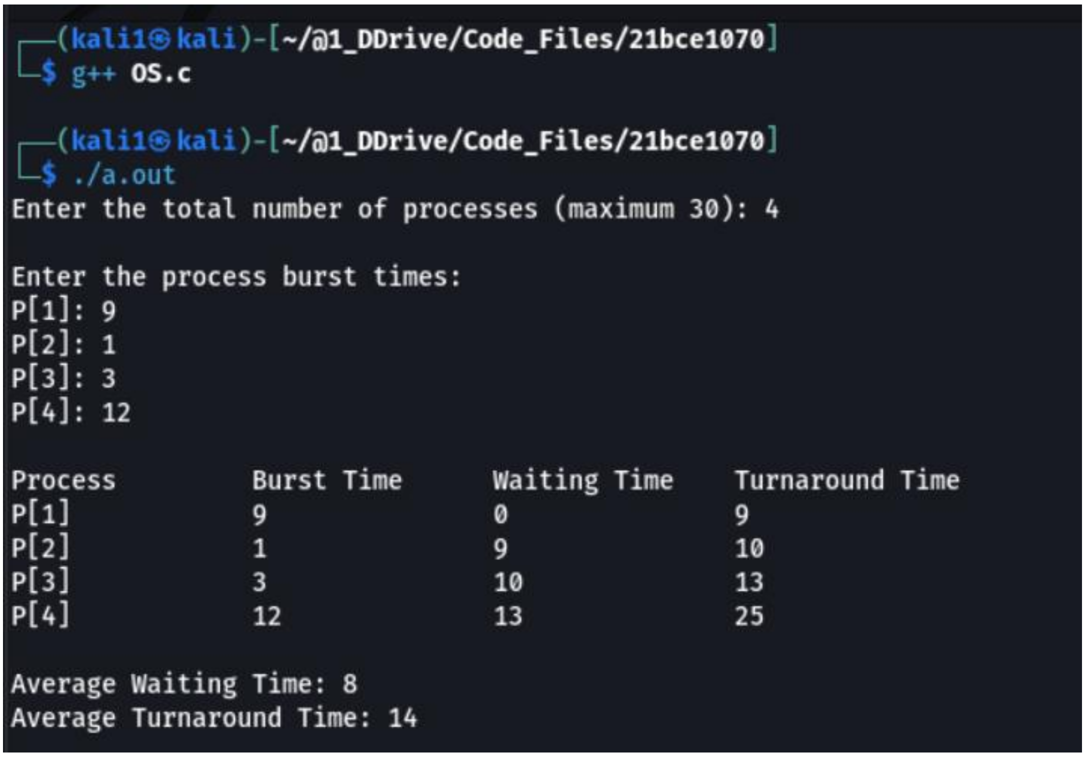
Process Scheduling

FCFS

Code:

```
#include <stdio.h>
int main() {
  int n, bt[30], wt[30], tt[30], avg_wt = 0, avg_tt = 0, i, j;

  printf("Enter the total number of processes (maximum 30): ");
  scanf("%d", &n);

  printf("\nEnter the process burst times:\n");
  for (i = 0; i < n; i++) {
    printf("P[%d]: ", i + 1);
    scanf("%d", &bt[i]);
  }

  wt[0] = 0;

  for (i = 1; i < n; i++) {
    wt[i] = 0;
    for (j = 0; j < i; j++) {
      wt[i] += bt[j];
    }
  }

  printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (i = 0; i < n; i++) {
    tt[i] = bt[i] + wt[i];
    avg_wt += wt[i];
    avg_tt += tt[i];
    printf("P[%d]\t\t%d\t\t%d\t\t%d\n", i + 1, bt[i], wt[i], tt[i]);
  }

  avg_wt /= i;
  avg_tt /= i;
  printf("\nAverage Waiting Time: %d\n", avg_wt);
  printf("Average Turnaround Time: %d\n", avg_tt);

  return 0;
}
```

Output:



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ g++ OS.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the total number of processes (maximum 30): 4

Enter the process burst times:
P[1]: 9
P[2]: 1
P[3]: 3
P[4]: 12

Process         Burst Time      Waiting Time    Turnaround Time
P[1]            9               0               9
P[2]            1               9               10
P[3]            3               10              13
P[4]            12              13              25

Average Waiting Time: 8
Average Turnaround Time: 14
```

Priority

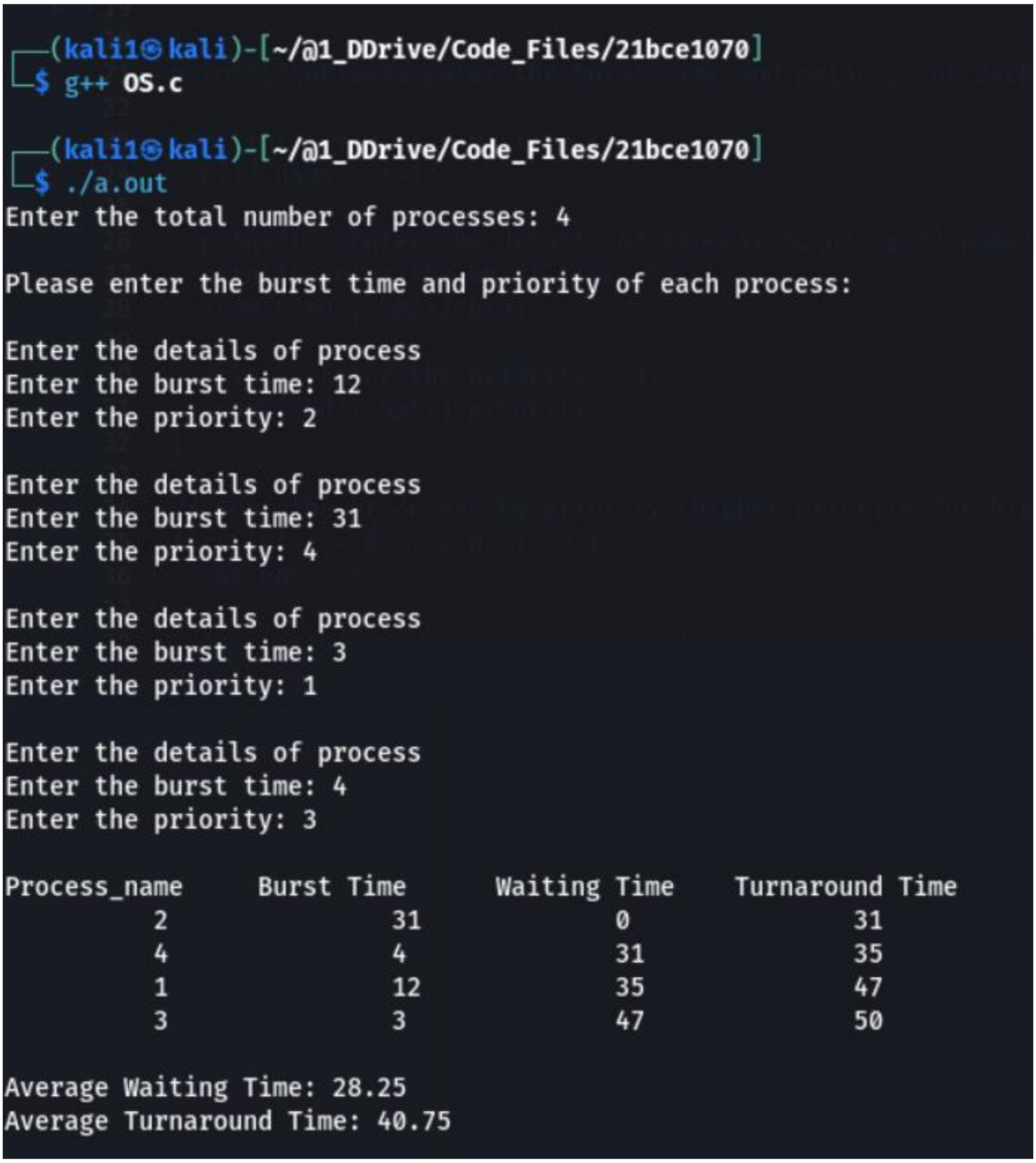
Code:

```
#include<stdio.h>
```

```
struct Process {
  int name;
  int bt;
  int wt;
  int tat;
  int priority;
};

int main() {
  int n, twt = 0, ttat = 0;
  float awt, atat;

  printf("Enter the total number of processes: ");
  scanf("%d", &n);

  struct Process p[n];


  printf("\nPlease enter the burst time and priority of each process:\n");

  for (int i = 0; i < n; i++) {
    p[i].name = i+1;

    printf("\nEnter the details of process %c\n", p[i].name);
    printf("Enter the burst time: ");
    scanf("%d", &p[i].bt);

    printf("Enter the priority: ");
    scanf("%d", &p[i].priority);
  }
```

```
// Sort the processes by priority (higher priority for higher number)
for (int i = 0; i < n; i++) {
  int pos = i;

  for (int j = i + 1; j < n; j++) {
    if (p[j].priority > p[pos].priority)
      pos = j;
  }

  struct Process temp = p[i];
  p[i] = p[pos];
  p[pos] = temp;
}

p[0].wt = 0;

for (int i = 1; i < n; i++) {
  p[i].wt = 0;
  for (int j = 0; j < i; j++) {
    p[i].wt += p[j].bt;
  }
  twt += p[i].wt;
}

awt = (float) twt / (float) n;

printf("\nProcess_name \t Burst Time \t Waiting Time \t Turnaround Time\n");



for (int i = 0; i < n; i++) {
    p[i].tat = p[i].bt + p[i].wt;
    ttat += p[i].tat;

    printf("\t  %d \t\t  %d \t\t %d \t\t %d\n", p[i].name, p[i].bt, p[i].wt, p[i].tat);

  }

  atat = (float) ttat / (float) n;

  printf("\nAverage Waiting Time: %.2f", awt);
  printf("\nAverage Turnaround Time: %.2f\n", atat);

  return 0;
}
```

Output:



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ g++ OS.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the total number of processes: 4

Please enter the burst time and priority of each process:

Enter the details of process
Enter the burst time: 12
Enter the priority: 2

Enter the details of process
Enter the burst time: 31
Enter the priority: 4

Enter the details of process
Enter the burst time: 3
Enter the priority: 1

Enter the details of process
Enter the burst time: 4
Enter the priority: 3

Process_name     Burst Time      Waiting Time    Turnaround Time
          2               31              0               31
          4               4               31              35
          1               12              35              47
          3               3               47              50

Average Waiting Time: 28.25
Average Turnaround Time: 40.75
```

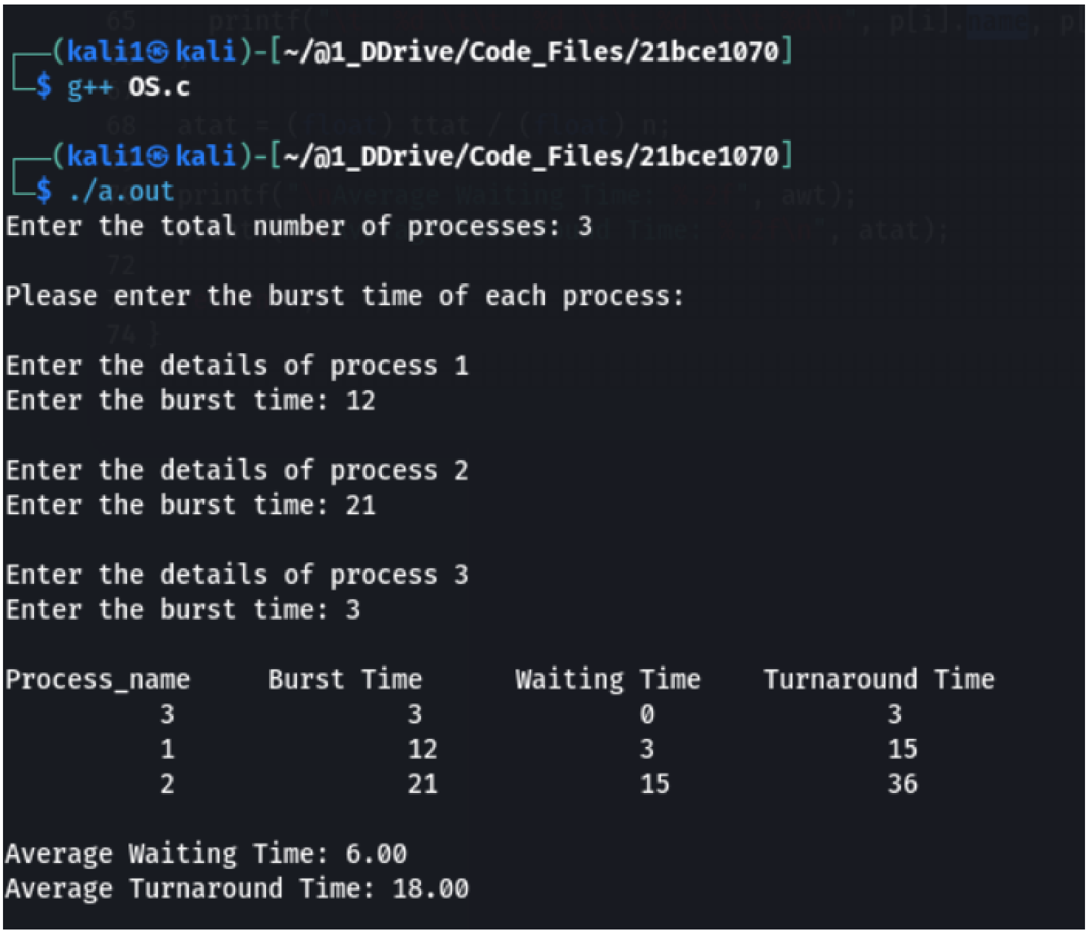
Shortest Job First

```
#include<stdio.h>


struct Process {
  int name;
  int bt;
```

```
 int wt;
 int tat;
 int priority;
};

int main() {
 int n, twt = 0, ttat = 0;
 float awt, atat;

 printf("Enter the total number of processes: ");
 scanf("%d", &n);

 struct Process p[n];

 printf("\nPlease enter the burst time of each process:\n");

 for (int i = 0; i < n; i++) {
  p[i].name = i + 1;

  printf("\nEnter the details of process %d\n", p[i].name);
  printf("Enter the burst time: ");
  scanf("%d", &p[i].bt);

  // Assigning priority based on burst time (lower burst time has higher priority)
  p[i].priority = p[i].bt;
 }

 // Sort the processes by burst time (SJF)
 for (int i = 0; i < n; i++) {
  int pos = i;
  for (int j = i + 1; j < n; j++) {
    if (p[j].bt < p[pos].bt)
      pos = j;
  }

  struct Process temp = p[i];
  p[i] = p[pos];
  p[pos] = temp;
}

p[0].wt = 0;

for (int i = 1; i < n; i++) {
  p[i].wt = 0;
  for (int j = 0; j < i; j++) {
    p[i].wt += p[j].bt;
  }
  twt += p[i].wt;
}

awt = (float) twt / (float) n;

printf("\nProcess_name \t Burst Time \t Waiting Time \t Turnaround Time\n");

for (int i = 0; i < n; i++) {
  p[i].tat = p[i].bt + p[i].wt;
  ttat += p[i].tat;

  printf("\t  %d \t\t  %d \t\t %d \t\t %d\n", p[i].name, p[i].bt, p[i].wt, p[i].tat);
}
```

```
  atat = (float) ttat / (float) n;

  printf("\nAverage Waiting Time: %.2f", awt);
  printf("\nAverage Turnaround Time: %.2f\n", atat);

  return 0;
}
```

Output:



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ g++ OS.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the total number of processes: 3

Please enter the burst time of each process:

Enter the details of process 1
Enter the burst time: 12

Enter the details of process 2
Enter the burst time: 21

Enter the details of process 3
Enter the burst time: 3

Process_name     Burst Time      Waiting Time     Turnaround Time
          3               3               0                3
          1               12              3                15
          2               21              15               36

Average Waiting Time: 6.00
Average Turnaround Time: 18.00
```