C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\os1.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```cpp
//dharshini j
//192324059

#include<stdio.h>
#include<unistd.h>
int main()
{
 printf("Process ID: %d\n", getpid() );
 printf("Parent Process ID: %d\n", getpid() );
 return 0;
}
```

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\os2.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
1  //dharshini j
2  //192324059
3  #include <stdio.h>
4  #include <stdlib.h>
5  int main()
6  {
7  FILE *fptr1, *fptr2;
8  char filename[100], c;
9  printf("Enter the filename to open for reading \n");
10 scanf("%s", filename);
11 fptr1 = fopen(filename, "r");
12 if (fptr1 == NULL)
13 {
14 printf("Cannot open file %s \n", filename);
15 exit(0);
16 }
17 printf("Enter the filename to open for writing \n");
18 scanf("%s", filename);
19 fptr2 = fopen(filename, "w");
20 if (fptr2 == NULL)
21 {
22 printf("Cannot open file %s \n", filename);
23 exit(0);
24 }
25 c = fgetc(fptr1);
26 while (c != EOF)
27 {
28 fputc(c, fptr2);
29 c = fgetc(fptr1);
30 }
31 printf("\nContents copied to %s", filename);
32 fclose(fptr1);
33 fclose(fptr2);
34 return 0;
35 }
```

Line: 1 Col: 1 Sel: 26 Lines: 35 Length: 682 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os3.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
1  //dharshini j
2  //192324059
3
4  #include <stdio.h>
5  int main()
6  {
7  int A[100][4];
8  int i, j, n, total = 0, index, temp;
9  float avg_wt, avg_tat;
10 printf("Enter number of process: ");
11 scanf("%d", &n);
12 printf("Enter Burst Time:\n");
13 for (i = 0; i < n; i++) {
14 printf("P%d: ", i + 1);
15 scanf("%d", &A[i][1]);
16 A[i][0] = i + 1;
17 }
18 for (i = 0; i < n; i++) {
19 index = i;
20 for (j = i + 1; j < n; j++)
21 if (A[j][1] < A[index][1])
22 index = j;
23 temp = A[i][1];
24 A[i][1] = A[index][1];
25 A[index][1] = temp;
26 temp = A[i][0];
27 A[i][0] = A[index][0];
28 A[index][0] = temp;
29 }
30 A[0][2] = 0;
31 for (i = 1; i < n; i++) {
32 A[i][2] = 0;
33 for (j = 0; j < i; j++)
34 A[i][2] += A[j][1];
35 total += A[i][2];
```

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os4.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```cpp
//dharshini j
//192324059

#include <stdio.h>

int main() {
    int bt[20], p[20], wt[20], tat[20], i, j, n, total = 0, pos, temp;
    float avg_wt, avg_tat;

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

    printf("\nEnter Burst Time:\n");
    for (i = 0; i < n; i++) {
        printf("p%d: ", i + 1);
        scanf("%d", &bt[i]);
        p[i] = i + 1;
    }

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

        temp = bt[i];
        bt[i] = bt[pos];
        bt[pos] = temp;

        temp = p[i];
        p[i] = p[pos];
        p[pos] = temp;
    }
```

Line: 3 Col: 2 Sel: 0 Lines: 63 Length: 1381 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os4.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
30          bt[pos] = temp;
31
32          temp = p[i];
33          p[i] = p[pos];
34          p[pos] = temp;
35      }
36
37      wt[0] = 0;
38      for (i = 1; i < n; i++) {
39          wt[i] = 0;
40          for (j = 0; j < i; j++) {
41              wt[i] += bt[j];
42          }
43          total += wt[i];
44      }
45
46      avg_wt = (float)total / n;
47      total = 0;
48
49      printf("\nProcess\t Burst Time \tWaiting Time \tTurnaround Time\n");
50      for (i = 0; i < n; i++) {
51          tat[i] = bt[i] + wt[i];
52          total += tat[i];
53          printf("p%d\t\t %d\t\t %d\t\t %d\n", p[i], bt[i], wt[i], tat[i]);
54      }
55
56      avg_tat = (float)total / n;
57
58      printf("\nAverage Waiting Time = %.2f", avg_wt);
59      printf("\nAverage Turnaround Time = %.2f\n", avg_tat);
60
61      return 0;
62  }
63
```

Line: 3 Col: 2 Sel: 0 Lines: 63 Length: 1381 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os5.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
1  //dharshini j
2  //192324059
3
4  #include<stdio.h>
5  struct priority_scheduling {
6   char process_name;
7   int burst_time;
8   int waiting_time;
9   int turn_around_time;
10  int priority;
11 };
12 int main() {
13  int number_of_process;
14  int total = 0;
15  struct priority_scheduling temp_process;
16  int ASCII_number = 65;
17  int position;
18  float average_waiting_time;
19  float average_turnaround_time;
20  printf("Enter the total number of Processes: ");
21  scanf("%d", & number_of_process);
22  struct priority_scheduling process[number_of_process];
23  printf("\nPlease Enter the Burst Time and Priority of each process:\n");
24  for (int i = 0; i < number_of_process; i++) {
25  process[i].process_name = (char) ASCII_number;
26  printf("\nEnter the details of the process %c \n", process[i].process_name);
27  printf("Enter the burst time: ");
28  scanf("%d", & process[i].burst_time);
29  printf("Enter the priority: ");
30  scanf("%d", & process[i].priority);
31  ASCII_number++;
32  }
33  for (int i = 0; i < number_of_process; i++) {
34  position = i;
35  for (int j = i + 1; j < number_of_process; j++) {
```

Line: 3 Col: 1 Sel: 0 Lines: 66 Length: 2342 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os5.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
33  for (int i = 0; i < number_of_process; i++) {
34  position = i;
35  for (int j = i + 1; j < number_of_process; j++) {
36  if (process[j].priority > process[position].priority)
37  position = j;
38  }
39  temp_process = process[i];
40  process[i] = process[position];
41  process[position] = temp_process;
42  }
43  process[0].waiting_time = 0;
44  for (int i = 1; i < number_of_process; i++) {
45  process[i].waiting_time = 0;
46  for (int j = 0; j < i; j++) {
47  process[i].waiting_time += process[j].burst_time;
48  }
49  total += process[i].waiting_time;
50  }
51  average_waiting_time = (float) total / (float) number_of_process;
52  total = 0;
53  printf("\n\nProcess_name \t Burst Time \t Waiting Time \t Turnaround Time\n");
54  printf("-----------------------------------------------------------\n");
55  for (int i = 0; i < number_of_process; i++) {
56  process[i].turn_around_time = process[i].burst_time + process[i].waiting_time;
57  total += process[i].turn_around_time;
58  printf("\t %c \t\t %d \t\t %d \t\t %d", process[i].process_name,
59 process[i].burst_time, process[i].waiting_time, process[i].turn_around_time);
60  printf("\n-----------------------------------------------------------\n");
61  }
62  average_turnaround_time = (float) total / (float) number_of_process;
63  printf("\n\n Average Waiting Time : %f", average_waiting_time);
64  printf("\n Average Turnaround Time: %f\n", average_turnaround_time);
65  return 0;
66 }
```

Line: 3 Col: 1 Sel: 0 Lines: 66 Length: 2342 Insert Done parsing in 0.031 seconds

Tomorrow's high
Near record

Search

ENG
IN

19:33
3 December

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os6.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```c
//dharshini j
//192324059
 #include <stdio.h>

struct Process {
    int process_id;
    int arrival_time;
    int burst_time;
    int priority;
    int remaining_time;
};

void swap(struct Process *xp, struct Process *yp) {
    struct Process temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(struct Process processes[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int min_index = i;
        for (int j = i + 1; j < n; j++) {
            if (processes[j].priority < processes[min_index].priority)
                min_index = j;
        }
        swap(&processes[min_index], &processes[i]);
    }
}

int main() {
    int n, current_time = 0;

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

```

Line: 4 Col: 1 Sel: 0 Lines: 82 Length: 2456 Insert Done parsing in 0.031 seconds

Tomorrow's high
Near record

Search

ENG
IN

19:33
3 December

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os6.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
40          scanf("%d %d %d", &processes[i].arrival_time, &processes[i].burst_time, &processes[i].priority);
41          processes[i].process_id = i + 1;
42          processes[i].remaining_time = processes[i].burst_time;
43      }
44
45      for (int i = 0; i < n - 1; i++) {
46          for (int j = 0; j < n - i - 1; j++) {
47              if (processes[j].arrival_time > processes[j + 1].arrival_time)
48                  swap(&processes[j], &processes[j + 1]);
49          }
50      }
51
52      printf("\nGantt Chart:\n-----------\n");
53      while (1) {
54          int flag = 1;
55          for (int i = 0; i < n; i++) {
56              if (processes[i].remaining_time > 0 && processes[i].arrival_time <= current_time) {
57                  flag = 0;
58                  printf("| P%d ", processes[i].process_id);
59                  processes[i].remaining_time--;
60                  if (processes[i].remaining_time == 0) {
61                      printf("|");
62                      printf("\n");
63                  }
64              }
65          }
66          if (flag)
67              break;
68          current_time++;
69      }
70
71      int total_waiting_time = 0, total_turnaround_time = 0;
72      for (int i = 0; i < n; i++) {
73          total_turnaround_time += current_time - processes[i].arrival_time;
74          total_waiting_time += (current_time - processes[i].arrival_time) - processes[i].burst_time;
```

Line: 4 Col: 1 Sel: 0 Lines: 82 Length: 2456 Insert Done parsing in 0.031 seconds

Tomorrow's high
Near record

Search

ENG
IN

19:33
3 December

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os6.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
49          }
50      }
51
52      printf("\nGantt Chart:\n----------\n");
53      while (1) {
54          int flag = 1;
55          for (int i = 0; i < n; i++) {
56              if (processes[i].remaining_time > 0 && processes[i].arrival_time <= current_time) {
57                  flag = 0;
58                  printf("| P%d ", processes[i].process_id);
59                  processes[i].remaining_time--;
60                  if (processes[i].remaining_time == 0) {
61                      printf("|");
62                      printf("\n");
63                  }
64              }
65          }
66          if (flag)
67              break;
68          current_time++;
69      }
70
71      int total_waiting_time = 0, total_turnaround_time = 0;
72      for (int i = 0; i < n; i++) {
73          total_turnaround_time += current_time - processes[i].arrival_time;
74          total_waiting_time += (current_time - processes[i].arrival_time) - processes[i].burst_time;
75      }
76
77      printf("\nAverage Waiting Time: %.2f\n", (float)total_waiting_time / n);
78      printf("Average Turnaround Time: %.2f\n", (float)total_turnaround_time / n);
79
80      return 0;
81  }
82
```

Line: 4 Col: 1 Sel: 0 Lines: 82 Length: 2456 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os9.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
1  //dharshini j
2  //192324059
3
4  #include<stdio.h>
5  #include<stdlib.h>
6  #include<unistd.h>
7  #include<sys/shm.h>
8  #include<string.h>
9  int main()
10 {
11 int i;
12 void *shared_memory;
13 char buff[100];
14 int shmid;
15 shmid=shmget((key_t)2345, 1024, 0666|IPC_CREAT);
16 printf("Key of shared memory is %d\n",shmid);
17 shared_memory=shmat(shmid,NULL,0);
18 printf("Process attached at %p\n",shared_memory);
19 printf("Enter some data to write to shared memory\n");
20 read(0,buff,100);
21 strcpy(shared_memory,buff);
22 printf("You wrote : %s\n",(char *)shared_memory);
23 }
```

Line: 3 Col: 1 Sel: 0 Lines: 23 Length: 565 Insert Done parsing in 0.031 seconds

C:\Users\rjana\OneDrive\Downloads\Documents\Desktop\program\OS\lab os\os10.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

os1.cpp os2.cpp os3.cpp os4.cpp os5.cpp os6.cpp os7.cpp os8.cpp [*] os9.cpp os10.cpp

```
1  //dharshini j
2  //192324059
3
4  #include <stdio.h>
5  struct clientData
6  {
7  unsigned int acctNum;
8  char lastName[ 15 ];
9  char firstName[ 10 ];
10 double balance;
11 };
12 int main( void )
13 {
14 unsigned int i;
15 struct clientData blankClient = { 0, "", "", 0.0 };
16 FILE *cfPtr;
17 if ( ( cfPtr = fopen( "credit.dat", "wb" ) ) == NULL )
18 {
19 puts( "File could not be opened." );
20 }
21 else
22 {
23 for ( i = 1; i <= 100; ++i )
24 {
25 fwrite( &blankClient, sizeof( struct clientData ), 1, cfPtr );
26 }
27 fclose ( cfPtr );
28 }
29 }
```

Line: 3 Col: 1 Sel: 0 Lines: 29 Length: 511 Insert Done parsing in 0.031 seconds