Xameleon protocols specification

Draft v5

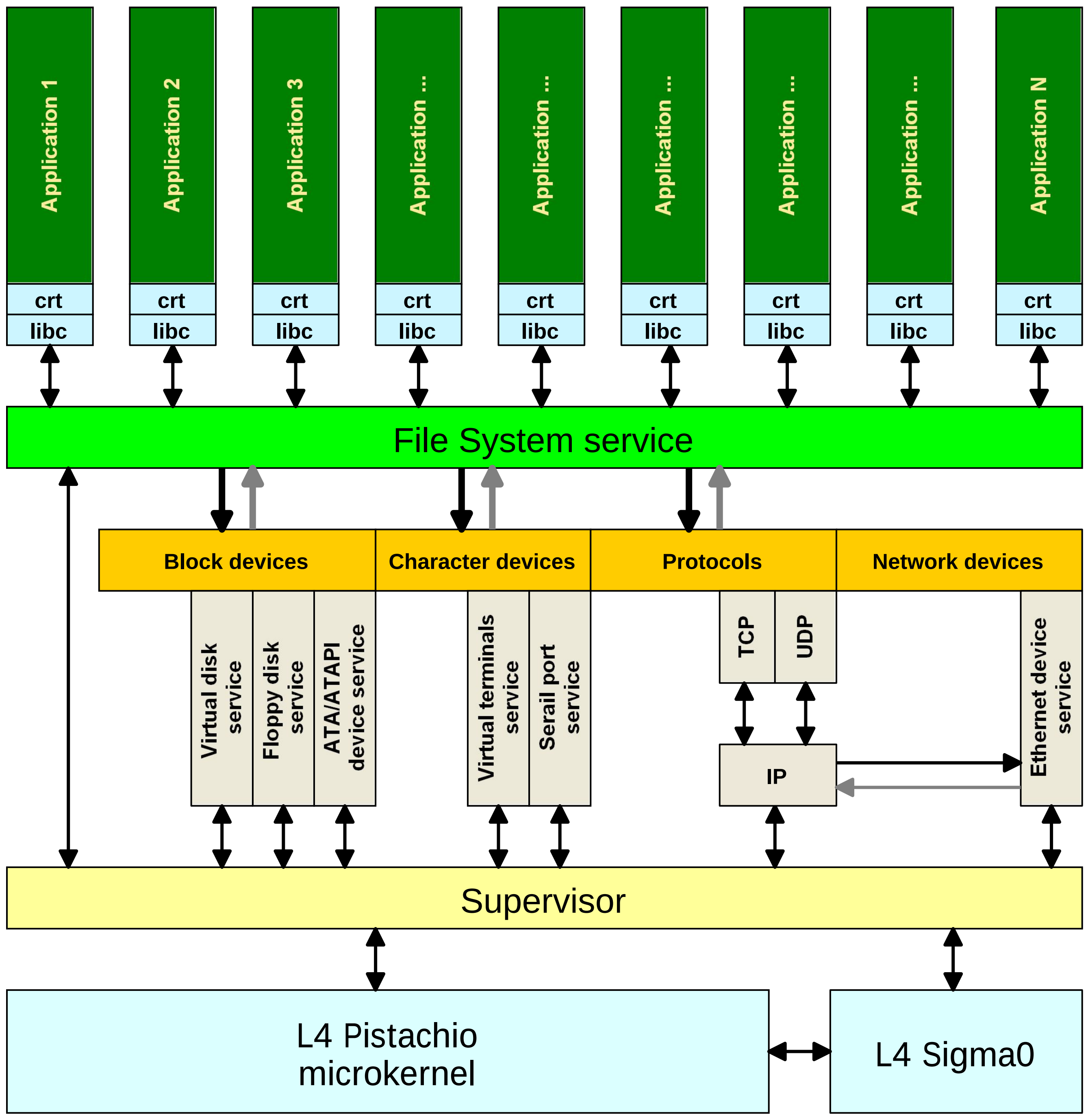
- Note that you have to be familiar with the L4 Specification Revision X2 (can be found at http://l4ka.org/projects/pistachio/l4-x2-r5.pdf) to understand these protocols.

Document revision history

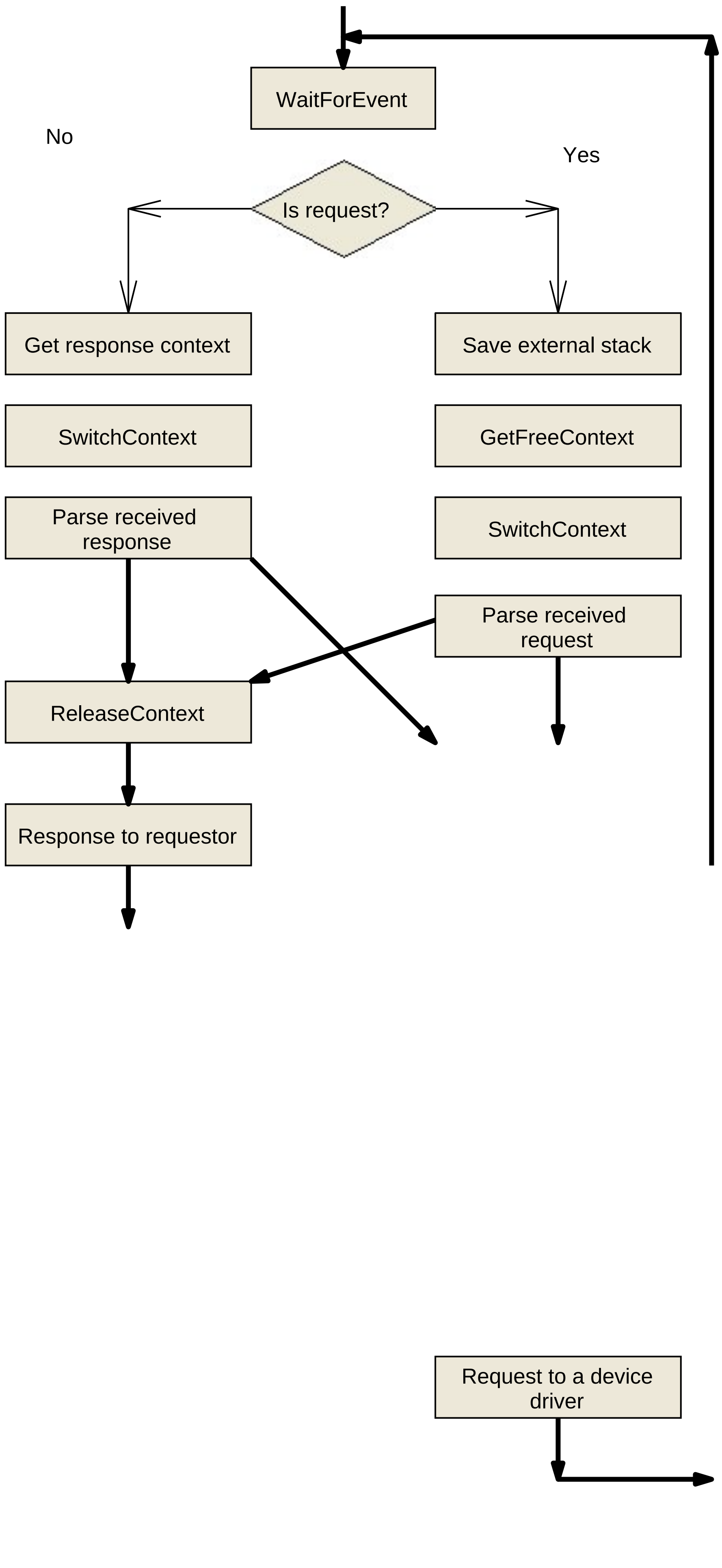
Revision 1	
11.24.2006	Initial document release
Revision 2	
11.26.2006	Added new protocols: FileSystem, BlockDevice, CharacterDevice
11.26.2006	Added new data type ThreadID
Revision 3	
11.27.2006	Rearranged supervisor's system calls.
11.28.2006	Added protocol layering model
Revision 4	
11.30.2006	Modified ReleaseSegment systemcall 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 semanitcs of StartDevice syscall
Revision 5	
10.11.2007	Add FileSync syscall description into filesystem section
10.11.2007	Change semantics of StartDevice syscall

Total 15 entries

Xameleon protocols layering model



Preliminary diagram of asynchronous over synchronous syscall implementation



Data types information

Following types shown here as a quick hint only.
Refer to L4 X2 specification for detail description of L4 data types.

Type	Description
Word	This data type is the machine word. It's size equal to the architecture word size.
String	The two-word value that represents some memory region within current address space.
Compound	Several pairs of the two-word values that repesents several memory regions within current address space. On the IPC receiver side these memory regions share a common message buffer.
MapItem	The two-word value represents L4 flexpage and send base. This item is a part of 'Map as part of message' operation.
ThreadID	The bitfileds that represents programm's thread. This type defined in L4 X2 spec as L4_ThreadId_t. It's size equal to the architecture word size.
Unspecified	This value depends on another argument and out of scope this document.
MsgTag	Message tag bitfiled. This structure identifies message and its content.
DRR	Device registration record as a string or an element of a compound string. This data type conveys information required for a device's startup procedure.
BIH	Xameleon binary image header
DeviceName	Sixteen bytes string that represnts a device name

Supervisor

Супервизор это процесс, который управляет основными ресурсами системы: процессами, программными потоками, таймерами, сигналами и виртуальной памятью. В этом разделе описаны системные вызовы супервизора, управляющие этими ресурсами.

Method	Description
AllocateSegment	Метод AllocateSegment выделяет составной сегмент памяти и отображает его в адресное пространство вызывающего процесса.
ReferencingSegment	Метод ReferencingSegment управляет количеством ссылок на сегмент. В момент создания составного сегмента счётчику ссылок присваивается единица. При удалении последней ссылки страницы сегмента и сегмент освобождаются.
QueryInterface	Метод QueryInterface используется для доступа к некоторым объектам супервизора из других модулей системы. Находится в стадии дизайна.
ExitProcess	Метод ExitProcess завершает вызывающий процесс и освобождает занятые им ресурсы. Обратите внимание, этот вызов освобождает только ресурсы, выделенные супервизором.
CreateThread	Метод CreateThread создаёт новый программный поток в адресном пространстве вызывающего процесса. Вновь созданный программный поток разделяет все ресурсы процесса, Вызывающему процесс возвращается статус операции и идентификатор созданного потока.
ExitThread	Метод ExitThread завершает вызывающий программный поток и освобождает запись в таблице программных потоков супервизора. Этот метод возвращает ошибку в случае удаления единственного программного потока.
ForkProcess	Метод ForkProcess клонирует и запускает вызывающий процесс. В результате этого вызова создаётся новый процесс, с общим сегментов кода, точной копией сегментов данных, BSS и стека вызывающего процесса. Обратите внимание, клонируется только вызывающий поток, остальные программные потоки вызываю процесса не клонируются.
ExecProcess	Метод ExecProcess запускает процесс, образ которого находится в составном сегменте памяти. Границы логических сегментов процесса, а также аргументы и среда окружения передаются в виде параметров. Обратите внимание, данный вызов частично совместим со стандартом POSIX. В случае если вызывающий поток является лидером процесса, то все потоки за исключением лидера уничтожаются, а лидер настраивается на запускаемый образ. В случае же, когда вызывающий поток не является лидером процесса, создаётся новое адресное пространство и поток мигриуте во вновь созданное адресное пространство, становится его лидером и примнимает значения запускаемого образа. При этом поток исключается из адресного пространства вызывающего процесса.
StartService	Метод StartDevice используется для регистрации сервисов в системе. При старте, сервис при помощи этого системного вызова посылает супервизору регистрационный пакет, с параметрами сервиса. Формат регистрационного пакета смотрите ниже в этом документе. Одной из возможностей этого системного вызова является возможость запуска сервиса (драйвера протокола/устройства) в выделенном адресном пространстве.
StopService	Метод находится в стадии дизайна и не используется в настоящее время.
DeviceEnumerator	Метод DeviceEnumearator используется для доступа к списку зарегистрированных сервисов. В настоящий метод используется сервисом файловой системы для регистрации устройств в пространстве /dev.
GetDeviceHandle	Метод GetDeviceHandle возвращает идентификатор сервиса(драйвера протокола/устройства) по его имени. В качестве идентификатора сервиса использует идентификатор программного потока, отвечающий на системные вызовы сервиса. Обратитве внимание, что имя сервиса не может быть более 16 символов длиной. Это поле может использоваться как UUID сервиса.

Method	Description
GetProcessID	Метод GetProcessID возвращает идентификатор процесса (pid_t) по идентификатору его программного потока (L4_ThreadId_t).
GetThreadID	Метод GetThreadID возвоащает идентификатор программного потока, являющегося лидером процесса, чей идентификатор (pid_t) передан параметром process_id
ChangeHeapSize	Метод ChangeHeapSize управляет размером области данных, называемой "Куча". Эта область данных используется фунциями динмического выделения памяти.
SetSignal	Метод SetSignal управляет обработчиками сигналов вызывающего процесса.
Signal	Метод Signal посылает сигнал потоку,.указанному параметром thread_id.
ProcessWait	Метод ProcessWait останавливает вызываюший поток до завершения процесса потомка.Ниже определены параметры ожидания. Выход из ожидания возможен также по сигналу.
SetTimer	Метод SetTimer управляет таймерами вызывающего процесса
SetNotificationHandler	Метод SetNotificationHandler устанавливает идентификатор потока, который принимает сообщения от супервизора. Например, в настоящее время сообщения от супервизора принимает сервис файловой системы, для корректного освобождения ресурсов удаляемого процесса.
SystemHalt	Метод находится в стадии дизайна и не используется в настоящее время.
LeaveSignal	Метод LeaveSignal вызывается обработчиком сигналов, для того чтобы корректно покинуть обработчик.

AllocateSegment

Метод AllocateSegment выделяет составной сегмент памяти и отображает его в адресное пространство вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**1**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	address	Virtual address of requested segment
MR2	Word	size	Size of requested segment

Output message register values			
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

Метод ReferencingSegment управляет количеством ссылок на сегмент. В момент создания составного сегмента счётчику ссылок присваивается единица. При удалении последней ссылки страницы сегмента и сегмент освобождаются.

Message register	Type	Label
Tag	MsgTag	**2**

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

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

QueryInterface

Метод QueryInterface используется для доступа к некоторым объектам супервизора из других модулей системы. Находится в стадии дизайна.

Message register	Type	Label
Tag	MsgTag	**3**

ExitProcess

Метод ExitProcess завершает вызывающий процесс и освобождает занятые им ресурсы. Обратите внимание, этот вызов освобождает только ресурсы, выделенные супервизором.

Message register	Type	Label
Tag	MsgTag	**4**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	status	Program termination status

CreateThread

Метод CreateThread создаёт новый программный поток в адресном пространстве вызывающего процесса. Вновь созданный программный поток разделяет все ресурсы процесса, Вызывающему процесс возвращается статус операции и идентификатор созданного потока.

Message register	Type	Label
Tag	MsgTag	**5**

Input message register values			
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

Output message register values			
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

Метод ExitThread завершает вызывающий программный поток и освобождает запись в таблице программных потоков супервизора. Этот метод возвращает ошибку в случае удаления единственного программного потока.

Message register	Type	Label
Tag	MsgTag	**6**

Input message register values			
Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of destroing thread

Output message register values			
Message registers	Type	Name	Description
MR1	Word	status	Operation status

ForkProcess

Метод ForkProcess клонирует и запускает вызывающий процесс. В результате этого вызова создаётся новый процесс, с общим сегментов кода, точной копией сегментов данных, BSS и стека вызывающего процесса. Обратите внимание, клонируется только вызывающий поток, остальные программные потоки вызываю процесса не клонируются.

Message register	Type	Label
Tag	MsgTag	**7**

Input message register values			
Message registers	Type	Name	Description
MR1	ThreadID	thread_id	Thread identifier of the forking thread

Output message register values			
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

Метод ExecProcess запускает процесс, образ которого находится в составном сегменте памяти. Границы логических сегментов процесса, а также аргументы и среда окружения передаются в виде параметров. Обратите внимание, данный вызов частично совместим со стандартом POSIX. В случае если вызывающий поток является лидером процесса, то все потоки за исключением лидера уничтожаются, а лидер настраивается на запускаемый образ. В случае же, когда вызывающий поток не является лидером процесса, создаётся новое адресное пространство и поток мигриуте во вновь созданное адресное пространство, становится его лидером и примнимает значения запускаемого образа. При этом поток исключается из адресного пространства вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**8**

Input message register values			
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 sgment
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 programm calling arguments
MR12,13	Compound	environment	String that represent process environment

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

StartService

Метод StartDevice используется для регистрации сервисов в системе. При старте, сервис при помощи этого системного вызова посылает супервизору регистрационный пакет, с параметрами сервиса. Формат регистрационного пакета смотрите ниже в этом документе. Одной из возможностей этого системного вызова является возможость запуска сервиса (драйвера протокола/устройства) в выделенном адресном пространстве.

Message register	Type	Label
Tag	MsgTag	**9**

Input message register values			
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 ot this document
MR4,5	String	driver_argument_string	a string that represent paramters to the driver executable.

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

StopService

Метод находится в стадии дизайна и не используется в настоящее время.

Message register	Type	Label
Tag	MsgTag	**10**

Input message register values			
Message registers	Type	Name	Description
MR1	Unspecified	…	Format is not designed yet

Output message register values			
Message registers	Type	Name	Description
MR1	Word	status	Operation status

DeviceEnumerator

Метод DeviceEnumearator используется для доступа к списку зарегистрированных сервисов. В настоящий метод используется сервисом файловой системы для регистрации устройств в пространстве /dev.

Message register	Type	Label
Tag	MsgTag	**11**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	device_no	Device number. Increment it unitl successful status

Output message register values			
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

GetDeviceHandle

Метод GetDeviceHandle возвращает идентификатор сервиса(драйвера протокола/устройства) по его имени. В качестве идентификатора сервиса использует идентификатор программного потока, отвечающий на системные вызовы сервиса. Обратитве внимание, что имя сервиса не может быть более 16 символов длиной. Это поле может использоваться как UUID сервиса.

Message register	Type	Label
Tag	MsgTag	**12**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	device_name	Name of requested device driver

Output message register values			
Message registers	Type	Name	Description
MR1	Word	status	Operation status
MR2	ThreadID	thread_id	Thread identifier of device driver or nilthread if device does not exist

GetProcessID

Метод GetProcessID возвращает идентификатор процесса (pid_t) по идентификатору его программного потока (L4_ThreadId_t).

Message register	Type	Label
Tag	MsgTag	**13**

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

Output message register values			
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

Метод GetThreadID возвоащает идентификатор программного потока, являющегося лидером процесса, чей идентификатор (pid_t) передан параметром process_id

Message register	Type	Label
Tag	MsgTag	**14**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	process_id	Process identifier

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

ChangeHeapSize

Метод ChangeHeapSize управляет размером области данных, называемой "Куча". Эта область данных используется фунциями динмического выделения памяти.

Message register	Type	Label
Tag	MsgTag	**15**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	increment	Signed value that indicates heap increment number bytes

Output message register values			
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

Метод SetSignal управляет обработчиками сигналов вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**16**

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

Output message register values			
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

Метод Signal посылает сигнал потоку,.указанному параметром thread_id.

Message register	Type	Label
Tag	MsgTag	**17**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	status	Signal delivery status
MR2	Word	reserver	Reserved value

ProcessWait

Метод ProcessWait останавливает вызываюший поток до завершения процесса потомка.Ниже определены параметры ожидания. Выход из ожидания возможен также по сигналу.

Message register	Type	Label
Tag	MsgTag	**18**

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

Output message register values			
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

Метод SetTimer управляет таймерами вызывающего процесса

Message register	Type	Label
Tag	MsgTag	**19**

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

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

SetNotificationHandler

Метод SetNotificationHandler устанавливает идентификатор потока, который принимает сообщения от супервизора. Например, в настоящее время сообщения от супервизора принимает сервис файловой системы, для корректного освобождения ресурсов удаляемого процесса.

Message register	Type	Label
Tag	MsgTag	**20**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Status of operation

SystemHalt

Метод находится в стадии дизайна и не используется в настоящее время.

Message register	Type	Label
Tag	MsgTag	**21**

LeaveSignal

Метод LeaveSignal вызывается обработчиком сигналов, для того чтобы корректно покинуть обработчик.

Message register	Type	Label
Tag	MsgTag	**22**

FileSystem

Сервис файловой системы предоставляет наборм методов для управления файлами: создание, запись, чтение, удаление, запуск и дргие операции. Любой ресурс, который может быть отображен в виде файла, может быть использован пользовательским приложением при наличии соответствующих прав доступа. Помимо этого, сервис файловой системы предоставляет более высокоуровневый интерфейс для операции клонирования процесса.

Method	Description
Mount	Отображает блочное устройство, отформатированное в формате одной из поддерживаемых файловых систем, в одну из директорий дерева файловой системы.
Unmount	Снимает отображение блочного устройства из дерева файловой системы.
GetCurrentDirectory	Возвращает полный путь к текущей директории вызывающего процесса.
ChangeDir	Изменяет текущую директорию вызывающего процесса.
OpenDirectory	Открывает директорию для чтения при помощи метода ReadDirectory. При этом устанавливается позиция на первый элемент директории.
ReadDirectory	Читает текущую запись в директории, открытой на чтение при помощи метода OpenDirectory. После чтения устанавливает позиция на следующую запись в директории.
CloseDirectory	Закрывает директорию, ранее открытую методом OpenDirectory.
GetFileStatus	Возвращает аттрибуты файла.
ChangeRootDirectory	Метод изменяет корневую директорию вызывающего процесса.
OpenFile	Открывает или создаёт и открывает файл.
CloseFile	Закрывает файл, ранее открытый при помощи метода OpenFile
ReadFile	Чтение из файла
WriteFile	Запись в файл.
RemoveFile	Удаление файла
RemoveDirectory	Удаление директории
CreateDirectory	Создание директории
CreateHardLink	Создание ссылки на существующий файл
CreateSybolicLink	Создание символьной ссылки на существующий файл
RenameFile	Переименование файла
SeekFile	Установка текущей позиции в открытом файле
GetSuperblockStatus	Возвращает статус файловой системы, отображенной в дерево файловой системы.
ChangeOwner	Изменяет владельца и группу файла
ChangeMode	Изменяет аттрибуты файла.
FlushCaches	Сбрасывает дисковые буфера на физический диск
DupDescriptor	Выделяет новый файловый дискриптор и устанавливает его на уже существующий
CopyDescriptor	Копирует файловый дескриптор. Если дескриптор-применик указывает на открытый файл, то файл предварительно закрывается

Method	Description
CreatePipe	Создаёт простейший канал и возвращает дескрипторы обоих концов канала.
ForkProcess	Создаёт копию вызывающего процесса. При этом оба процесса разделяют общий сегмент кода, а сегменты данных, .bss и стека копируются
Exec	Запускает на исполнение процесс, путь к которому указан в параметрах команды.
MakeNode	Создаёт элемент INODE
IOCTL	Упраление устройствами ввода/вывод. Этот системный вызов транслирует вложенные команды к низлежащим сервисам и возвращает полученный ответы вызывающему программному потоку.
Exit	Заверешние процесса и передача статуса завершения процесса.
ControlFile	Предоставляет сервис, аналогичный POSIX fcntl()
ChaneFileTimes	Предоставляет сервис, аналогичный POSIX utimes()
CreateSession	Предоставляет сервис, аналогичный POSIX setsid()
FileSync	Сбрасывет дисковые буфера и другие структуры данных, относящиеся к файловому дискриптору, переданному аргументом Handle, на диск.

Mount

Отображает блочное устройство, отформатированное в формате одной из поддерживаемых файловых систем, в одну из директорий дерева файловой системы.

Message register	Type	Label
Tag	MsgTag	**64**

Input message register values			
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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Mount operation status

Unmount

Снимает отображение блочного устройства из дерева файловой системы.

Message register	Type	Label
Tag	MsgTag	**65**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	Point	The mount point

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

GetCurrentDirectory

Возвращает полный путь к текущей директории вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**66**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Size	Receiver's buffer size.

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

ChangeDir

Изменяет текущую директорию вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**67**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	Path	New process path

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

OpenDirectory

Открывает директорию для чтения при помощи метода ReadDirectory. При этом устанавливается позиция на первый элемент директории.

Message register	Type	Label
Tag	MsgTag	**68**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	Path	Path to directory for open

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	Handle	Directory handle.

ReadDirectory

Читает текущую запись в директории, открытой на чтение при помощи метода OpenDirectory. После чтения устанавливает позиция на следующую запись в директории.

Message register	Type	Label
Tag	MsgTag	**69**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Handle	Handle to directory

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Status of operation
MR2,3	String	Record	Diectory record. Subsequent calls return next directory entry

CloseDirectory

Закрывает директорию, ранее открытую методом OpenDirectory.

Message register	Type	Label
Tag	MsgTag	**70**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Handle	Handle to directory

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

GetFileStatus

Возвращает аттрибуты файла.

Message register	Type	Label
Tag	MsgTag	**71**

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

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

ChangeRootDirectory

Метод изменяет корневую директорию вызывающего процесса.

Message register	Type	Label
Tag	MsgTag	**72**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	Path	Path to new root direcory of caller process

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Opeartion status

OpenFile

Открывает или создаёт и открывает файл.

Message register	Type	Label
Tag	MsgTag	**73**

Input message register values			
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

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

CloseFile

Закрывает файл, ранее открытый при помощи метода OpenFile

Message register	Type	Label
Tag	MsgTag	**74**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Handle	File handle to close

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operaion status

ReadFile

Чтение из файла

Message register	Type	Label
Tag	MsgTag	**75**

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

Output message register values			
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

Запись в файл.

Message register	Type	Label
Tag	MsgTag	**76**

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

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

RemoveFile

Удаление файла

Message register	Type	Label
Tag	MsgTag	**77**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Deletion status

RemoveDirectory

Удаление директории

Message register	Type	Label
Tag	MsgTag	**78**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	DirName	The directory name to delete

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateDirectory

Создание директории

Message register	Type	Label
Tag	MsgTag	**79**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateHardLink

Создание ссылки на существующий файл

Message register	Type	Label
Tag	MsgTag	**80**

Input message register values			
Message registers	Type	Name	Description
MR1,2	Compound	Source	Source filename
MR3,4	Compound	Target	Target filename

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateSybolicLink

Создание символьной ссылки на существующий файл

Message register	Type	Label
Tag	MsgTag	**81**

Input message register values			
Message registers	Type	Name	Description
MR1,2	Compound	Source	Source filename
MR3,4	Compound	Target	Target filename

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

RenameFile

Переименование файла

Message register	Type	Label
Tag	MsgTag	**82**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Opeation status

SeekFile

Установка текущей позиции в открытом файле

Message register	Type	Label
Tag	MsgTag	**83**

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

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

GetSuperblockStatus

Возвращает статус файловой системы, отображенной в дерево файловой системы.

Message register	Type	Label
Tag	MsgTag	**84**

Input message register values			
Message registers	Type	Name	Description
MR1,2	String	Path	Path

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

ChangeOwner

Изменяет владельца и группу файла

Message register	Type	Label
Tag	MsgTag	**85**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

ChangeMode

Изменяет аттрибуты файла.

Message register	Type	Label
Tag	MsgTag	**86**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Mode	File mode
MR2,3	String	Filename	Filename

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

FlushCaches

Сбрасывает дисковые буфера на физический диск

Message register	Type	Label
Tag	MsgTag	**87**

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

DupDescriptor

Выделяет новый файловый дискриптор и устанавливает его на уже существующий

Message register	Type	Label
Tag	MsgTag	**88**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Handle	File handle for duplication

Output message register values			
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

Копирует файловый дескриптор. Если дескриптор-применик указывает на открытый файл, то файл предварительно закрывается

Message register	Type	Label
Tag	MsgTag	**89**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Target	Target file descriptor
MR2	Word	Source	Source file descriptor

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreatePipe

Создаёт простейший канал и возвращает дескрипторы обоих концов канала.

Message register	Type	Label
Tag	MsgTag	**90**

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status
MR2	Word	PipeIn	Pipe input endpoint
MR3	Word	PipeOut	Pipe output endpont

ForkProcess

Создаёт копию вызывающего процесса. При этом оба процесса разделяют общий сегмент кода, а сегменты данных, .bss и стека копируются

Message register	Type	Label
Tag	MsgTag	**91**

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

Output message register values			
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

Запускает на исполнение процесс, путь к которому указан в параметрах команды.

Message register	Type	Label
Tag	MsgTag	**92**

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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Exec status. Status is nevere returned on success execution.

MakeNode

Создаёт элемент INODE

Message register	Type	Label
Tag	MsgTag	**96**

Input message register values			
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

Output message register values			
Message registers	Type	Name	Description
MR0	Word	Status	Operation status

IOCTL

Упраление устройствами ввода/вывод. Этот системный вызов транслирует вложенные команды к низлежащим сервисам и возвращает полученный ответы вызывающему программному потоку.

Message register	Type	Label
Tag	MsgTag	**97**

Input message register values			
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

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

Exit

Заверешние процесса и передача статуса завершения процесса.

Message register	Type	Label
Tag	MsgTag	**98**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Status that must be sent to waiting processes

ControlFile

Предоставляет сервис, аналогичный POSIX fcntl()

Message register	Type	Label
Tag	MsgTag	**99**

Input message register values			
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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Control operation status
MR2…	Unspecified	…	Format of these registers depends on Command mode

ChaneFileTimes

Предоставляет сервис, аналогичный POSIX utimes()

Message register	Type	Label
Tag	MsgTag	**100**

Input message register values			
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

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Operation status

CreateSession

Предоставляет сервис, аналогичный POSIX setsid()

Message register	Type	Label
Tag	MsgTag	**101**

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Status of operation
MR2	Word	ProcessID	pid_t of new session

FileSync

Сбрасывет дисковые буфера и другие структуры данных, относящиеся к файловому дискриптору, переданному аргументом Handle, на диск.

Message register	Type	Label
Tag	MsgTag	**102**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Handle	File handle to flush it's cashes

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Status	Status of operation

BlockDevice

Протокол обмена данными и управления блочным устройством.

Method	Description
BlockDevice_Open	Открывает блочное устройство и возвращает его параметры
BlockDevice_Close	Закрывает блочное устройство
BlockDevice_Read	Читает блок данных с блочного устройства
BlockDevice_Write	Пишет блок данных на блочное устройство
BlockDevice_Flush	Сбрасывает дисковые буфера на физическое устройство
BlockDevice_IOCTL	Предоставляет управление устройством, например режимы энергопотребления, кеширования, режимы DMA.

BlockDevice_Open

Открывает блочное устройство и возвращает его параметры

Message register	Type	Label
Tag	MsgTag	**144**

Input message register values			
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

Output message register values			
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

Закрывает блочное устройство

Message register	Type	Label
Tag	MsgTag	**145**

BlockDevice_Read

Читает блок данных с блочного устройства

Message register	Type	Label
Tag	MsgTag	**146**

Input message register values			
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

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

BlockDevice_Write

Пишет блок данных на блочное устройство

Message register	Type	Label
Tag	MsgTag	**147**

Input message register values			
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	Desrcribes data type on top of that a caching scheme is selecting by the device driver
MR6,7	String	Data	Writing data

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status

BlockDevice_Flush

Сбрасывает дисковые буфера на физическое устройство

Message register	Type	Label
Tag	MsgTag	**148**

Input message register values			
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.

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Context	Context
MR2	Word	Status	Operation status

BlockDevice_IOCTL

Предоставляет управление устройством, например режимы энергопотребления, кеширования, режимы DMA.

Message register	Type	Label
Tag	MsgTag	**149**

Input message register values			
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

Output message register values			
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

Протокол обмена данными и управления поточного (символьного) устройства.

Method	Description
StreamDevice_Open	Открывает символьное устройство
StreamDevice_Close	Закрывает ранее открытое символьное устройство
StreamDevice_Read	Читает данные с символьного устройства
StreamDevice_Write	Пишет данные в символьное устройство
StreamDevice_IOCTL	Управляет символьным устройством.
StreamDevice_Log	Используется для посылки данных непосредственно на устройство, минуя сервис файловой системы. Этот вызов широко используется для отладки сервисов.

StreamDevice_Open

Открывает символьное устройство

Message register	Type	Label
Tag	MsgTag	**130**

Input message register values			
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()

Output message register values			
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

Закрывает ранее открытое символьное устройство

Message register	Type	Label
Tag	MsgTag	**131**

Input message register values			
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

Output message register values			
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

Читает данные с символьного устройства

Message register	Type	Label
Tag	MsgTag	**132**

Input message register values			
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.

Output message register values			
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 copyied into receiver's buffer

StreamDevice_Write

Пишет данные в символьное устройство

Message register	Type	Label
Tag	MsgTag	**133**

Input message register values			
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

Output message register values			
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 wtiten to device. Negative value signaling error and error code

StreamDevice_IOCTL

Управляет символьным устройством.

Message register	Type	Label
Tag	MsgTag	**134**

Input message register values			
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

Output message register values			
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

Используется для посылки данных непосредственно на устройство, минуя сервис файловой системы. Этот вызов широко используется для отладки сервисов.

Message register	Type	Label
Tag	MsgTag	**135**

Input message register values			
Message registers	Type	Name	Description
MR1	Word	Context	Этот параметр используется для совместимости формата сообщений сервисов
MR2	Word	Minor	Minor device number
MR3,4	String	Data	String that represent information. Usually a text representation of a log message

Output message register values			
Message registers	Type	Name	Description
MR1	Word	Context	Этот параметр используется для совместимости формата сообщений сервисов
MR2	Word	Status	Command status

NetworkService

Протокол обмена данными и управления сервисом сетевой передачи данных

Method	Description
NetworkService_CreateSocket	Создаёт сетевой сокет
NetworkService_CloseSocket	Освобождает ранее созданный сокет

NetworkService_CreateSocket

Создаёт сетевой сокет

Message register	Type	Label
Tag	MsgTag	**160**

NetworkService_CloseSocket

Освобождает ранее созданный сокет

Message register	Type	Label
Tag	MsgTag	**161**

Device registration record as a string or an element of a compound string. This data type conveys information required for a device's startup procedure.

```
typedef struct DRR {
  Word        ProtocolVersion;   // Version of the startup protocol
  Word        Category;          // Driver category: 0 - RAW, 1 - Stream, 2 - Block, 3 -
                                 Protocol
  Word        Model;             // Driver execution model: 0 - Module, 1 - Process, 2 -
                                 Active thread
  Word        FirstMinor;        // First minor number of a device set provided by the
                                 driver
  Word        MinorDeviceCount;  // Minor device count
  BIH         ImageHeader;       // Binary image header
  Word        MaxHeapSize;       // Maximum size of the driver's heap
  DeviceName  DeviceName;        // Device name that sent this request
  Word        MagicNumber;       // This field must convey hex value 0xfeedface
} DRR;
```

Xameleon binary image header

```
typedef struct BIH {
  Word        EntryPoint;        // entry point to the code
  Word        TextStart;         // start address of the code segment. Note that a
                                 current implementation does not distinguish between code
                                 and read-only data sections,
  Word        TextSize;          // size of the code segment
  Word        DataStart;         // start address of the initialized writible data
                                 segment
  Word        DataSize;          // size of the the initialized writible data segment
  Word        BssStart;          // start address of the BSS
  Word        BssSize;           // size of the BSS
  Word        StackBottom;       // Semanthics of this field depends on StackSize
                                 argument. If stack size is not zero, this value
                                 describes the bottom stack address. In other case, this
                                 value describes a maximum limit of executable stack
  Word        StackSize;         // Zero value forces a system stack allocation policy.
                                 Any other value describes programm stack size.
} BIH;
```