

Xameleon

Draft v6

- Этот документ не является общественным достоянием и принадлежит его разработчикам.
- Документ имеет статус черновика и может быть изменён в любое время без предварительного уведомления.
- Этот документ основан на L4 Specification Revision X2 (<http://www.l4ka.org/pistachio>).

Revision 1

11.24.2006 Initial document release

Revision 2

11.26.2006 Added new protocols: FileSystem, BlockDevice, CharacterDevice

11.26.2006 Added new data type ThreadID

Revision 3

11.27.2006 Rearranged supervisor's system calls.

11.28.2006 Added protocol layering model

Revision 4

11.30.2006 Modified ReleaseSegment syscall to return segment's references count

11.30.2006 Added the handle argument to the CreateProcess syscall.

12.01.2006 Added Appendix_I with detail descriptions of a some data types

12.02.2006 Extended semantics of the fork() syscall by a new flag.

12.02.2006 Fixed semantics of the wait() syscall

12.09.2006 Introduced a context argument to the device driver syscalls

01.22.2007 Added ChangeFileTimes syscall

08.24.2007 Change semantics of StartDevice syscall

Revision 5

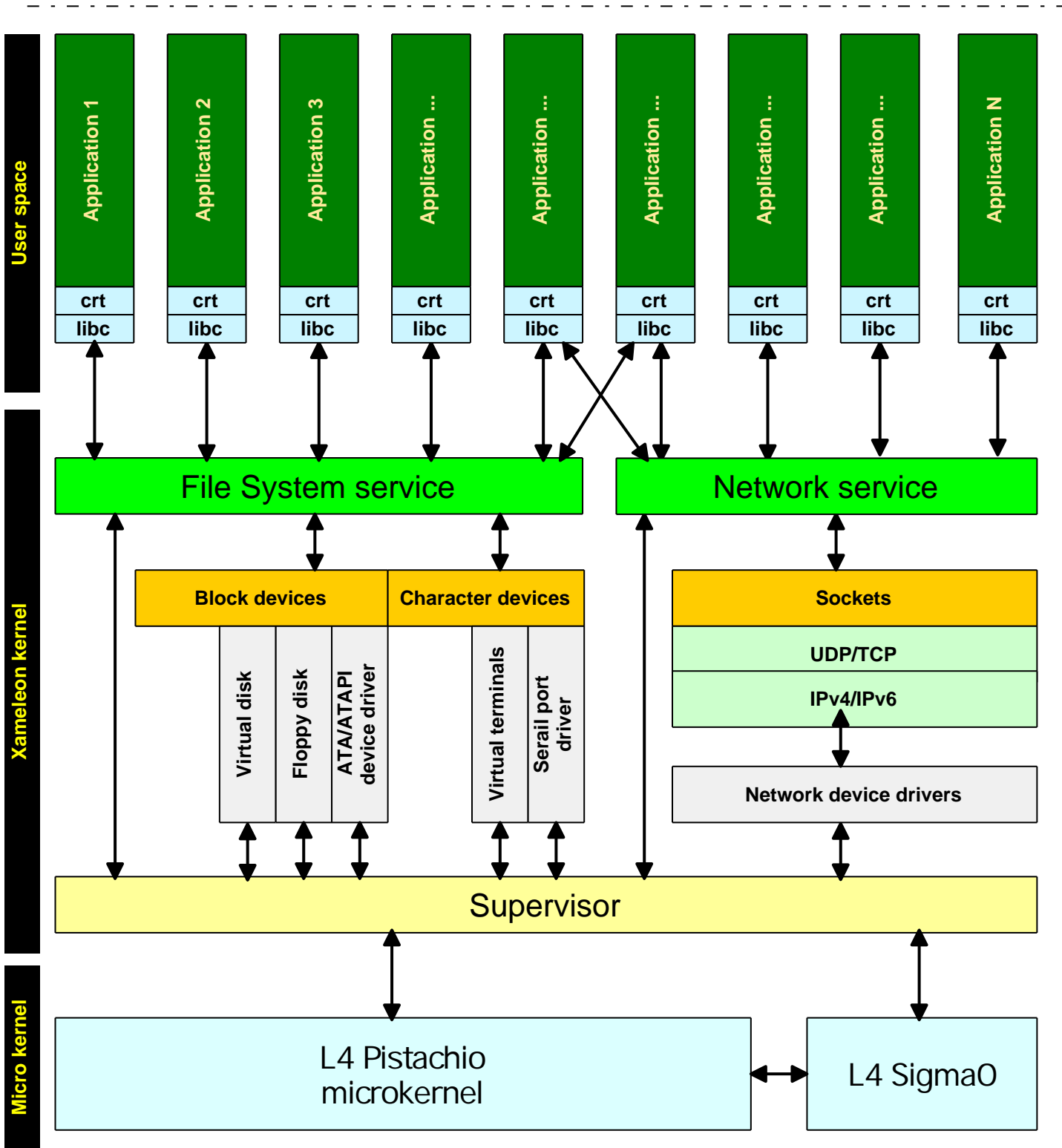
10.11.2007 Add FileSync syscall description into filesystem section

10.11.2007 Change semantics of StartDevice syscall

Revision 6

05.13.2011

Total 16 entries



L4x2.

Type	Description
Word	
String	
Compound	
MapItem	'Map as part of message' .
ThreadID	(,).
Unspecified	
MsgTag	,
DRR	/ .
BIH	.
DeviceName	16 , .

Supervisor

Method	Description
AllocateSegment	AllocateSegment
ReferencingSegment	ReferencingSegment (/)
QueryInterface	QueryInterface
ExitProcess	ExitProcess
CreateThread	CreateThread
ExitThread	ExitThread
ForkProcess	ForkProcess , BSS
ExecProcess	ExecProcess POSIX.
StartService	StartDevice (/)
StopService	
DeviceEnumerator	DeviceEnumearator /dev.

Method	Description
GetDeviceHandle	GetDeviceHandle (/)
	UUID 16
GetProcessID	GetProcessID (pid_t) (L4_ThreadID_t).
GetThreadID	GetThreadID (pid_t) process_id
ChangeHeapSize	ChangeHeapSize " "
SetSignal	SetSignal
Signal	Signal thread_id.
ProcessWait	ProcessWait
SetTimer	SetTimer
SetNotificationHandler	SetNotificationHandler
SystemHalt	
LeaveSignal	LeaveSignal

AllocateSegment

AllocateSegment

Message register	Type	Label
MR0	MsgTag	1

Message registers	Type	Name	Description
MR1	Word	address	Virtual address of requested segment
MR2	Word	size	Size of requested segment

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	Word	handle	Handle to allocated segment
MR3	Word	pages_count	Pages count to be mapped. Note that we using compound pages. That means that pages is an array of different size pages, which covers requested memory region.
MR4...n	MapItem	pages	Pages array. See description of the MapItem and mapping procedure in L4 X2 specification.

ReferencingSegment

ReferencingSegment (/)

Message register	Type	Label
MR0	MsgTag	2

Message registers	Type	Name	Description
MR1	Word	handle	Handle to releasing segment
MR2	Word	command	This field describes what to do with a segment

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	Word	ref_count	References count to this segment

QueryInterface

QueryInterface

Message register	Type	Label
MR0	MsgTag	3

ExitProcess

ExitProcess

Message register	Type	Label
MR0	MsgTag	4

Message registers	Type	Name	Description
MR1	Word	status	Program termination status

CreateThread

CreateThread

Message register	Type	Label
MR0	MsgTag	5

Message registers	Type	Name	Description
MR1	Word	ip	Entry point to the thread
MR2	Word	stack_bottom	Bottom address of the stack
MR3	Word	stack_size	Size of stack
MR4	Word	thread_options	Bitfield that describes thread attributes

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	ThreadID	thread_id	Thread identifier of a newly created thread or nilthread if process creation fails

ExitThread

ExitThread

Message register	Type	Label
MR0	MsgTag	6

Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of destroying thread

Message registers	Type	Name	Description
MR1	Word	status	Operation status

ForkProcess

ForkProcess

, BSS

Message register	Type	Label
MR0	MsgTag	7

Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of the forking thread

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	ThreadID	thread_id	Thread identifier of forked process or nilthread if fork is failed
MR3	Word	process_id	Process identifier (pid_t) of newly created process

ExecProcess

ExecProcess

POSIX.

Message register	Type	Label
MR0	MsgTag	8

Message registers	Type	Name	Description
MR 1	Word	thread_id	Thread identifier that describes an overlapping thread
MR 2	Word	handle	Handle to a compound memory segment that covers all sections of the executable image.
MR 3	Word	entry_point	Entry point of the executable image.
MR 4	Word	text_start	Virtual address of text segment
MR 5	Word	text_size	Size of text segment
MR 6	Word	data_start	Virtual address of data segment
MR 7	Word	data_size	Size of data segment
MR 8	Word	bss_start	Virtual address of BSS segment
MR 9	Word	bss_size	Size of BSS segment
MR10,11	Compound	arguments	String that represent program calling arguments
MR12,13	Compound	environment	String that represent process environment

Message registers	Type	Name	Description
MR1	Word	status	Operation status. Note that this value is used by filesystem driver

StartService

StartDevice

(/)

Message register	Type	Label
MR0	MsgTag	9

Message registers	Type	Name	Description
MR1	Word	space	0 - run in driver recommended address space, 1 - run in Supervisor's address space, 2 - run in dedicated address space
MR2,3	String	device_record	a string that represents a device registration record. See description of Device Registration Record at end of this document
MR4,5	String	driver_argument_string	a string that represent paramters to the driver executable.

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	ThreadID	ThreadID	ThreadID of a main therad of the driver

StopService

Message register	Type	Label
MR0	MsgTag	10

Message registers	Type	Name	Description
MR1	Unspecified	...	Format is not designed yet

Message registers	Type	Name	Description
MR1	Word	status	Operation status

DeviceEnumerator

DeviceEnumearator

/dev.

Message register	Type	Label
MR0	MsgTag	11

Message registers	Type	Name	Description
MR1	Word	device_no	Device number. Increment it until successful status

Message registers	Type	Name	Description
MR1	Word	status	Operation status code
MR2	Word	total	Total device count
MR3,4	String	dev_info	Public device record

GetProcessID

GetProcessID

(L4_ThreadId_t).

(pid_t)

Message register	Type	Label
MR0	MsgTag	13

Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of process that ID is requested

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	Word	process_id	Process id for thread_id argument. See POSIX pid_t
MR3	Word	parent_id	Parent process id for thread argument. See POSIX ppid_t

GetThreadID

GetThreadID
, (pid_t) , process_id

Message register	Type	Label
MR0	MsgTag	14

Message registers	Type	Name	Description
MR1	Word	process_id	Process identifier

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	ThreadID	thread_id	Main thread identifier of requested process

ChangeHeapSize

ChangeHeapSize

Message register	Type	Label
MR0	MsgTag	15

Message registers	Type	Name	Description
MR1	Word	increment	Signed value that indicates heap increment number bytes

Message registers	Type	Name	Description
MR1	Word	status	Operation status code.
MR2	Word	bottom	Heap start virtual address
MR3	Word	old_top	Heap top virtual address
MR4	Word	new_top	Old heap top value

SetSignal

SetSignal

Message register	Type	Label
MR0	MsgTag	16

Message registers	Type	Name	Description
MR1	Word	signal_number	Signal number
MR2	String	handler	String that represents the POSIX sigaction structure

Message registers	Type	Name	Description
MR1	Word	status	Status of setting signal handler operation
MR2	String	old_handler	String that represents the POSIX sigaction structure that describes previous settings

Signal

Signal

..

thread_id.

Message register	Type	Label
MR0	MsgTag	17

Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of destination thread
MR2	Word	signal_number	Signal number

Message registers	Type	Name	Description
MR1	Word	status	Signal delivery status
MR2	Word	reserver	Reserved value

ProcessWait

ProcessWait

Message register	Type	Label
MR0	MsgTag	18

Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of waitng process or anythread for waiting any child thread.
MR2	Word	options	Waiting options. See POSIX wait

Message registers	Type	Name	Description
MR1	Word	status	Status of wait operation
MR2	Word	exit_status	Exit status of terminated process or signal number
MR3	ThreadID	thread_id	Thread identifier of terminated process
MR4	Word	process_id	Process identifier of terminated process

SetTimer

SetTimer

Message register	Type	Label
MR0	MsgTag	19

Message registers	Type	Name	Description
MR1	Word	timer_id	Timer identifier
MR2,3	String	itimerval	POSIX itimerval structure. Empty string resets timer timer_id.

Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2,3	String	itimerval	Previously timer values as POSIX ittimerrval structure

SetNotificationHandler

SetNotificationHandler

Message register	Type	Label
MR0	MsgTag	20

Message registers	Type	Name	Description
MR1	Word	EventID	Value reserved for future use (must be 0)
MR2	ThreadID	ThreadID	An event handling thread identifier

Message registers	Type	Name	Description
MR1	Word	Status	Status of operation

SystemHalt

Message register	Type	Label
MR0	MsgTag	21

LeaveSignal

LeaveSignal

Message register	Type	Label
MRO	MsgTag	22

FileSystem

Method	Description
Mount	
Unmount	
GetCurrentDirectory	
ChangeDir	
OpenDirectory	ReadDirectory.
ReadDirectory	OpenDirectory.
CloseDirectory	OpenDirectory.
GetFileStatus	
ChangeRootDirectory	
OpenFile	
CloseFile	OpenFile
ReadFile	
WriteFile	
RemoveFile	
RemoveDirectory	
CreateDirectory	
CreateHardLink	
CreateSybolicLink	
RenameFile	
SeekFile	
GetSuperblockStatus	
ChangeOwner	
ChangeMode	
FlushCaches	
DupDescriptor	

Method	Description
CopyDescriptor	
CreatePipe	
ForkProcess	, .bss
Exec	
MakeNode	INODE
IOCTL	/
Exit	
ControlFile	, POSIX fcntl()
ChaneFileTimes	, POSIX utimes()
CreateSession	, POSIX setsid()
FileSync	C Handle,

Mount

Message register	Type	Label
MR0	MsgTag	64

Message registers	Type	Name	Description
MR1	Word	Flags	Mount flags
MR2,3	Compound	Device	String that represents the device name to be mounted
MR4,5	Compound	Point	String that represent the mount point within a filesystem tree
MR6,7	Compound	Type	Filesystem type

Message registers	Type	Name	Description
MR1	Word	Status	Mount operation status

Unmount

Message register	Type	Label
MR0	MsgTag	65

Message registers	Type	Name	Description
MR1,2	String	Point	The mount point

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

GetCurrentDirectory

Message register	Type	Label
MR0	MsgTag	66

Message registers	Type	Name	Description
MR1	Word	Size	Receiver's buffer size.

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2,3	String	Path	String that represent current process' path.

ChangeDir

Message register	Type	Label
MR0	MsgTag	67

Message registers	Type	Name	Description
MR1,2	String	Path	New process path

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

OpenDirectoryReadDirectory.
.

Message register	Type	Label
MR0	MsgTag	68

Message registers	Type	Name	Description
MR1,2	String	Path	Path to directory for open

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	Handle	Directory handle.

ReadDirectory

OpenDirectory.

Message register	Type	Label
MR0	MsgTag	69

Message registers	Type	Name	Description
MR1	Word	Handle	Handle to directory

Message registers	Type	Name	Description
MR1	Word	Status	Status of operation
MR2,3	String	Record	Directory record. Subsequent calls return next directory entry

CloseDirectory

, OpenDirectory.

Message register	Type	Label
MR0	MsgTag	70

Message registers	Type	Name	Description
MR1	Word	Handle	Handle to directory

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

GetFileStatus

Message register	Type	Label
MR0	MsgTag	71

Message registers	Type	Name	Description
MR1	Word	Parameters	0-follow symlinks, 1-allow symlink, any other value is open file descriptor
MR2,3	String	FileName	Name of requested file

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2,3	String	Attributes	File attributes

ChangeRootDirectory

Message register	Type	Label
MR0	MsgTag	72

Message registers	Type	Name	Description
MR1,2	String	Path	Path to new root direcorry of caller process

Message registers	Type	Name	Description
MR1	Word	Status	Opeartion status

OpenFile

Message register	Type	Label
MR0	MsgTag	73

Message registers	Type	Name	Description
MR1	Word	Flags	File open flags. Flags comatible with POSIX open() flags
MR2	Word	Mode	File open mode. Mode compatible with POSIX file modes.
MR3,4	String	Filename	The fname of file to open/create

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	Handle	Positive value is file handle, negative value is error code

CloseFile

OpenFile

Message register	Type	Label
MR0	MsgTag	74

Message registers	Type	Name	Description
MR1	Word	Handle	File handle to close

Message registers	Type	Name	Description
MR1	Word	Status	Operaion status

ReadFile

Message register	Type	Label
MR0	MsgTag	75

Message registers	Type	Name	Description
MR1	Word	Handle	File handle to read
MR2	Word	BytesCount	Count bytes to read

Message registers	Type	Name	Description
MR1	Word	Status	Read status. Positive value is bytes read count, negative is error code
MR2,3	String	Data	Data that have been read from file

WriteFile

Message register	Type	Label
MR0	MsgTag	76

Message registers	Type	Name	Description
MR1	Word	Handle	File handle to write
MR2,3	String	Data	Data for writing to file

Message registers	Type	Name	Description
MR1	Word	Status	Write status. Positive value is bytes write count, negative is error code

RemoveFile

Message register	Type	Label
MR0	MsgTag	77

Message registers	Type	Name	Description
MR1,2	String	Filename	The name of file to delete

Message registers	Type	Name	Description
MR1	Word	Status	Deletion status

RemoveDirectory

Message register	Type	Label
MR0	MsgTag	78

Message registers	Type	Name	Description
MR1,2	String	DirName	The directory name to delete

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateDirectory

Message register	Type	Label
MR0	MsgTag	79

Message registers	Type	Name	Description
MR1	Word	Mode	Creation mode
MR2,3	String	DirName	The new directory name

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateHardLink

Message register	Type	Label
MR0	MsgTag	80

Message registers	Type	Name	Description
MR1,2	Compound	Source	Source filename
MR3,4	Compound	Target	Target filename

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateSybolicLink

Message register	Type	Label
MR0	MsgTag	81

Message registers	Type	Name	Description
MR1,2	Compound	Source	Source filename
MR3,4	Compound	Target	Target filename

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

RenameFile

Message register	Type	Label
MR0	MsgTag	82

Message registers	Type	Name	Description
MR1,2	Compound	OldName	File name to change
MR3,4	MapItem	NewName	New name of the file

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

SeekFile

Message register	Type	Label
MR0	MsgTag	83

Message registers	Type	Name	Description
MR1	Word	Handle	File handle
MR2	Word	Offset	Seek offset (signed)
MR3	Word	Whence	Seek direction

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	Position	New position within file

GetSuperblockStatus

Message register	Type	Label
MR0	MsgTag	84

Message registers	Type	Name	Description
MR1,2	String	Path	Path

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2,3	String	Record	Superblock information record

ChangeOwner

Message register	Type	Label
MR0	MsgTag	85

Message registers	Type	Name	Description
MR1	Word	Owner	New file owner
MR2	Word	Group	New file group
MR3,4	String	Filename	Filename

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

ChangeMode

Message register	Type	Label
MR0	MsgTag	86

Message registers	Type	Name	Description
MR1	Word	Mode	File mode
MR2,3	String	Filename	Filename

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

FlushCaches

Message register	Type	Label
MR0	MsgTag	87

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

DupDescriptor

Message register	Type	Label
MR0	MsgTag	88

Message registers	Type	Name	Description
MR1	Word	Handle	File handle for duplication

Message registers	Type	Name	Description
MR1	Word	Status	Operation status.
MR2	Word	NewHandle	New descriptor that points to the same file as the input handle

CopyDescriptor

Message register	Type	Label
MR0	MsgTag	89

Message registers	Type	Name	Description
MR1	Word	Target	Target file descriptor
MR2	Word	Source	Source file descriptor

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreatePipe

Message register	Type	Label
MR0	MsgTag	90

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	Pipeln	Pipe input endpoint
MR3	Word	PipeOut	Pipe output endpoint

ForkProcess

, .bss

Message register	Type	Label
MR0	MsgTag	91

Message registers	Type	Name	Description
MR1	Word	Flags	These flags describes a fork() extension modes. Provide zero value to follow the POSIX behaviour

Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	ProcessID	Process identifier of newly created child process
MR3	ThreadID	ThreadID	Thread identifier of new created process

Exec

Message register	Type	Label
MR0	MsgTag	92

Message registers	Type	Name	Description
MR1,2	Compound	ProcessPath	Path to executable file
MR3,4	Compound	Arguments	Programm command line arguments

Message registers	Type	Name	Description
MR1	Word	Status	Exec status. Status is never returned on success execution.

MakeNode

INODE

Message register	Type	Label
MR0	MsgTag	96

Message registers	Type	Name	Description
MR0	Word	Mode	File mode
MR1	ThreadID	Major	Major device number
MR2	Word	Minor	Minor devcie number
MR3	Word	Size	Object size
MR4,5	String	Name	Path and name new node

Message registers	Type	Name	Description
MR0	Word	Status	Operation status

IOCTL

Message register	Type	Label
MR0	MsgTag	97

Message registers	Type	Name	Description
MR1	Word	Handle	File descriptor
MR2	Word	Command	Control command ID
MR3...	Unspecified	...	Format of these registers depends on Command type

Message registers	Type	Name	Description
MR1	Word	Status	Control command execution status
MR2...	Unspecified	Format of these registers depends on Command type

Exit

Message register	Type	Label
MR0	MsgTag	98

Message registers	Type	Name	Description
MR1	Word	Status	Status that must be sent to waiting processes

ControlFile

, POSIX fcntl()

Message register	Type	Label
MR0	MsgTag	99

Message registers	Type	Name	Description
MR1	Word	Handle	File handle to control
MR2	Word	Mode	Control mode
MR3...	Unspecified	...	Format of these registers depends on command mode

Message registers	Type	Name	Description
MR1	Word	Status	Control operation status
MR2...	Unspecified	...	Format of these registers depends on Command mode

ChaneFileTimes

POSIX utimes()

Message register	Type	Label
MR0	MsgTag	100

Message registers	Type	Name	Description
MR1,2	String	FileName	The name of file which times will be set by following structure
MR3,4	String	FileTimes	This string conveys an utimbuf_t structure

Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateSession

, POSIX setsid()

Message register	Type	Label
MR0	MsgTag	101

Message registers	Type	Name	Description
MR1	Word	Status	Status of operation
MR2	Word	ProcessID	pid_t of new session

FileSync

C

Handle,

Message register	Type	Label
MR0	MsgTag	102

Message registers	Type	Name	Description
MR1	Word	Handle	File handle to flush it's caches

Message registers	Type	Name	Description
MR1	Word	Status	Status of operation

BlockDevice

Method	Description
BlockDevice_Open	
BlockDevice_Close	
BlockDevice_Read	
BlockDevice_Write	
BlockDevice_Flush	
BlockDevice_IOCTL	DMA.

BlockDevice_Open

Message register	Type	Label
MR0	MsgTag	144

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Minor	Minor device number
MR3	ThreadID	Listenr	An ID of the response listener thread

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status
MR3	Word	ID	Filesystem ID. This value conforms to MBR .
MR4	Word	SectorSize	Device's sector size
MR5	Word	FirstSector	First sector number of partition/session relative to the first sector of a physical media

BlockDevice_Close

Message register	Type	Label
MR0	MsgTag	145

BlockDevice_Read

Message register	Type	Label
MR0	MsgTag	146

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Minor	Minor device number
MR3	Word	BlockNumber	Identifies requested block number
MR4	Word	BlockSize	Identifies block size
MR5	Word	CachingType	Determines caching algorithm for requested block

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status
MR3,4	String	Data	Requested data

BlockDevice_Write

Message register	Type	Label
MR0	MsgTag	147

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Minor	Minor device number
MR3	Word	BlockNumber	Identifies writing block number
MR4	Word	BlockSize	Identifies block size
MR5	Word	CachingType	Describes data type on top of that a caching scheme is selecting by the device driver
MR6,7	String	Data	Writing data

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status

BlockDevice_Flush

Message register	Type	Label
MR0	MsgTag	148

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Minor	Minor device number
MR3	Word	reserved	Do we need flush caches of special type? Like inodes blocks, file data bocks, inodes bitmap blocks, free block bitmap blocks, system area? If so, then this argument would be nice for cache type description.

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status

BlockDevice_IOCTL

DMA.

Message register	Type	Label
MR0	MsgTag	149

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Minor	Minor device number
MR3	Word	Command	Control command ID
MR4...	Unspecified	...	Format of these registers depends on Command type

Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Control command execution status
MR3...	Unspecified	...	Format of these registers depends on Command type

CharacterDevice

()

Method	Description
StreamDevice_Open	
StreamDevice_Close	
StreamDevice_Read	
StreamDevice_Write	
StreamDevice_IOCTL	
StreamDevice_Log	

StreamDevice_Open

Message register	Type	Label
MR0	MsgTag	130

Message registers	Type	Name	Description
MR1	Word	Context	Requestor's context, this value must be returned within response and it unique identifies response
MR2	Word	Minor	Minor device number
MR3	ThreadID	Listener	An ID of the response listener thread
MR4	Word	Flags	Flags for method open()

Message registers	Type	Name	Description
MR1	Word	Context	This argument must provide the same value as the context argument of the input message
MR2	Word	Status	Operation status

StreamDevice_Close

Message register	Type	Label
MR0	MsgTag	131

Message registers	Type	Name	Description
MR1	Word	Context	Requestor's context, this value must be returned within response and it unique identifies response
MR2	Word	Minor	Minor device number

Message registers	Type	Name	Description
MR1	Word	Context	This argument must provide the same value as the context argument of the input message
MR2	Word	Status	Operation status

StreamDevice_Read

Message register	Type	Label
MR0	MsgTag	132

Message registers	Type	Name	Description
MR1	Word	Context	Requestor's context, this value must be returned within response and it unique identifies response
MR2	Word	Minor	Minor device number
MR3	Word	Size	Maximum bytes to read.

Message registers	Type	Name	Description
MR1	Word	Context	This argument must provide the same value as the context argument of the input message
MR2	Word	Status	Count bytes have been read from device. Negative value signalling error and error code
MR3,4	String	Data	Data that have been read from device. This data will be copied into receiver's buffer

StreamDevice_Write

Message register	Type	Label
MR0	MsgTag	133

Message registers	Type	Name	Description
MR1	Word	Context	Requestor's context, this value must be returned within response and it unique identifies response
MR2	Word	Minor	Minor device number
MR3,4	String	Data	String that represents data for writing to device

Message registers	Type	Name	Description
MR1	Word	Context	This argument must provide the same value as the context argument of the input message
MR2	Word	Status	Count bytes that have been witten to device. Negative value signaling error and error code

StreamDevice_IOCTL

Message register	Type	Label
MR0	MsgTag	134

Message registers	Type	Name	Description
MR1	Word	Context	Requestor's context, this value must be returned within response and it unique identifies response
MR2	Word	Minor	Minor device number
MR3	Word	Command	Control command ID
MR4...	Unspecified	...	Format of these registers depends on Command type

Message registers	Type	Name	Description
MR1	Word	Context	This argument must provide the same value as the context argument of the input message
MR2	Word	Status	Control command execution status
MR3...	Unspecified	...	Format of these registers depends on Command type

StreamDevice_Log

Message register	Type	Label
MR0	MsgTag	135

Message registers	Type	Name	Description
MR1	Word	Context	
MR2	Word	Minor	Minor device number
MR3,4	String	Data	String that represent information. Usually a text representation of a log message

Message registers	Type	Name	Description
MR1	Word	Context	
MR2	Word	Status	Command status

NetworkService

Method	Description
CreateSocket	
CloseSocket	
BindSocket	BINDED
ListenSocket	LISTEN ()
Accept	
Connect	
Recv	
Send	
Shutdown	
ControlRoute	
IfConfig	
//ConnectResponse	
AcceptResponse	
//RecvResponse	
//SendResponse	
GetSocketName	

CreateSocket

Message register	Type	Label
MR0	MsgTag	160

Message registers	Type	Name	Description
MR1	Word	nDomain	Protocol definition: IPv4, IPv6, etc
MR2	Word	nTransportType	Transport type: stream, datagram, etc
MR3	Word	nProtocol	Protocol definition

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status
MR2	Word	nSocket	Socket index

CloseSocket

Message register	Type	Label
MR0	MsgTag	161

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket indentifier

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status

BindSocket

BINDED

Message register	Type	Label
MR0	MsgTag	162

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket indentifier
MR2	String	pLocalAddress	Local socket address identifier

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status

ListenSocket

LISTEN ()

Message register	Type	Label
MR0	MsgTag	163

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket indentifier
MR2	String	nBacklog	Maximum number incoming connections

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status

Accept

Message register	Type	Label
MR0	MsgTag	164

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket identifier
MR2	Word	nAddressSize	Requestor buffer size

Message registers	Type	Name	Description
MR1	String	RemoteAddr	Remote endpoint address

Connect

Message register	Type	Label
MR0	MsgTag	165

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket identifier
MR2	Word	nAddrSize	Size of address structure
MR3	String	RemoteAddr	Connection address

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status

Recv

Message register	Type	Label
MR0	MsgTag	166

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket identifier
MR2	Word	nFlags	Operation attributes
MR3	Word	nLength	Receiver buffer size
MR4	Word	nAddrSize	Size of address buffer

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status
MR2	Compound	Data	Received data and peer address

Send

Message register	Type	Label
MR0	MsgTag	167

Message registers	Type	Name	Description
MR1	Word	nIndex	Socket identier
MR2	Compound	Data	Sending data and optionally remote address (sendto)

Message registers	Type	Name	Description
MR1	Word	nStatus	Operation status

Shutdown

Message register	Type	Label
MR0	MsgTag	168

ControlRoute

Message register	Type	Label
MR0	MsgTag	169

IfConfig

Message register	Type	Label
MR0	MsgTag	170

//ConnectResponse

Message register	Type	Label
MR0	MsgTag	171

AcceptResponse

Message register	Type	Label
MR0	MsgTag	172

//RecvResponse

Message register	Type	Label
MR0	MsgTag	173

//SendResponse

Message register	Type	Label
MR0	MsgTag	174

GetSocketName

Message register	Type	Label
MR0	MsgTag	175

NetworkDevice

Method	Description
SendFrame	Ethernet
IfConfig	
NetStat	
SentFrameConfirm	
ReceiveFrame	
DispatchIPv4Packet	(legacy)
ParseProtocol	(legacy)
SendPacket	(legacy)
Control	

SendFrame

Ethernet

Message register	Type	Label
MR0	MsgTag	176

Message registers	Type	Name	Description
MR1	Word	nPacketID	
MR2	Word	nCardNo	()
MR3	String	Packet	

IfConfig

Message register	Type	Label
MR0	MsgTag	177

NetStat

Message register	Type	Label
MR0	MsgTag	178

SentFrameConfirm

Message register	Type	Label
MR0	MsgTag	179

ReceiveFrame

Message register	Type	Label
MR0	MsgTag	180

DispatchIPv4Packet

(legacy)

Message register	Type	Label
MR0	MsgTag	181

ParseProtocol

(legacy)

Message register	Type	Label
MR0	MsgTag	182

SendPacket

(legacy)

Message register	Type	Label
MR0	MsgTag	183

Control

Message register	Type	Label
MR0	MsgTag	184



```
typedef struct DRR {  
    Word // startup  
    Word // : , , ,  
    Word // : , ,  
    ;  
    Word // ,  
    Word // .  
    BIH // ,  
    Word // Heap  
    DeviceName //  
    Word // : 0xfeedface  
} DRR;
```



```
typedef struct BIH {  
    Word //  
    Word // .  
    Word //  
    Word //  
    Word //  
    Word // BSS  
    Word // BSS  
    Word // StackSize.  
    Word //  
    - .  
} BIH;
```