



main.cpp

Run

Output

```
1 // C program for the above approach
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 // Initialize a mutex to 1
6 int mutex = 1;
7
8 // Number of full slots as 0
9 int full = 0;
10
11 // Number of empty slots as size
12 // of buffer
13 int empty = 10, x = 0;
14
15 // Function to produce an item and
16 // add it to the buffer
17 void producer()
18 {
19     // Decrease mutex value by 1
20     --mutex;
```

/tmp/DKtEf8M7J1.o

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

== Code Execution Successful ==

Activate Window
Go to Settings to activate

EN

34°C Partly cloudy

HCL

programiz

C++ Online Compiler

Premium Coding Courses by Programiz

Programiz PRO

main.cpp

Run

Output

```
1 // C++ implementation of FIFO page replacement
2 // in Operating Systems.
3 #include<bits/stdc++.h>
4 using namespace std;
5
6 // Function to find page faults using FIFO
7 int pageFaults(int pages[], int n, int capacity)
8 {
9     // To represent set of current pages. We use
10    // an unordered_set so that we quickly check
11    // if a page is present in set or not
12    unordered_set<int> s;
13
14    // To store the pages in FIFO manner
15    queue<int> indexes;
16
17    // Start from initial page
18    int page_faults = 0;
19    for (int i=0; i<n; i++)
20    {
```

/tmp/erv2J5beX5.o

6

== Code Execution Successful ==

Activate Wind
Go to Settings to a

Type here to search

EN

34°C Partly cloudy

Gmail YouTube Maps

Programiz C Online Compiler

FRONTEND NATION June 4-7 JOIN THE LARGEST FREE ONLINE FRONTEND EVENT. EVER. Register today →

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <arpa/inet.h>
6
7 // Function to add two numbers
8 int add(int a, int b) {
9     return a + b;
10 }
11
12 // Function to subtract two numbers
13 int subtract(int a, int b) {
14     return a - b;
15 }
16
17 // Function to handle client requests
18 void handle_client_request(int client_socket) {
19     int choice, num1, num2, result;
```

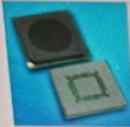
Run Output

/tmp/6DulE7Ll0f.o
Server listening on port 2000...

Type here to search Earnings upcoming 08:50 09-05-2024 HCL

gramiz

Online Compiler



TJA1081GTS/0 - Order Electronic Components...

SPONSORED BY WWW.ARIATECH.COM/ELECTRONIC/COMPONENTS

LEARN MORE

main.cpp

```
1 #include <iostream>
2 #include <thread>
3 #include <mutex>
4 #include <chrono>
5 #include <vector>
6
7 using namespace std;
8
9 const int NUM_PHILOSOPHERS = 5;
10 const int NUM_MEALS = 3;
11
12 mutex forks[NUM_PHILOSOPHERS];
13 vector<int> meals_eaten(NUM_PHILOSOPHERS, 0);
14
15 void philosopher(int id) {
16     int left_fork = id;
17     int right_fork = (id + 1) % NUM_PHILOSOPHERS;
18
19     for (int i = 0; i < NUM_MEALS; i++) {
20         // Thinking
```



Run

Output

```
Philosopher 0 is thinking
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 0 picked up fork 0 (left)
Philosopher 0 picked up fork 1 (right)
Philosopher 1 picked up fork 1 (left)
Philosopher 1 picked up fork 2 (right)
Philosopher 2 picked up fork 2 (left)
Philosopher 2 picked up fork 3 (right)
Philosopher 3 picked up fork 3 (left)
Philosopher 3 picked up fork 4 (right)
Philosopher 4 picked up fork 4 (left)
Philosopher 4 picked up fork 0 (right)
Philosopher 0 is eating
Philosopher 1 is eating
Philosopher 2 is eating
Philosopher 3 is eating
Philosopher 4 is eating
```

Activate Windows
Go to Settings to activate Windows
09-22 09-05-2024

HCL

Tube Maps

amiz

line Compiler

TJA1081GTS/0 - Order Electronic Components...

SPONSORED BY WWW.ARIAT-TECH.COM/ELECTRONIC/COMPONENTS

LEARN MORE

Programiz PR

in.cpp

```
#include <iostream>
#include <thread>
#include <mutex>
#include <chrono>
#include <vector>
using namespace std;
const int NUM_PHILOSOPHERS = 5;
const int NUM_MEALS = 3;
mutex forks[NUM_PHILOSOPHERS];
vector<int> meals_eaten(NUM_PHILOSOPHERS, 0);
void philosopher(int id) {
    int left_fork = id;
    int right_fork = (id + 1) % NUM_PHILOSOPHERS;
    for (int i = 0; i < NUM_MEALS; i++) {
        // Thinking
        cout << "Philosopher " << id << " is thinking" << endl;
        this_thread::sleep_for(chrono::seconds(1));
        cout << "Philosopher " << id << " picked up fork " << left_fork << endl;
        cout << "Philosopher " << id << " picked up fork " << right_fork << endl;
        cout << "Philosopher " << id << " put down fork " << right_fork << endl;
        cout << "Philosopher " << id << " put down fork " << left_fork << endl;
    }
}
```

Output

```
Philosopher 3 is eating
Philosopher 4 is eating
Philosopher 0 put down fork 1 (right)
Philosopher 0 put down fork 0 (left)
Philosopher 1 put down fork 2 (right)
Philosopher 1 put down fork 1 (left)
Philosopher 2 put down fork 3 (right)
Philosopher 2 put down fork 2 (left)
Philosopher 3 put down fork 4 (right)
Philosopher 3 put down fork 3 (left)
Philosopher 4 put down fork 0 (right)
Philosopher 4 put down fork 4 (left)
Philosopher 0 is thinking
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 0 picked up fork 0 (left)
Philosopher 0 picked up fork 1 (right)
Philosopher 1 picked up fork 1 (left)
```

Type here to search

EN 33°C Partly cloudy 08:23 08-05-2014

gramiz
Online Compiler

TJA1081GTS/0 - Order Electronic Components...

SPONSORED BY WWW.ARIATECH.COM/ELECTRONIC/COMPONENTS

[LEARN MORE](#)

[Programiz Pro](#)

main.cpp

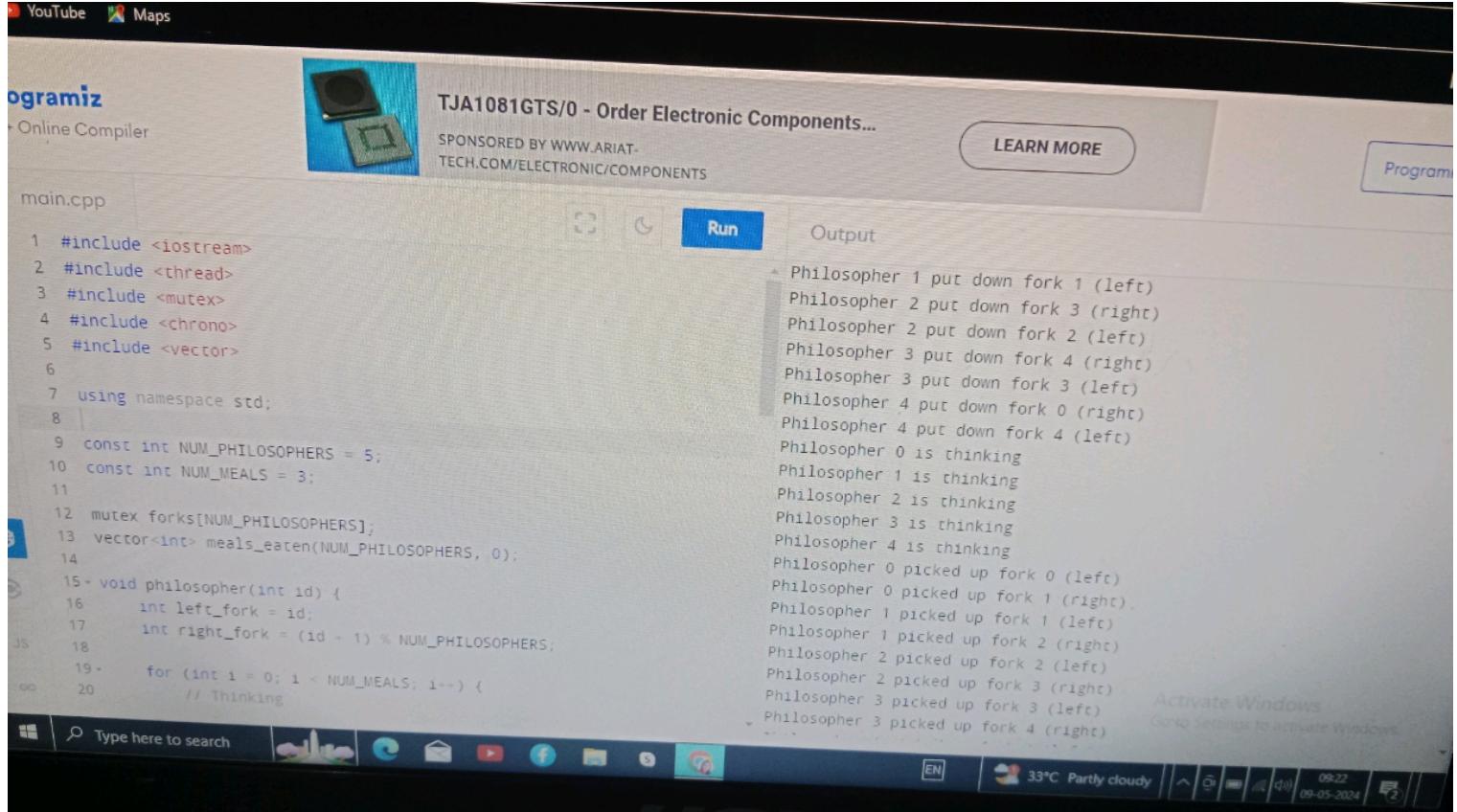
Run Output

```
1 #include <iostream>
2 #include <thread>
3 #include <mutex>
4 #include <chrono>
5 #include <vector>
6
7 using namespace std;
8
9 const int NUM_PHILOSOPHERS = 5;
10 const int NUM_MEALS = 3;
11
12 mutex forks[NUM_PHILOSOPHERS];
13 vector<int> meals_eaten(NUM_PHILOSOPHERS, 0);
14
15 void philosopher(int id) {
16     int left_fork = id;
17     int right_fork = (id + 1) % NUM_PHILOSOPHERS;
18
19     for (int i = 0; i < NUM_MEALS; i++) {
20         // Thinking
```

Philosopher 0 picked up fork 0 (left)
Philosopher 0 picked up fork 1 (right)
Philosopher 1 picked up fork 1 (left)
Philosopher 1 picked up fork 2 (right)
Philosopher 2 picked up fork 2 (left)
Philosopher 2 picked up fork 3 (right)
Philosopher 3 picked up fork 3 (left)
Philosopher 3 picked up fork 4 (right)
Philosopher 4 picked up fork 4 (left)
Philosopher 4 picked up fork 0 (right)
Philosopher 0 is eating
Philosopher 1 is eating
Philosopher 2 is eating
Philosopher 3 is eating
Philosopher 4 is eating
Philosopher 0 put down fork 1 (right)
Philosopher 0 put down fork 0 (left)
Philosopher 1 put down fork 2 (right)
Philosopher 1 put down fork 1 (left)
Philosopher 2 put down fork 3 (right)
Philosopher 2 put down fork 2 (left)

Type here to search 33°C Partly cloudy 09:22 09-05-2024

HCL



The screenshot shows a terminal window on a Windows desktop. The terminal displays the output of a C++ program that simulates the Dining Philosophers problem. The program uses mutexes to ensure that only one philosopher can eat at a time. The output shows five philosophers (labeled 0 to 4) picking up their left and right forks, eating, and then putting the forks down. The total number of meals eaten by all philosophers is displayed at the end.

```
YouTube Maps
Programiz
Online Compiler
FRONTEND NATION.COM
June 4-7
JOIN THE LARGEST FREE ONLINE FRONTEND EVENT. EVER.
Register today →
Programiz

main.cpp
1 #include <iostream>
2 #include <thread>
3 #include <mutex>
4 #include <chrono>
5 #include <vector>
6
7 using namespace std;
8
9 const int NUM_PHILOSOPHERS = 5;
10 const int NUM_MEALS = 3;
11
12 mutex forks[NUM_PHILOSOPHERS];
13 vector<int> meals_eaten(NUM_PHILOSOPHERS, 0);
14
15 void philosopher(int id) {
16     int left_fork = id;
17     int right_fork = (id + 1) % NUM_PHILOSOPHERS;
18
19     for (int i = 0; i < NUM_MEALS; i++) {
20         // Thinking
21         cout << "Philosopher " << id << " is thinking..." << endl;
22
23         // Picking up left fork
24         cout << "Philosopher " << id << " picked up fork " << left_fork << endl;
25         this_thread::sleep_for(chrono::seconds(1));
26
27         // Picking up right fork
28         cout << "Philosopher " << id << " picked up fork " << right_fork << endl;
29         this_thread::sleep_for(chrono::seconds(1));
30
31         // Eating
32         cout << "Philosopher " << id << " is eating" << endl;
33         this_thread::sleep_for(chrono::seconds(2));
34
35         // Putting down right fork
36         cout << "Philosopher " << id << " put down fork " << right_fork << endl;
37         this_thread::sleep_for(chrono::seconds(1));
38
39         // Putting down left fork
40         cout << "Philosopher " << id << " put down fork " << left_fork << endl;
41         this_thread::sleep_for(chrono::seconds(1));
42
43         // Thinking
44         cout << "Philosopher " << id << " is thinking..." << endl;
45     }
46
47     // Total meals eaten by philosopher
48     cout << "Meals eaten by each philosopher: " << meals_eaten[id] << endl;
49 }
50
51 int main() {
52     thread t[5];
53
54     for (int i = 0; i < 5; i++) {
55         t[i] = thread(philosopher, i);
56     }
57
58     for (int i = 0; i < 5; i++) {
59         t[i].join();
60     }
61
62     return 0;
63 }
```

Output:

```
Philosopher 3 picked up fork 3 (left)
Philosopher 3 picked up fork 4 (right)
Philosopher 4 picked up fork 4 (left)
Philosopher 4 picked up fork 0 (right)
Philosopher 0 is eating
Philosopher 1 is eating
Philosopher 2 is eating
Philosopher 3 is eating
Philosopher 4 is eating
Philosopher 0 put down fork 1 (right)
Philosopher 0 put down fork 0 (left)
Philosopher 1 put down fork 2 (right)
Philosopher 1 put down fork 1 (left)
Philosopher 2 put down fork 3 (right)
Philosopher 2 put down fork 2 (left)
Philosopher 3 put down fork 4 (right)
Philosopher 3 put down fork 3 (left)
Philosopher 4 put down fork 0 (right)
Philosopher 4 put down fork 4 (left)
Meals eaten by each philosopher: 3
```

Activate Windows
Go to Settings to activate Windows

G programiz.com/cpp-programming/online-compiler/

mail YouTube Maps

Programiz

C++ Online Compiler

FRONTEND NATION.COM June 4-7 JOIN THE LARGEST FREE ONLINE FRONTEND EVENT. EVER. Register today →

main.cpp

```
1 #include <iostream>
2 #include <thread>
3 #include <mutex>
4 #include <chrono>
5 #include <vector>
6
7 using namespace std;
8
9 const int NUM_PHILOSOPHERS = 5;
10 const int NUM_MEALS = 3;
11
12 mutex forks[NUM_PHILOSOPHERS];
13 vector<int> meals_eaten(NUM_PHILOSOPHERS, 0);
14
15 void philosopher(int id) {
16     int left_fork = id;
17     int right_fork = (id + 1) % NUM_PHILOSOPHERS;
18
19     for (int i = 0; i < NUM_MEALS; i++) {
20         // Thinking
```

Run Output

```
Philosopher 2 is eating
Philosopher 3 is eating
Philosopher 4 is eating
Philosopher 0 put down fork 1 (right)
Philosopher 0 put down fork 0 (left)
Philosopher 1 put down fork 2 (right)
Philosopher 1 put down fork 1 (left)
Philosopher 2 put down fork 3 (right)
Philosopher 2 put down fork 2 (left)
Philosopher 3 put down fork 4 (right)
Philosopher 3 put down fork 3 (left)
Philosopher 4 put down fork 0 (right)
Philosopher 4 put down fork 4 (left)
Meals eaten by each philosopher:
Philosopher 0: 3 meals
Philosopher 1: 3 meals
Philosopher 2: 3 meals
Philosopher 3: 3 meals
Philosopher 4: 3 meals
```

Type here to search

Activate Windows Go to Settings to activate Windows

EN 33°C Partly cloudy 09:23

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Premium Coding Courses by Programiz

Programiz PRO

main.c

```
1 #include <windows.h>
2 #include <stdio.h>
3
4 HANDLE mutexReadCount;
5 HANDLE mutexResource;
6 int readCount = 0;
7
8 DWORD WINAPI reader(LPVOID Param) {
9     int readerId = *(int*)Param;
10
11     WaitForSingleObject(mutexReadCount, INFINITE);
12     readCount++;
13
14     if (readCount == 1) {
15         WaitForSingleObject(mutexResource, INFINITE);
16         ReleaseMutex(mutexReadCount);
17
18         // Reading section
19         printf("Reader %d is reading\n", readerId);
20         Sleep(1000); // Simulate reading time
21         printf("Reader %d finished reading\n", readerId);
22     }
}
```

Run Output

```
Reader 1 is reading
Reader 2 is reading
Reader 3 is reading
Reader 4 is reading
Reader 5 is reading
Reader 6 is reading
Reader 7 is reading
Reader 8 is reading
Reader 9 is reading
Reader 10 is reading
Writer 1 is trying to write
Writer 1 is writing
Writer 2 is trying to write
Reader 1 finished reading
Reader 2 finished reading
Reader 3 finished reading
Reader 4 finished reading
Reader 5 finished reading
Reader 6 finished reading
Reader 7 finished reading
Reader 8 finished reading
Reader 9 finished reading
```

Activate Windows
Go to Settings to activate Windows.

EN 31°C Partly cloudy 09-05-2024 09:40

Programiz
C Online Compiler

Premium Coding Courses by Programiz

Programiz PRO

main.c

```
20 Sleep(1000); // Simulate reading time
21 printf("Reader %d finished reading\n", readerId);
22
23 WaitForSingleObject(mutexReadCount, INFINITE);
24 readCount--;
25 if (readCount == 0) {
26 ReleaseMutex(mutexResource);
27 }
28 ReleaseMutex(mutexReadCount);
29
30 return 0;
31
32 }
33
34 DWORD WINAPI writer(LPVOID Param) {
35 int writerId = *((int*)Param);
36
37 printf("Writer %d is trying to write\n", writerId);
38 WaitForSingleObject(mutexResource, INFINITE);
39 printf("Writer %d is writing\n", writerId);
40 Sleep(2000); // Simulate writing time
41 printf("Writer %d finished writing\n", writerId);
```

Run Output

```
Reader 3 finished reading
Reader 4 finished reading
Reader 5 finished reading
Reader 6 finished reading
Reader 7 finished reading
Reader 8 finished reading
Reader 9 finished reading
Reader 10 finished reading
Writer 1 finished writing
Reader 11 is reading
Reader 12 is reading
Reader 13 is reading
Reader 14 is reading
Reader 15 is reading
Reader 11 finished reading
Reader 12 finished reading
Reader 13 finished reading
Reader 14 finished reading
Reader 15 finished reading
Writer 2 is writing
Writer 2 finished writing
```

Activate Windows
Go to Settings to activate Windows.

Type here to search

EN 31°C Partly cloudy 09:40 09-05-2024

programiz

Online Compiler

Programiz PRO

Premium Coding
Courses by Programiz

Learn More

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/socket.h>
5 #include <netinet/in.h>
6
7 int add(int a, int b) {
8     return a + b;
9 }
10
11 int main() {
12     int server_sock, client_sock;
13     struct sockaddr_in server_addr, client_addr;
14     socklen_t addr_size;
15     int buffer[1024];
16
17     // Create socket
18     server_sock = socket(AF_INET, SOCK_STREAM, 0);
19     if (server_sock < 0) {
20         perror("[-]Socket error");
21         exit(1);
22     }
```

Run

Output

```
[+]Server socket created.
[+]Bind to the port number: 7799
[+]Listening...
[+]Client connected.
[+]Data sent: <result>
[+]Connection closed.
```

Type here to search

EN

31°C Partly cloudy

Activate Windows
Go to Settings to activate Windows

C Online Compiler

Run

Output

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/socket.h>
5 #include <netinet/in.h>
6 #include <arpa/inet.h>
7
8 int main() {
9     int client_sock;
10    struct sockaddr_in server_addr;
11    int buffer[1024], result;
12
13    // Create socket
14    client_sock = socket(AF_INET, SOCK_STREAM, 0);
15    if (client_sock < 0) {
16        perror("[-]Socket error");
17        exit(1);
18    }
19    printf("[+]Client socket created.\n");
20
21    // Configure settings
22    memset(&server_addr, 0, sizeof(server_addr));
```

[+]Client socket created.
[+]Connected to server.
[+]Data sent.
[+]Result received: <result>
[+]Connection closed.

Type here to search

EN

Activate Wind
Go to Settings to a

Weather alert