

OUTPUT · MNagakeshava/CSA04 x Online C Compiler - online editor x +

onlinegdb.com/online_c_compiler

Gmail YouTube Maps News Translate 2

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP

Have fun taking surveys
and get paid!
ADS VIA CARBON

main.c

```
1 #include<stdio.h>
2 #include<unistd.h>
3 #include<sys/types.h>
4 int main()
5 {
6     pid_t p;
7     printf("before fork\n");
8     p=fork();
9     if(p==0)
10    {
11         printf("I am child having id %d\n",getpid());
12         printf("My parent's id is %d\n",getppid());
13     }
14    else{
15        printf("My child's id is %d\n",p);
16        printf("I am parent having id %d\n",getpid());
17    }
18    printf("Common\n");
19 }
```

before fork
My child's id is 628
I am parent having id 624
Common

...Program finished with exit code 0
Press ENTER to exit console.[]

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

D:\systemcall copy.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug [*] OS.C systemcall.c systemcall copy.c

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

D:\systemcall copy.exe

Enter the filename to open for reading :operating system
Cannot open file operating

Process exited after 13.6 seconds with return value 0
Press any key to continue . . . |

Compiler (3) Resources Compile Log Debug Find Results

Line: 28 Col: 49 Sel: 0 Lines: 32 Length: 666 Insert Done parsing in 0 seconds

D:\fcfs.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug systemcall copy.c [*] fcfs.c

```
1 #include <stdio.h>
2 int main()
3 {
4     int pid[15];
5     int bt[15];
6     int n,i;
7     printf("Enter the number of processes: ");
8     scanf("%d",&n);
9     printf("Enter process id of all the processes: ");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&pid[i]);
13    }
14    printf("Enter burst time of all the processes: ");
15    for(i=0;i<n;i++)
16    {
17        scanf("%d",&bt[i]);
18    }
19    int wt[n];
20    wt[0]=0;
21    for(i=1; i<n; i++)
22    {
23        wt[i]= bt[i-1]+ wt[i-1];
24    }
25    printf("Process ID      Burst Time      Waiting Time      TurnAround Time\n");
26    float twt=0.0;
27    float tat= 0.0;
28    for(i=0; i<n; i++)
29    {
30        printf("%d\t\t", pid[i]);
31        printf("%d\t\t", bt[i]);
32        printf("%d\t\t", wt[i]);
33        printf("%d\t\t", bt[i]+wt[i]);
34        printf("\n");
35        twt += wt[i];
36        tat += (wt[i]+bt[i]);
37    }
38    float att,awt;
```

D:\fcfs.exe

```
Enter the number of processes: 5
Enter process id of all the processes: 11
12
13
14
15
Enter burst time of all the processes: 16
17
18
19
20
Process ID      Burst Time      Waiting Time      TurnAround Time
11              16                  0                  16
12              17                  16                 33
13              18                  33                 51
14              19                  51                 70
15              20                  70                 90
Avg. waiting time= 34.000000
Avg. turnaround time= 52.000000
-----
Process exited after 24.38 seconds with return value 31
Press any key to continue . . . |
```

Compiler Resources Compile Log Debug Find Results

Line: 38 Col: 19 Sel: 0 Lines: 43 Length: 1026 Insert Done parsing in 0.015 seconds

D:\smallestexecution.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug systemcall copy.c fcfs.c [*] smallestexecution.c

```
1 #include <stdio.h>
2 int main()
3 {
4     int A[100][4];
5     int i, j, b, total = 0, index, temp;
6     float avg_wt, avg_tat;
7     printf("Enter number of process: ");
8     scanf("%d", &b);
9     printf("Enter Burst Time:\n");
10    for (i = 0; i < b; i++)
11    {
12        printf("P%d: ", i + 1);
13        scanf("%d", &A[i][1]);
14        A[i][0] = i + 1;
15    }
16    for (i = 0; i < b; i++)
17    {
18        index = i;
19        for (j = i + 1; j < b; j++)
20            if (A[j][1] < A[index][1])
21                index = j;
22        temp = A[i][1];
23        A[i][1] = A[index][1];
24        A[index][1] = temp;
25        temp = A[i][0];
26        A[i][0] = A[index][0];
27        A[index][0] = temp;
28    }
29    A[0][2] = 0;
30    for (i = 1; i < b; i++)
31    {
32        A[i][2] = 0;
33        for (j = 0; j < i; j++)
34            A[i][2] += A[j][1];
35        total += A[i][2];
36    }
37    avg_wt = (float)total / b;
38    total = 0;
39    printf("P   BT   WT   TAT\n");
40    for (i = 0; i < b; i++)
41    {
42        A[i][3] = A[i][1] + A[i][2];
43        total += A[i][3];
44        printf("P%d %d %d %d\n", A[i][0],
45               A[i][1], A[i][2], A[i][3]);
46    }
47    avg_tat = (float)total / b;
48    printf("Average Waiting Time= %f", avg_wt);
49    printf("\nAverage Turnaround Time= %f", avg_tat);
```

D:\smallestexecution.exe

Enter number of process: 6

Enter Burst Time:

P1: 3

P2: 9

P3: 5

P4: 11

P5: 55

P6: 66

P	BT	WT	TAT
P1	3	0	3
P3	5	3	8
P2	9	8	17
P4	11	17	28
P5	55	28	83
P6	66	83	149

Average Waiting Time= 23.166666

Average Turnaround Time= 48.000000

Process exited after 6.341 seconds with return value 35

Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 34 Col: 32 Sel: 0 Lines: 50 Length: 1081 Insert Done parsing in 0.031 seconds

D:\highest priority.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug systemcall copy.c fcfs.c [*] smallestexecution.c [*] highest priority.c

```
1 #include<stdio.h>
2 struct priority_scheduling
3 {
4     char process_name;
5     int burst_time,waiting_time,turn_around_time,priority;
6 };
7 int main()
8 {
9     int number_of_process,total = 0,ASCII_number = 65,position,i,j;
10    struct priority_scheduling temp_process;
11    float average_waiting_time;
12    float average_turnaround_time;
13    printf("Enter the total number of Processes: ");
14    scanf("%d", & number_of_process);
15    struct priority_scheduling process[number_of_process];
16    printf("\nPlease Enter the Burst Time and Priority of each process:\n");
17    for (i = 0; i < number_of_process; i++)
18    {
19        process[i].process_name = (char) ASCII_number;
20        printf("\nEnter the details of the process %c \n", process[i].process_name);
21        printf("Enter the burst time: ");
22        scanf("%d", & process[i].burst_time);
23        printf("Enter the priority: ");
24        scanf("%d", & process[i].priority);
25        ASCII_number++;
26    }
27    for (i = 0; i < number_of_process; i++)
28    {
29        position = i;
30        for (j = i + 1; j < number_of_process; j++)
31        {
32            if (process[j].priority > process[position].priority)
33                position = j;
34        }
35        temp_process = process[i];
36        process[i] = process[position];
37        process[position] = temp_process;
38    }
39    process[0].waiting_time = 0;
40    for (i = 1; i < number_of_process; i++)
41    {
42        process[i].waiting_time = 0;
43        for (j = 0; j < i; j++)
44            process[i].waiting_time += process[j].burst_time;
45        total += process[i].waiting_time;
46    }
47    average_waiting_time = (float) total / (float) number_of_process;
48    total = 0;
49    printf("\nProcess_name \t Burst Time \t Waiting Time \t Turnaround Time\n");
50    printf("-----\n");
51    for (i = 0; i < number_of_process; i++)
52    {
53    }
```

D:\highest priority.exe

Enter the burst time: 5
Enter the priority: 6

Enter the details of the process C
Enter the burst time: 9
Enter the priority: 8

Enter the details of the process D
Enter the burst time: 4
Enter the priority: 6

Process_name	Burst Time	Waiting Time	Turnaround Time
C	9	0	9
B	5	9	14
D	4	14	18
A	1	18	19

Average Waiting Time : 10.250000
Average Turnaround Time: 15.000000

Process exited after 9.423 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 48 Col: 68 Sel: 0 Lines: 63 Length: 2409 Insert Done parsing in 0.109 seconds

D:\pre-emptive.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug systemcall copy.c fcfs.c [*] smallestexecution.c [*] highest priority.c pre-emptive.c

```
1 #include<stdio.h>
2 struct process
3 {
4     int WT,AT,BT,TAT,PT;
5 };
6 struct process a[10];
7 int main()
8 {
9     int n,temp[10],t,count=0,short_p,i;
10    float total_WT=0,total_TAT=0,Avg_WT,Avg_TAT;
11    printf("Enter the number of the process\n");
12    scanf("%d",&n);
13    printf("Enter the arrival time , burst time and priority of the process\n");
14    printf("AT BT PT\n");
15    for(i=0;i<n;i++)
16    {
17        scanf("%d%d%d",&a[i].AT,&a[i].BT,&a[i].PT);
18        temp[i]=a[i].BT;
19    }
20    a[9].PT=10000;
21
22    for(t=0;count!=n;t++)
23    {
24        short_p=9;
25        for(i=0;i<n;i++)
26        {
27            if(a[short_p].PT>a[i].PT && a[i].AT<=t && a[i].BT>0)
28            {
29                short_p=i;
30            }
31
32            a[short_p].BT=a[short_p].BT-1;
33            if(a[short_p].BT==0)
34            {
35                count++;
36                a[short_p].WT=t+1-a[short_p].AT-temp[short_p];
37                a[short_p].TAT=t+1-a[short_p].AT;
38                total_WT=total_WT+a[short_p].WT;
39                total_TAT=total_TAT+a[short_p].TAT;
40            }
41        }
42        Avg_WT=total_WT/n;
43        Avg_TAT=total_TAT/n;
44        printf("ID WT TAT\n");
45        for(i=0;i<n;i++)
46        {
47            printf("%d %d\t%d\n",i+1,a[i].WT,a[i].TAT);
48        }
49        printf("Avg waiting time of the process is %f\n",Avg_WT);
50        printf("Avg turn around time of the process is %f\n",Avg_TAT);
51    }
52    return 0;
53 }
```

D:\pre-emptive.exe

Enter the number of the process
5
Enter the arrival time , burst time and priority of the process
AT BT PT
1 5 6
11 6 7
4 9 10
5 2 32
11 56 5
ID WT TAT
1 0 5
2 56 62
3 64 73
4 72 74
5 0 56
Avg waiting time of the process is 38.400002
Avg turn around time of the process is 54.000000

Process exited after 36.51 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 51 Col: 67 Sel: 0 Lines: 53 Length: 1397 Insert Done parsing in 0.016 seconds

D:\Non Preemptive SJF.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug systemcall copy.c fcfs.c [*] smallestexecution.c [*] highest priority.c pre-emptive.c Non Preemptive SJF.c

```
1 #include<stdio.h>
2 int main()
3 {
4     int time, burst_time[10], at[10], sum_burst_time = 0, smallest, n, i, sumt = 0, sumw = 0;
5     printf("enter the number of processes : ");
6     scanf("%d", &n);
7     for (i = 0; i < n; i++)
8     {
9         printf("the arrival time for process P %d : ", i + 1);
10        scanf("%d", &at[i]);
11        printf("the burst time for process P %d : ", i + 1);
12        scanf("%d", &burst_time[i]);
13        sum_burst_time += burst_time[i];
14    }
15    burst_time[9] = 9999;
16    for (time = 0; time < sum_burst_time;)
17    {
18        smallest = 9;
19        for (i = 0; i < n; i++)
20        {
21            if (at[i] <= time && burst_time[i] > 0 && burst_time[i] < burst_time[smallest])
22                smallest = i;
23        }
24        printf("P[%d]\t|\t%d\t|\t%d\n", smallest + 1, time + burst_time[smallest] - at[smallest], time - at[smallest]);
25        sumt += time + burst_time[smallest] - at[smallest];
26        sumw += time - at[smallest];
27        time += burst_time[smallest];
28        burst_time[smallest] = 0;
29    }
30    printf("\n\n average waiting time = %f", sumw * 1.0 / n);
31    printf("\n\n average turnaround time = %f", sumt * 1.0 / n);
32    return 0;
33 }
```

D:\Non Preemptive SJF.exe

enter the number of processes : 4
the arrival time for process P 1 : 5
the burst time for process P 1 : 6
the arrival time for process P 2 : 9
the burst time for process P 2 : 3
the arrival time for process P 3 : 8
the burst time for process P 3 : 6
the arrival time for process P 4 : 4
the burst time for process P 4 : 8
P[10] | 9999 | 0

average waiting time = 0.000000
average turnaround time = 2499.750000

Process exited after 9.363 seconds with return value
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 20 Col: 6 Sel: 0 Lines: 33 Length: 1113 Insert Done parsing in 0.015 seconds

D:\Roundrobin.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug systemcall copy.c fcfs.c [*] smallestexecution.c [*] highest priority.c pre-emptive.c Non Preemptive SJF.c Roundrobin.c

```

1 #include<stdio.h>
2 int main()
3 {
4     int i, limit, total = 0, x, counter = 0, time_quantum;
5     int wait_time = 0, turnaround_time = 0, arrival_time[10], burst_time[10], temp[10];
6     float average_wait_time, average_turnaround_time;
7     printf("\nEnter Total Number of Processes:\n");
8
9     public int __cdecl printf (const char * __restrict__ _Format, ...)
10    for(i = 0; i < limit; i++)
11    {
12        printf("\nEnter Details of Process[%d]\n", i + 1);
13        scanf("%d", &arrival_time[i]);
14        printf("Burst Time:\t");
15        scanf("%d", &burst_time[i]);
16        temp[i] = burst_time[i];
17    }
18
19    printf("\nEnter Time Quantum:\t");
20    scanf("%d", &time_quantum);
21    printf("\nProcess ID\tBurst Time\t Turnaround Time\t Waiting Time\n");
22    for(total = 0, i = 0; x != 0;
23    {
24        if(temp[i] <= time_quantum && temp[i] > 0)
25        {
26            total = total + temp[i];
27            temp[i] = 0;
28            counter = 1;
29        }
30        else if(temp[i] > 0)
31        {
32            temp[i] = temp[i] - time_quantum;
33            total = total + time_quantum;
34        }
35        if(temp[i] == 0 && counter == 1)
36        {
37            x--;
38            printf("\nProcess[%d]\t%d\t%d\t %d\t %d", i + 1, burst_time[i], total,
39            wait_time = wait_time + total - arrival_time[i] - burst_time[i];
40            turnaround_time = turnaround_time + total - arrival_time[i];
41            counter = 0;
42        }
43        if(i == limit - 1)
44        {
45            i = 0;
46        }
47        else if(arrival_time[i + 1] <= total)
48        {
49            i++;
50        }
51        else
52        {
53            i = 0;
54        }
55    }
56
57    printf("Average Wait Time: %.2f\n", average_wait_time);
58    printf("Average Turnaround Time: %.2f\n", average_turnaround_time);
59}

```

D:\Roundrobin.exe

Enter Total Number of Processes:
5

Enter Details of Process[1]
Arrival Time: 2
Burst Time: 3

Enter Details of Process[2]
Arrival Time: 6
Burst Time: 5

Enter Details of Process[3]
Arrival Time: 9
Burst Time: 7

Enter Details of Process[4]
Arrival Time: 3
Burst Time: 6

Enter Details of Process[5]
Arrival Time: 2
Burst Time: 6

Enter Time Quantum: 5

Process ID	Burst Time	Turnaround Time	Waiting Time
Process[1]	3	1	-2

Compiler Resources Compile Log Debug Find Results

Line: 7 Col: 51 Sel: 0 Lines: 61 Length: 2271 Insert Done parsing in 0.016 seconds

GDB online Debugger | Compiler x +

onlinegdb.com/#

Gmail YouTube Maps

OnlineGDB beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Sign Up

Login

f 199K

GOT AN OPINION? SHARE AND GET REWARDED. Rakuten AIP

Have fun taking surveys and get paid!

ADS VIA CARBON

main.c

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

input

```
Enter some data to write to shared memory
operating system
You wrote : operating system

...Program finished with exit code 0
Press ENTER to exit console.
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

GDB online Debugger | Compiler x +

onlinegdb.com/#

Gmail YouTube Maps

OnlineGDB beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Sign Up

Login

f 199K

GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP

Have fun taking surveys
and get paid!

ADS VIA CARBON

main.c

```
1 #include <stdio.h>
2 #include <sys/ipc.h>
3 #include <sys/msg.h>
4 #define MAX 10
5
6 struct mesg_buffer {
7     long mesg_type;
8     char mesg_text[100];
9 } message;
10 int main()
11 {
12     key_t key;
13     int msgid;
14     key = ftok("progsfile", 65);
15     msgid = msgget(key, 0666 | IPC_CREAT);
16     message.mesg_type = 1;
17
18     printf("Write Data : ");
19     fgets(message.mesg_text,MAX,stdin);
20     msgsnd(msgid, &message, sizeof(message), 0);
21     printf("Data send is : %s \n", message.mesg_text);
22 }

```

Language C

Run Debug Stop Share Save Beautify

input

```
Write Data : OPERATING SYSTEM
Data send is : OPERATING

...Program finished with exit code 0
Press ENTER to exit console.■
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

D:\inter process.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug fcfs.c inter process.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <semaphore.h>
5 #include <unistd.h>
6 int sum = 0;
7 sem_t mutex;
8 void *add(void *arg)
9 {
10     int *ptr = (int *) arg;
11     while(*ptr != -1)
12     {
13         sem_wait(&mutex);
14         sum += *ptr;
15         printf("value: %d sum %d\n", *ptr,sum );
16         sem_post(&mutex);
17         ptr++;
18     }
19     return NULL;
20 }
21 int main(int argc, char *args[])
22 {
23     int A[4] = {1,3,4,-1};
24     int B[4] = {4,3,5,-2};
25     pthread_t t_a, t_b;
26     sem_init(&mutex, 0, 1);
27     pthread_create(&t_a , NULL, add, A);
28     pthread_create(&t_b, NULL, add, B);
29     pthread_join(t_a, NULL);
30     pthread_join(t_b, NULL);
31     printf("Total: %d\n", sum);
32     return 0;
33 }
```

D:\inter process.exe

```
value: 1 sum 1
value: 4 sum 5
value: 3 sum 8
value: 3 sum 11
value: 4 sum 15
value: 5 sum 20
value: -2 sum 18
value: 1 sum 19
value: 3 sum 22
value: 4 sum 26
Total: 26
-----
Process exited after 0.01607 seconds with return value 0
Press any key to continue . . . |
```

Compiler Resources Compile Log Debug Find Results

Line: 28 Col: 40 Sel: 0 Lines: 33 Length: 731 Insert Done parsing in 0.031 seconds

D:\Dining Philosophers.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug fcfs.c MULTITHREADING.C Dining Philosophers.C

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<pthread.h>
4 #include<semaphore.h>
5 #include<unistd.h>
6 sem_t room;
7 sem_t chopstick[5];
8 void * philosopher(void * );
9 void eat(int);
10 int main()
11 {
12     int i,a[5];
13     pthread_t tid[5];
14     sem_init(&room,0,3);
15     for(i=0;i<5;i++)
16         sem_init(&chopstick[i],0,1);
17     for(i=0;i<5;i++){
18         a[i]=i;
19         pthread_create(&tid[i],NULL,philosopher,(void *)&a[i]);
20     }
21     for(i=0;i<5;i++)
22         pthread_join(tid[i],NULL);
23 }
24 void * philosopher(void * num)
25 {
26     int phil=*(int *)num;
27     sem_wait(&room);
28     printf("\nPhilosopher %d has entered room",phil);
29     sem_wait(&chopstick[phil]);
30     sem_wait(&chopstick[(phil+1)%5]);
31     eat(phil);
32     sleep(2);
33     printf("\nPhilosopher %d has finished eating",phil);
34     sem_post(&chopstick[(phil+1)%5]);
35     sem_post(&chopstick[phil]);
36     sem_post(&room);
37 }
38 void eat(int phil)
```

D:\Dining Philosophers.exe

```
Philosopher 0 has entered room
Philosopher 0 is eating
Philosopher 4 has entered room
Philosopher 2 has entered room
Philosopher 2 is eating
Philosopher 0 has finished eating
Philosopher 2 has finished eating
Philosopher 4 is eating
Philosopher 3 has entered room
Philosopher 1 has entered room
Philosopher 1 is eating
Philosopher 4 has finished eating
Philosopher 1 has finished eating
Philosopher 3 is eating
Philosopher 3 has finished eating
-----
Process exited after 6.052 seconds with return value 0

Press any key to continue . . .
```

Compiler Resources Compile Log Debug Find Results

Line: 35 Col: 32 Sel: 0 Lines: 41 Length: 887 Insert Done parsing in 0.016 seconds

D:\implementation of memory allocation.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug fcfs.c MULTITHREADING.C Dining Philosophers.C implementation of memory allocation.C

```
1 #include<stdio.h>
2 void firstFit(int blockSize[], int m, int processSize[], int n)
3 {
4     int i, j;
5     int allocation[n];
6     for(i = 0; i < n; i++)
7     {
8         allocation[i] = -1;
9     }
10    for (i = 0; i < n; i++)
11    {
12        for (j = 0; j < m; j++)
13        {
14            if (blockSize[j] >= processSize[i])
15            {
16                allocation[i] = j;
17                blockSize[j] -= processSize[i];
18                break;
19            }
20        }
21    }
22    printf("\nProcess No.\tProcess Size\tBlock no.\n");
23    for (int i = 0; i < n; i++)
24    {
25        printf(" %i\t\t", i+1);
26        printf("%i\t\t", processSize[i]);
27        if (allocation[i] != -1)
28            printf("%i", allocation[i] + 1);
29        else
30            printf("Not Allocated");
31        printf("\n");
32    }
33 }
34 int main()
35 {
36     int m;
37     int n;
38     int blockSize[] = {100,500,200,300,500};
39     int processSize[] = {212, 417, 112, 426};
40     m = sizeof(blockSize) / sizeof(blockSize[0]);
41     n = sizeof(processSize) / sizeof(processSize[0]);
42     firstFit(blockSize, m, processSize, n);
43     return 0 ;
44 }
```

D:\implementation of memor X + v

Process No.	Process Size	Block no.
1	212	2
2	417	5
3	112	2
4	426	Not Allocated

Process exited after 0.01655 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 32 Col: 6 Sel: 0 Lines: 44 Length: 1153 Insert Done parsing in 0.031 seconds

D:\Organize The File Using Single Level Directory.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug Organize The File Using Single Level Directory.C

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<string.h>
4 int main()
5 {
6     int nf=0,i=0,j=0,ch;
7     char mdname[10],fname[10][10],name[10];
8     printf("Enter the directory name:");
9     scanf("%s",mdname);
10    printf("Enter the number of files:");
11    scanf("%d",&nf);
12    do
13    {
14        printf("Enter file name to be created:");
15        scanf("%s",name);
16        for(i=0;i<nf;i++)
17        {
18            if(!strcmp(name,fname[i]))
19                break;
20        }
21        if(i==nf)
22        {
23            strcpy(fname[j++],name);
24            nf++;
25        }
26        else
27            printf("There is already %s\n",name);
28        printf("Do you want to enter another file(yes - 1 or no - 0):");
29        scanf("%d",&ch);
30    }
31    while(ch==1);
32    printf("Directory name is:%s\n",mdname);
33    printf("Files names are:");
34    for(i=0;i<j;i++)
35        printf("\n%s",fname[i]);
36    getch();
37 }
```

D:\Organize The File Using Sir + ×

```
Enter the directory name:system
Enter the number of files:3
Enter file name to be created:deadlocks
Do you want to enter another file(yes - 1 or no - 0):1
Enter file name to be created:processes
Do you want to enter another file(yes - 1 or no - 0):1
Enter file name to be created:operating
Do you want to enter another file(yes - 1 or no - 0):1
Enter file name to be created:mobiles
Do you want to enter another file(yes - 1 or no - 0):0
Directory name is:system
Files names are:
deadlocks
processes
operating
mobiles
-----
Process exited after 121.2 seconds with return value 0
Press any key to continue . . . |
```

Compiler Resources Compile Log Debug Find Results

Line: 26 Col: 5 Sel: 0 Lines: 37 Length: 707 Insert Done parsing in 0.016 seconds

D:\two level program.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug Organize The File Using Single Level Directory.C two level program.c

```
1 #include<string.h>
2 #include<stdlib.h>
3 #include<stdio.h>
4 struct
5 {
6     char dname[10], fname[10][10];
7     int fcnt;
8 }dir[10];
9 int main()
10 {
11     int i, ch, dcnt, k;
12     char f[30], d[30];
13     dcnt=0;
14     while(1)
15     {
16         printf("\n\n1. Create Directory\t2. Create File\t3. Delete File");
17         printf("\n4. Search File\t5. Display\t6. Exit\tEnter your choice -- ");
18         scanf("%d", &ch);
19         switch(ch)
20         {
21             case 1: printf("\nEnter name of directory -- ");
22             scanf("%s", dir[dcnt].dname);
23             dir[dcnt].fcnt=0;
24             dcnt++;
25             printf("Directory created");
26             break;
27             case 2: printf("\nEnter name of the directory -- ");
28             scanf("%s", d);
29             for(i=0; i<dcnt; i++)
30                 if(strcmp(d, dir[i].dname)==0)
31                 {
32                     printf("Enter name of the file -- ");
33                     scanf("%s", dir[i].fname[dir[i].fcnt]);
34                     printf("File created");
35                     break;
36                 }
37             if(i==dcnt)
38                 printf("Directory %s not found", d);
        }
```

D:\two level program.exe

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit Enter your choice -- 1

Enter name of directory -- system
Directory created

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit Enter your choice -- 2

Enter name of the directory -- system
Enter name of the file -- desktop
File created

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit Enter your choice -- 3

Enter name of the directory -- system
Enter name of the file -- desktop
File desktop not found

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit Enter your choice -- 4

Enter name of the directory -- system
Enter the name of the file -- desktop
File desktop not found

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit Enter your choice -- 5

Directory Files
system

Compiler Resources Compile Log Debug Find Results

Line: 93 Col: 33 Sel: 0 Lines: 102 Length: 1948 Insert Done parsing in 0.016 seconds

D:\implementing random access file for processing the employee details.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug Organize The File Using Single Level Directory.C two level program.c implementing random access file for processing the employee details.C

```
1 #include <stdio.h>
2 struct employee
3 {
4     char    name[30];
5     int     empId;
6     float   salary;
7 }
8 int main()
9 {
10    struct employee emp;
11    printf("\nEnter details :\n");
12    printf("Name ?:");
13    gets(emp.name);
14    printf("ID ?:");
15    scanf("%d",&emp.empId);
16    printf("Salary ?:");
17    scanf("%f",&emp.salary);
18    printf("\nEntered detail is \n:");
19    printf("Name: %s \n ",emp.name);
20    printf("Id: %d \n ",emp.empId);
21    printf("Salary: %f \n",emp.salary);
22    return 0;
23 }
```

D:\implementing random access file for processing the employee details.C

```
Enter details :
Name ?:sir
ID ?:0498
Salary ?:2000000

Entered detail is
:Name: sir
Id: 498
Salary: 2000000.000000

-----
Process exited after 9.497 seconds with return value 0
Press any key to continue . . .
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\implementing random access file for processing the employee details.exe
- Output Size: 129.5078125 KiB
- Compilation Time: 0.20s

Line: 18 Col: 39 Sel: 0 Lines: 23 Length: 533 Insert Done parsing in 0.015 seconds

D:\dead lock avoid with bankers algorithm.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug Organize The File Using Single Level Directory.C two level program.c implementing random access file for processing the employee details.C dead lock avoid with bankers algorithm.C

```
1 #include<stdio.h>
2 #include<conio.h>
3 int max[100][100];
4 int alloc[100][100];
5 int need[100][100];
6 int avail[100];
7 int n,r;
8 void input();
9 void show();
10 void cal();
11 int main()
12 {
13     int i,j;
14     printf("***** Banker's Algo *****\n");
15     input();
16     show();
17     cal();
18     getch();
19     return 0;
20 }
21 void input()
22 {
23     int i,j;
24     printf("Enter the no of Processes:");
25     scanf("%d",&n);
26     printf("Enter the no of resources instances:");
27     scanf("%d",&r);
28     printf("Enter the Max Matrix:\n");
29     for(i=0;i<n;i++)
30     {
31         for(j=0;j<r;j++)
32         {
33             scanf("%d",&max[i][j]);
34         }
35     }
36     printf("Enter the Allocation Matrix\n");
37     for(i=0;i<n;i++)
38     {
```

***** Banker's Algo *****

Enter the no of Processes:2

Enter the no of resources instances:3

Enter the Max Matrix:

6

9

10

15

19

20

Enter the Allocation Matrix

3

6

9

10

15

20

Enter the available Resources

25

26

29

Process	Allocation	Max	Available
P1	3 6 9 6 9 10	25 26 29	
P2	10 15 20	15 19 20	

P0->P1->

The system is in safe state

Process exited after 24.97 seconds with return value 0

Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 130 Col: 19 Sel: 0 Lines: 142 Length: 1845 Insert Done parsing in 0 seconds

D:\producer consumer problem.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug Organize The File Using Single Level Directory.C two level program.c implementing random access file for processing the employee details.C dead lock avoid with bankers algorithm.C producer consumer problem.C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int mutex = 1;
4 int full = 0;
5 int empty = 10, x = 0;
6 void producer()
7 {
8     --mutex;
9     ++full;
10    --empty;
11    x++;
12    printf("\nProducer produces"
13          " item %d",
14          x);
15    ++mutex;
16 }
17 void consumer()
18 {
19     --mutex;
20     --full;
21     ++empty;
22     printf("\nConsumer consumes "
23           " item %d",
24           x);
25     x--;
26     ++mutex;
27 }
28 int main()
29 {
30     int n, i;
31     printf("\n1. Press 1 for Producer"
32           "\n2. Press 2 for Consumer"
33           "\n3. Press 3 for Exit");
34     for (i = 1; i > 0; i++) {
35
36         printf("\nEnter your choice:");
37         scanf("%d", &n);
38         switch (n) {
```

D:\producer consumer problem

1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit
Enter your choice:1

Producer produces item 1
Enter your choice:2

Consumer consumes item 1
Enter your choice:3

Process exited after 18.54 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 58 Col: 16 Sel: 0 Lines: 63 Length: 899 Insert Done parsing in 0.032 seconds

K:\c++\19.Process Synchronisation.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 32-bit Profiling

(globals)

Project Classes Debug 17.Bankers algorithm.c 18.Producer and consumer semaphroes.c 1.system call.cpp 19.Process Synchronisation.c

```
1 #include<pthread.h>
2 #include<stdio.h>
3 #include<unistd.h>
4 void *fun1();
5 void *fun2();
6 int shared=1;
7 pthread_mutex_t l;
8 int main()
9 {
10     pthread_mutex_init(&l, NULL);
11     pthread_t thread1, thread2;
12     pthread_create(&thread1, NULL, fun1, NULL);
13     pthread_create(&thread2, NULL, fun2, NULL);
14     pthread_join(thread1, NULL);
15     pthread_join(thread2, NULL);
16     printf("Final value of shared is %d\n",shared);
17 }
18 void *fun1()
19 {
20     int x;
21     printf("Thread1 trying to acquire lock\n");
22     pthread_mutex_lock(&l);
23     printf("Thread1 acquired lock\n");
24     x=shared;
25     printf("Thread1 reads the value of shared variable as %d\n",x);
26     x++;
27     printf("Local updation by Thread1: %d\n",x);
28     sleep(1);
29     shared=x;
30     printf("Value of shared variable updated by Thread1 is: %d\n",shared);
31     pthread_mutex_unlock(&l);
32     printf("Thread1 released the lock\n");
33 }
34 void *fun2()
35 {
36     int y;
37     printf("Thread2 trying to acquire lock\n");

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

Abort Compilation

Shorten compiler paths

- Errors: 0
- Warnings: 0
- Output Filename: K:\c++\19.Process Synchronisation.exe
- Output Size: 176.8583984375 KiB
- Compilation Time: 0.28s

Line: 33 Col: 3 Sel: 0 Lines: 49 Length: 1387 Insert Done parsing in 0.047 seconds

K:\c++\19.Process Synchronisation.exe

Thread1 trying to acquire lock
Thread1 acquired lock
Thread1 reads the value of shared variable as 1
Local updation by Thread1: 2
Thread2 trying to acquire lock
Value of shared variable updated by Thread1 is: 2
Thread1 released the lock
Thread2 acquired lock
Thread2 reads the value as 2
Local updation by Thread2: 1
Value of shared variable updated by Thread2 is: 1
Thread2 released the lock
Final value of shared is 1

Process exited after 5.524 seconds with return value 27
Press any key to continue . . .

suryasaidurgesh0008/CSA0487-C | Create Ex.20 Reader writer problem | Online C Compiler - online editor

onlinegdb.com/online_c_compiler

Gmail YouTube Maps

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

f  t  + 96.7K

 GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP

Have fun taking surveys
and get paid!

ADS VIA CARBON

main.c

```
1 #include<stdio.h>
2 #include<pthread.h>
3 #include<semaphore.h>
4
5 sem_t mutex,writeblock;
6 int data = 0,rcount = 0;
7
8 void *reader(void *arg)
9 {
10     int f;
11     f = ((int)arg);
12     sem_wait(&mutex);
13     rcount = rcount + 1;
14     if(rcount==1)
15         sem_wait(&writeblock);
16     sem_post(&mutex);
17     printf("Data read by the reader%d is %d\n",f,data);
18     sleep(1);
19     sem_wait(&mutex);
20     rcount = rcount - 1;
21     if(rcount==0)
22         sem_post(&writeblock);
23     sem_post(&mutex);
24 }
25
26 void *writer(void *arg)
27 {
28     int f;
29     f = ((int) arg);
```

input

```
Data written by the writer0 is 1
Data written by the writer1 is 2
Data read by the reader1 is 2
Data read by the reader0 is 2
Data read by the reader2 is 2
Data written by the writer2 is 3
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

D:\worst fit algorithm.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug worst fit algorithm.C

```
1 #include <stdio.h>
2 void implementWorstFit(int blockSize[], int blocks, int processSize[], int processes)
3 {
4     int allocation[processes];
5     int occupied[blocks];
6     for(int i = 0; i < processes; i++)
7     {
8         allocation[i] = -1;
9     }
10    for(int i = 0; i < blocks; i++)
11    {
12        occupied[i] = 0;
13    }
14    for (int i=0; i < processes; i++)
15    {
16        int indexPlaced = -1;
17        for(int j = 0; j < blocks; j++)
18        {
19            if(blockSize[j] >= processSize[i] && !occupied[j])
20            {
21                if (indexPlaced == -1)
22                {
23                    indexPlaced = j;
24                }
25            }
26        }
27        if (indexPlaced != -1)
28        {
29            allocation[i] = indexPlaced;
30            occupied[indexPlaced] = 1;
31            blockSize[indexPlaced] -= processSize[i];
32        }
33    }
34    printf("\nProcess No.\tProcess Size\tBlock no.\n");
35    for (int i = 0; i < processes; i++)
36    {
37        printf("%d\t%d\t%d\n", i+1, processSize[i], allocation[i]);
38        if (allocation[i] != -1)
39        {
40            printf("%d\n", allocation[i] + 1);
41        }
42    }
43 }
44 int main()
45 {
46     int blockSize[] = {100, 50, 100, 130, 45};
47     int processSize[] = {40, 10, 30, 50};
48     int blocks = sizeof(blockSize)/sizeof(blockSize[0]);
49     int processes = sizeof(processSize)/sizeof(processSize[0]);
}
```

D:\worst fit algorithm.exe

Process No.	Process Size	Block no.
1	40	4
2	10	1
3	30	3
4	50	2

Process exited after 0.5603 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 47 Col: 42 Sel: 0 Lines: 52 Length: 1533 Insert Done parsing in 0.031 seconds

D:\best fit.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug worst fit algorithm.C best fit.c

```
1 #include<stdio.h>
2 #include<conio.h>
3 #define max 25
4 void main()
5 {
6     int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000;
7     static int bf[max],ff[max];
8
9     printf("\nEnter the number of blocks:");
10    scanf("%d",&nb);
11    printf("Enter the number of files:");
12    scanf("%d",&nf);
13    printf("\nEnter the size of the blocks:-\n");
14    for(i=1;i<=nb;i++)
15    {
16        printf("Block %d:",i);
17        scanf("%d",&b[i]);
18    }
19    printf("Enter the size of the files :-\n");
20    for(i=1;i<=nf;i++)
21    {
22        printf("File %d:",i);
23        scanf("%d",&f[i]);
24    }
25    for(i=1;i<=nf;i++)
26    {
27        for(j=1;j<=nb;j++)
28        {
29            if(bf[j]!=1)
30            {
31                temp=b[j]-f[i];
32                if(temp>=0)
33                if(lowest>temp)
34                {
35                    ff[i]=j;
36                    lowest=temp;
37                }
38            }
39        }
40    }
41    frag[i]=lowest;
42    bf[ff[i]]=1;
43    lowest=10000;
44 }
45 printf("\nFile No\tFile Size\tBlock No\tBlock Size\tFragment");
46 for(i=1;i<=nf && ff[i]!=0;i++)
47 printf("\n%d\t%d\t%d\t%d\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
48 getch();
}
```

D:\best fit.exe

Enter the number of blocks:6
Enter the number of files:4

Enter the size of the blocks:-
Block 1:6
Block 2:9
Block 3:3
Block 4:5
Block 5:2
Block 6:6

Enter the size of the files :-
File 1:3
File 2:6
File 3:10
File 4:2

File No File Size Block No Block Size Fragment
1 3 3 3 0
2 6 1 6 0

Process exited after 11.29 seconds with return value 51
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 40 Col: 2 Sel: 0 Lines: 49 Length: 886 Insert Could not find corresponding header file

D:\first fit algorithm.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug worst fit algorithm.C best fit.c first fit algorithm.c

```

1 #include<stdio.h>
2 #include<conio.h>
3 #define max 25
4 void main()
5 {
6     int frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0;
7     static int bf[max],ff[max];
8
9     printf("\nEnter the number of blocks:");
10    scanf("%d",&nb);
11    printf("Enter the number of files:");
12    scanf("%d",&nf);
13    printf("\nEnter the size of the blocks:-\n");
14    for(i=1;i<=nb;i++)
15    {
16        printf("Block %d:",i);
17        scanf("%d",&b[i]);
18    }
19    printf("Enter the size of the files :-\n");
20    for(i=1;i<=nf;i++)
21    {
22        printf("File %d:",i);
23        scanf("%d",&f[i]);
24    }
25    for(i=1;i<=nf;i++)
26    {
27
28        for(j=1;j<=nb;j++)
29        {
30            if(bf[j]!=1)
31            {
32                temp=b[j]-f[i];
33                if(temp>=0)
34                if(highest<temp)
35                {
36                    ff[i]=j;
37                    highest=temp;
38                }
39            }
40        }
41        frag[i]=highest;
42        bf[ff[i]]=1;
43        highest=0;
44    }
45    printf("\nFile_no:\tFile_size :\tBlock_no:\tBlock_size:\tFragement");
46    for(i=1;i<=nf;i++)
47    printf("\n%d\t%d\t%d\t%d\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
48    getch();
49 }

```

D:\first fit algorithm.exe

Enter the number of blocks:3
Enter the number of files:6
Enter the size of the blocks:-
Block 1:6
Block 2:3
Block 3:6
Enter the size of the files :-
File 1:36
File 2:3
File 3:6
File 4:3
File 5:63
File 6:33

File_no:	File_size :	Block_no:	Block_size:	Fragement
1	36	0	3	0
2	3	1	6	3
3	6	0	3	0
4	3	3	6	3
5	63	0	3	0
6	33	0	3	0

Process exited after 12.83 seconds with return value 13
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 30 Col: 14 Sel: 0 Lines: 49 Length: 877 Insert Done parsing in 0 seconds

D:\UNIX system calls .c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug worst fit algorithm.C best fit.c first fit algorithm.c UNIX system calls .c

```
1 #include<stdio.h>
2 #include<fcntl.h>
3 #include<errno.h>
4 extern int errno;
5 int main()
6 {
7     int fd = open("foo.txt", O_RDONLY | O_CREAT);
8     printf("fd = %d\n", fd);
9     if (fd == -1)
10    {
11        printf("Error Number % d\n", errno);
12        perror("Program");
13    }
14 }
15 return 0;
16
```

D:\UNIX system calls .exe

```
fd = 3

-----
Process exited after 0.01457 seconds with return value 0
Press any key to continue . . . |
```

Compiler (2) Resources Compile Log Debug Find Results

Line: 11 Col: 45 Sel: 0 Lines: 16 Length: 314 Insert Done parsing in 0.015 seconds

D:\system calls of UNIX.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug worst fit algorithm.C best fit.c first fit algorithm.c UNIX system calls .c [*] system calls of UNIX.C

```
1 #include<stdio.h>
2 #include <fcntl.h>
3 int main()
4 {
5     int fd1 = open("foo.txt", O_RDONLY);
6     if (fd1 < 0)
7     {
8         perror("c1");
9     }
10    printf("opened the fd = % d\n", fd1);
11    if (close(fd1) < 0)
12    {
13        perror("c1");
14    }
15    printf("closed the fd.\n");
16 }
17
```

D:\system calls of UNIX.exe

opened the fd = 3
closed the fd.

Process exited after 0.01931 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\system calls of UNIX.exe
- Output Size: 129.1328125 KiB
- Compilation Time: 0.27s

Line: 1 Col: 1 Sel: 0 Lines: 17 Length: 255 Insert Done parsing in 0.016 seconds

D:\file management operations.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file management operations.c

```
1 # include <stdio.h>
2 # include <string.h>
3 int main( )
4 {
5     FILE *filePointer ;
6     char dataToBeWritten[50]
7     = "GeeksforGeeks-A Computer Science Portal for Geeks";
8     filePointer = fopen("GfgTest.c", "w") ;
9     if ( filePointer == NULL )
10    {
11        printf( "GfgTest.c file failed to open." ) ;
12    }
13    else
14    {
15        printf("The file is now opened.\n");
16        if ( strlen ( dataToBeWritten ) > 0 )
17        {
18            fputs(dataToBeWritten, filePointer) ;
19            fputs("\n", filePointer) ;
20        }
21        fclose(filePointer) ;
22        printf("Data successfully written in file GfgTest.c\n");
23        printf("The file is now closed." ) ;
24    }
25
26 }
```

The file is now opened.
Data successfully written in file GfgTest.c
The file is now closed.

Process exited after 0.01589 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\file management operations.exe
- Output Size: 129.8095703125 KiB
- Compilation Time: 2.24s

Line: 26 Col: 2 Sel: 0 Lines: 26 Length: 613 Insert Done parsing in 0.047 seconds

D:\function of ls UNIX command.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug file management operations.c function of ls UNIX command.C

```
1 #include<stdio.h>
2 #include<dirent.h>
3 main()
4 {
5     char dirname[10];
6     DIR*p;
7     struct dirent *d;
8     printf("Enter directory name\n");
9     scanf("%s",dirname);
10    p=opendir(dirname);
11    if(p==NULL)
12    {
13        perror("Cannot find directory");
14    }
15    while(d=readdir(p))
16        printf("%s\n",d->d_name);
17 }
```

D:\function of ls UNIX command.C

Enter directory name
system
Cannot find directory: No such file or directory

Process exited after 10.94 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\function of ls UNIX command.exe
- Output Size: 138.9208984375 KiB
- Compilation Time: 0.22s

Line: 13 Col: 35 Sel: 0 Lines: 17 Length: 286 Insert Done parsing in 0 seconds

D:\GREP UNIX commands.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char fn[30],pat[30],temp[200];
6     FILE *fp;
7     printf("Enter file name\n");
8     scanf("%s",fn);
9     printf("Enter pattern to be searched\n");
10    scanf("%s",pat);
11    fp=fopen(fn,"r");
12    while(!feof(fp))
13    {
14        fgets(temp,1000,fp);
15        if strstr(temp,pat)!=NULL)
16            printf("%s",temp);
17    }
18    fclose(fp);
19 }
```

D:\GREP UNIX commands.C

Enter file name
program.c++
Enter pattern to be searched
if

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\GREP UNIX commands.exe
- Output Size: 129.6416015625 KiB
- Compilation Time: 0.22s

Line: 3 Col: 4 Sel: 0 Lines: 19 Length: 334 Insert Done parsing in 0 seconds

D:\Classical process synchronization.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int mutex = 1;
4 int full = 0;
5 int empty = 10, x = 0;
6 void producer()
7 {
8     --mutex;
9     ++full;
10    --empty;
11    x++;
12    printf("\nProducer produces item %d",x);
13    ++mutex;
14 }
15 void consumer()
16 {
17     --mutex;
18     --full;
19     ++empty;
20     printf("\nConsumer consumes item %d",x);
21     x--;
22     ++mutex;
23 }
24 int main()
25 {
26     int n, i;
27     printf("\n1. Press 1 for Producer"
28           "\n2. Press 2 for Consumer"
29           "\n3. Press 3 for Exit");
30     #pragma omp critical
31     for (i = 1; i > 0; i++)
32     {
33         printf("\nEnter your choice:");
34         scanf("%d", &n);
35         switch (n)
36         {
37             case 1:
38                 if ((mutex == 1)&& (empty != 0))
39                 {
40                     producer();
41                 }
42                 else
43                 {
44                     printf("Buffer is full!");
45                 }
46                 break;
47             case 2:
48                 if ((mutex == 1)&& (full != 0))
49                 {
```

D:\Classical process synchronization.C

1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit
Enter your choice:1

Producer produces item 1
Enter your choice:2

Consumer consumes item 1
Enter your choice:3

Process exited after 3.491 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 53 Col: 14 Sel: 0 Lines: 62 Length: 912 Insert Done parsing in 0.016 seconds

D:\thread related concepts.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C thread related concepts.C

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 void* func(void* arg)
6 {
7     pthread_detach(pthread_self());
8
9     printf("Inside the thread\n");
10    pthread_exit(NULL);
11 }
12
13 void fun()
14 {
15     pthread_t ptid;
16     pthread_create(&ptid, NULL, &func, NULL);
17     printf("This line may be printed"
18           " before thread terminates\n");
19     if(pthread_equal(ptid, pthread_self()))
20         printf("Threads are equal\n");
21     else
22         printf("Threads are not equal\n");
23     pthread_join(ptid, NULL);
24
25     printf("This line will be printed"
26           " after thread ends\n");
27
28     pthread_exit(NULL);
29 }
30 int main()
31 {
32     fun();
33     return 0;
34 }
```

D:\thread related concepts.exe

This line may be printed before thread terminates
Inside the thread
Threads are not equal
This line will be printed after thread ends

Process exited after 0.0166 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\thread related concepts.exe
- Output Size: 183.5234375 KiB
- Compilation Time: 0.23s

Line: 30 Col: 11 Sel: 0 Lines: 34 Length: 617 Insert Done parsing in 0.016 seconds

D:\FIFO paging technique.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C thread related concepts.C FIFO paging technique.C

```
1 #include<stdio.h>
2 int main()
3 {
4     int incomingStream[] = {5,10,12,6,22,44};
5     int pageFaults = 0;
6     int frames = 3;
7     int m, n, s, pages;
8     pages = sizeof(incomingStream)/sizeof(incomingStream[0]);
9     printf(" Incoming \t Frame 1 \t Frame 2 \t Frame 3 ");
10    int temp[ frames ];
11    for(m = 0; m < frames; m++)
12    {
13        temp[m] = -1;
14    }
15    for(m = 0; m < pages; m++)
16    {
17        s = 0;
18        for(n = 0; n < frames; n++)
19        {
20            if(incomingStream[m] == temp[n])
21            {
22                s++;
23                pageFaults--;
24            }
25        }
26        pageFaults++;
27        if((pageFaults <= frames) && (s == 0))
28        {
29            temp[m] = incomingStream[m];
30        }
31        else if(s == 0)
32        {
33            temp[(pageFaults - 1) % frames] = incomingStream[m];
34        }
35        printf("\n");
36        printf("%d\t\t\t",incomingStream[m]);
37        for(n = 0; n < frames; n++)
38        {
39            if(temp[n] != -1)
40            printf(" %d\t\t\t", temp[n]);
41            else
42            printf(" - \t\t\t");
43        }
44    }
45    printf("\nTotal Page Faults:\t%d\n", pageFaults);
46    return 0;
47 }
```

D:\FIFO paging technique.exe

Incoming	Frame 1	Frame 2	Frame 3
5	5	-	-
10	5	10	-
12	5	10	12
6	6	10	12
22	6	22	12
44	6	22	44

Total Page Faults: 6

Process exited after 0.01403 seconds with return value 0
Press any key to continue . . . |

Compiler (4) Resources Compile Log Debug Find Results

Line: 34 Col: 12 Sel: 0 Lines: 47 Length: 1302 Insert Done parsing in 0 seconds

D:\LRU.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C thread related concepts.C FIFO paging technique.C [*] LRU.c

```
1 #include<stdio.h>
2 int findLRU(int time[], int n)
3 {
4     int i, minimum = time[0], pos = 0;
5     for(i = 1; i < n; ++i)
6     {
7         if(time[i] < minimum)
8         {
9             minimum = time[i];
10            pos = i;
11        }
12    }
13    return pos;
14 }
15 int main()
16 {
17     int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j, pos, faults = 0;
18     printf("Enter number of frames: ");
19     scanf("%d", &no_of_frames);
20     printf("Enter number of pages: ");
21     scanf("%d", &no_of_pages);
22     printf("Enter reference string: ");
23     for(i = 0; i < no_of_pages; ++i)
24     {
25         scanf("%d", &pages[i]);
26     }
27     for(i = 0; i < no_of_frames; ++i)
28     {
29         frames[i] = -1;
30     }
31     for(i = 0; i < no_of_pages; ++i)
32     {
33         flag1 = flag2 = 0;
34         for(j = 0; j < no_of_frames; ++j)
35         {
36             if(frames[j] == pages[i])
37             {
38                 counter++;
39                 time[j] = counter;
40                 flag1 = flag2 = 1;
41                 break;
42             }
43             if(flag1 == 0)
44             {
45                 for(j = 0; j < no_of_frames; ++j)
46                 {
47                     if(frames[j] == -1)
48                     {
49                         counter++;
50                         frames[j] = pages[i];
51                         time[j] = counter;
52                         flag1 = 1;
53                         break;
54                     }
55                 }
56             }
57         }
58     }
59     for(i = 0; i < no_of_frames; ++i)
60     {
61         printf("%d %d %d %d\n", pages[i], frames[i], time[i], flag1);
62     }
63     printf("Total Page Faults = %d", faults);
64 }
```

D:\LRU.exe

Enter number of frames: 4
Enter number of pages: 7
Enter reference string: 8
5
6
3
2
11
20
8 -1 -1 -1
8 5 -1 -1
8 5 6 -1
8 5 6 3
2 5 6 3
2 11 6 3
2 11 20 3

Total Page Faults = 7

Process exited after 8.35 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 67 Col: 6 Sel: 0 Lines: 73 Length: 1458 Insert Done parsing

D:\optimal paging.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C thread related concepts.C FIFO paging technique.C LRU.c optimal paging.c

```
1 #include<stdio.h>
2 int main()
3 {
4     int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos, max, faults = 0;
5     printf("Enter number of frames: ");
6     scanf("%d", &no_of_frames);
7     printf("Enter number of pages: ");
8     scanf("%d", &no_of_pages);
9     printf("Enter page reference string: ");
10    for(i = 0; i < no_of_pages; ++i){
11        scanf("%d", &pages[i]);
12    }
13    for(i = 0; i < no_of_frames; ++i){
14        frames[i] = -1;
15    }
16    for(i = 0; i < no_of_pages; ++i){
17        flag1 = flag2 = 0;
18
19        for(j = 0; j < no_of_frames; ++j){
20            if(frames[j] == pages[i]){
21                flag1 = flag2 = 1;
22                break;
23            }
24        }
25        if(flag1 == 0){
26            for(j = 0; j < no_of_frames; ++j){
27                if(frames[j] == -1){
28                    if(faults++ > max)
29                        max = faults;
30                    frames[j] = pages[i];
31                    flag2 = 1;
32                    break;
33                }
34            }
35        if(flag2 == 0){
36            flag3 = 0;
37            for(j = 0; j < no_of_frames; ++j){
38                temp[j] = -1;
39
40                for(k = i + 1; k < no_of_pages; ++k){
41                    if(frames[j] == pages[k]){
42                        temp[j] = k;
43                        break;
44                    }
45                }
46            }
47            for(j = 0; j < no_of_frames; ++j){
48                if(temp[j] == -1){
49                    pos = j;
50                }
51            }
52        }
53    }
54 }
```

D:\optimal paging.exe

Enter number of frames: 3
Enter number of pages: 5
Enter page reference string: 1
2
3
6
5

1 -1 -1
1 2 -1
1 2 3
6 2 3
5 2 3

Total Page Faults = 5

Process exited after 39.27 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 71 Col: 10 Sel: 0 Lines: 75 Length: 2087 Insert Done parsing in 0 seconds

D:\file allocation files are sorted both physically and logically.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file management operations.c function of ls UNIX command.C GREP UNIX commands.C Classical process synchronization.C thread related concepts.C FIFO paging technique.C LRU.c optimal paging.c file allocation files are sorted both physically and logically.c

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 int main()
5 {
6     int f[50], i, st, len, j, c, k, count = 0;
7     for(i=0;i<50;i++)
8         f[i]=0;
9     printf("Files Allocated are : \n");
10    scanf("%d", &a);
11    x : count=0;
12    printf("Enter starting block and length of files: ");
13    scanf("%d%d", &st, &len);
14    for(k=st;k<(st+len);k++)
15        if(f[k]==0)
16            count++;
17        if(len==count)
18        {
19            for(j=st;j<(st+len);j++)
20                if(f[j]==0)
21                {
22                    f[j]=1;
23                    printf("%d\t%d\n", j, f[j]);
24                }
25            if(j!=(st+len-1))
26                printf("The file is allocated to disk\n");
27        }
28        else
29        printf("The file is not allocated \n");
30        printf("Do you want to enter more file(Yes - 1/No - 0)");
31        scanf("%d", &c);
32        if(c==1)
33            goto x;
34        else
35            exit(0);
36        getch();
37 }
```

D:\file allocation files are sort

Files Allocated are :
333
Enter starting block and length of files: 3
6
3 1
4 1
5 1
6 1
7 1
8 1
The file is allocated to disk
Do you want to enter more file(Yes - 1/No - 0)1
Enter starting block and length of files: 2
5
The file is not allocated
Do you want to enter more file(Yes - 1/No - 0)1
Enter starting block and length of files: 1
3
The file is not allocated
Do you want to enter more file(Yes - 1/No - 0)0

Process exited after 15.8 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 26 Col: 43 Sel: 0 Lines: 37 Length: 684 Insert Done parsing in 0 seconds

D:\file pointers together to index block.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug file pointers together to index block.C

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 int main()
5 {
6     int f[50], index[50], i, n, st, len, j, c, k, ind, count=0;
7     for(i=0;i<50;i++)
8         f[i]=0;
9     x:printf("Enter the index block: ");
10    scanf("%d",&ind);
11    if(f[ind]!=1)
12    {
13        printf("Enter no of blocks needed and no of files for the index %d on the disk : \n", ind);
14        scanf("%d",&n);
15    }
16    else
17    {
18        printf("%d index is already allocated \n",ind);
19        goto x;
20    }
21    y: count=0;
22    for(i=0;i<n;i++)
23    {
24        scanf("%d", &index[i]);
25        if(f[index[i]]==0)
26            count++;
27    }
28    if(count==n)
29    {
30        for(j=0;j<n;j++)
31            f[index[j]]=1;
32        printf("Allocated\n");
33        printf("File Indexed\n");
34        for(k=0;k<n;k++)
35            printf("%d----->%d : %d\n",ind,index[k],f[index[k]]);
36    }
37    else
38    {
39        printf("File in the index is already allocated \n");
40        printf("Enter another file indexed");
41        goto y;
42    }
43    printf("Do you want to enter more file(Yes - 1/No - 0)");
44    scanf("%d", &c);
45    if(c==1)
46        goto x;
47    else
48        exit(0);
49    getch();
```

D:\file pointers together to in X + ↻

Enter the index block: 5
Enter no of blocks needed and no of files for the index 5 on the disk :
5
2
3
6
9
2
Allocated
File Indexed
5----->2 : 1
5----->3 : 1
5----->6 : 1
5----->9 : 1
5----->2 : 1
Do you want to enter more file(Yes - 1/No - 0)1
Enter the index block: 3
3 index is already allocated
Enter the index block: 9
9 index is already allocated
Enter the index block: 10
Enter no of blocks needed and no of files for the index 10 on the disk
2
6
9
File in the index is already allocated
Enter another file indexed11
15
Allocated
File Indexed

Compiler Resources Compile Log Debug Find Results

Line: 31 Col: 15 Sel: 0 Lines: 50 Length: 930 Insert Done parsing in 0.109 seconds

D:\linked allocation.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug [*] file pointers together to index block.C linked allocation.C [*] Untitled3

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 int main()
5 {
6     int f[50], p, i, st, len, j, c, k, a;
7
8     for(i=0;i<50;i++)
9     f[i]=0;
10    printf("Enter how many blocks already allocated: ");
11    scanf("%d",&p);
12    printf("Enter blocks already allocated: ");
13    for(i=0;i<p;i++)
14    {
15        scanf("%d",&a);
16        f[a]=1;
17    }
18    x: printf("Enter index starting block and length: ");
19    scanf("%d%d", &st,&len);
20    k=len;
21    if(f[st]==0)
22    {
23        for(j=st;j<(st+k);j++)
24        {
25            if(f[j]==0)
26            {
27                f[j]=1;
28                printf("%d----->%d\n",j,f[j]);
29            }
30        else
31        {
32            printf("%d Block is already allocated \n",j);
33            k++;
34        }
35    }
36}
37 else
38 printf("%d starting block is already allocated \n",st);
39 printf("Do you want to enter more file(Yes - 1/No - 0)");
40 scanf("%d", &c);
41 if(c==1)
42 goto x;
43 else
44 exit(0);
45 getch();
46 }
```

D:\linked allocation.exe

Enter how many blocks already allocated: 5
Enter blocks already allocated: 3
1
2
6
9
Enter index starting block and length: 10
11
10----->1
11----->1
12----->1
13----->1
14----->1
15----->1
16----->1
17----->1
18----->1
19----->1
20----->1
Do you want to enter more file(Yes - 1/No - 0)1
Enter index starting block and length: 2
1
2 starting block is already allocated
Do you want to enter more file(Yes - 1/No - 0)0

Process exited after 30.77 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 4 Col: 4 Sel: 0 Lines: 46 Length: 762 Insert Done parsing in 0 seconds

D:\DISK SCHEDULING.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug [*] file pointers together to index block.C linked allocation.C FCFS disk scheduling.C DISK SCHEDULING.C

(globals)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int main()
4 {
5     int RQ[100],i,n,TotalHeadMoment=0,initial;
6     printf("Enter the number of Requests\n");
7     scanf("%d",&n);
8     printf("Enter the Requests sequence\n");
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&RQ[i]);
12    }
13    printf("Enter initial head position\n");
14    scanf("%d",&initial);
15    for(i=0;i<n;i++)
16    {
17        TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
18        initial=RQ[i];
19    }
20    printf("Total head moment is %d",TotalHeadMoment);
21    return 0;
22 }
```

D:\DISK SCHEDULING.exe

Enter the number of Requests
3
Enter the Requests sequence
1
5
19
Enter initial head position
5
Total head moment is 22

Process exited after 9.248 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 22 Col: 2 Sel: 0 Lines: 22 Length: 546 Insert Done parsing in 0.032 seconds

D:\SCAN disk scheduling.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug [*] file pointers together to index block.C linked allocation.C FCFS disk scheduling.C DISK SCHEDULING.C SCAN disk scheduling.C

```
1 #include <stdio.h>
2 #include <math.h>
3 int main()
4 {
5     int queue[20], n, head, i, j, k, seek = 0, max, diff, temp, queue1[20],
6     queue2[20], temp1 = 0, temp2 = 0;
7     float avg;
8     printf("Enter the max range of disk\n");
9     scanf("%d", &max);
10    printf("Enter the initial head position\n");
11    scanf("%d", &head);
12    printf("Enter the size of queue request\n");
13    scanf("%d", &n);
14    printf("Enter the queue of disk positions to be read\n");
15    for (i = 1; i <= n; i++)
16    {
17        scanf("%d", &temp);
18        if (temp >= head)
19        {
20            queue1[temp1] = temp;
21            temp1++;
22        }
23        else
24        {
25            queue2[temp2] = temp;
26            temp2++;
27        }
28    }
29    for (i = 0; i < temp1 - 1; i++)
30    {
31        for (j = i + 1; j < temp1; j++)
32        {
33            if (queue1[i] > queue1[j])
34            {
35                temp = queue1[i];
36                queue1[i] = queue1[j];
37                queue1[j] = temp;
38            }
39        }
40    }
41    for (i = 0; i < temp2 - 1; i++)
42    {
43        for (j = i + 1; j < temp2; j++)
44        {
45            if (queue2[i] < queue2[j])
46            {
47                temp = queue2[i];
48                queue2[i] = queue2[j];
49            }
50        }
51    }
52    for (i = 0; i < temp1; i++)
53    {
54        for (j = i + 1; j < temp2; j++)
55        {
56            if (queue1[i] < queue2[j])
57            {
58                temp = queue1[i];
59                queue1[i] = queue2[j];
60                queue2[j] = temp;
61            }
62        }
63    }
64    for (i = 0; i < temp1; i++)
65    {
66        for (j = i + 1; j < temp2; j++)
67        {
68            if (queue1[i] > queue2[j])
69            {
70                temp = queue1[i];
71                queue1[i] = queue2[j];
72                queue2[j] = temp;
73            }
74        }
75    }
76    for (i = 0; i < temp1; i++)
77    {
78        for (j = i + 1; j < temp2; j++)
79        {
80            if (queue1[i] < queue2[j])
81            {
82                temp = queue1[i];
83                queue1[i] = queue2[j];
84                queue2[j] = temp;
85            }
86        }
87    }
88    for (i = 0; i < temp1; i++)
89    {
90        for (j = i + 1; j < temp2; j++)
91        {
92            if (queue1[i] > queue2[j])
93            {
94                temp = queue1[i];
95                queue1[i] = queue2[j];
96                queue2[j] = temp;
97            }
98        }
99    }
100   for (i = 0; i < temp1; i++)
101   {
102       for (j = i + 1; j < temp2; j++)
103       {
104           if (queue1[i] < queue2[j])
105           {
106               temp = queue1[i];
107               queue1[i] = queue2[j];
108               queue2[j] = temp;
109           }
110       }
111   }
112   for (i = 0; i < temp1; i++)
113   {
114       for (j = i + 1; j < temp2; j++)
115       {
116           if (queue1[i] > queue2[j])
117           {
118               temp = queue1[i];
119               queue1[i] = queue2[j];
120               queue2[j] = temp;
121           }
122       }
123   }
124   for (i = 0; i < temp1; i++)
125   {
126       for (j = i + 1; j < temp2; j++)
127       {
128           if (queue1[i] < queue2[j])
129           {
130               temp = queue1[i];
131               queue1[i] = queue2[j];
132               queue2[j] = temp;
133           }
134       }
135   }
136   for (i = 0; i < temp1; i++)
137   {
138       for (j = i + 1; j < temp2; j++)
139       {
140           if (queue1[i] > queue2[j])
141           {
142               temp = queue1[i];
143               queue1[i] = queue2[j];
144               queue2[j] = temp;
145           }
146       }
147   }
148   for (i = 0; i < temp1; i++)
149   {
150       for (j = i + 1; j < temp2; j++)
151       {
152           if (queue1[i] < queue2[j])
153           {
154               temp = queue1[i];
155               queue1[i] = queue2[j];
156               queue2[j] = temp;
157           }
158       }
159   }
160   for (i = 0; i < temp1; i++)
161   {
162       for (j = i + 1; j < temp2; j++)
163       {
164           if (queue1[i] > queue2[j])
165           {
166               temp = queue1[i];
167               queue1[i] = queue2[j];
168               queue2[j] = temp;
169           }
170       }
171   }
172   for (i = 0; i < temp1; i++)
173   {
174       for (j = i + 1; j < temp2; j++)
175       {
176           if (queue1[i] < queue2[j])
177           {
178               temp = queue1[i];
179               queue1[i] = queue2[j];
180               queue2[j] = temp;
181           }
182       }
183   }
184   for (i = 0; i < temp1; i++)
185   {
186       for (j = i + 1; j < temp2; j++)
187       {
188           if (queue1[i] > queue2[j])
189           {
190               temp = queue1[i];
191               queue1[i] = queue2[j];
192               queue2[j] = temp;
193           }
194       }
195   }
196   for (i = 0; i < temp1; i++)
197   {
198       for (j = i + 1; j < temp2; j++)
199       {
200           if (queue1[i] < queue2[j])
201           {
202               temp = queue1[i];
203               queue1[i] = queue2[j];
204               queue2[j] = temp;
205           }
206       }
207   }
208   for (i = 0; i < temp1; i++)
209   {
210       for (j = i + 1; j < temp2; j++)
211       {
212           if (queue1[i] > queue2[j])
213           {
214               temp = queue1[i];
215               queue1[i] = queue2[j];
216               queue2[j] = temp;
217           }
218       }
219   }
220   for (i = 0; i < temp1; i++)
221   {
222       for (j = i + 1; j < temp2; j++)
223       {
224           if (queue1[i] < queue2[j])
225           {
226               temp = queue1[i];
227               queue1[i] = queue2[j];
228               queue2[j] = temp;
229           }
230       }
231   }
232   for (i = 0; i < temp1; i++)
233   {
234       for (j = i + 1; j < temp2; j++)
235       {
236           if (queue1[i] > queue2[j])
237           {
238               temp = queue1[i];
239               queue1[i] = queue2[j];
240               queue2[j] = temp;
241           }
242       }
243   }
244   for (i = 0; i < temp1; i++)
245   {
246       for (j = i + 1; j < temp2; j++)
247       {
248           if (queue1[i] < queue2[j])
249           {
250               temp = queue1[i];
251               queue1[i] = queue2[j];
252               queue2[j] = temp;
253           }
254       }
255   }
256   for (i = 0; i < temp1; i++)
257   {
258       for (j = i + 1; j < temp2; j++)
259       {
260           if (queue1[i] > queue2[j])
261           {
262               temp = queue1[i];
263               queue1[i] = queue2[j];
264               queue2[j] = temp;
265           }
266       }
267   }
268   for (i = 0; i < temp1; i++)
269   {
270       for (j = i + 1; j < temp2; j++)
271       {
272           if (queue1[i] < queue2[j])
273           {
274               temp = queue1[i];
275               queue1[i] = queue2[j];
276               queue2[j] = temp;
277           }
278       }
279   }
280   for (i = 0; i < temp1; i++)
281   {
282       for (j = i + 1; j < temp2; j++)
283       {
284           if (queue1[i] > queue2[j])
285           {
286               temp = queue1[i];
287               queue1[i] = queue2[j];
288               queue2[j] = temp;
289           }
290       }
291   }
292   for (i = 0; i < temp1; i++)
293   {
294       for (j = i + 1; j < temp2; j++)
295       {
296           if (queue1[i] < queue2[j])
297           {
298               temp = queue1[i];
299               queue1[i] = queue2[j];
300               queue2[j] = temp;
301           }
302       }
303   }
304   for (i = 0; i < temp1; i++)
305   {
306       for (j = i + 1; j < temp2; j++)
307       {
308           if (queue1[i] > queue2[j])
309           {
310               temp = queue1[i];
311               queue1[i] = queue2[j];
312               queue2[j] = temp;
313           }
314       }
315   }
316   for (i = 0; i < temp1; i++)
317   {
318       for (j = i + 1; j < temp2; j++)
319       {
320           if (queue1[i] < queue2[j])
321           {
322               temp = queue1[i];
323               queue1[i] = queue2[j];
324               queue2[j] = temp;
325           }
326       }
327   }
328   for (i = 0; i < temp1; i++)
329   {
330       for (j = i + 1; j < temp2; j++)
331       {
332           if (queue1[i] > queue2[j])
333           {
334               temp = queue1[i];
335               queue1[i] = queue2[j];
336               queue2[j] = temp;
337           }
338       }
339   }
340   for (i = 0; i < temp1; i++)
341   {
342       for (j = i + 1; j < temp2; j++)
343       {
344           if (queue1[i] < queue2[j])
345           {
346               temp = queue1[i];
347               queue1[i] = queue2[j];
348               queue2[j] = temp;
349           }
350       }
351   }
352   for (i = 0; i < temp1; i++)
353   {
354       for (j = i + 1; j < temp2; j++)
355       {
356           if (queue1[i] > queue2[j])
357           {
358               temp = queue1[i];
359               queue1[i] = queue2[j];
360               queue2[j] = temp;
361           }
362       }
363   }
364   for (i = 0; i < temp1; i++)
365   {
366       for (j = i + 1; j < temp2; j++)
367       {
368           if (queue1[i] < queue2[j])
369           {
370               temp = queue1[i];
371               queue1[i] = queue2[j];
372               queue2[j] = temp;
373           }
374       }
375   }
376   for (i = 0; i < temp1; i++)
377   {
378       for (j = i + 1; j < temp2; j++)
379       {
380           if (queue1[i] > queue2[j])
381           {
382               temp = queue1[i];
383               queue1[i] = queue2[j];
384               queue2[j] = temp;
385           }
386       }
387   }
388   for (i = 0; i < temp1; i++)
389   {
390       for (j = i + 1; j < temp2; j++)
391       {
392           if (queue1[i] < queue2[j])
393           {
394               temp = queue1[i];
395               queue1[i] = queue2[j];
396               queue2[j] = temp;
397           }
398       }
399   }
300 }
```

D:\SCAN disk scheduling.exe

20
Enter the initial head position
10
Enter the size of queue request
5
Enter the queue of disk positions to be read
6
3
19
22
26
Disk head moves from 10 to 19 with seek 9
Disk head moves from 19 to 22 with seek 3
Disk head moves from 22 to 26 with seek 4
Disk head moves from 26 to 20 with seek 6
Disk head moves from 20 to 6 with seek 14
Disk head moves from 6 to 3 with seek 3
Disk head moves from 3 to 0 with seek 3
Total seek time is 42
Average seek time is 8.400000

Process exited after 9.912 seconds with return value 0
Press any key to continue . . .

Line: 66 Col: 29 Sel: 0 Lines: 72 Length: 1912 Insert Done parsing in 0.015 seconds

D:\C_SCAN disk scheduling.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug C_SCAN disk scheduling.C

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int main()
4 {
5     int RQ[100],i,j,n,TotalHeadMoment=0,initial,size,move;
6     printf("Enter the number of Requests\n");
7     scanf("%d",&n);
8     printf("Enter the Requests sequence\n");
9     for(i=0;i<n;i++)
10    scanf("%d",&RQ[i]);
11    printf("Enter initial head position\n");
12    scanf("%d",&initial);
13    printf("Enter total disk size\n");
14    scanf("%d",&size);
15    printf("Enter the head movement direction for high 1 and for low 0\n");
16    scanf("%d",&move);
17    for(i=0;i<n;i++)
18    {
19        for( j=0;j<n-i-1;j++)
20        {
21            if(RQ[j]>RQ[j+1])
22            {
23                int temp;
24                temp=RQ[j];
25                RQ[j]=RQ[j+1];
26                RQ[j+1]=temp;
27            }
28        }
29    }
30
31    int index;
32    for(i=0;i<n;i++)
33    {
34        if(initial<RQ[i])
35        {
36            index=i;
37            break;
38        }
39    }
40
41    if(move==1)
42    {
43        for(i=index;i<n;i++)
44        {
45            TotalHeadMoment=TotalHeadMoment+abs(RQ[i]-initial);
46            initial=RQ[i];
47        }
48        TotalHeadMoment=TotalHeadMoment+abs(size-RQ[i-1]-1);
49        TotalHeadMoment=TotalHeadMoment+abs(size-1-0);
50    }
51 }
```

D:\C_SCAN disk scheduling.exe

Enter the number of Requests
10
Enter the Requests sequence
5
9
12
16
19
22
23
30
36
55
Enter initial head position
19
Enter total disk size
8
Enter the head movement direction for high 1 and for low 0
1
Total head movement is 110

Process exited after 38.29 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 76 Length: 1948 Insert Done parsing in 0.078 seconds

D:\File access permission.C - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug [*] file pointers together to index block.C linked allocation.C FCFS disk scheduling.C DISK SCHEDULING.C SCAN disk scheduling.C C_SCAN disk scheduling.C File access permission.C

(globals)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 int main(int argc, char **argv)
6 {
7     int result;
8     char *filename = (char *)malloc(512);
9     if (argc < 2)
10    {
11        strcpy(filename, "/usr/bin/adb");
12    }
13    else
14    {
15        strcpy(filename, argv[1]);
16    }
17    result = access (filename, R_OK);
18    if ( result == 0 )
19    {
20        printf("%s is readable\n",filename);
21    }
22    else
23    {
24        printf("%s is not readable\n",filename);
25    }
26    result = access (filename, W_OK);
27    if ( result == 0 )
28    {
29        printf("%s is Writeable\n",filename);
30    }
31    else
32    {
33        printf("%s is not Writeable\n",filename);
34    }
35    result = access (filename, X_OK);
36    if ( result == 0 )
37    {
38        printf("%s is executable\n",filename);
39    }
40    else
41    {
42        printf("%s is not executable\n",filename);
43    }
44    free(filename);
45    return 0;
46 }
```

D:\File access permission.exe

/usr/bin/adb is not readable
/usr/bin/adb is not Writeable
/usr/bin/adb is not executable

Process exited after 0.01437 seconds with return value 0
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results

Line: 39 Col: 7 Sel: 0 Lines: 46 Length: 953 Insert Done parsing in 0.016 seconds