CS 446 Project 4

4

Generated by Doxygen 1.8.9.1

Wed Nov 15 2017 20:07:58

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 ConfigFileInput Class Reference

Public Member Functions

- **ConfigFileInput** (char ∗fileName)
- **ConfigFileInput** (const **ConfigFileInput** &copyInput)
- int **GetNumberOfProcesses** ()
- int **GetProcessValue** (const char processName[ ])
- int **GetProcessNumber** (const char processName[ ])
- char ∗ **GetProcessName** (const int position)
- int **GetProcessQuantity** (const char processName[ ])
- **ConfigFileInputNode** ∗ **GetProcess** (const char processName[ ])
- char **GetLogOutputSpecification** ()
- char ∗ **GetFilePath** ()
- char ∗ **GetLogFilePath** ()
- char ∗ **GetCPUSchedulingCode** ()
- int **GetProcessorQuantumNum** ()
- unsigned int **GetSystemMemory** ()
- unsigned int **GetMemoryBlockSize** ()

Protected Member Functions

- bool **ParseLine** (char lineToParse[ ])
- void **RemoveSpaces** (char lineToRemoveSpaces[ ])
- void **AdjustLineElements** (char lineToAdjust[ ], int positionToAdjust)

3.1.1 Member Function Documentation

3.1.1.1 void ConfigFileInput::AdjustLineElements (char *lineToAdjust[ ]*, int *positionToAdjust*) `[protected]`

< Helper function to make removing spaces easier

Precondition

None.

Postcondition

The specified string will have all leading characters from the position specified moved back one space.

Parameters

lineToAdjust	The line that will be adjusted.
positionToAdjust	The position in the string to which we are adjusting.

Returns

void

3.1.1.2 int ConfigFileInput::GetProcessValue (const char *processName[ ]*)

< Searches for the specified process name in the list of processes to get it's value Simply searches through the list to find the position of the process we are looking for and uses that position to get the value.

Precondition

None.

Postcondition

The value of the process with the given name is returned.

Parameters

processName	The name of the process to search for.

Returns

The value of the specified process.

3.1.1.3 bool ConfigFileInput::ParseLine (char *lineToParse[ ]*) [protected]

< Helper function that parses the lines in the config file and stores the relevent information.

Precondition

None.

Postcondition

The line will be parsed through and the tokens will be extracted and saved in the nodes in the list.

Parameters

lineToParse	The line to parse through.

Returns

The status of the parse line function.

3.1.1.4 void ConfigFileInput::RemoveSpaces (char *lineToRemoveSpaces[ ]*) [protected]

< Helper function to make parsing easier

Precondition

None.

Postcondition

The specified string will have all space characters removed.

Parameters

lineToRemove↩ Spaces	The string who's spaces will be removed.

Returns

void

The documentation for this class was generated from the following files:

- **ConfigFileInput.hh**
- ConfigFileInput.cc

3.2 ConfigFileInputNode Struct Reference

Public Member Functions

- **ConfigFileInputNode** (const **ConfigFileInputNode** &copyNode)

Public Attributes

- char **aProcessName** [STR_MAX_LENGTH]
- int **aProcessValue**
- int **aQuantity**

The documentation for this struct was generated from the following file:

- **ConfigFileInput.hh**

3.3 MetaDataInfo Class Reference

Public Member Functions

- **MetaDataInfo** (char ∗fileName, char ∗schedulingCode)
- void **ProcessData** (**ConfigFileInput** &configFile, **PCB** &state)

Protected Member Functions

- bool **ParseLine** (char lineToParse[ ], **processData** &pD)
- void **RemoveSpaces** (char lineToRemoveSpaces[ ])
- void **AdjustLineElements** (char lineToAdjust[ ], int positionToAdjust)
- void **ProcessErrorCode** (char logSpecification, char errorCode, ofstream &logFile)
- void **LogOutput** (char logSpecification, char ∗logMessage, ofstream &logFile)
- void **LogTime** (char logSpecification, double time, ofstream &logFile)
- char ∗ **itoa** (int inputValue, char ∗outputString, int base)
- void **ReverseString** (char ∗string, int size)

3.3.1 Member Function Documentation

3.3.1.1 void MetaDataInfo::AdjustLineElements (char *lineToAdjust[ ]*, int *positionToAdjust*) [protected]

< Helper function to make removing spaces easier

Precondition

None.

Postcondition

The specified string will have all leading characters from the position specified moved back one space.

Parameters

lineToAdjust	The line that will be adjusted.
positionToAdjust	The position in the string to which we are adjusting.

Returns

void

3.3.1.2 char ∗ MetaDataInfo::itoa (int *inputValue*, char ∗ *outputString*, int *base*) [protected]

< Helper function to convert from an integer to a string

Precondition

None.

Postcondition

The integer value passed into the function will be converted into a string representation to make logging easier.

Parameters

inputValue	The integer value that will be converted to a string.
ouputString	The string where the converted value will be stored.
base	The base to convert to. i.e. 10 = decimal, 16 = hexadecimal.

Returns

void

3.3.1.3 bool MetaDataInfo::ParseLine (char *lineToParse[ ]*, processData & *pD*) [protected]

< Parses the line from Meta Data and stores the tokens in nodes in the queue

Precondition

None.

Postcondition

Meta Data line will be parsed and all important information will be stored in nodes within the queue of processes to be simulated later.

Parameters

lineToParse	The line from the MetaData file that will be parsed

Returns

The status of the parse, indicating errors in the parsing process.

Note

Will also set error codes for any potentially troublesome processes so that they may be delt with later, without crashing the program.

3.3.1.4 void MetaDataInfo::ProcessData (ConfigFileInput & *configFile,* PCB & *state*)

< Processes the data from the Meta Data file and logs it to appropriate output

Precondition

ConfigFile has all necessary information to process.

Postcondition

Simulation will have been run and output logged to appropriate outlets.

Parameters

configFile	The configuration file with necessary info to preform the simulation.

Returns

void

3.3.1.5 void MetaDataInfo::ProcessErrorCode (char *logSpecification,* char *errorCode,* ofstream & *logFile*) [protected]

< Helper function make dealing with errors easier

Precondition

None.

Postcondition

The specified error code will be handled and logged appropriately.

Parameters

logSpecification	The specification code indicating where to log error.
errorCode	The code that needs to be handled.
logFile	The file where the error will be logged if specified to log to file.

Returns

void Helper function to make logging messages easier

Precondition

None.

Postcondition

The message to be logged will be logged to the appropriate destinations.

Parameters

logSpecification	The specification code indicating where to log error.
logMessage	The message that needs to be logged.
logFile	The file where the error will be logged if specified to log to file.

Returns

void

3.3.1.6 void MetaDataInfo::RemoveSpaces (char *lineToRemoveSpaces[ ]*) [protected]

< Helper function to make parsing easier

Precondition

None.

Postcondition

The specified string will have all space characters removed.

Parameters

lineToRemove←↩ Spaces	The string who's spaces will be removed.

Returns

void

3.3.1.7 void MetaDataInfo::ReverseString (char ∗ *string,* int *size*) [protected]

< Helper function to make converting from int to string easier.

Precondition

None.

Postcondition

The string passed in will be reversed.

Parameters

string	The string to be reversed.
size	The size of the string passed in.

Returns

void

The documentation for this class was generated from the following files:

- **MetaDataInfo.hh**
- **MetaDataInfo.cc**

3.4 MetaDataInfoNode Struct Reference

Public Member Functions

- **MetaDataInfoNode** (const **MetaDataInfoNode** &copyNode)

Public Attributes

- char **aMetaDataCode**
- char **aMetaDataDescriptor** [MAX_STR_LEN]
- int **aNumberOfCycles**
- int **aErrorCode**

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

3.5 mutex Struct Reference

Public Member Functions

- void **lock** ()
- void **release** ()

Public Attributes

- bool **available**

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

3.6 PCB Class Reference

Public Attributes

- int **processState**

The documentation for this class was generated from the following file:

- PCB.hh

3.7 process Struct Reference

Public Member Functions

- **process** (const **process** &p2)

Public Attributes

- int **processNum** = -1
- int **priority** = -1
- queue< **MetaDataInfoNode** > **metaDataQueue**

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

3.8 processData Struct Reference

Collaboration diagram for processData:

Public Attributes

- int **processNum** = 0
- int **taskNum** = 0
- int **inputOutputNum** = 0
- **process** * **tempProcess** = NULL

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

3.9 semaphore Struct Reference

Collaboration diagram for semaphore:

Public Attributes

- int **value**
- int **size**
- **mutex m**

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

3.10 thread_data Struct Reference

Collaboration diagram for thread_data:

Public Attributes

- timeval **initTime**
- int **tempProcessRunTime**
- double **executionTime**
- **semaphore S**

The documentation for this struct was generated from the following file:

- **MetaDataInfo.hh**

Chapter 4

File Documentation

4.1 ConfigFileInput.hh File Reference

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <stdlib.h>
#include <list>
```

Include dependency graph for ConfigFileInput.hh:

4.2 MetaDataInfo.cc File Reference

```
#include "MetaDataInfo.hh"
#include "MemoryFunction.hh"
```

Include dependency graph for MetaDataInfo.cc:

Functions

- bool **operator>** (const **process** &P1, const **process** &P2)
- bool **operator<** (const **process** &P1, const **process** &P2)
- void ∗ **threadInput** (void ∗threadarg)
- void ∗ **threadOutput** (void ∗threadarg)
- double **timer** (long timeToWait, timeval &initTime)
- long **getWaitTime** (timeval &start)
- void **wait** (**semaphore** ∗S)
- void **signal** (**semaphore** ∗S)

4.2.1 Detailed Description

Author

Eugene Nelson The implamentation file for the **MetaDataInfo** (p. 7) class.

Version

1.0 Eugene Nelson Originally developed (9 - 19 - 17)

4.3 MetaDataInfo.hh File Reference

```
#include <iostream>
#include <queue>
#include <iomanip>
#include <sys/time.h>
#include <pthread.h>
#include <fstream>
#include <cstring>
#include <stdlib.h>
#include <cstdlib>
#include "ConfigFileInput.hh"
#include "PCB.hh"
```

Include dependency graph for MetaDataInfo.hh: This graph shows which files directly or indirectly include this file:

Classes

- struct **MetaDataInfoNode**
- struct **mutex**
- struct **semaphore**
- struct **thread_data**
- struct **process**
- struct **processData**
- class **MetaDataInfo**

Macros

- #define **MAX_STR_LEN** 30
- #define **LINE_MAX_LEGNTH** 300
- #define **MESSAGE_MAX_LENGTH** 500

Functions

- bool **operator**> (const **process** &P1, const **process** &P2)
- bool **operator**< (const **process** &P1, const **process** &P2)
- void ∗ **threadInput** (void ∗threadarg)
- void ∗ **threadOutput** (void ∗threadarg)
- double **timer** (long timeToWait, timeval &initTime)
- long **getWaitTime** (timeval &start)
- void **wait** (**semaphore** ∗S)
- void **signal** (**semaphore** ∗S)

4.3.1 Detailed Description

Author

Eugene Nelson The header file for the **MetaDataInfo** (p. 7) class.

Version

1.0 Eugene Nelson Originally developed (9 - 19 - 17)

Index