

# The Problem of Dining Philosophers

1.0

Generated by Doxygen 1.7.4

Thu May 24 2012 02:39:37



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	diningphilos.c File Reference . . . . .	3
2.1.1	Detailed Description . . . . .	3
2.1.2	Function Documentation . . . . .	4
2.1.2.1	checkForB . . . . .	4
2.1.2.2	eat . . . . .	4
2.1.2.3	main . . . . .	4
2.1.2.4	philos . . . . .	4
2.1.2.5	think . . . . .	5
2.2	diningphilos.h File Reference . . . . .	5
2.2.1	Detailed Description . . . . .	7
2.2.2	Function Documentation . . . . .	7
2.2.2.1	checkForB . . . . .	7
2.2.2.2	convertStates . . . . .	7
2.2.2.3	disp_philos_states . . . . .	8
2.2.2.4	eat . . . . .	8
2.2.2.5	get_sticks . . . . .	8
2.2.2.6	philos . . . . .	8
2.2.2.7	put_sticks . . . . .	9
2.2.2.8	think . . . . .	9
2.3	monitor.c File Reference . . . . .	9
2.3.1	Detailed Description . . . . .	10
2.3.2	Function Documentation . . . . .	10

2.3.2.1	<a href="#">convertStates</a>	10
2.3.2.2	<a href="#">disp_philo_states</a>	10
2.3.2.3	<a href="#">get_sticks</a>	10
2.3.2.4	<a href="#">put_sticks</a>	11

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">diningphilos.c</a> (Problem of dining philosophers using monitor-concept with pthreads-mutex and pthreads-condvars ) . . . . .	3
<a href="#">diningphilos.h</a> (All the includes, constants, macros, variables and declarations )	5
<a href="#">monitor.c</a> (Monitor functions for diningphilos ) . . . . .	9



## Chapter 2

# File Documentation

### 2.1 diningphilos.c File Reference

Problem of dining philosophers using monitor-concept with pthreads- mutex and pthreads-condvars.

```
#include "diningphilos.h"
```

#### Functions

- int `main` (void)  
*Main function that starts threads and listens for keyboard input.*
- void \* `philo` (void \*pID)  
*The main philosopher function.*
- void `think` (int philoID)  
*Simulates thinking by going through an empty loop.*
- void `eat` (int philoID)  
*Simulates eating by going through an empty loop.*
- void `checkForB` (int philoID)  
*Checks if Thread is blocked.*

#### 2.1.1 Detailed Description

Problem of dining philosophers using monitor-concept with pthreads- mutex and pthreads-condvars.

#### Author

repat, <repat@repat.de>

#### Note

All comments for doxygen

## 2.1.2 Function Documentation

### 2.1.2.1 void checkForB ( int *philoid* )

Checks if Thread is blocked.

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

### 2.1.2.2 void eat ( int *philoid* )

Simulates eating by going through an empty loop.

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

### 2.1.2.3 int main ( void )

Main function that starts threads and listens for keyboard input.

#### Returns

0 if programm exits normally

### 2.1.2.4 void\* philo ( void \* *pid* )

The main philosopher function.

#### Parameters

<i>pid</i>	philosoher ID from thread creation
------------	------------------------------------

#### Returns

nothing



### 2.1.2.5 void think ( int *philoid* )

Simulates thinking by going through an empty loop.

#### Parameters

<i>philoid</i>	philosopher ID from thread creation
----------------	-------------------------------------

#### Returns

nothing

## 2.2 diningphilos.h File Reference

all the includes, constants, macros, variables and declarations

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <assert.h>
#include <semaphore.h>
```

#### Defines

- #define **NPHILO** 5  
*Number of philosophers.*
- #define **ASCII** 48  
*Helps to cope with ASCII.*
- #define **THINK\_LOOP** 100000000  
*Used for the think-loop in [think\(\)](#) as a really large number.*
- #define **EAT\_LOOP** 500000000  
*Used for the eat-loop in [eat\(\)](#) as a really large number.*
- #define **KEYKOMBO** 3  
*For Keyboard-Input.*
- #define **RIGHT**(philoid) ((philoid+1)%NPHILO)  
*this is how the stick IDs match the philosophers on the right side*
- #define **LEFT**(philoid) philoid  
*this is how the stick IDs match the philosophers on the left side*
- #define **RIGHTNEIGHB**(philoid) ((philoid == NPHILO-1)? 0 : philoid+1)  
*this is how neighbours are defined at a round table - right side*
- #define **LEFTNEIGHB**(philoid) ((philoid == 0)? NPHILO-1 : philoid-1)  
*this is how neighbours are defined at a round table - left side*

## Enumerations

- enum **Bool** { **FALSE** = 0, **TRUE** = 1 }  
*boolean variables true and false*
- enum **State** { **THINK** = 0, **HUNGRY** = 1, **EAT** = 2 }  
*definition of the states of the philosophers*
- enum **Sticks** { **FREE** = 0, **IN\_USE** = 1 }  
*definition of the states of the sticks*

## Functions

- void \* **philo** (void \*arg)  
*The main philosopher function.*
- void **think** (int philoID)  
*Simulates thinking by going through an empty loop.*
- void **eat** (int philoID)  
*Simulates eating by going through an empty loop.*
- void **get\_sticks** (int philoID)  
*philosopher tries to get both sticks or waits for sticks to become available(on HIS cond-var), then eats*
- void **put\_sticks** (int philoID)  
*philosopher puts down the sticks, nudges his fellow philo-buddies and enters thinking phase again*
- void **disp\_philo\_states** ()  
*Displays what happens inside the monitor like this OT 1H 2E 3T 4T in which T stands for THINK, H for HUNGRY and E for EAT.*
- char **convertStates** (State philoState)  
*Converts the states into their first letter.*
- void **checkForB** (int philoID)  
*Checks if Thread is blocked.*

## Variables

- State **philoStates** [NPHILO]  
*states of the N philosophers*
- Sticks **stickStates** [NPHILO]  
*states of the N sticks*
- pthread\_mutex\_t **mutex**  
*mutual exclusion with pthreads*
- pthread\_cond\_t **cond** [NPHILO]  
*condition variables with pthreads - one for every philosopher*
- sem\_t **semaphore** [NPHILO]  
*semaphors with pthreads - one for every philosopher*

- int `tmp` [NPHILO]  
*philoids*
- char `keyinput` [KEYKOMBO]  
*For keyboard input.*
- char `listen` [NPHILO]  
*For transmitting b, u or p to the philosophers.*

### 2.2.1 Detailed Description

all the includes, constants, macros, variables and declarations

#### Author

repat, [repat@repat.de](mailto:repat@repat.de)

#### Note

All comments for doxygen

### 2.2.2 Function Documentation

#### 2.2.2.1 void checkForB ( int *philoid* )

Checks if Thread is blocked.

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

#### 2.2.2.2 char convertStates ( State *philostate* )

Converts the states into their first letter.

#### Parameters

<i>philostate</i>	state the philosoher is in(THINK, HUNGRY or EAT)
-------------------	--

#### Returns

T/H/E for THINK, HUNGRY or EAT

**2.2.2.3 void disp\_philo\_states ( )**

Displays what happens inside the monitor like this OT 1H 2E 3T 4T in which T stands for THINK, H for HUNGRY and E for EAT.

**Returns**

nothing

**2.2.2.4 void eat ( int *philoid* )**

Simulates eating by going through an empty loop.

**Parameters**

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

**Returns**

nothing

**2.2.2.5 void get\_sticks ( int *philoid* )**

philosopher tries to get both sticks or waits for sticks to become available(on HIS cond-var), then eats

**Parameters**

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

**Returns**

nothing

**2.2.2.6 void\* philo ( void \* *pID* )**

The main philosopher function.

**Parameters**

<i>pID</i>	philosoher ID from thread creation
------------	------------------------------------

**Returns**

nothing

### 2.2.2.7 void put\_sticks ( int *philoid* )

philosopher puts down the sticks, nudges his fellow philo-buddies and enters thinking phase again

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

### 2.2.2.8 void think ( int *philoid* )

Simulates thinking by going through an empty loop.

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

## 2.3 monitor.c File Reference

monitor functions for diningphilos

```
#include "diningphilos.h"
```

### Functions

- void [get\\_sticks](#) (int philoID)  
*philosophe tries to get both sticks or waits for sticks to become available(on HIS cond-var), then eats*
- void [put\\_sticks](#) (int philoID)  
*philosopher puts down the sticks, nudges his fellow philo-buddies and enters thinking phase again*
- void [disp\\_philo\\_states](#) ()  
*Displays what happens inside the monitor like this OT 1H 2E 3T 4T in which T stands for THINK, H for HUNGRY and E for EAT.*
- char [convertStates](#) (State philoState)  
*Converts the states into their first letter.*

### 2.3.1 Detailed Description

monitor functions for diningphilos

#### Author

repat, [repat@repat.de](mailto:repat@repat.de)

#### Note

All comments for doxygen

### 2.3.2 Function Documentation

#### 2.3.2.1 char convertStates ( State *philoState* )

Converts the states into their first letter.

#### Parameters

<i>philoState</i>	state the philosoher is in(THINK, HUNGRY or EAT)
-------------------	--

#### Returns

T/H/E for THINK, HUNGRY or EAT

#### 2.3.2.2 void disp\_philo\_states ( )

Displays what happens inside the monitor like this OT 1H 2E 3T 4T in which T stands for THINK, H for HUNGRY and E for EAT.

#### Returns

nothing

#### 2.3.2.3 void get\_sticks ( int *philoid* )

philosopher tries to get both sticks or waits for sticks to become available(on HIS cond-var), then eats

#### Parameters

<i>philoid</i>	philosoher ID from thread creation
----------------	------------------------------------

#### Returns

nothing

#### 2.3.2.4 void put\_sticks ( int *philoid* )

philosopher puts down the sticks, nudges his fellow philo-buddies and enters thinking phase again

#### Parameters

<i>philoid</i>	philosopher ID from thread creation
----------------	-------------------------------------

#### Returns

nothing

# Index

- checkForB
  - diningphilos.c, [4](#)
  - diningphilos.h, [7](#)
- convertStates
  - diningphilos.h, [7](#)
  - monitor.c, [10](#)
- diningphilos.c, [3](#)
  - checkForB, [4](#)
  - eat, [4](#)
  - main, [4](#)
  - philos, [4](#)
  - think, [4](#)
- diningphilos.h, [5](#)
  - checkForB, [7](#)
  - convertStates, [7](#)
  - disp\_philo\_states, [7](#)
  - eat, [8](#)
  - get\_sticks, [8](#)
  - philos, [8](#)
  - put\_sticks, [8](#)
  - think, [9](#)
- disp\_philo\_states
  - diningphilos.h, [7](#)
  - monitor.c, [10](#)
- eat
  - diningphilos.c, [4](#)
  - diningphilos.h, [8](#)
- get\_sticks
  - diningphilos.h, [8](#)
  - monitor.c, [10](#)
- main
  - diningphilos.c, [4](#)
- monitor.c, [9](#)
  - convertStates, [10](#)
  - disp\_philo\_states, [10](#)
  - get\_sticks, [10](#)
  - put\_sticks, [10](#)
- philos
  - diningphilos.c, [4](#)
  - diningphilos.h, [8](#)
- put\_sticks
  - diningphilos.h, [8](#)
  - monitor.c, [10](#)
- think
  - diningphilos.c, [4](#)
  - diningphilos.h, [9](#)