Memory Manager

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

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

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 virual 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

int	Virtual address to read from.

Return values

char	value from mem[addr]

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

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

Parameters

int	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 PageIsValid (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

int	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

int	page

Return values

int	frame

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

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

Parameters

int	Page to Lookup

Return values

int	Frame at

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

Determines if a page is loaded into physical memory.

Parameters

int	Page number to check

Return values

bool	True if in memory (hit), False if not (miss)

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

Page out the table.

Parameters

int	The page to pageout.

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

Update the LRU list.

Parameters

int	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

int	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

int	Frame #
int	Offset in bytes

Return values

char	Byte at (f, d)

2.4.2.3 bool PhysicalMemory::isFull ( )

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

Return values

bool	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

int	Frame # to page into
char[FRAME_SIZE]	Contents of the frame

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

Page out a frame.

Parameters

int	Frame to page out

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

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

Index