The Problem of Dining Philosophers

1.0

Generated by Doxygen 1.7.4

Thu May 24 2012 02:39:37

Contents

Chapter 1

File Index

1.1 File List

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

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

philoID	philosoher ID from thread creation

Returns

nothing

2.1.2.2 void eat (int *philoID*)

Simulates eating by going through an empty loop.

Parameters

philoID	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

pID	philosoher ID from thread creation

Returns

nothing

2.1.2.5 void think (int *philoID*)

Simulates thinking by going through an empty loop.

Parameters

philoID	philosoher 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`

Note

All comments for doxygen

2.2.2 Function Documentation

2.2.2.1 void checkForB (int *philoID*)

Checks if Thread is blocked.

Parameters

philoID	philosoher ID from thread creation

Returns

nothing

2.2.2.2 char convertStates (State *philoState*)

Converts the states into their first letter.

Parameters

philoState	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

philoID	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

philoID	philosoher ID from thread creation

Returns

nothing

2.2.2.6 void* philo (void * *pID*)

The main philosopher function.

Parameters

pID	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

philoID	philosoher ID from thread creation

Returns

nothing

2.2.2.8 void think (int *philoID*)

Simulates thinking by going through an empty loop.

Parameters

philoID	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)

  philosopher tries to get both sticks or waits for sticks to become available(on HIS condvar), 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

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

philoState	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 condvar), then eats

Parameters

philoID	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

philoID	philosoher ID from thread creation

Returns

nothing

Index