SmartOS

0.3.3

Generated by Doxygen 1.8.14

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

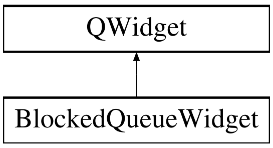
Chapter 4

Class Documentation

4.1 BlockedQueueWidget Class Reference

```
#include <BlockedQueueWidget.h>
```

Inheritance diagram for BlockedQueueWidget:



Public Member Functions

- BlockedQueueWidget (QWidget ∗parent=nullptr)
- void update ()

4.1.1 Detailed Description

Definition at line 7 of file BlockedQueueWidget.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 BlockedQueueWidget()

```
BlockedQueueWidget::BlockedQueueWidget (
            QWidget ∗ parent = nullptr )
```

Definition at line 13 of file BlockedQueueWidget.cpp.

4.1.3 Member Function Documentation

4.1.3.1 update()

```
void BlockedQueueWidget::update ( )
```

Definition at line 29 of file BlockedQueueWidget.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.cpp

4.2 CentralProcessingUnit Class Reference

The CentralProcessingUnit class holds the current process.

```
#include <CentralProcessingUnit.h>
```

Public Member Functions

- CentralProcessingUnit ()

  CentralProcessingUnit construct a new Central Processing Unit.
- ProcessControlBlockPtr setActiveProcess (ProcessControlBlockPtr pcb)

  setActiveProcess sets the active process
- ProcessControlBlockPtr & currentProcess ()

  currentProcess returns a reference to the current process.

4.2.1 Detailed Description

The CentralProcessingUnit class holds the current process.

Definition at line 9 of file CentralProcessingUnit.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 CentralProcessingUnit()

```
CentralProcessingUnit::CentralProcessingUnit ( )
```

CentralProcessingUnit construct a new Central Processing Unit.

Definition at line 3 of file CentralProcessingUnit.cpp.

4.2.3 Member Function Documentation

4.2.3.1 currentProcess()

```
ProcessControlBlockPtr & CentralProcessingUnit::currentProcess ( )
```

currentProcess returns a reference to the current process.

Returns

a reference to the current process

Definition at line 13 of file CentralProcessingUnit.cpp.

4.2.3.2 setActiveProcess()

```
ProcessControlBlockPtr CentralProcessingUnit::setActiveProcess (
             ProcessControlBlockPtr pcb )
```

setActiveProcess sets the active process

Parameters

pcb	the process to make active

Returns

the old process that was in the CPU, or nullptr

Definition at line 6 of file CentralProcessingUnit.cpp.

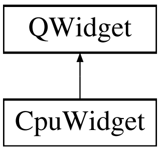
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.h
- /home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.cpp

4.3 CpuWidget Class Reference

```
#include <CpuWidget.h>
```

Inheritance diagram for CpuWidget:



Public Member Functions

- CpuWidget (QWidget ∗parent=nullptr)
- void update ()

4.3.1 Detailed Description

Definition at line 13 of file CpuWidget.h.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 CpuWidget()

```
CpuWidget::CpuWidget (
              QWidget ∗ parent = nullptr )   [explicit]
```

Definition at line 11 of file CpuWidget.cpp.

4.3.3 Member Function Documentation

4.3.3.1 update()

```
void CpuWidget::update ( )
```

Definition at line 24 of file CpuWidget.cpp.

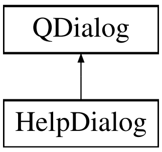
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.cpp

4.4 HelpDialog Class Reference

```
#include <HelpDialog.h>
```

Inheritance diagram for HelpDialog:



Public Member Functions

- HelpDialog ()

4.4.1 Detailed Description

Definition at line 6 of file HelpDialog.h.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 HelpDialog()

```
HelpDialog::HelpDialog ( )
```

Definition at line 3 of file HelpDialog.cpp.

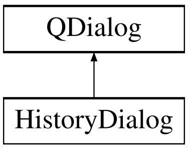
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.cpp

4.5 HistoryDialog Class Reference

```
#include <HistoryDialog.h>
```

Inheritance diagram for HistoryDialog:



Public Member Functions

- HistoryDialog (QStringList &historyList)

4.5.1 Detailed Description

Definition at line 7 of file HistoryDialog.h.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 HistoryDialog()

```
HistoryDialog::HistoryDialog (
            QStringList & historyList )
```

Definition at line 7 of file HistoryDialog.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.cpp

4.6 IOEvent Class Reference

```
#include <IOEvent.h>
```

Public Types

- enum Type { Type::NONE, Type::HARD_DRIVE, Type::USER_IO }

  The Type enum represents the different types of IOEvents.

Public Member Functions

- IOEvent (Type type, size_t cycleStamp)

  IOEvent constructs a new IOEvent with the specified type and cycleStamp.
- Type type () const

  type returns the type of event
- size_t cycleStamp () const

  cycleStamp returns the time cycle this event was added.

4.6.1 Detailed Description

Definition at line 6 of file IOEvent.h.

4.6.2 Member Enumeration Documentation

4.6.2.1 Type

```
enum IOEvent::Type  [strong]
```

The Type enum represents the different types of IOEvents.

Enumerator

NONE	
HARD_DRIVE	
USER_IO	

Definition at line 12 of file IOEvent.h.

4.6.3 Constructor & Destructor Documentation

4.6.3.1 IOEvent()

```
IOEvent::IOEvent (
            Type type,
            size_t cycleStamp )
```

IOEvent constructs a new IOEvent with the specified type and cycleStamp.

Parameters

type	the type of IOEvent
cycleStamp	the time cycle this event was added

Definition at line 3 of file IOEvent.cpp.

4.6.4 Member Function Documentation

4.6.4.1 cycleStamp()

```
size_t IOEvent::cycleStamp ( ) const
```

cycleStamp returns the time cycle this event was added.

Returns

the time cycle this event was added

Definition at line 13 of file IOEvent.cpp.

4.6.4.2 type()

```
IOEvent::Type IOEvent::type ( ) const
```

type returns the type of event

Returns

the type of event

Definition at line 8 of file IOEvent.cpp.

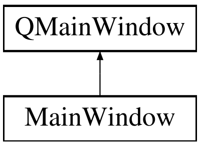
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.h
- /home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.cpp

4.7 MainWindow Class Reference

```
#include <MainWindow.h>
```

Inheritance diagram for MainWindow:



Public Slots

- void showProcessScheduler ()

  showProcessScheduler shows the view to deal with process scheduling.
- void showUserDocumentation ()

  showUserDocumentation shows the user documentation PDF.

Public Member Functions

- MainWindow (QWidget ∗parent=0)
- ∼MainWindow ()
- void addHistory (const QString &info)

  addHistory add the following information to the history

4.7.1 Detailed Description

Definition at line 10 of file MainWindow.h.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 MainWindow()

```
MainWindow::MainWindow (
            QWidget * parent = 0 )  [explicit]
```

Definition at line 24 of file MainWindow.cpp.

4.7.2.2 ∼MainWindow()

```
MainWindow::∼MainWindow ( )
```

Definition at line 37 of file MainWindow.cpp.

4.7.3 Member Function Documentation

4.7.3.1 addHistory()

```
void MainWindow::addHistory (
            const QString & info )
```

addHistory add the following information to the history

Parameters

info	the info to add to the history

Definition at line 40 of file MainWindow.cpp.

4.7.3.2 showProcessScheduler

```
void MainWindow::showProcessScheduler ( )  [slot]
```

showProcessScheduler shows the view to deal with process scheduling.

Definition at line 113 of file MainWindow.cpp.

4.7.3.3 showUserDocumentation

```
void MainWindow::showUserDocumentation ( )  [slot]
```

showUserDocumentation shows the user documentation PDF.

Definition at line 132 of file MainWindow.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.cpp

4.8 ProcessControlBlock Class Reference

The ProcessControlBlock class represents a Process Control BLock in the simulator.

```
#include <ProcessControlBlock.h>
```

Public Member Functions

- ProcessControlBlock (size_t pid, size_t memory)

  ProcessControlBlock constructs a new Process Control Block with the specified pid and memory required.
- size_t pid () const

  pid returns the process's id
- size_t cpuUsageTerm () const

  cpuUsageTerm returns the time spent using the cpu
- size_t ioReqTerm () const

  ioReqTerm returns the time waiting for IO
- size_t waitTerm () const

  waitTerm returns the time waiting in the ready queue
- size_t memory () const

  memory returns the memory required by the process
- size_t priority () const

  priority returns the priority for this process
- IOEvent ioEvent () const

  ioEvent returns the type of IOEvent that the process is waiting on.
- ProcessType processType () const

  processType returns the type of process this is
- void updateCpuUsageTerm (size_t elapsed)

  updateCpuUsageTerm adds the elapsed time to the cpu usage.
- void updateIoReqTerm (size_t elapsed)

  updateIoReqTerm adds the elapsed time to the io waiting time
- void updateWaitTerm (size_t elapsed)

  updateWaitTerm adds the elapsed time to the time waiting in the ready queue.
- void setWaitEvent (IOEvent event)

  setWaitEvent sets the type of event this process is waiting for
- void clearWaitEvent ()

  clearWaitEvent clears the event this process is waiting for
- void setProcessType (ProcessType type)

  setProcessType sets the type of process this is
- void setPriority (size_t priority)

  setPriority sets this process's priority

4.8.1 Detailed Description

The ProcessControlBlock class represents a Process Control BLock in the simulator.

Definition at line 16 of file ProcessControlBlock.h.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 ProcessControlBlock()

```
ProcessControlBlock::ProcessControlBlock (
             size_t pid,
             size_t memory )  [explicit]
```

ProcessControlBlock constructs a new Process Control Block with the specified pid and memory required.

Parameters

pid	the id of the process
memory	the amount of memory needed by the process

Definition at line 3 of file ProcessControlBlock.cpp.

4.8.3 Member Function Documentation

4.8.3.1 clearWaitEvent()

```
void ProcessControlBlock::clearWaitEvent ( )
```

clearWaitEvent clears the event this process is waiting for

Definition at line 74 of file ProcessControlBlock.cpp.

4.8.3.2 cpuUsageTerm()

```
size_t ProcessControlBlock::cpuUsageTerm ( ) const
```

cpuUsageTerm returns the time spent using the cpu

Returns

the time spent using the cpu

Definition at line 19 of file ProcessControlBlock.cpp.

4.8.3.3 ioEvent()

```
IOEvent ProcessControlBlock::ioEvent ( ) const
```

ioEvent returns the type of IOEvent that the process is waiting on.

Returns

the type of IOEvent the process is waiting on

Definition at line 44 of file ProcessControlBlock.cpp.

4.8.3.4 ioReqTerm()

```
size_t ProcessControlBlock::ioReqTerm ( ) const
```

ioReqTerm returns the time waiting for IO

Returns

the time waiting for IO

Definition at line 24 of file ProcessControlBlock.cpp.

4.8.3.5 memory()

```
size_t ProcessControlBlock::memory ( ) const
```

memory returns the memory required by the process

Returns

the memory required by the process

Definition at line 34 of file ProcessControlBlock.cpp.

4.8.3.6 pid()

```
size_t ProcessControlBlock::pid ( ) const
```

pid returns the process's id

Returns

the process's id

Definition at line 14 of file ProcessControlBlock.cpp.

4.8.3.7 priority()

```
size_t ProcessControlBlock::priority ( ) const
```

priority returns the priority for this process

Returns

the priority of the process

Definition at line 39 of file ProcessControlBlock.cpp.

4.8.3.8 processType()

```
ProcessType ProcessControlBlock::processType ( ) const
```

processType returns the type of process this is

Returns

the type of process this is

Definition at line 49 of file ProcessControlBlock.cpp.

4.8.3.9 setPriority()

```
void ProcessControlBlock::setPriority (
             size_t priority )
```

setPriority sets this process's priority

Parameters

priority	the new priority

Definition at line 84 of file ProcessControlBlock.cpp.

4.8.3.10 setProcessType()

```
void ProcessControlBlock::setProcessType (
             ProcessType type )
```

setProcessType sets the type of process this is

Parameters

type	the type of process this is

Definition at line 79 of file ProcessControlBlock.cpp.

4.8.3.11 setWaitEvent()

```
void ProcessControlBlock::setWaitEvent (
            IOEvent event )
```

setWaitEvent sets the type of event this process is waiting for

Parameters

event	the type of event this process is waiting for

Definition at line 69 of file ProcessControlBlock.cpp.

4.8.3.12 updateCpuUsageTerm()

```
void ProcessControlBlock::updateCpuUsageTerm (
            size_t elapsed )
```

updateCpuUsageTerm adds the elapsed time to the cpu usage.

Parameters

elapsed	the time elapsed

Definition at line 54 of file ProcessControlBlock.cpp.

4.8.3.13 updateIoReqTerm()

```
void ProcessControlBlock::updateIoReqTerm (
            size_t elapsed )
```

updateIoReqTerm adds the elapsed time to the io waiting time

Parameters

elapsed	the time elapsed

Definition at line 59 of file ProcessControlBlock.cpp.

4.8.3.14 updateWaitTerm()

```
void ProcessControlBlock::updateWaitTerm (
             size_t elapsed )
```

updateWaitTerm adds the elapsed time to the time waiting in the ready queue.

Parameters

elapsed	the time elapsed

Definition at line 64 of file ProcessControlBlock.cpp.

4.8.3.15 waitTerm()

```
size_t ProcessControlBlock::waitTerm ( ) const
```

waitTerm returns the time waiting in the ready queue

Returns

the time waiting in the ready queue

Definition at line 29 of file ProcessControlBlock.cpp.

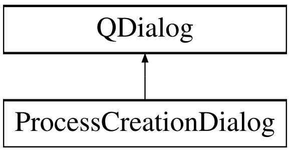
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.h
- /home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.cpp

4.9 ProcessCreationDialog Class Reference

```
#include <ProcessCreationDialog.h>
```

Inheritance diagram for ProcessCreationDialog:



Public Member Functions

- ProcessCreationDialog ()
- size_t pid () const
- size_t memoryRequired () const
- ProcessType processType () const

4.9.1 Detailed Description

Definition at line 10 of file ProcessCreationDialog.h.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 ProcessCreationDialog()

```
ProcessCreationDialog::ProcessCreationDialog ( )
```

Definition at line 10 of file ProcessCreationDialog.cpp.

4.9.3 Member Function Documentation

4.9.3.1 memoryRequired()

```
size_t ProcessCreationDialog::memoryRequired ( ) const
```

Definition at line 21 of file ProcessCreationDialog.cpp.

4.9.3.2 pid()

```
size_t ProcessCreationDialog::pid ( ) const
```

Definition at line 16 of file ProcessCreationDialog.cpp.

4.9.3.3 processType()

```
ProcessType ProcessCreationDialog::processType ( ) const
```

Definition at line 26 of file ProcessCreationDialog.cpp.

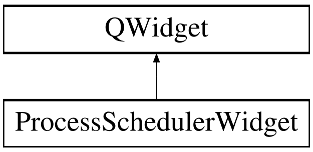
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessCreationDialog.cpp

4.10 ProcessSchedulerWidget Class Reference

```
#include <ProcessSchedulerWidget.h>
```

Inheritance diagram for ProcessSchedulerWidget:



Public Slots

- void addProcessControlBlock ()
- void addRandomProcessControlBlocks ()
- void execute ()
- void pause ()
- void stop ()
- void executeStep ()

Public Member Functions

- ProcessSchedulerWidget (MainWindow ∗parent=nullptr)
- void paintEvent (QPaintEvent ∗paintEvent) override

4.10.1 Detailed Description

Definition at line 17 of file ProcessSchedulerWidget.h.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 ProcessSchedulerWidget()

```
ProcessSchedulerWidget::ProcessSchedulerWidget (
            MainWindow * parent = nullptr )  [explicit]
```

Definition at line 36 of file ProcessSchedulerWidget.cpp.

4.10.3 Member Function Documentation

4.10.3.1 addProcessControlBlock

```
void ProcessSchedulerWidget::addProcessControlBlock ( )  [slot]
```

Definition at line 115 of file ProcessSchedulerWidget.cpp.

4.10.3.2 addRandomProcessControlBlocks

```
void ProcessSchedulerWidget::addRandomProcessControlBlocks ( )  [slot]
```

Definition at line 130 of file ProcessSchedulerWidget.cpp.

4.10.3.3 execute

```
void ProcessSchedulerWidget::execute ( )  [slot]
```

Definition at line 163 of file ProcessSchedulerWidget.cpp.

4.10.3.4 executeStep

```
void ProcessSchedulerWidget::executeStep ( )  [slot]
```

Definition at line 206 of file ProcessSchedulerWidget.cpp.

4.10.3.5 paintEvent()

```
void ProcessSchedulerWidget::paintEvent (
            QPaintEvent * paintEvent )  [override]
```

Definition at line 110 of file ProcessSchedulerWidget.cpp.

4.10.3.6 pause

```
void ProcessSchedulerWidget::pause ( )  [slot]
```

Definition at line 177 of file ProcessSchedulerWidget.cpp.

4.10.3.7 stop

```
void ProcessSchedulerWidget::stop ( )  [slot]
```

Definition at line 190 of file ProcessSchedulerWidget.cpp.

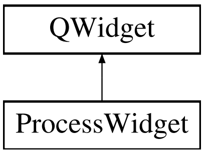
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/ProcessSchedulerWidget.cpp

4.11 ProcessWidget Class Reference

```
#include <ProcessWidget.h>
```

Inheritance diagram for ProcessWidget:



Public Member Functions

- ProcessWidget (ProcessControlBlockPtr &pcb, bool detailed=false, QWidget ∗parent=nullptr)
- void paintEvent (QPaintEvent ∗event) override
- void update ()

4.11.1 Detailed Description

Definition at line 9 of file ProcessWidget.h.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 ProcessWidget()

```
ProcessWidget::ProcessWidget (
            ProcessControlBlockPtr & pcb,
            bool detailed = false,
            QWidget * parent = nullptr )
```

Definition at line 9 of file ProcessWidget.cpp.

4.11.3 Member Function Documentation

4.11.3.1 paintEvent()

```
void ProcessWidget::paintEvent (
            QPaintEvent * event ) [override]
```

Definition at line 69 of file ProcessWidget.cpp.

4.11.3.2 update()

```
void ProcessWidget::update ( )
```

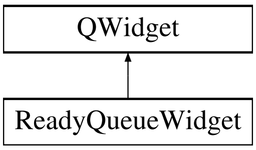
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.cpp

4.12 ReadyQueueWidget Class Reference

```
#include <ReadyQueueWidget.h>
```

Inheritance diagram for ReadyQueueWidget:



Public Member Functions

- ReadyQueueWidget (QWidget ∗parent=nullptr)
- void update ()

4.12.1 Detailed Description

Definition at line 7 of file ReadyQueueWidget.h.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 ReadyQueueWidget()

```
ReadyQueueWidget::ReadyQueueWidget (
             QWidget ∗ parent = nullptr )
```

Definition at line 13 of file ReadyQueueWidget.cpp.

4.12.3 Member Function Documentation

4.12.3.1 update()

```
void ReadyQueueWidget::update ( )
```

Definition at line 29 of file ReadyQueueWidget.cpp.

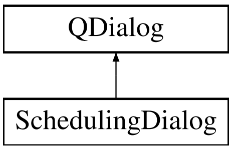
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.cpp

4.13 SchedulingDialog Class Reference

```
#include <SchedulingDialog.h>
```

Inheritance diagram for SchedulingDialog:



Public Member Functions

- SchedulingDialog ()
- SchedulerType schedulerType () const
- size_t quantum () const
- size_t priorityQueues () const

4.13.1 Detailed Description

Definition at line 10 of file SchedulingDialog.h.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 SchedulingDialog()

```
SchedulingDialog::SchedulingDialog ( )
```

Definition at line 8 of file SchedulingDialog.cpp.

4.13.3 Member Function Documentation

4.13.3.1 priorityQueues()

```
size_t SchedulingDialog::priorityQueues ( ) const
```

Definition at line 62 of file SchedulingDialog.cpp.

4.13.3.2 quantum()

```
size_t SchedulingDialog::quantum ( ) const
```

Definition at line 57 of file SchedulingDialog.cpp.

4.13.3.3 schedulerType()

```
SchedulerType SchedulingDialog::schedulerType ( ) const
```

Definition at line 46 of file SchedulingDialog.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/SchedulingDialog.cpp

4.14 SmartOS Class Reference

The SmartOS class is the heart of the SmartOS operating system simulator. All functions to manipulate processes are found here.

```
#include <SmartOS.h>
```

Public Member Functions

- SmartOS (size_t memory)

  SmartOS construct a new simulator with the specified maximum memory.
- size_t nextSequentialPID ()

  nextSequentialPID the next process id that is available starting from 0.
- void createProcessControlBlock (size_t pid, size_t memory, ProcessType type=ProcessType::RANDOM)

  createProcessControlBlock creates a new Process Control Block and adds it to the ready queue.
- bool deleteProcessControlBlock (size_t pid)

  deleteProcessControlBlock deletes the process control block with the specified pid.
- bool blockProcessControlBlock (size_t pid, IOEvent ioEvent)

  blockProcessControlBlock moves the specified process to the block queue
- bool unblockProcessControlBlock (size_t pid)

  unblockProcessControlBlock removes the process from the block queue and inserts it into the ready queue
- bool setActiveProcess (size_t pid)

  setActiveProcess sets the active process to the process with the specified pid.
- CentralProcessingUnit & cpu ()

  cpu returns a reference to the Central Processing Unit.
- PCBQueue & readyQueue ()

  readyQueue returns a reference to the Ready Queue
- PCBQueue & blockedQueue ()

  blockedQueue returns a reference to the Blocked Queue
- const ProcessControlBlockPtr & findProcessControlBlock (size_t pid)

  findProcessControlBlock finds a reference to the Process Control Block with the specified id.
- const IOEventQueue & ioEventQueue ()

  ioEventQueue returns a reference to the io event queue
- size_t maxMemory () const

  maxMemory returns the maximum amount of memory in the operating system.
- size_t usedMemory ()

  memory returns the amount of memory used in the operating system.
- void execute ()

execute execute a single step in the operating system.

- void updateCurrentProcessControlBlock ()

  updateCurrent sets the next active process in the CPU.
- void addOperatingSystemProcess ()

  addOperatingSystemProcess adds the operating system process
- ProcessControlBlockPtr determineNextProcess ()

  determineNextProcess determines the next process in the ready queue to run.
- size_t cycleCount () const

  cycleCount returns the current time cycle count
- void reset ()

  reset resets the operating system.
- void setScheduler (SchedulerType type)

  setScheduler sets the type of scheduler to use.
- void setTimeQuantum (size_t quantum)

  setTimeQuantum sets the time quantum
- void setMaximumPriority (size_t priority)

  setMaximumPriority sets the maximum priority level.

Static Public Member Functions

- static std::string getVersionNumber ()

  getVersionNumber returns the version number

4.14.1 Detailed Description

The SmartOS class is the heart of the SmartOS operating system simulator. All functions to manipulate processes are found here.

Definition at line 36 of file SmartOS.h.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 SmartOS()

```
SmartOS::SmartOS (
            size_t memory )
```

SmartOS construct a new simulator with the specified maximum memory.

Parameters

memory	the maximum amount of memory

Definition at line 13 of file SmartOS.cpp.

4.14.3 Member Function Documentation

4.14.3.1 addOperatingSystemProcess()

```
void SmartOS::addOperatingSystemProcess ( )
```

addOperatingSystemProcess adds the operating system process

Definition at line 323 of file SmartOS.cpp.

4.14.3.2 blockedQueue()

```
PCBQueue & SmartOS::blockedQueue ( )
```

blockedQueue returns a reference to the Blocked Queue

Returns

a reference to the Blocked Queue

Definition at line 185 of file SmartOS.cpp.

4.14.3.3 blockProcessControlBlock()

```
bool SmartOS::blockProcessControlBlock (
             size_t pid,
             IOEvent ioEvent )
```

blockProcessControlBlock moves the specified process to the block queue

Parameters

pid	the process's id
ioEvent	the event that being waited on

Returns

true if successful

Definition at line 76 of file SmartOS.cpp.

4.14.3.4 cpu()

```
CentralProcessingUnit & SmartOS::cpu ( )
```

cpu returns a reference to the Central Processing Unit.

Returns

a reference to the Central Processing Unit

Definition at line 175 of file SmartOS.cpp.

4.14.3.5 createProcessControlBlock()

```
void SmartOS::createProcessControlBlock (
             size_t pid,
             size_t memory,
             ProcessType type = ProcessType::RANDOM )
```

createProcessControlBlock creates a new Process Control Block and adds it to the ready queue.

Parameters

pid	the pid
memory	the amount of memory needed for the processes
type	the type of process.

Definition at line 34 of file SmartOS.cpp.

4.14.3.6 cycleCount()

```
size_t SmartOS::cycleCount ( ) const
```

cycleCount returns the current time cycle count

Returns

the current time cycle count

Definition at line 450 of file SmartOS.cpp.

4.14.3.7 deleteProcessControlBlock()

```
bool SmartOS::deleteProcessControlBlock (
             size_t pid )
```

deleteProcessControlBlock deletes the process control block with the specified pid.

Parameters

pid	the process's id

Returns

true if successful

Definition at line 50 of file SmartOS.cpp.

4.14.3.8 determineNextProcess()

```
ProcessControlBlockPtr SmartOS::determineNextProcess ( )
```

determineNextProcess determines the next process in the ready queue to run.

Returns

the next process in the ready queue to run.

4.14.3.9 execute()

```
void SmartOS::execute ( )
```

execute execute a single step in the operating system.

Definition at line 343 of file SmartOS.cpp.

4.14.3.10 findProcessControlBlock()

```
const ProcessControlBlockPtr & SmartOS::findProcessControlBlock (
             size_t pid )
```

findProcessControlBlock finds a reference to the Process Control Block with the specified id.

Parameters

pid	the process's id

Returns

a reference to the Process Control Block if found, otherwise a null pcb.

Definition at line 190 of file SmartOS.cpp.

4.14.3.11 getVersionNumber()

```
static std::string SmartOS::getVersionNumber ( )  [inline], [static]
```

getVersionNumber returns the version number

Returns

the version number

Definition at line 43 of file SmartOS.h.

4.14.3.12 ioEventQueue()

```
const IOEventQueue & SmartOS::ioEventQueue ( )
```

ioEventQueue returns a reference to the io event queue

Returns

a reference to the io event queue

Definition at line 217 of file SmartOS.cpp.

4.14.3.13 maxMemory()

```
size_t SmartOS::maxMemory ( ) const
```

maxMemory returns the maximum amount of memory in the operating system.

Returns

the maximum amount of memory in the operating system

Definition at line 480 of file SmartOS.cpp.

4.14.3.14 nextSequentialPID()

```
size_t SmartOS::nextSequentialPID ( )
```

nextSequentialPID the next process id that is available starting from 0.

Returns

the next process id

Definition at line 23 of file SmartOS.cpp.

4.14.3.15 readyQueue()

```
PCBQueue & SmartOS::readyQueue ( )
```

readyQueue returns a reference to the Ready Queue

Returns

a reference to the Ready Queue

Definition at line 180 of file SmartOS.cpp.

4.14.3.16 reset()

```
void SmartOS::reset ( )
```

reset resets the operating system.

Definition at line 455 of file SmartOS.cpp.

4.14.3.17 setActiveProcess()

```
bool SmartOS::setActiveProcess (
              size_t pid )
```

setActiveProcess sets the active process to the process with the specified pid.

Parameters

pid	the process's pid

Returns

true if successful

Definition at line 135 of file SmartOS.cpp.

4.14.3.18 setMaximumPriority()

```
void SmartOS::setMaximumPriority (
             size_t priority )
```

setMaximumPriority sets the maximum priority level.

Parameters

priority	the priority level

Definition at line 475 of file SmartOS.cpp.

4.14.3.19 setScheduler()

```
void SmartOS::setScheduler (
             SchedulerType type )
```

setScheduler sets the type of scheduler to use.

Parameters

type	the scheduler type

Definition at line 465 of file SmartOS.cpp.

4.14.3.20 setTimeQuantum()

```
void SmartOS::setTimeQuantum (
             size_t quantum )
```

setTimeQuantum sets the time quantum

Parameters

quantum	the time quantum

Definition at line 470 of file SmartOS.cpp.

4.14.3.21 unblockProcessControlBlock()

```
bool SmartOS::unblockProcessControlBlock (
             size_t pid )
```

unblockProcessControlBlock removes the process from the block queue and inserts it into the ready queue

Parameters

pid	the process's id

Returns

true if successful

Definition at line 109 of file SmartOS.cpp.

4.14.3.22 updateCurrentProcessControlBlock()

```
void SmartOS::updateCurrentProcessControlBlock ( )
```

updateCurrent sets the next active process in the CPU.

Definition at line 245 of file SmartOS.cpp.

4.14.3.23 usedMemory()

```
size_t SmartOS::usedMemory ( )
```

memory returns the amount of memory used in the operating system.

Returns

the amount of memory used in the operating system

Definition at line 222 of file SmartOS.cpp.

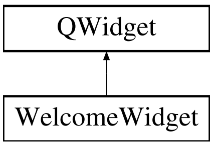
The documentation for this class was generated from the following files:

- /home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.h
- /home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.cpp

4.15 WelcomeWidget Class Reference

```
#include <WelcomeWidget.h>
```

Inheritance diagram for WelcomeWidget:



Public Member Functions

- WelcomeWidget (MainWindow ∗mainWindow, QWidget ∗parent=nullptr)

4.15.1 Detailed Description

Definition at line 8 of file WelcomeWidget.h.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 WelcomeWidget()

```
WelcomeWidget::WelcomeWidget (
            MainWindow ∗ mainWindow,
            QWidget ∗ parent = nullptr )
```

Definition at line 8 of file WelcomeWidget.cpp.

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

- /home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.h
- /home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.cpp

Chapter 5

File Documentation

5.1 /home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.cpp File Reference

```
#include "CentralProcessingUnit.h"
```

5.2 /home/nicholas/qtcreator/smartos/SmartOS_Core/CentralProcessingUnit.h File Reference

```
#include "ProcessControlBlock.h"
```

Classes

- class CentralProcessingUnit

  The CentralProcessingUnit class holds the current process.

5.3 /home/nicholas/qtcreator/smartos/SmartOS_Core/Global.h File Reference

```
#include <QtCore/qglobal.h>
```

Macros

- #define SMARTOS_CORESHARED_EXPORT Q_DECL_IMPORT

5.3.1 Macro Definition Documentation

5.3.1.1 SMARTOS_CORESHARED_EXPORT

```
#define SMARTOS_CORESHARED_EXPORT Q_DECL_IMPORT
```

Definition at line 9 of file Global.h.

5.4 /home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.cpp File Reference

```
#include "IOEvent.h"
```

5.5 /home/nicholas/qtcreator/smartos/SmartOS_Core/IOEvent.h File Reference

```
#include <memory>
```

Classes

- class IOEvent

5.6 /home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.cpp File Reference

```
#include "ProcessControlBlock.h"
```

5.7 /home/nicholas/qtcreator/smartos/SmartOS_Core/ProcessControlBlock.h File Reference

```
#include "IOEvent.h"
#include <memory>
```

Classes

- class ProcessControlBlock

  The ProcessControlBlock class represents a Process Control BLock in the simulator.

Typedefs

- typedef std::unique_ptr< ProcessControlBlock > ProcessControlBlockPtr

Enumerations

- enum ProcessType {
  ProcessType::RANDOM, ProcessType::INTERACTIVE, ProcessType::CPU_BOUND, ProcessType::MIXED, ProcessType::LAST }

  The ProcessType enum represents the type of process.

5.7.1 Typedef Documentation

5.7.1.1 ProcessControlBlockPtr

```
typedef std::unique_ptr<ProcessControlBlock> ProcessControlBlockPtr
```

Definition at line 127 of file ProcessControlBlock.h.

5.7.2 Enumeration Type Documentation

5.7.2.1 ProcessType

```
enum ProcessType  [strong]
```

The ProcessType enum represents the type of process.

Enumerator

RANDOM	
INTERACTIVE	
CPU_BOUND	
MIXED	
LAST	

Definition at line 11 of file ProcessControlBlock.h.

5.8 /home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.cpp File Reference

```
#include "SmartOS.h"
#include <random>
#include <iostream>
```

5.9 /home/nicholas/qtcreator/smartos/SmartOS_Core/SmartOS.h File Reference

```
#include "Global.h"
#include "CentralProcessingUnit.h"
#include "IOEvent.h"
#include "ProcessControlBlock.h"
#include <list>
#include <fstream>
```

Classes

- class SmartOS

  The SmartOS class is the heart of the SmartOS operating system simulator. All functions to manipulate processes are found here.

Typedefs

- typedef std::list< ProcessControlBlockPtr > PCBQueue

  PCBQueue a data structure used to iterate proceses.
- typedef std::list< IOEvent > IOEventQueue

  IOEventQueue a data structure used to iterate IO events.

Enumerations

- enum SchedulerType { SchedulerType::DEFAULT, SchedulerType::ROUND_ROBIN, SchedulerType::MLFQ }

  The SchedulerType enum represents the different types of process schedulers.

5.9.1 Typedef Documentation

5.9.1.1 IOEventQueue

```
typedef std::list<IOEvent> IOEventQueue
```

IOEventQueue a data structure used to iterate IO events.

Definition at line 22 of file SmartOS.h.

5.9.1.2 PCBQueue

```
typedef std::list<ProcessControlBlockPtr> PCBQueue
```

PCBQueue a data structure used to iterate proceses.

Definition at line 17 of file SmartOS.h.

5.9.2 Enumeration Type Documentation

5.9.2.1 SchedulerType

```
enum SchedulerType  [strong]
```

The SchedulerType enum represents the different types of process schedulers.

Enumerator

DEFAULT	
ROUND_ROBIN	
MLFQ	

Definition at line 27 of file SmartOS.h.

5.10 /home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.cpp File Reference

```
#include "BlockedQueueWidget.h"
#include "Globals.h"
#include "ProcessWidget.h"
#include <QLabel>
#include <QListWidgetItem>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

5.11 /home/nicholas/qtcreator/smartos/SmartOS_GUI/BlockedQueueWidget.h File Reference

```
#include <QListWidget>
#include <QWidget>
```

Classes

- class BlockedQueueWidget

5.12 /home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.cpp File Reference

```
#include "CpuWidget.h"
#include "Globals.h"
#include <QLabel>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

5.13 /home/nicholas/qtcreator/smartos/SmartOS_GUI/CpuWidget.h File Reference

```
#include "ProcessWidget.h"
#include <CentralProcessingUnit.h>
#include <ProcessControlBlock.h>
#include <QPaintEvent>
#include <QStackedWidget>
#include <QWidget>
```

Classes

- class CpuWidget

5.14 /home/nicholas/qtcreator/smartos/SmartOS_GUI/Globals.h File Reference

```
#include <SmartOS.h>
#include <memory>
```

Variables

- std::unique_ptr< SmartOS > g_SmartOS

5.14.1 Variable Documentation

5.14.1.1 g_SmartOS

```
std::unique_ptr<SmartOS> g_SmartOS
```

Definition at line 7 of file main.cpp.

5.15 /home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.cpp File Reference

```
#include "HelpDialog.h"
```

5.16 /home/nicholas/qtcreator/smartos/SmartOS_GUI/HelpDialog.h File Reference

```
#include <QDialog>
```

Classes

- class HelpDialog

5.17 /home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.cpp File Reference

```
#include "HistoryDialog.h"
#include <QListWidget>
#include <QPushButton>
#include <QVBoxLayout>
```

5.18 /home/nicholas/qtcreator/smartos/SmartOS_GUI/HistoryDialog.h File Reference

```
#include <QDialog>
#include <QStringList>
```

Classes

- class HistoryDialog

5.19 /home/nicholas/qtcreator/smartos/SmartOS_GUI/main.cpp File Reference

```
#include "Globals.h"
#include "MainWindow.h"
#include <QApplication>
```

Functions

- int main (int argc, char ∗argv[ ])

Variables

- const size_t MEMORY = 1024
- std::unique_ptr< SmartOS > g_SmartOS = std::make_unique<SmartOS>(MEMORY)

5.19.1 Function Documentation

5.19.1.1 main()

```
int main (
            int argc,
            char * argv[] )
```

Definition at line 9 of file main.cpp.

5.19.2 Variable Documentation

5.19.2.1 g_SmartOS

```
std::unique_ptr<SmartOS> g_SmartOS = std::make_unique<SmartOS>(MEMORY)
```

Definition at line 7 of file main.cpp.

5.19.2.2 MEMORY

```
const size_t MEMORY = 1024
```

Definition at line 6 of file main.cpp.

5.20 /home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.cpp File Reference

```
#include "MainWindow.h"
#include "Globals.h"
#include "HistoryDialog.h"
#include "ProcessScheduler/ProcessSchedulerWidget.h"
#include "ProcessScheduler/SchedulingDialog.h"
#include "WelcomeWidget.h"
#include <QDesktopServices>
#include <QFileInfo>
#include <QHBoxLayout>
#include <QListWidget>
#include <QListWidgetItem>
#include <QMenu>
#include <QMenuBar>
#include <QMessageBox>
#include <QPushButton>
#include <QVBoxLayout>
#include <SmartOS.h>
#include <QDebug>
```

5.21 /home/nicholas/qtcreator/smartos/SmartOS_GUI/MainWindow.h File Reference

```
#include <QMainWindow>
#include <QStackedWidget>
#include <QStringList>
#include <QWidget>
```

Classes

- class MainWindow

5.22 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Process↩CreationDialog.cpp File Reference

```
#include "ProcessCreationDialog.h"
#include "Globals.h"
#include <QDebug>
#include <QGridLayout>
#include <QLabel>
#include <QMessageBox>
#include <QPushButton>
```

5.23 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Process↩CreationDialog.h File Reference

```
#include <ProcessControlBlock.h>
#include <QComboBox>
#include <QDialog>
#include <QLineEdit>
```

Classes

- class ProcessCreationDialog

5.24 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Process↩ SchedulerWidget.cpp File Reference

```
#include "ProcessSchedulerWidget.h"
#include "ProcessCreationDialog.h"
#include "MainWindow.h"
#include <QAction>
#include <QBoxLayout>
#include <QHBoxLayout>
#include <QIcon>
#include <QInputDialog>
#include <QLabel>
#include <QListView>
#include <QPushButton>
#include <QSlider>
#include <QStringListModel>
#include <QToolBar>
#include <QVBoxLayout>
#include <memory>
#include <random>
#include <SmartOS.h>
#include <QDebug>
```

Variables

- std::unique_ptr< SmartOS > g_SmartOS

5.24.1 Variable Documentation

5.24.1.1 g_SmartOS

```
std::unique_ptr<SmartOS> g_SmartOS
```

Definition at line 7 of file main.cpp.

5.25 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Process↩ SchedulerWidget.h File Reference

```
#include "BlockedQueueWidget.h"
#include "CpuWidget.h"
#include "ReadyQueueWidget.h"
#include <QAction>
#include <QLabel>
#include <QPaintEvent>
#include <QSlider>
#include <QTimer>
#include <QWidget>
```

Classes

- class ProcessSchedulerWidget

5.26 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Scheduling↩ Dialog.cpp File Reference

```
#include "SchedulingDialog.h"
#include <QGridLayout>
#include <QMessageBox>
#include <QPushButton>
#include <QRadioButton>
```

5.27 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessScheduler/Scheduling↩ Dialog.h File Reference

```
#include <SmartOS.h>
#include <QDialog>
#include <QRadioButton>
#include <QSpinBox>
```

Classes

- class SchedulingDialog

5.28 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.cpp File Reference

```
#include "ProcessWidget.h"
#include <QGridLayout>
#include <QLabel>
#include <QPainter>
#include <QDebug>
```

5.29 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ProcessWidget.h File Reference

```
#include <ProcessControlBlock.h>
#include <QLabel>
#include <QWidget>
```

Classes

- class ProcessWidget

5.30 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.cpp File Reference

```
#include "ReadyQueueWidget.h"
#include "Globals.h"
#include "ProcessWidget.h"
#include <QLabel>
#include <QListWidgetItem>
#include <QPainter>
#include <QVBoxLayout>
#include <QDebug>
```

5.31 /home/nicholas/qtcreator/smartos/SmartOS_GUI/ReadyQueueWidget.h File Reference

```
#include <QListWidget>
#include <QWidget>
```

Classes

- class ReadyQueueWidget

5.32 /home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.cpp File Reference

```
#include "WelcomeWidget.h"
#include <QHBoxLayout>
#include <QIcon>
#include <QPushButton>
#include <QVBoxLayout>
```

5.33 /home/nicholas/qtcreator/smartos/SmartOS_GUI/WelcomeWidget.h File Reference

```
#include "MainWindow.h"
#include <QWidget>
```

Classes

- class WelcomeWidget

Index