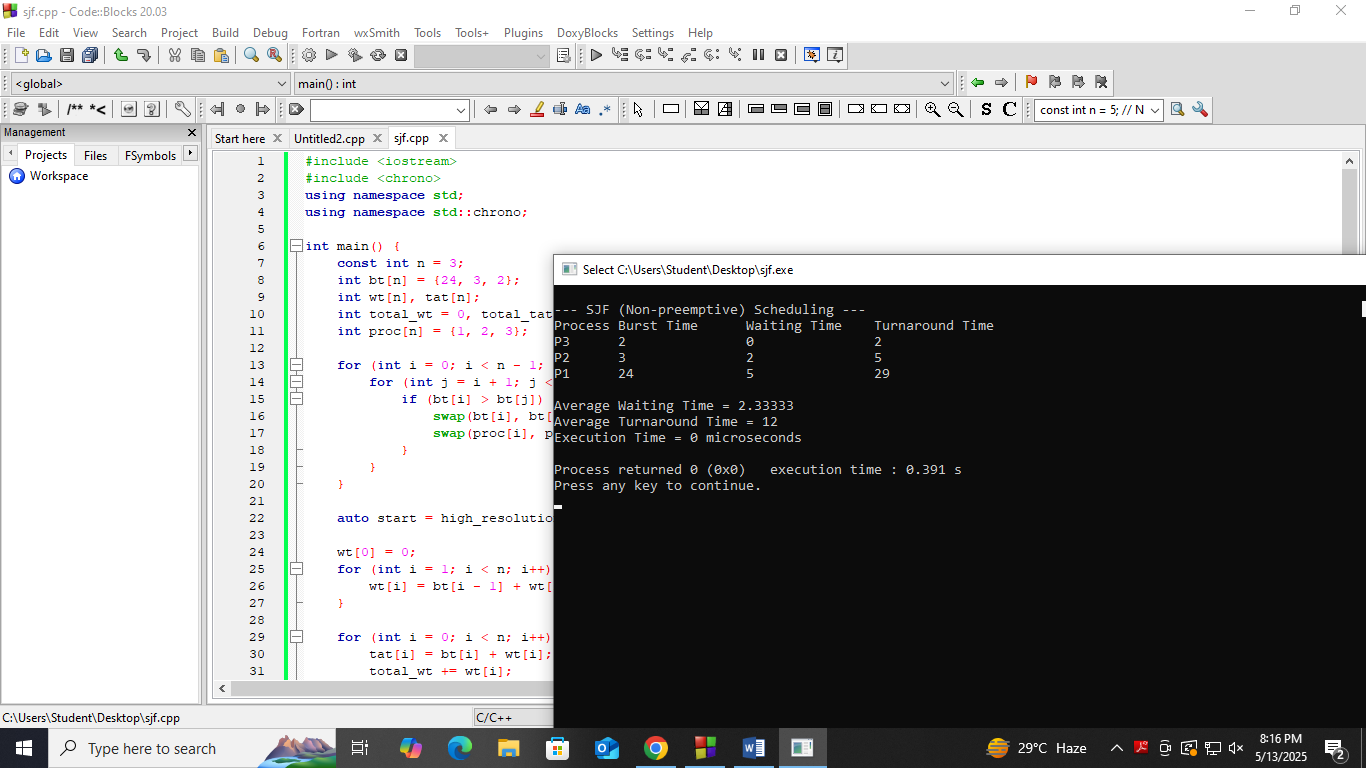
auto duration = duration\_cast<microseconds>(stop - start);

cout << "\nExecution Time = " << duration.count() << " microseconds" << endl;

return 0;

}

**Screenshots of output**:for SJF



**Source code :** for no preemption SJF

#include <iostream>

#include <climits>

using namespace std;

int main() {

int n = 4;

int at[] = {0, 1, 2, 3};

int bt[] = {8, 4, 9, 5};

int rt[4];

for (int i = 0; i < n; i++)

rt[i] = bt[i];

int complete = 0, t = 0, minm = INT\_MAX;

int shortest = 0, finish\_time;

bool check = false;

int wt[4] = {0}, tat[4] = {0};

while (complete != n) {

minm = INT\_MAX;

check = false;

for (int j = 0; j < n; j++) {

if ((at[j] <= t) && (rt[j] < minm) && rt[j] > 0) {

minm = rt[j];

shortest = j;

check = true;

}

}

if (!check) {

t++;

continue;

}

rt[shortest]--;

if (rt[shortest] == 0) {

complete++;

finish\_time = t + 1;

wt[shortest] = finish\_time - bt[shortest] - at[shortest];

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

}

t++;

}

for (int i = 0; i < n; i++) {

tat[i] = bt[i] + wt[i]; }

cout << "Process\tArrival\tBurst\tWaiting\tTurnaround\n";

int total\_wt = 0, total\_tat = 0;

for (int i = 0; i < n; i++) {

total\_wt += wt[i];

total\_tat += tat[i];

cout << "P" << i + 1 << "\t" << at[i] << "\t" << bt[i]

<< "\t" << wt[i] << "\t" << tat[i] << endl;

}

cout << "\nAverage Waiting Time = " << (float)total\_wt / n;

cout << "\nAverage Turnaround Time = " << (float)total\_tat / n << endl;

return 0;

}

**Screenshots of Output**: for no preemption of SJF

