

# SyscallCapability Reference Manual

Generated by Doxygen 1.5.3

Tue Feb 19 19:52:33 2008



# Contents

<b>1</b>	<b>SyscallCapability Namespace Index</b>	<b>1</b>
1.1	SyscallCapability Namespace List	1
<b>2</b>	<b>SyscallCapability Hierarchical Index</b>	<b>3</b>
2.1	SyscallCapability Class Hierarchy	3
<b>3</b>	<b>SyscallCapability Class Index</b>	<b>5</b>
3.1	SyscallCapability Class List	5
<b>4</b>	<b>SyscallCapability Namespace Documentation</b>	<b>7</b>
4.1	full_system Namespace Reference	7
4.2	full_system::plugins Namespace Reference	8
4.3	full_system::plugins::os Namespace Reference	9
4.4	full_system::plugins::os::linux_os Namespace Reference	10
4.5	full_system::plugins::os::linux_os::arm Namespace Reference	11
4.6	full_system::plugins::os::linux_os::powerpc Namespace Reference	12
<b>5</b>	<b>SyscallCapability Class Documentation</b>	<b>13</b>
5.1	full_system::plugins::os::linux_os::arm::ARMLinux Class Reference	13
5.2	full_system::plugins::os::linux_os::arm::ARMLinuxOS Class Reference	15
5.3	full_system::plugins::os::linux_os::arm::ARMLinuxOSInterface Class Reference	18
5.4	full_system::plugins::os::linux_os::arm::CPUARMLinuxOSInterface Class Reference	19
5.5	full_system::plugins::os::linux_os::CPULinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > Class Template Reference	20
5.6	full_system::plugins::os::CPUOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > Class Template Reference	22
5.7	full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE > Class Template Reference	24
5.8	full_system::plugins::os::linux_os::LinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > Class Template Reference	27
5.9	full_system::plugins::os::OSInterface Class Reference	29

---

5.10	<a href="#">full_system::plugins::os::linux_os::powerpc::PowerPCLinux Class Reference . . . . .</a>	30
5.11	<a href="#">full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS Class Reference . . . . .</a>	32
5.12	<a href="#">full_system::plugins::os::linux_os::powerpc::PowerPCSyscall Class Reference . . . . .</a>	36
5.13	<a href="#">full_system::plugins::os::linux_os::powerpc::StandalonePowerPCLoader Class Reference . . . . .</a>	37
5.14	<a href="#">full_system::plugins::os::linux_os::SyscallJournal Class Reference . . . . .</a>	38
5.15	<a href="#">full_system::plugins::os::linux_os::SyscallJournalOpenedFile Class Reference . . . . .</a>	39
5.16	<a href="#">full_system::plugins::os::linux_os::SyscallMemoryDispatcher&lt; nCPU &gt; Class Template Reference . . . . .</a>	40
5.17	<a href="#">full_system::plugins::os::linux_os::UnimplementedSystemCall Class Reference . . . . .</a>	42

# Chapter 1

## SyscallCapability Namespace Index

### 1.1 SyscallCapability Namespace List

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

<a href="#">full_system</a> . . . . .	7
<a href="#">full_system::plugins</a> . . . . .	8
<a href="#">full_system::plugins::os</a> . . . . .	9
<a href="#">full_system::plugins::os::linux_os</a> . . . . .	10
<a href="#">full_system::plugins::os::linux_os::arm</a> . . . . .	11
<a href="#">full_system::plugins::os::linux_os::powerpc</a> . . . . .	12



## Chapter 2

# SyscallCapability Hierarchical Index

### 2.1 SyscallCapability Class Hierarchy

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

full_system::plugins::os::linux_os::arm::ARMLinux . . . . .	13
full_system::plugins::os::linux_os::arm::ARMLinuxOS . . . . .	15
full_system::plugins::os::linux_os::arm::ARMLinuxOSInterface . . . . .	18
full_system::plugins::os::linux_os::arm::CPUARMLinuxOSInterface . . . . .	19
full_system::plugins::os::CPUOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > . . . . .	22
full_system::plugins::os::linux_os::CPULinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > . . . . .	20
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE > . . . . .	24
full_system::plugins::os::OSInterface . . . . .	29
full_system::plugins::os::linux_os::LinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE > . . . . .	27
full_system::plugins::os::linux_os::powerpc::PowerPCLinux . . . . .	30
full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS . . . . .	32
full_system::plugins::os::linux_os::powerpc::PowerPCSyscall . . . . .	36
full_system::plugins::os::linux_os::powerpc::StandalonePowerPCLoader . . . . .	37
full_system::plugins::os::linux_os::SyscallJournal . . . . .	38
full_system::plugins::os::linux_os::SyscallJournalOpenedFile . . . . .	39
full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU > . . . . .	40
full_system::plugins::os::linux_os::UnimplementedSystemCall . . . . .	42



# Chapter 3

## SyscallCapability Class Index

### 3.1 SyscallCapability Class List

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

<code>full_system::plugins::os::linux_os::arm::ARMLinux</code> . . . . .	13
<code>full_system::plugins::os::linux_os::arm::ARMLinuxOS</code> . . . . .	15
<code>full_system::plugins::os::linux_os::arm::ARMLinuxOSInterface</code> . . . . .	18
<code>full_system::plugins::os::linux_os::arm::CPUARMLinuxOSInterface</code> . . . . .	19
<code>full_system::plugins::os::linux_os::CPULinuxOSInterface&lt; ADDRESS_TYPE, PARAMETER_TYPE &gt;</code> . . . . .	20
<code>full_system::plugins::os::CPUOSInterface&lt; ADDRESS_TYPE, PARAMETER_TYPE &gt;</code> . . . . .	22
<code>full_system::plugins::os::linux_os::LinuxOS&lt; ADDRESS_TYPE, PARAMETER_TYPE &gt;</code> . . . . .	24
<code>full_system::plugins::os::linux_os::LinuxOSInterface&lt; ADDRESS_TYPE, PARAMETER_TYPE &gt;</code> . . . . .	27
<code>full_system::plugins::os::OSInterface</code> . . . . .	29
<code>full_system::plugins::os::linux_os::powerpc::PowerPCLinux</code> . . . . .	30
<code>full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS</code> . . . . .	32
<code>full_system::plugins::os::linux_os::powerpc::PowerPCSyscall</code> . . . . .	36
<code>full_system::plugins::os::linux_os::powerpc::StandalonePowerPCLoader</code> . . . . .	37
<code>full_system::plugins::os::linux_os::SyscallJournal</code> . . . . .	38
<code>full_system::plugins::os::linux_os::SyscallJournalOpenedFile</code> . . . . .	39
<code>full_system::plugins::os::linux_os::SyscallMemoryDispatcher&lt; nCPU &gt;</code> . . . . .	40
<code>full_system::plugins::os::linux_os::UnimplementedSystemCall (Exception : Unimplemented Syscall)</code> . . . . .	42



## Chapter 4

# SyscallCapability Namespace Documentation

### 4.1 full\_system Namespace Reference

#### Namespaces

- namespace [plugins](#)

## 4.2 full\_system::plugins Namespace Reference

### Namespaces

- namespace [os](#)

---

## 4.3 full\_system::plugins::os Namespace Reference

### Classes

- class [CPUOSInterface](#)
- class [OSInterface](#)

### Namespaces

- namespace [linux\\_os](#)

## 4.4 full\_system::plugins::os::linux\_os Namespace Reference

### Classes

- class [CPUlinuxOSInterface](#)
- class [UnimplementedSystemCall](#)  
*Exception : Unimplemented Syscall.*
  
- class [LinuxOS](#)
- class [LinuxOSInterface](#)
- class [SyscallJournalOpenedFile](#)
- class [SyscallJournal](#)
- class [SyscallMemoryDispatcher](#)

### Namespaces

- namespace [arm](#)
- namespace [powerpc](#)

### Functions

- ostream & [operator<<](#) (ostream &os, const [SyscallJournalOpenedFile](#) &f)  
*Pretty printer of the [SyscallJournalOpenedFile](#) class.*

## 4.5 full\_system::plugins::os::linux\_os::arm Namespace Reference

### Classes

- class [ARMLinux](#)
- class [ARMLinuxOS](#)
- class [ARMLinuxOSInterface](#)
- class [CPUARMLinuxOSInterface](#)

## 4.6 full\_system::plugins::os::linux\_os::powerpc Namespace Reference

### Classes

- class [PowerPCLinux](#)
- class [PowerPCSyscall](#)
- class [PowerPCLinuxOS](#)
- class [StandalonePowerPCLoader](#)

# Chapter 5

## SyscallCapability Class Documentation

### 5.1 full\_system::plugins::os::linux\_os::arm::ARMLinux Class Reference

Collaboration diagram for full\_system::plugins::os::linux\_os::arm::ARMLinux:



#### Public Member Functions

- [ARMLinux](#) (const char \*name, Object \*parent=0)  
*Class constructor.*
- [~ARMLinux](#) ()  
*Class destructor.*

#### Public Attributes

- `ServiceImport< MemoryInterface< physical_address_t > >` [memory\\_import](#)  
*Service port to memory.*
- `ServiceImport< BusInterface< physical_address_t > >` [bus\\_import](#)  
*Service port to bus.*
- `ServiceImport< ARMLinuxOSInterface, CPUARMLinuxOSInterface >` [armlinuxos\\_imp](#)  
*Service port from/to [LinuxOS](#).*
- `ServiceExport< ARMLinuxOSInterface, CPUARMLinuxOSInterface >` [armlinuxos\\_exp](#)  
*Service port from/to [LinuxOS](#).*
- `ServiceExport< SymbolTableLookupInterface< address_t > >` [syntablookup\\_exp](#)

*Service port from symbol table.*

### 5.1.1 Detailed Description

Definition at line 76 of file arm-linux.hh.

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

- linux/arm/arm-linux.hh

## 5.2 full\_system::plugins::os::linux\_os::arm::ARMLinuxOS Class Reference

### Public Member Functions

- [ARMLinuxOS](#) (const char \*name, Object \*parent=0)  
*Class constructor.*
- virtual [~ARMLinuxOS](#) ()
- virtual void [OnDisconnect](#) ()
- virtual bool [ClientIndependentSetup](#) ()  
*Service interface : initialization.*
- virtual bool [ClientDependentSetup](#) ()  
*Service interface : initialization.*
- virtual void [ExecuteSystemCall](#) (int id)  
*Request to [LinuxOS](#) to execute the current system call.*
- virtual void \* [ConvertStat](#) (struct stat \*)  
*Convert the native "stat" structure to target architecture struct.*
- virtual unsigned int [StatSize](#) (void)  
*Returns the size of the 'stat' struct on the target architecture.*
- virtual void \* [ConvertTime](#) (struct timeval \*)  
*Convert the native "timeval" structure to target architecture struct.*
- virtual unsigned int [TimeSize](#) (void)  
*Returns the size of the 'timesize' struct on the target architecture.*
- virtual void \* [ConvertStat64](#) (struct stat64 \*)  
*Convert the native "stat64" structure to target architecture struct.*
- virtual unsigned int [Stat64Size](#) (void)  
*Returns the size of the 'stat64' struct on the target architecture.*
- virtual void \* [ConvertTms](#) (struct tms \*)  
*Convert the native "tms" structure to target architecture struct.*
- virtual unsigned int [TmsSize](#) (void)  
*Returns the size of the 'tms' struct on the target architecture.*
- virtual int [GetSystemCallNumber](#) ()  
*Retrieve the number of the syscall.*
- virtual address\_t [GetMmapBase](#) () const  
*Retrieve the mmap base address.*

- virtual void [SetMmapBase](#) (address\_t base)  
*Set the mmap base address.*
- virtual address\_t [GetMmapBrkPoint](#) () const  
*Retrieve the mmap breakpoint.*
- virtual void [SetMmapBrkPoint](#) (address\_t brk\_point)  
*Set the mmap breakpoint.*
- virtual address\_t [GetBrkPoint](#) () const  
*Retrieve the breakpoint.*
- virtual void [SetBrkPoint](#) (address\_t brk\_point)  
*Set the breakpoint.*
- virtual reg\_t [GetSystemCallParam](#) (int id)  
*Retrieve one of the syscall parameters.*
- virtual void [PerformExit](#) (int ret)  
*Called upon reaching the "exit" syscall.*
- virtual void [SetSystemCallStatus](#) (int ret, bool error)  
*Set the system call exit status.*
- virtual endian\_type [GetEndianness](#) ()  
*Returns the endianness of the target architecture.*
- virtual void [Reset](#) ()  
*Memory Interface : reset.*
- virtual bool [ReadMemory](#) (address\_t addr, void \*buffer, uint32\_t size)  
*Memory interface : Read from memory.*
- virtual bool [WriteMemory](#) (address\_t addr, const void \*buffer, uint32\_t size)  
*Memory interface : Write to memory.*

## Public Attributes

- ServiceImport< LoaderInterface< address\_t > > [linux\\_loader](#)  
*Service port to the linux loader.*
- ServiceImport< [LinuxOSInterface](#)< address\_t, reg\_t >, [CPULinuxOSInterface](#)< address\_t, reg\_t > > [linux\\_os](#)  
*Service port from/to the CPU.*
- ServiceExport< [ARMLinuxOSInterface](#), [CPUARMLinuxOSInterface](#) > [exp](#)  
*Service port from/to LinuxOS.*

## Protected Attributes

- endian\_type [endianess](#)  
*Enduaness of the target architecture.*
- reg\_t [memory\\_page\\_size](#)  
*Memory page size.*

## Classes

- struct [arm\\_stat64\\_t](#)
- struct [arm\\_timespec\\_t](#)
- struct [arm\\_tms\\_t](#)

### 5.2.1 Detailed Description

Definition at line 70 of file arm\_linux\_os.hh.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 full\_system::plugins::os::linux\_os::arm::ARMLinuxOS::~~ARMLinuxOS () [virtual]

Class destructor.

Definition at line 255 of file arm\_linux\_os.hh.

### 5.2.3 Member Function Documentation

#### 5.2.3.1 void full\_system::plugins::os::linux\_os::arm::ARMLinuxOS::OnDisconnect () [virtual]

Method to execute when [ARMLinuxOS](#) is disconnected from its client.

Definition at line 262 of file arm\_linux\_os.hh.

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

- linux/arm/arm\_linux\_os.hh

## 5.3 full\_system::plugins::os::linux\_os::arm::ARMLinuxOSInterface Class Reference

### Public Member Functions

- virtual void [ExecuteSystemCall](#) (int id)=0  
*Execute the system call natively.*

### 5.3.1 Detailed Description

Definition at line 49 of file arm\_linux\_os\_interface.hh.

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

- linux/arm/arm\_linux\_os\_interface.hh

## 5.4 full\_system::plugins::os::linux\_os::arm::CPUARMLinuxOSInterface Class Reference

### Public Member Functions

- virtual void ARMLinuxOSExit (int ret)=0

#### 5.4.1 Detailed Description

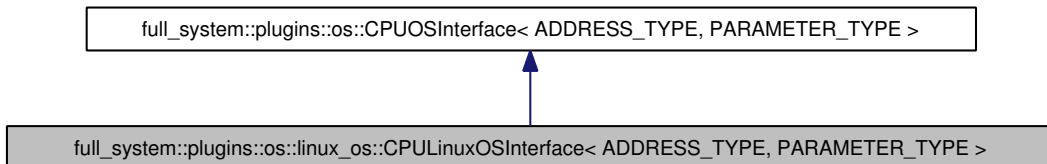
Definition at line 56 of file cpu\_arm\_linux\_os\_interface.hh.

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

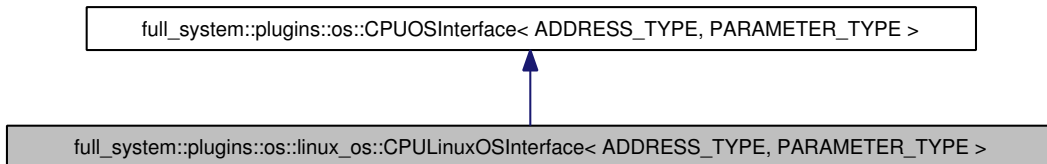
- linux/arm/cpu\_arm\_linux\_os\_interface.hh

## 5.5 full\_system::plugins::os::linux\_os::CPUlinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE > Class Template Reference

Inheritance diagram for full\_system::plugins::os::linux\_os::CPUlinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE >:



Collaboration diagram for full\_system::plugins::os::linux\_os::CPUlinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE >:



### Public Member Functions

- virtual void \* [ConvertTime](#) (struct timeval \*)=0  
*Convert native "timeval" structure to target architecture struct.*
- virtual unsigned int [TimeSize](#) (void)=0  
*Return the target architecture size of the "time" structure.*
- virtual void \* [ConvertStat](#) (struct stat \*)=0  
*Convert native "stat" structure to target architecture struct.*
- virtual unsigned int [StatSize](#) (void)=0  
*Return the target architecture size of the "stat" structure.*
- virtual void \* [ConvertStat64](#) (struct stat64 \*)=0  
*Convert native "stat64" structure to target architecture struct.*
- virtual unsigned int [Stat64Size](#) (void)=0  
*Return the target architecture size of the "stat64" structure.*
- virtual void \* [ConvertTms](#) (struct tms \*)=0  
*Convert native "tms" structure to target architecture struct.*
- virtual unsigned int [TmsSize](#) (void)=0  
*Return the target architecture size of the "tms" structure.*

### 5.5.1 Detailed Description

```
template<class ADDRESS_TYPE, class PARAMETER_TYPE> class full_
system::plugins::os::linux_os::CPUlinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE
>
```

Definition at line 53 of file cpu\_linux\_os\_interface.hh.

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

- linux/cpu\_linux\_os\_interface.hh

## 5.6 full\_system::plugins::os::CPUOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE > Class Template Reference

Inheritance diagram for full\_system::plugins::os::CPUOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE >:



### Public Member Functions

- virtual int [GetSystemCallNumber](#) ()=0  
*Retrieve the syscall number.*
- virtual ADDRESS\_TYPE [GetMmapBase](#) () const =0  
*Retrieve the base address form mmap.*
- virtual void [SetMmapBase](#) (ADDRESS\_TYPE base)=0  
*Set the base address for mmap.*
- virtual ADDRESS\_TYPE [GetMmapBrkPoint](#) () const =0  
*Retrieve the mmap breakpoint.*
- virtual void [SetMmapBrkPoint](#) (ADDRESS\_TYPE brk\_point)=0  
*Set the mmap breakpoint.*
- virtual ADDRESS\_TYPE [GetBrkPoint](#) () const =0  
*Retrieve the breakpoint.*
- virtual void [SetBrkPoint](#) (ADDRESS\_TYPE brk\_point)=0  
*Set the breakpoint.*
- virtual PARAMETER\_TYPE [GetSystemCallParam](#) (int id)=0  
*Retrieve one of the syscall parameters.*
- virtual void [PerformExit](#) (int ret)=0  
*Action that should be performed upon reaching the Exit syscall.*
- virtual void [SetSystemCallStatus](#) (int ret, bool error)=0  
*Set the syscall return status.*
- virtual PARAMETER\_TYPE [GetSystemCallParam](#) (unsigned int parm\_num)=0  
*Gets the register identified by name.*
- virtual unsigned int [GetSystemCallNumber](#) ()=0  
*Retrieve the syscall number.*
- virtual void [SetSystemCallStatus](#) (PARAMETER\_TYPE value, bool is\_error)=0  
*Set the system call return status.*

- virtual void [Exit](#) (int ret)=0

*To report to the main simulation loop that the program is finished.*

### 5.6.1 Detailed Description

```
template<class ADDRESS_TYPE, class PARAMETER_TYPE> class full_
system::plugins::os::CPUOSInterface< ADDRESS_TYPE, PARAMETER_TYPE >
```

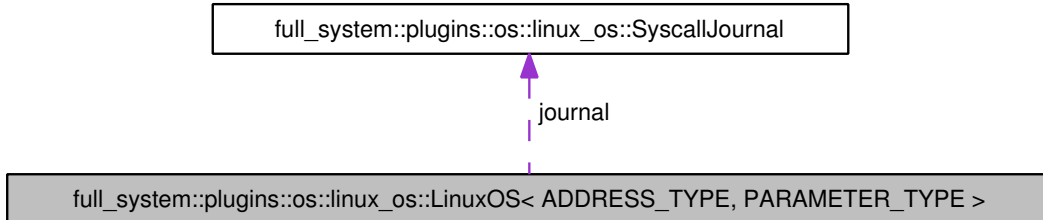
Definition at line 47 of file cpu\_os\_interface.hh.

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

- cpu\_os\_interface.hh
- os.hh

## 5.7 full\_system::plugins::os::linux\_os::LinuxOS< ADDRESS\_TYPE, PARAMETER\_TYPE > Class Template Reference

Collaboration diagram for full\_system::plugins::os::linux\_os::LinuxOS< ADDRESS\_TYPE, PARAMETER\_TYPE >:



### Public Member Functions

- [LinuxOS](#) (const char \*name, Object \*parent=0)
- virtual [~LinuxOS](#) ()
- virtual void [OnDisconnect](#) ()
- virtual bool [ClientIndependentSetup](#) ()
- virtual bool [ClientDependentSetup](#) ()
- virtual void [SetSyscallId](#) (const string &syscall\_name, int syscall\_id)
- virtual void [ExecuteSystemCall](#) ()
- virtual void [save\\_checkpoint](#) (std::ostream &os)

*Save the syscall journal to a checkpoint file.*

- virtual void [load\\_checkpoint](#) (std::istream &is)

*Load the syscall journal from a checkpoint file.*

- void [set\\_stdin](#) (const string &filename)

*Mimic "stdin" from an input file.*

- void [set\\_stdout](#) (const string &filename)

*Mimic "stdout" to an output file.*

- void [set\\_stderr](#) (const string &filename)

*Mimic "stderr" to an output file.*

### Public Attributes

- ServiceExport< [LinuxOSInterface](#)< ADDRESS\_TYPE, PARAMETER\_TYPE >, [CPULinuxOSInterface](#)< ADDRESS\_TYPE, PARAMETER\_TYPE > > [exp](#)

*Service port from/to the CPU.*

- ServiceImport< [PthreadInterface](#) > [pthread\\_port](#)

*Service port to the pthread enabled CPU.*

## Protected Attributes

- endian\_type [endianess](#)  
*Endianess of the connected processor.*
- bool [verbose](#)  
*Set the verbosity level.*

### 5.7.1 Detailed Description

```
template<class ADDRESS_TYPE, class PARAMETER_TYPE> class full_ -  
system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE >
```

Definition at line 114 of file linux\_os.hh.

### 5.7.2 Constructor & Destructor Documentation

```
5.7.2.1 template<class ADDRESS_TYPE, class PARAMETER_TYPE>  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::LinuxOS (const char * name, Object * parent = 0) [inline]
```

Constructor.

Definition at line 328 of file linux\_os.hh.

```
5.7.2.2 template<class ADDRESS_TYPE, class PARAMETER_TYPE>  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::~~LinuxOS () [inline, virtual]
```

Destructor.

Definition at line 348 of file linux\_os.hh.

### 5.7.3 Member Function Documentation

```
5.7.3.1 template<class ADDRESS_TYPE, class PARAMETER_TYPE> void  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::OnDisconnect () [inline, virtual]
```

Method to execute when the [LinuxOS](#) is disconnected from its client.

Definition at line 354 of file linux\_os.hh.

```
5.7.3.2 template<class ADDRESS_TYPE, class PARAMETER_TYPE> bool  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::ClientIndependentSetup () [inline, virtual]
```

Client independent setup.

Definition at line 367 of file linux\_os.hh.

**5.7.3.3** `template<class ADDRESS_TYPE, class PARAMETER_TYPE> bool  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::ClientDependentSetup () [inline, virtual]`

Client dependent setup.

Definition at line 376 of file linux\_os.hh.

References `full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::exp`.

**5.7.3.4** `template<class ADDRESS_TYPE, class PARAMETER_TYPE> void  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::SetSyscallId (const string & syscall_name, int syscall_id) [inline, virtual]`

Associates an id (integer) to a the implementation of the syscall specified by `syscall_name`.

**Parameters:**

*syscall\_name*

*syscall\_id*

Definition at line 413 of file linux\_os.hh.

**5.7.3.5** `template<class ADDRESS_TYPE, class PARAMETER_TYPE> void  
full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::ExecuteSystemCall () [inline, virtual]`

Gets from the cpu calling this method the syscall id to execute and emulates it if it exists.

Definition at line 438 of file linux\_os.hh.

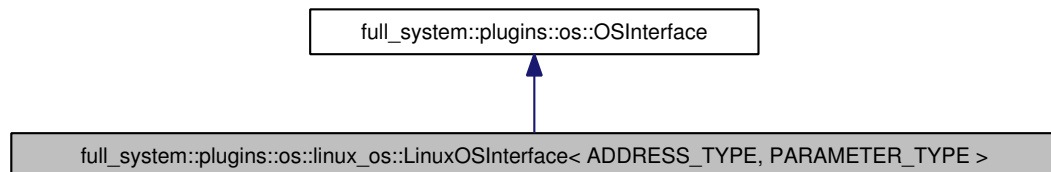
References `full_system::plugins::os::linux_os::LinuxOS< ADDRESS_TYPE, PARAMETER_TYPE  
>::exp`.

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

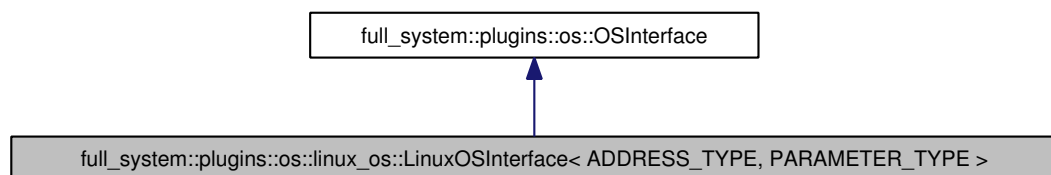
- linux/linux\_os.hh

## 5.8 full\_system::plugins::os::linux\_os::LinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE > Class Template Reference

Inheritance diagram for full\_system::plugins::os::linux\_os::LinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE >:



Collaboration diagram for full\_system::plugins::os::linux\_os::LinuxOSInterface< ADDRESS\_TYPE, PARAMETER\_TYPE >:



### Public Member Functions

- virtual void [SetSyscallId](#) (const string &syscall\_name, int syscall\_id)=0

#### 5.8.1 Detailed Description

```
template<class ADDRESS_TYPE, class PARAMETER_TYPE> class full_system::plugins::os::linux_os::LinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE >
```

Definition at line 49 of file linux\_os\_interface.hh.

#### 5.8.2 Member Function Documentation

**5.8.2.1** `template<class ADDRESS_TYPE, class PARAMETER_TYPE> virtual void full_system::plugins::os::linux_os::LinuxOSInterface< ADDRESS_TYPE, PARAMETER_TYPE >::SetSyscallId (const string & syscall_name, int syscall_id) [pure virtual]`

Set the binding between a system call implementation defined by its name and its identifier.

##### Parameters:

*syscall\_name* The name of the system call we want to bind.

*syscall\_id* The number identifier that the system call will have.

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

- linux/linux\_os\_interface.hh

## 5.9 full\_system::plugins::os::OSInterface Class Reference

Inheritance diagram for full\_system::plugins::os::OSInterface:



### Public Member Functions

- virtual void [ExecuteSystemCall](#) ()=0

#### 5.9.1 Detailed Description

Definition at line 44 of file os\_interface.hh.

#### 5.9.2 Member Function Documentation

##### 5.9.2.1 virtual void full\_system::plugins::os::OSInterface::ExecuteSystemCall () [pure virtual]

Execute an [os](#) system call.

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

- os\_interface.hh

## 5.10 full\_system::plugins::os::linux\_os::powerpc::PowerPCLinux Class Reference

Collaboration diagram for full\_system::plugins::os::linux\_os::powerpc::PowerPCLinux:



### Public Member Functions

- [PowerPCLinux](#) (const char \*name, Object \*parent=0)  
*Class constructor.*
- [~PowerPCLinux](#) ()  
*Class destructor.*
- void [set\\_stdin](#) (const string &filename)  
*Set the input file to be used as stdin.*
- void [set\\_stdout](#) (const string &filename)  
*Set the output file to be used as stdout.*
- void [set\\_stderr](#) (const string &filename)  
*Set the output file to be used as stderr.*

### Public Attributes

- ServiceImport< MemoryInterface< physical\_address\_t > > [memory\\_import](#)  
*Service port to memory.*
- ServiceImport< BusInterface< physical\_address\_t > > [bus\\_import](#)  
*Service port to bus.*
- ServiceImport< [ARMLinuxOSInterface](#), [CPUARMLinuxOSInterface](#) > [ppclinuxos\\_imp](#)  
*Service port from/to ppclinuxos.*
- ServiceExport< [ARMLinuxOSInterface](#), [CPUARMLinuxOSInterface](#) > [ppclinuxos\\_exp](#)  
*Service port from/to ppclinuxos.*
- ServiceExport< SymbolTableLookupInterface< address\_t > > [symtablelookup\\_exp](#)  
*Service port from symbol table.*
- [LinuxOS](#)< address\_t, reg\_t > \* [linux\\_os](#)  
*LinuxOS.*
- [PowerPCLinuxOS](#) \* [ppc\\_linux\\_os](#)  
*PowerPC LinuxOS.*

- `SymbolTable< address_t > * symbol_table`

*Symbol table.*

### 5.10.1 Detailed Description

Definition at line 73 of file powerpc-linux.hh.

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

- linux/powerpc/powerpc-linux.hh

## 5.11 full\_system::plugins::os::linux\_os::powerpc::PowerPCLinuxOS Class Reference

### Public Member Functions

- [PowerPCLinuxOS](#) (const char \*name, Object \*parent=0)
- virtual [~PowerPCLinuxOS](#) ()
- virtual void [OnDisconnect](#) ()
- virtual bool [ClientIndependentSetup](#) ()
- virtual bool [ClientDependentSetup](#) ()
- virtual void [ExecuteSystemCall](#) (int id)
- virtual void \* [ConvertStat](#) (struct stat \*)  
*Convert the native "stat" structure to the target architecture struct.*
- virtual unsigned int [StatSize](#) (void)  
*Return the size of the "stat" struct on target architecture.*
- virtual void \* [ConvertTime](#) (struct timeval \*)  
*Convert the native "tms" structure to the target architecture struct.*
- virtual unsigned int [TimeSize](#) (void)  
*Return the size of the "timespec" struct on target architecture.*
- virtual void \* [ConvertStat64](#) (struct stat64 \*)  
*Convert the native "stat64" structure to the target architecture struct.*
- virtual unsigned int [Stat64Size](#) (void)  
*Returns the size of the "stat64" struct on target architecture.*
- virtual void \* [ConvertTms](#) (struct tms \*)  
*Convert the native "tms" structure to the target architecture struct.*
- virtual unsigned int [TmsSize](#) (void)  
*Return the size of the "tms" struct on target architecture.*
- virtual int [GetSystemCallNumber](#) ()  
*Get the current syscall number.*
- virtual address\_t [GetMmapBase](#) () const  
*Retrieve the mmap base address.*
- virtual void [SetMmapBase](#) (address\_t base)  
*Set the mmap base address.*
- virtual address\_t [GetMmapBrkPoint](#) () const  
*Retrieve the mmap breakpoint.*
- virtual void [SetMmapBrkPoint](#) (address\_t brk\_point)  
*Set the mmap breakpoint.*

- virtual address\_t [GetBrkPoint](#) () const  
*Retrieve the breakpoint.*
- virtual void [SetBrkPoint](#) (address\_t brk\_point)  
*Set the breakpoint.*
- virtual reg\_t [GetSystemCallParam](#) (int id)  
*Retrieve one of the system call parameters.*
- virtual void [PerformExit](#) (int ret)  
*Raised upon reaching the "exit" syscall.*
- virtual void [SetSystemCallStatus](#) (int ret, bool error)  
*Set the exit status of the system call.*
- virtual endian\_type [GetEndianness](#) ()  
*Get the endianness of the target architecture.*
- virtual void [Reset](#) ()  
*Memory interface : Reset.*
- virtual bool [ReadMemory](#) (address\_t addr, void \*buffer, uint32\_t size)  
*Memory interface : Read from memory.*
- virtual bool [WriteMemory](#) (address\_t addr, const void \*buffer, uint32\_t size)  
*Memory interface : Write to memory.*

## Public Attributes

- ServiceImport< LoaderInterface< address\_t > > [linux\\_loader](#)  
*Service port to Linux Loader.*
- ServiceImport< [LinuxOSInterface](#)< address\_t, reg\_t >, [CPULinuxOSInterface](#)< address\_t, reg\_t > > [linux\\_os](#)  
*Service port from/to LinuxOS.*
- ServiceExport< [ARMLinuxOSInterface](#), [CPUARMLinuxOSInterface](#) > [exp](#)  
*Service port from/to LinuxOS.*

## Protected Attributes

- endian\_type [endianness](#)  
*Endianness of the target architecture.*
- reg\_t [memory\\_page\\_size](#)  
*Memory page size.*

## Classes

- struct `powerpc_stat64_t`
- struct `powerpc_stat_t`
- struct `powerpc_timespec_t`
- struct `powerpc_tms_t`

### 5.11.1 Detailed Description

Definition at line 74 of file `powerpc_linux_os.hh`.

### 5.11.2 Constructor & Destructor Documentation

#### 5.11.2.1 `full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::PowerPCLinuxOS (const char * name, Object * parent = 0)`

Constructor.

Definition at line 271 of file `powerpc_linux_os.hh`.

#### 5.11.2.2 `full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::~~PowerPCLinuxOS ()` [virtual]

Destructor.

Definition at line 288 of file `powerpc_linux_os.hh`.

### 5.11.3 Member Function Documentation

#### 5.11.3.1 `void full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::OnDisconnect ()` [virtual]

Method to execute when `PowerPCLinuxOS` is disconnected from its client.

Definition at line 293 of file `powerpc_linux_os.hh`.

#### 5.11.3.2 `bool full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::ClientIndependentSetup ()` [virtual]

Client independent setup.

Definition at line 298 of file `powerpc_linux_os.hh`.

References `linux_loader`, `linux_os`, and `memory_page_size`.

#### 5.11.3.3 `bool full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::ClientDependentSetup ()` [virtual]

Client dependent setup.

Definition at line 400 of file `powerpc_linux_os.hh`.

References `exp`, and `linux_loader`.

#### 5.11.3.4 `void full_system::plugins::os::linux_os::powerpc::PowerPCLinuxOS::ExecuteSystemCall(int id) [virtual]`

Request to [LinuxOS](#) to execute the current system call.

Definition at line 426 of file `powerpc_linux_os.hh`.

References `linux_os`.

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

- `linux/powerpc/powerpc_linux_os.hh`

## 5.12 full\_system::plugins::os::linux\_os::powerpc::PowerPCSyscall Class Reference

Collaboration diagram for full\_system::plugins::os::linux\_os::powerpc::PowerPCSyscall:



### Public Member Functions

- [PowerPCSyscall](#) (const char \*name, Object \*parent=0)  
*Class constructor.*
- [~PowerPCSSyscall](#) ()  
*Class destructor.*

### Public Attributes

- [ServiceImport< ARMLinuxOSInterface, CPUARMLinuxOSInterface >](#) `ppclinuxos_imp`  
*Service port to PPCLinuxOS.*
- [ServiceExport< ARMLinuxOSInterface, CPUARMLinuxOSInterface >](#) `ppclinuxos_exp`  
*Service port from PPCLinuxOS.*
- [LinuxOS< address\\_t, reg\\_t >](#) \* `linux_os`  
*LinuxOS.*
- [PowerPCLinuxOS](#) \* `ppc_linux_os`  
*PowerPC LinuxOS.*

#### 5.12.1 Detailed Description

Definition at line 68 of file `powerpc-syscall.hh`.

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

- `linux/powerpc/powerpc-syscall.hh`

## 5.13 `full_system::plugins::os::linux_os::powerpc::StandalonePowerPCLoader` Class Reference

### Public Member Functions

- [StandalonePowerPCLoader](#) (const char \*name, Object \*parent=0)  
*Class constructor.*

### Public Attributes

- `ServiceImport< MemoryInterface< physical_address_t > >` [memory\\_import](#)  
*Service port to memory.*

#### 5.13.1 Detailed Description

Definition at line 60 of file `standalone_powerpc_loader.hh`.

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

- `linux/powerpc/standalone_powerpc_loader.hh`

## 5.14 full\_system::plugins::os::linux\_os::SyscallJournal Class Reference

### Public Member Functions

- void [NotifyOpen](#) (int fd, const char \*pathname, int flags, mode\_t mode)  
*Notify an "open" syscall, adding a new entry in the list of opened files.*
- void [NotifyClose](#) (int fd)  
*Notify an "close" syscall, removing an entry from the list of opened files.*
- void [NotifyRead](#) (int fd, void \*buf, size\_t count)  
*Notify an "read" syscall.*
- void [NotifyWrite](#) (int fd, void \*buf, size\_t count)  
*Notify an "write" syscall.*
- void [load\\_checkpoint](#) (std::istream &is, int &fd\_stdin, int &stdout, int &stderr)  
*Load from a checkpoint file the part related to opened files.*
- void [save\\_checkpoint](#) (std::ostream &os)  
*Save in the checkpoint file the part related to opened files.*

### 5.14.1 Detailed Description

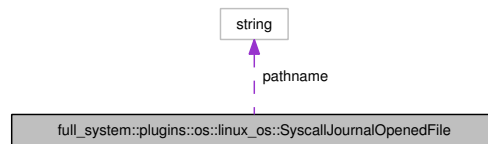
Definition at line 60 of file syscall\_journal.hh.

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

- linux/syscall\_journal.hh
- linux/syscall\_journal.cpp

## 5.15 full\_system::plugins::os::linux\_os::SyscallJournalOpenedFile Class Reference

Collaboration diagram for full\_system::plugins::os::linux\_os::SyscallJournalOpenedFile:



### Public Member Functions

- [SyscallJournalOpenedFile \(\)](#)  
*Empty class constructor, needed to build list of objects.*
- [SyscallJournalOpenedFile \(const char \\*\\_pathname, int \\_flags, mode\\_t \\_mode\)](#)

### Friends

- `std::ostream & operator<< (std::ostream &os, const SyscallJournalOpenedFile &f)`  
*Class pretty printer.*

#### 5.15.1 Detailed Description

Definition at line 49 of file `syscall_journal.hh`.

#### 5.15.2 Constructor & Destructor Documentation

##### 5.15.2.1 full\_system::plugins::os::linux\_os::SyscallJournalOpenedFile::SyscallJournalOpenedFile (const char \* \_pathname, int \_flags, mode\_t \_mode)

Class constructor. Creates a new entry for a opened file

Definition at line 62 of file `syscall_journal.cpp`.

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

- `linux/syscall_journal.hh`
- `linux/syscall_journal.cpp`

## 5.16 full\_system::plugins::os::linux\_os::SyscallMemoryDispatcher<nCPU> Class Template Reference

### Public Member Functions

- [SyscallMemoryDispatcher](#) (const char \*name, Object \*parent=0)  
*Create a new SyscallMemoryDispatcher service.*
- [~SyscallMemoryDispatcher](#) ()  
*Class destructor.*
- virtual void [Reset](#) ()  
*Service interface : reset.*
- bool [ReadMemory](#) (address\_t addr, void \*buffer, uint32\_t size)  
*Direct memory read.*
- bool [WriteMemory](#) (address\_t addr, const void \*buffer, uint32\_t size)  
*Direct memory write.*
- bool [ClientIndependentSetup](#) ()  
*Service interface : initialization.*
- bool [ClientDependentSetup](#) ()  
*Service interface : initialization.*

### Public Attributes

- [ServiceExport<MemoryInterface<address\\_t>>](#) [exp](#)  
*Service port from the local CPU.*
- [ServiceImport<MemoryInterface<address\\_t>>](#) [imp](#)  
*Service port to the local cache.*
- [ServiceImport<MemoryInterface<address\\_t>>](#) \* [cache\\_imp](#) [nCPU-1]  
*Service ports to the distant caches.*

#### 5.16.1 Detailed Description

```
template<int nCPU> class full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU
>
```

Definition at line 64 of file syscall\_memory\_dispatcher.hh.

## 5.16.2 Member Function Documentation

### 5.16.2.1 `template<int nCPU> bool full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::ReadMemory (address_t addr, void * buffer, uint32_t size) [inline]`

Direct memory read.

Memory reads are forwarded to the memory port

Definition at line 114 of file `syscall_memory_dispatcher.hh`.

References `full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::cache_imp`, and `full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::imp`.

### 5.16.2.2 `template<int nCPU> bool full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::WriteMemory (address_t addr, const void * buffer, uint32_t size) [inline]`

Direct memory write.

Memory write are forwarded to the connected cache, and some flushes are sent to every non-connected cache.

Definition at line 131 of file `syscall_memory_dispatcher.hh`.

References `full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::cache_imp`, and `full_system::plugins::os::linux_os::SyscallMemoryDispatcher< nCPU >::imp`.

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

- `linux/syscall_memory_dispatcher.hh`

## 5.17 full\_system::plugins::os::linux\_os::UnimplementedSystemCall Class Reference

Exception : Unimplemented Syscall.

```
#include <linux_os.hh>
```

### Public Member Functions

- [UnimplementedSystemCall](#) (const char \*function, const char \*file, int line, unsigned int number)
- [~UnimplementedSystemCall](#) () throw ()  
*Class destructor.*
- const char \* [what](#) () const throw ()  
*Exception helper string.*

### 5.17.1 Detailed Description

Exception : Unimplemented Syscall.

Definition at line 95 of file linux\_os.hh.

### 5.17.2 Constructor & Destructor Documentation

#### 5.17.2.1 UnimplementedSystemCall::UnimplementedSystemCall (const char \* *function*, const char \* *file*, int *line*, unsigned int *number*)

brief Class constructor

Definition at line 45 of file linux\_os.cpp.

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

- linux/linux\_os.hh
- linux/linux\_os.cpp

# Index

- ~ARMLinuxOS
  - full\_system::plugins::os::linux\_  
os::arm::ARMLinuxOS, 17
- ~LinuxOS
  - full\_system::plugins::os::linux\_os::LinuxOS, 25
- ~PowerPCLinuxOS
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 34
- ClientDependentSetup
  - full\_system::plugins::os::linux\_os::LinuxOS, 25
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 34
- ClientIndependentSetup
  - full\_system::plugins::os::linux\_os::LinuxOS, 25
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 34
- ExecuteSystemCall
  - full\_system::plugins::os::linux\_os::LinuxOS, 26
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 35
  - full\_system::plugins::os::OSInterface, 29
- full\_system, 7
- full\_system::plugins, 8
- full\_system::plugins::os, 9
- full\_system::plugins::os::CPUOSInterface, 22
- full\_system::plugins::os::linux\_os, 10
- full\_system::plugins::os::linux\_os::arm, 11
- full\_system::plugins::os::linux\_  
os::arm::ARMLinux, 13
- full\_system::plugins::os::linux\_  
os::arm::ARMLinuxOS, 15
  - ~ARMLinuxOS, 17
  - OnDisconnect, 17
- full\_system::plugins::os::linux\_  
os::arm::ARMLinuxOSInterface, 18
- full\_system::plugins::os::linux\_  
os::arm::CPUARMLinuxOSInterface, 19
- full\_system::plugins::os::linux\_  
os::CPULinuxOSInterface, 20
- full\_system::plugins::os::linux\_os::LinuxOS, 24
  - ~LinuxOS, 25
  - ClientDependentSetup, 25
  - ClientIndependentSetup, 25
  - ExecuteSystemCall, 26
  - LinuxOS, 25
  - OnDisconnect, 25
  - SetSyscallId, 26
- full\_system::plugins::os::linux\_  
os::LinuxOSInterface, 27
  - SetSyscallId, 27
- full\_system::plugins::os::linux\_os::powerpc, 12
- full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinux, 30
- full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 32
  - ~PowerPCLinuxOS, 34
  - ClientDependentSetup, 34
  - ClientIndependentSetup, 34
  - ExecuteSystemCall, 35
  - OnDisconnect, 34
  - PowerPCLinuxOS, 34
- full\_system::plugins::os::linux\_  
os::powerpc::PowerPCSyscall, 36
- full\_system::plugins::os::linux\_  
os::powerpc::StandalonePowerPCLoader, 37
- full\_system::plugins::os::linux\_os::SyscallJournal, 38
- full\_system::plugins::os::linux\_  
os::SyscallJournalOpenedFile, 39
  - SyscallJournalOpenedFile, 39
- full\_system::plugins::os::linux\_  
os::SyscallMemoryDispatcher, 40
  - ReadMemory, 41
  - WriteMemory, 41
- full\_system::plugins::os::linux\_  
os::UnimplementedSystemCall, 42
  - UnimplementedSystemCall, 42
- full\_system::plugins::os::OSInterface, 29
  - ExecuteSystemCall, 29
- LinuxOS

- full\_system::plugins::os::linux\_os::LinuxOS,  
25
- OnDisconnect
  - full\_system::plugins::os::linux\_  
os::arm::ARMLinuxOS, 17
  - full\_system::plugins::os::linux\_os::LinuxOS,  
25
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 34
- PowerPCLinuxOS
  - full\_system::plugins::os::linux\_  
os::powerpc::PowerPCLinuxOS, 34
- ReadMemory
  - full\_system::plugins::os::linux\_  
os::SyscallMemoryDispatcher, 41
- SetSyscallId
  - full\_system::plugins::os::linux\_os::LinuxOS,  
26
  - full\_system::plugins::os::linux\_  
os::LinuxOSInterface, 27
- SyscallJournalOpenedFile
  - full\_system::plugins::os::linux\_  
os::SyscallJournalOpenedFile, 39
- UnimplementedSystemCall
  - full\_system::plugins::os::linux\_  
os::UnimplementedSystemCall, 42
- WriteMemory
  - full\_system::plugins::os::linux\_  
os::SyscallMemoryDispatcher, 41