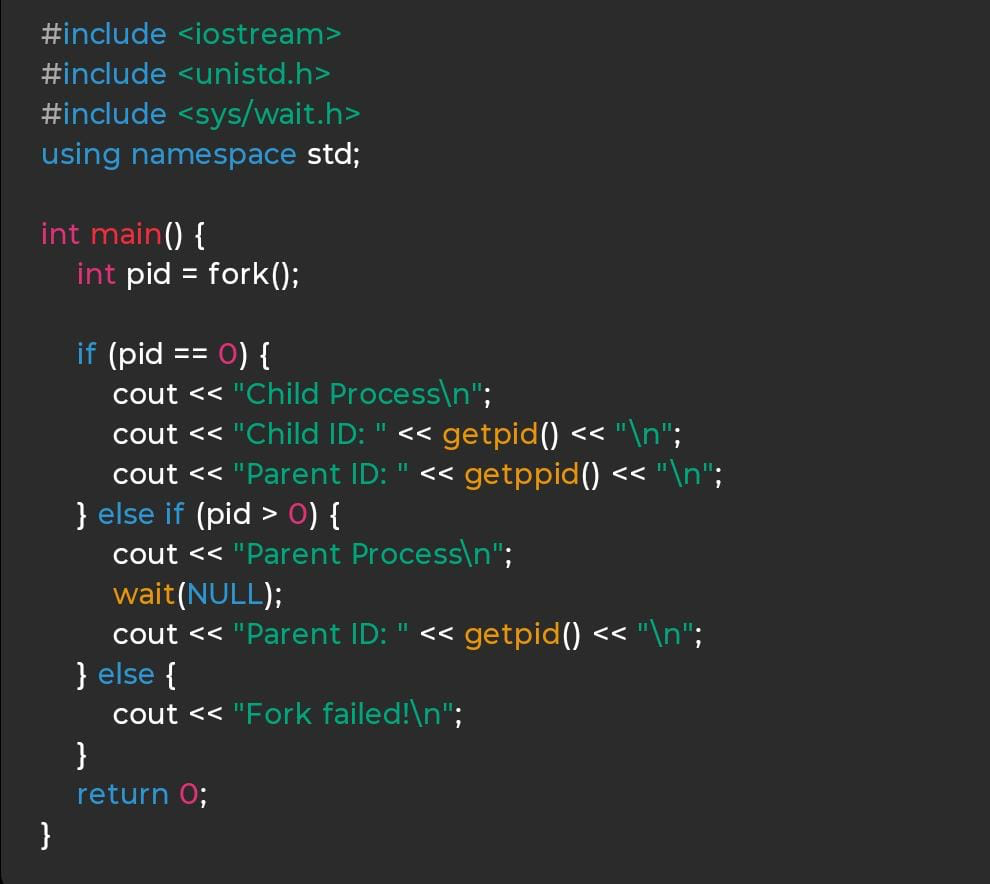
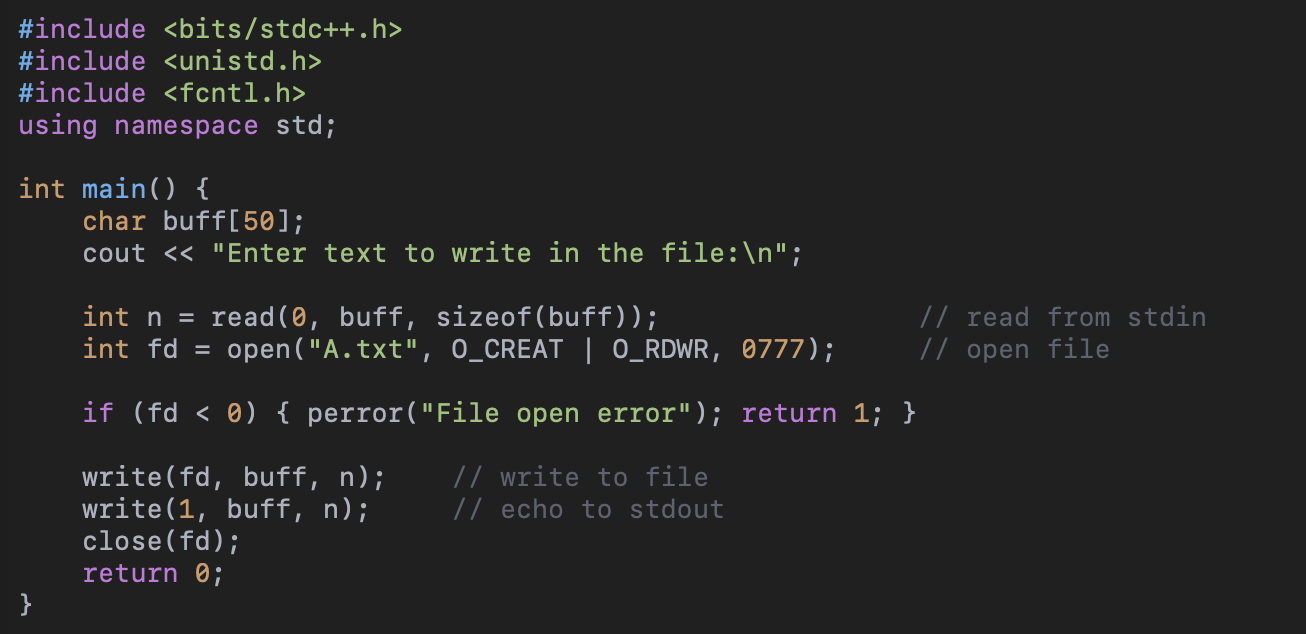
2.  




3.

Sjf  
#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter number of processes: ";

cin >> n;

int pid[50], bt[50], wt[50], tat[50];

cout << "Enter process IDs and burst times:\n";

for (int i = 0; i < n; i++) cin >> pid[i] >> bt[i];

// Sort processes by burst time

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

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

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

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

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

}

}

}

cout << "\nPID\tBT\tWT\tTAT\n";

int totalWT = 0, totalTAT = 0, prevWT = 0;

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

wt[i] = prevWT;

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

prevWT += bt[i];

totalWT += wt[i];

totalTAT += tat[i];

cout << pid[i] << "\t" << bt[i] << "\t" << wt[i] << "\t" << tat[i] << "\n";

}

cout << "\nAverage WT = " << (float)totalWT / n;

cout << "\nAverage TAT = " << (float)totalTAT / n << endl;

}  
  
  
fcfs  
  
#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter number of processes: ";

cin >> n;

int pid[50], bt[50], wt[50], tat[50];

cout << "Enter process IDs and burst times:\n";

for (int i = 0; i < n; i++) cin >> pid[i] >> bt[i];

cout << "\nPID\tBT\tWT\tTAT\n";

int totalWT = 0, totalTAT = 0, prevWT = 0;

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

wt[i] = prevWT;

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

prevWT += bt[i];

totalWT += wt[i];

totalTAT += tat[i];

cout << pid[i] << "\t" << bt[i] << "\t" << wt[i] << "\t" << tat[i] << "\n";

}

cout << "\nAverage WT = " << (float)totalWT / n;

cout << "\nAverage TAT = " << (float)totalTAT / n << endl;

}

Rr

#include <iostream>

using namespace std;

int main() {

int n, tq;

cout << "Enter number of processes: ";

cin >> n;

int pid[50], bt[50], rem[50], wt[50] = {0}, tat[50];

cout << "Enter process IDs and burst times:\n";

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

cin >> pid[i] >> bt[i];

rem[i] = bt[i]; // remaining burst time

}

cout << "Enter time quantum: ";

cin >> tq;

int time = 0, done = 0;

while (done < n) {

done = 0;

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

if (rem[i] > 0) {

if (rem[i] > tq) {

time += tq;

rem[i] -= tq;

} else {

time += rem[i];

wt[i] = time - bt[i]; // waiting time

rem[i] = 0;

}

}

if (rem[i] == 0) done++;

}

}

int totalWT = 0, totalTAT = 0;

cout << "\nPID\tBT\tWT\tTAT\n";

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

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

totalWT += wt[i];

totalTAT += tat[i];

cout << pid[i] << "\t" << bt[i] << "\t" << wt[i] << "\t" << tat[i] << "\n";

}

cout << "\nAverage WT = " << (float)totalWT / n;

cout << "\nAverage TAT = " << (float)totalTAT / n << endl;

}