A Stressful Machine

Generated by Doxygen 1.8.13

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

Class Documentation

3.1 GameLevel Class Reference

```
#include <GameLevel.h>
```

Public Member Functions

- **GameLevel** (const string &level_name)
- bool **attempt_one_input** (const string &program, int verbose_level)
- void **text_play** ()

3.1.1 Detailed Description

This class enable the player to complete a level.

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

- /home/emile/etudes/LIFAP4/AStressfulMachine/src/GameLevel.h
- /home/emile/etudes/LIFAP4/AStressfulMachine/src/GameLevel.cpp

3.2 GameSequence Class Reference

```
#include <GameSequence.h>
```

Public Member Functions

- **GameSequence** (const string &savename)

3.2.1 Detailed Description

This class is used to represent a game of A Stressful Virtual Machine once the player choose a savefile.

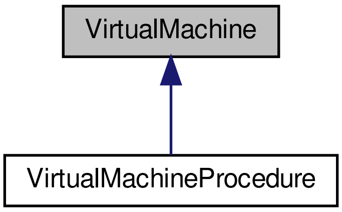
The documentation for this class was generated from the following file:

- /home/emile/etudes/LIFAP4/AStressfulMachine/src/GameSequence.h

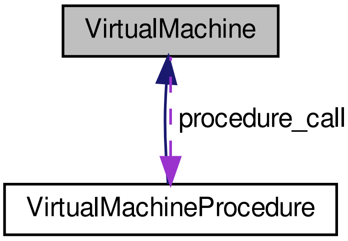
3.3 VirtualMachine Class Reference

```
#include <VirtualMachine.h>
```

Inheritance diagram for VirtualMachine:



Collaboration diagram for VirtualMachine:



Public Member Functions

- VirtualMachine (const string &program, istream *in, ostream *out, size_t size=30000, int *memory=nullptr)
- void do_one_iteration (bool advance=true)
- virtual void loop ()
- size_t get_size ()
- int get_current_operator ()
- int get_status ()
- int * get_memory ()
- int * get_memory_ptr ()
- void be_verbose ()
- void stop_verbose ()
- void be_verbose_procedure ()
- void stop_verbose_procedure ()
- virtual operator string ()
- ostream & **operator<<** (ostream &o)

Protected Member Functions

- void **initialize_anchor_map** ()
- void **ptr_incr** ()
- void **ptr_dincr** ()
- void **val_incr** ()
- void **val_dincr** ()
- virtual void **val_out** ()
- void **char_out** ()
- virtual void **val_in** ()
- void **handle_bracket** ()
- void **go_to_cond** ()
- void **go_to** ()
- void **go_to_anchor** (int anchor)
- void **exit_goto** ()
- void **ptr_jump** ()
- void **ptr_reset** ()
- void **val_reset** ()
- void **do_n_time** ()
- void **call_procedure** ()
- string **file_to_string** (string filename)
- void **loop_procedure** ()
- void **terminate_procedure** ()
- virtual void **error** (int code)
- int **extract_number_from_program** (unsigned int start_address, size_t *t=nullptr)
- virtual void **message** (const string &message)
- virtual string **memory_to_string** ()
- virtual string **program_to_string** ()

Protected Attributes

- string **program**
- istream ∗ **in**
- ostream ∗ **out**
- size_t **size**
- int ∗ **memory**
- int ∗ **memory_ptr**
- unsigned int **current_operator**
- int **status**
- bool **verbose**
- bool **verbose_procedure**
- map< unsigned int, unsigned int > **anchor_map**
- int **depth**
- VirtualMachineProcedure ∗ **procedure_call**

3.3.1 Detailed Description

This is the language interpreter class

3.3.2 Constructor & Destructor Documentation

3.3.2.1 VirtualMachine()

```
VirtualMachine::VirtualMachine (
              const string & program,
              istream * in,
              ostream * out,
              size_t size = 30000,
              int * memory = nullptr )
```

Constructor that initializes all the fields

Parameters

program	The code to be executed
in	The input stream
out	The output stream
size	The size of the memory. If the program starts with a number, this will be ignored
memory	The memory to be used by the machine. Allocated automatically if not specified.

3.3.3 Member Function Documentation

3.3.3.1 be_verbose()

```
void VirtualMachine::be_verbose ( )
```

Makes the VM verbose.

3.3.3.2 be_verbose_procedure()

```
void VirtualMachine::be_verbose_procedure ( )
```

Make the VM and its procedures verbose.

3.3.3.3 do_one_iteration()

```
void VirtualMachine::do_one_iteration (
             bool advance = true )
```

This do one iteration of the execution

Parameters

advance	This bool is here to tell the VM if it should advance in the program or redo the same operator next time.

3.3.3.4 get_current_operator()

```
int VirtualMachine::get_current_operator ( )
```

Getter for member current_operator

Returns

current_operator

3.3.3.5 get_memory()

```
int * VirtualMachine::get_memory ( )
```

Getter for member memory

Returns

memory

3.3.3.6 get_memory_ptr()

```
int * VirtualMachine::get_memory_ptr ( )
```

Getter for member memory_ptr

Returns

memory_ptr

3.3.3.7 get_size()

```
size_t VirtualMachine::get_size ( )
```

Getter for member size

Returns

size

3.3.3.8 get_status()

```
int VirtualMachine::get_status ( )
```

Getter for member status

Returns

status

3.3.3.9 loop()

```
void VirtualMachine::loop ( ) [virtual]
```

This execute the program until it halts.

3.3.3.10 operator string()

```
VirtualMachine::operator string ( ) [explicit], [virtual]
```

Convert the VM's current state into a string.

Returns

The VM's string representation

Reimplemented in VirtualMachineProcedure.

3.3.3.11 stop_verbose()

```
void VirtualMachine::stop_verbose ( )
```

Makes the VM silent.

3.3.3.12 stop_verbose_procedure()

```
void VirtualMachine::stop_verbose_procedure ( )
```

Make the VM's procedures silent.

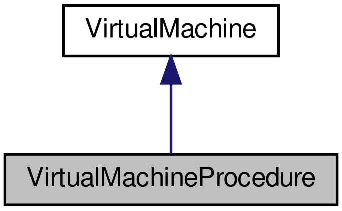
The documentation for this class was generated from the following files:

- /home/emile/etudes/LIFAP4/AStressfulMachine/src/VirtualMachine.h
- /home/emile/etudes/LIFAP4/AStressfulMachine/src/VirtualMachine.cpp

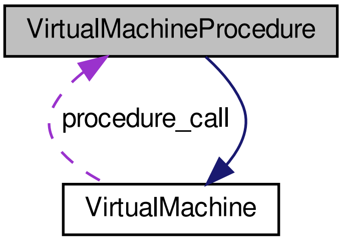
3.4 VirtualMachineProcedure Class Reference

```
#include <VirtualMachineProcedure.h>
```

Inheritance diagram for VirtualMachineProcedure:



Collaboration diagram for VirtualMachineProcedure:



Public Member Functions

- VirtualMachineProcedure (const string &program, istream ∗in, ostream ∗out, int depth, size_t size=30000, int ∗memory=nullptr)
- int get_output ()
- void **input** (int inpt)
- operator string () override

Public Attributes

- int **depth**

Protected Member Functions

- void **val_out** () override
- void **val_in** () override
- void **error** (int code) override
- void **message** (const string &message) override

Protected Attributes

- int **output**

3.4.1 Detailed Description

This class is used by VirtualMachine to perform procedure calls. It is only the input and output and the verbose/error printing that are changed.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 VirtualMachineProcedure()

```
VirtualMachineProcedure::VirtualMachineProcedure (
             const string & program,
             istream ∗ in,
             ostream ∗ out,
             int depth,
             size_t size = 30000,
             int ∗ memory = nullptr )
```

This call the VirtualMachine constructor as a delegate constructor and initialize the new fields.

Parameters

program	
in	
out	
depth	The recursive depth of the procedure.
size	
memory	

3.4.3 Member Function Documentation

3.4.3.1 get_output()

```
int VirtualMachineProcedure::get_output ( )
```

This should be used to access the output of the procedure.

Returns

The output of the procedure

3.4.3.2 operator string()

```
VirtualMachineProcedure::operator string ( )  [explicit], [override], [virtual]
```

Convert the VM's current state into a string.

Returns

The VM's string representation

Reimplemented from VirtualMachine.

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

- /home/emile/etudes/LIFAP4/AStressfulMachine/src/VirtualMachineProcedure.h
- /home/emile/etudes/LIFAP4/AStressfulMachine/src/VirtualMachineProcedure.cpp

Index