

Memory Manager

Generated by Doxygen 1.8.11

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	MemoryManager Class Reference	3
2.1.1	Detailed Description	3
2.1.2	Member Function Documentation	3
2.1.2.1	ReadMemory(int addr)	3
2.1.2.2	TranslateAddress(int addr)	4
2.2	MemoryPairAddress_t Struct Reference	4
2.3	PageTable Class Reference	4
2.3.1	Detailed Description	5
2.3.2	Member Function Documentation	5
2.3.2.1	GetLRUPage()	5
2.3.2.2	LookupPage(int pagenum)	5
2.3.2.3	LookupPage_no_LRU(int pagenum)	6
2.3.2.4	PagelsValid(int pagenum)	6
2.3.2.5	PageOut_table(int pagenum)	6
2.3.2.6	UpdateLRUList(int last_used)	6
2.4	PhysicalMemory Class Reference	7
2.4.1	Detailed Description	7
2.4.2	Member Function Documentation	7
2.4.2.1	FindFirstFrame()	7
2.4.2.2	GetMemoryContents(int frame, int offset)	7
2.4.2.3	isFull()	8
2.4.2.4	PageIn(int frame, char pagein[FRAME_SIZE])	8
2.4.2.5	PageOut(int frame)	8
	Index	9

Chapter 1

Class Index

1.1 Class List

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

MemoryManager		
	A memory management unit	3
MemoryPairAddress_t	4
PageTable		
	Page table holding page/frame pairs	4
PhysicalMemory		
	Imitates a physical memory	7

Chapter 2

Class Documentation

2.1 MemoryManager Class Reference

A memory management unit.

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

Public Member Functions

- [MemoryManager](#) ()
Constructor.
- char [ReadMemory](#) (int addr)
Read a value from memory.
- int [TranslateAddress](#) (int addr)
Translate a virtual address (P, d) to a physical address (f, d). Doesn't implement any p.
- void [PrintPageTable](#) ()
Print the page table.

2.1.1 Detailed Description

A memory management unit.

2.1.2 Member Function Documentation

2.1.2.1 char MemoryManager::ReadMemory (int *addr*)

Read a value from memory.

Parameters

<i>int</i>	Virtual address to read from.
------------	-------------------------------

Return values

<i>char</i>	value from mem[addr]
-------------	----------------------

2.1.2.2 int MemoryManager::TranslateAddress (int *addr*)

Translate a virtual address (P, d) to a physical address (f, d). Doesn't implement any p.

Parameters

<i>int</i>	Virtual address to translate.
------------	-------------------------------

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

- src/memory.h
- src/memory.cpp

2.2 MemoryPairAddress_t Struct Reference

Public Attributes

- int **P**
- int **d**

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

- src/memory.h

2.3 PageTable Class Reference

Page table holding page/frame pairs.

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


Public Member Functions

- [PageTable](#) ()
Constructor for [PageTable](#) object.
- int [LookupPage](#) (int pagenum)
Lookup a page number and return the corresponding frame.
- int [LookupPage_no_LRU](#) (int pagenum)
Lookup a page number, but don't update LRU calculations.
- void [SetPageToFrame](#) (int pagenum, int framenum)
Set a page table entry to a given frame.
- bool [PagelsValid](#) (int pagenum)
Determines if a page is loaded into physical memory.
- void [PrintPageTable](#) ()
Print out the page table.
- int [GetLRUPage](#) ()
Get the LRU page.
- void [UpdateLRUList](#) (int last_used)
Update the LRU list.
- void [PageOut_table](#) (int pagenum)
Page out the table.

2.3.1 Detailed Description

Page table holding page/frame pairs.

2.3.2 Member Function Documentation

2.3.2.1 int PageTable::GetLRUPage ()

Get the LRU page.

Return values

<i>int</i>	The integer value of the LRU page
------------	-----------------------------------

2.3.2.2 int PageTable::LookupPage (int pagenum)

Lookup a page number and return the corresponding frame.

Parameters

<i>int</i>	page
------------	------

Return values

<i>int</i>	frame
------------	-------

2.3.2.3 int PageTable::LookupPage_no_LRU (int *pagenum*)

Lookup a page number, but don't update LRU calculations.

Parameters

<i>int</i>	Page to Lookup
------------	----------------

Return values

<i>int</i>	Frame at
------------	----------

2.3.2.4 bool PageTable::PagelsValid (int *pagenum*)

Determines if a page is loaded into physical memory.

Parameters

<i>int</i>	Page number to check
------------	----------------------

Return values

<i>bool</i>	True if in memory (hit), False if not (miss)
-------------	--

2.3.2.5 void PageTable::PageOut_table (int *pagenum*)

Page out the table.

Parameters

<i>int</i>	The page to pageout.
------------	----------------------

2.3.2.6 void PageTable::UpdateLRUList (int *last_used*)

Update the LRU list.

Parameters

<i>int</i>	The latest used element
------------	-------------------------

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

- src/memory.h
- src/memory.cpp

2.4 PhysicalMemory Class Reference

Imitates a physical memory.

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

Public Member Functions

- [PhysicalMemory](#) ()
Constructor. Initializes memory to zero.
- int [FindFirstFrame](#) ()
Finds the first available frame in the memory.
- char [GetMemoryContents](#) (int frame, int offset)
Gets the byte at position (f, d)
- bool [isFull](#) ()
Returns true/false if the memory is full/empty.
- void [PageIn](#) (int frame, char pagein[FRAME_SIZE])
Pages a page into frame f.
- void [PageOut](#) (int frame)
Page out a frame.

2.4.1 Detailed Description

Imitates a physical memory.

2.4.2 Member Function Documentation

2.4.2.1 int PhysicalMemory::FindFirstFrame ()

Finds the first available frame in the memory.

Return values

<i>int</i>	Integer position of the first available frame.
------------	--

2.4.2.2 char PhysicalMemory::GetMemoryContents (int frame, int offset)

Gets the byte at position (f, d)

Parameters

<i>int</i>	Frame #
<i>int</i>	Offset in bytes

Return values

<i>char</i>	Byte at (f, d)
-------------	----------------

2.4.2.3 bool PhysicalMemory::isFull ()

Returns true/false if the memory is full/empty.

Return values

<i>bool</i>	True if memory is full, False otherwise
-------------	---

2.4.2.4 void PhysicalMemory::PageIn (int *frame*, char *pagein*[FRAME_SIZE])

Pages a page into frame f.

Parameters

<i>int</i>	Frame # to page into
<i>char</i> [FRAME_SIZE]	Contents of the frame

2.4.2.5 void PhysicalMemory::PageOut (int *frame*)

Page out a frame.

Parameters

<i>int</i>	Frame to page out
------------	-------------------

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

- src/memory.h
- src/memory.cpp

Index

- FindFirstFrame
 - PhysicalMemory, [7](#)
- GetLRUPage
 - PageTable, [5](#)
- GetMemoryContents
 - PhysicalMemory, [7](#)
- isFull
 - PhysicalMemory, [8](#)
- LookupPage
 - PageTable, [5](#)
- LookupPage_no_LRU
 - PageTable, [6](#)
- MemoryManager, [3](#)
 - ReadMemory, [3](#)
 - TranslateAddress, [4](#)
- MemoryPairAddress_t, [4](#)
- PageIn
 - PhysicalMemory, [8](#)
- PageIsValid
 - PageTable, [6](#)
- PageOut
 - PhysicalMemory, [8](#)
- PageOut_table
 - PageTable, [6](#)
- PageTable, [4](#)
 - GetLRUPage, [5](#)
 - LookupPage, [5](#)
 - LookupPage_no_LRU, [6](#)
 - PageIsValid, [6](#)
 - PageOut_table, [6](#)
 - UpdateLRUList, [6](#)
- PhysicalMemory, [7](#)
 - FindFirstFrame, [7](#)
 - GetMemoryContents, [7](#)
 - isFull, [8](#)
 - PageIn, [8](#)
 - PageOut, [8](#)
- ReadMemory
 - MemoryManager, [3](#)
- TranslateAddress
 - MemoryManager, [4](#)
- UpdateLRUList
 - PageTable, [6](#)