

## **Функциональные характеристики экземпляра программного обеспечения блока стационарной связи**

Функциональные характеристики экземпляра программного обеспечения блока стационарной связи (ПО БСС) представляют из себя характеристики образа операционной системы Linux и интегрированного в него ПО БСС.

Образ ПО БСС представляет из себя файл с расширением «img» и является образом диска, сформированным посредством утилиты «genimage». Образ несет в себе точные копии данных ПО БСС для соответствующего комплекта на определенном объекте применения и применяются для записи нового контента на требуемые дисковые носители.

Имя образа `bss_№объекта_комплект.img`, где «№объекта» — номер объекта применения, «комплект» - расположение комплекта в БСС (ориентация: левый, правый).

Размер образа 1213 Мб.

После успешного развертывания образа на процессорной плате ПО имеет вид, описанный ниже.

### **Структура и назначение исполняемых файлов и скриптов ПО БСС**

Прикладное программное обеспечение ПО БСС и скрипты запуска с настройками для исполняемых модулей ПО БСС находится в каталоге «/ksdserve/server\_saksd», внутри-машинная база данных в каталоге «/ksdserve/dbase\_bin», загружаемые модули ядра для задачи ПО БСС «/lib/modules/[версия\_ядра]/kernel/drivers/bss/», системные скрипты для автоматического запуска ПО и файлы настроек операционной системы (ОС) находятся в каталогах «/etc», «/etc/network» и «/etc/init.d».

Размеры модулей, исполняемых файлов и библиотек, размеры файлов внутри-машинной базы данных, а также затрачиваемые ресурсы при выполнении ПО представлены на момент создания документа для определенной версии ПО, используемого на конкретном объекте. На разных объектах применения, затрачиваемые ресурсы и список бинарных файлов внутри-машинной базы данных могут отличаться.

Перечень загружаемых исполняемых файлов и библиотек, а также и пути к ним, представлен в таблице 1.

Таблица 1. Перечень загружаемых исполняемых файлов и библиотек  
ПО БСС

Перечень файлов	Размер, байт
/ksdserv/server_saksd/bin/db_bin	33052
/ksdserv/server_saksd/bin/ksdbus	77468
/ksdserv/server_saksd/bin/server	341200
/ksdserv/server_saksd/bin/wdProc	16532
/ksdserv/server_saksd/libReadHosts.so	18084
/ksdserv/server_saksd/libSRRssv.so	13624
/ksdserv/server_saksd/libUDPTransfer.so	33156
/ksdserv/server_saksd/libUDPsr.so	71512
/ksdserv/server_saksd/libethernetM_Ext.so	59368
/ksdserv/server_saksd/libconvert.so	8384
/ksdserv/server_saksd/libdelay.so	13044
/ksdserv/server_saksd/liblog.so	25372
/ksdserv/server_saksd/libc_eventgl.so	24628
/ksdserv/server_saksd/libcrc16.so	13220
/ksdserv/server_saksd/libsystemer.so	10864
/ksdserv/server_saksd/libtimervirt.so	13060
/ksdserv/server_saksd/libtimerSelect.so	8700
/lib/modules/[версия_ядра]/kernel/drivers/bss/chstime_ hr_n/mod_chsthr_multi_fops.ko	15192
/lib/modules/[версия_ядра]/kernel/drivers/bss/multi_ev ent/mod_event_fops_once.ko	13052

Перечень загрузочных и вспомогательных скриптов представлен в таблице 2.

Таблица 2. Перечень загрузочных и вспомогательных скриптов

Директорий	Перечень файлов	Размер, байт
/ksdserv/server_saksd	killserv.sh	668
	startserv.sh	160
	startservAssembly.sh	4358
/etc/init.d/ (или /etc/)	S90start_serv ( или rc.local )	68

Перечень файлов внутри-машинной базы данных представлен в таблице 3.

Таблица 3. Перечень файлов внутри-машинной базы данных

Директорий	Перечень файлов	Размер, байт
/ksdserv/dbase_bin/228	OwnFile.bin	72
/ksdserv/dbase_bin	xmr228.bin	7190
	ars228.bin	5525
	pds.bin	4854
	xsv.bin	2275
	xps.bin	1412
	xps.bin	1412
	tvm	1232
	rsb2.bin	470
	xpr.bin	462
	zsm	442
	xrc228.bin	390

Таблица 3 (окончание)

	stan.bin	361
	xmk.bin	308
	lines.bin	286
	xbr.bin	283
	avtostop.bin	268
	bars1.bin	234
	rof.bin	217
	xkp.bin	124
	xbfm.bin	96
	xsp.bin	82
	OwnFile.bin	72
	zum.bin	65
	krb.bin	41
	tpui.bin	38

Затрачиваемые ресурсы выполняемого ПО БСС ОС при выполнении задачи на процессорной плате встраиваемой системы представлены на рисунке 1.

### **Задачи исполняемых модулей ПО БСС**

В данном разделе кратко описаны задачи выполняемого модуля, входные и выходные данные.

Затрачиваемые ресурсы ОС представлены выше на рисунке 1 и в таблице 1.



- занимается организацией обмена по локальной сети системы «Движение» с помощью использования специфичного протокола обмена, построенного на использовании собственных типов кадров Ethernet.

- взаимодействует с центральным процессом-менеджером (модуль «server») с помощью общей разделяемой памяти: атомарно записывает в память данные, полученные от абонентов локальной сети для модуля «server» и атомарно считывает данные из общей разделяемой памяти, которые подготовил модуль «server» для передачи абонентам локальной сети.

Входные данные:

- данные, читаемые из аргументов командной строки при старте процесса;
- принимаемые данные от абонентов локальной сети Ethernet;
- прочитанные данные из общей разделяемой памяти от модуля «server».

Выходные данные:

- передаваемые данные для абонентов локальной сети Ethernet;
- записываемые данные в общую разделяемую память от модуля «server».

### **Исполняемый модуль «server»**

Модуль «server» - программное обеспечение центрального процесса-менеджера (ПО ЦПМ).

Задачи модуля «server»:

- служит для организации взаимодействия абонентов локальной сети системы и абонентов внешней сети (перенаправление данных между абонентами в соответствии с их протоколам обмена), выполнения функций резервирования и безопасности;

- обеспечивает проверку и достоверность получаемой информации от абонентов, анализирует состояние абонентов по их диагностике, анализирует наличие абонентов в сети, проверяет однозначность передаваемой информации, сверяет на совпадение ВМБД абонентов, которые ее используют, сверяет на совпадение версий исполняемых модулей троированной вычислительной системы по их контрольным суммам;

- взаимодействие с процессом и потоками организации обмена по локальной и внешней вычислительным сетям через общие разделяемые ресурсы для каждого направления обмена данными.

Входные данные:

- данные, читаемые из аргументов командной строки при старте процесса;
- данные из общей разделяемой памяти, подготовленные модулем «ksdbus», от абонентов локальной сети;
- данные от абонентов внешней сети, читаемые из общих буферов ввода-вывода библиотеки «libUDPTransfer.so»;

Выходные данные:

- данные для абонентов локальной сети, записанные в общую разделяемую память с модулем «ksdbus»;
- данные для абонентов внешней сети, записываемые в общие буфера ввода-вывода библиотеки «libUDPTransfer.so»;

### **Исполняемый модуль «wdProc»**

Модуль «wdProc» - процесс, который служит для контроля факта выполнения процессов модулей «server» и «ksdbus».

Задачи модуля «wdProc»:

- отслеживает факт выполнения наличия подконтрольного процесса по его идентификатору в каталоге «/proc».

Входные данные:

- данные, читаемые из аргументов командной строки при старте процесса.

### **Задачи подключаемых библиотек ПО БСС**

#### **Библиотека «libReadHosts.so»**

Библиотека «libReadHosts.so» является динамически подключаемой библиотекой.

Задачи библиотеки «libReadHosts.so»:

- предоставляет ip-адрес по доменному имени с учетом специфики наименования абонентов сети.

Входные данные:

- имя абонента.

Выходные данные:

- ip-адрес абонента (адрес назначения) и режим работы абонента, прочитанные из файла «/etc/hosts».

### **Библиотека «libSRRssv.so»**

Библиотека «libSRRssv.so» является динамически подключаемой библиотекой.

Библиотека «libSRRssv.so» реализует обмен данными по принципу обмена «Send-блокированными» и «Receive-блокированными» сообщениями для обмена данными между двумя процессами с помощью объектов библиотеки «libc\_eventgl.so».

Задачи библиотеки «libSRRssv.so»:

- организация обмена данными через блокируемые функции «Send», «Reply» и «Receive»;

Входные данные:

- передаваемые данные другому процессу.

Выходные данные:

- принимаемые данные от другого процесса.

### **Библиотека «libUDPTransfer.so»**

Библиотека «libUDPTransfer.so» является динамически подключаемой библиотекой.

Библиотека «libUDPTransfer.so» создает и инициализирует объекты для приема и передачи данных абонентов по протоколу UDP, организует атомарный доступ к общим буферам ввода-вывода.

Задачи библиотеки «libUDPTransfer.so»:

- чтение исходных данных из системного файла ОС «/etc/services»;
- создание динамических объектов с помощью библиотеки «libUDPr.so», обеспечивающих организацию приема и/или передачи данных по протоколу UDP в соответствии с указанным портом службы, доменным именем и локальным ip-адресом интерфейсов, через которые будет организованы операции приема и передачи данных.

Входные данные:

- передаваемые данные объекту, который взаимодействует с абонентом.

Выходные данные:

- принимаемые данные от объекта, который взаимодействует с абонентом;
- ошибки приема-передачи данных от объекта, который взаимодействует с абонентом.

### **Библиотека «libUDPsr.so»**

Библиотека «libUDPsr.so» является динамически подключаемой библиотекой.

Библиотека «libUDPsr.so» организует прием и передачу данных по сетевому протоколу UDP под двум сетевым интерфейсам с абонентом.

Задачи библиотеки «libUDPsr.so»:

- организует прием и передачу данных по сетевому протоколу UDP под двум сетевым интерфейсам с абонентом, предоставляет результат успешно открытого соединения с абонентом, предоставляет результат успешной выдачи данных и приема данных от абонента, предоставляет информацию об ошибках открытия соединения и ошибках приема и передачи данных по каждому обслуживаемому сетевому интерфейсу.

Входные данные:

- передаваемые данные абоненту.

Выходные данные:

- принимаемые данные абонента;
- ошибки приема-передачи данных от абонента по двум сетевым интерфейсам.

### **Библиотека «libethernetM\_Ext.so»**

Библиотека «libethernetM\_Ext.so» является динамически подключаемой библиотекой.

Библиотека «libethernetM\_Ext.so» реализует обмен данными по внутреннему специфичному протоколу канального уровня сети Ethernet.

Задачи библиотеки «libethernetM\_Ext .so»:

- организует и отслеживает прием данных по специфичному протоколу канального уровня и конкретному сетевому интерфейсу;
- формирует диагностику по наличию активности приема данных на обслуживаемом сетевом интерфейсе;

- организует передачу данных по специфичному протоколу с определенным таймаутом;

- уведомляет о физическом отключении патч-корда от обслуживаемого объектом сетевого интерфейса.

Входные данные:

- передаваемые данные абоненту.

Выходные данные:

- принимаемые данные абонента;

- ошибки приема-передачи данных от абонента по двум сетевым интерфейсам.

### **Библиотека «libconvert.so»**

Библиотека «libconvert.so» является динамически подключаемой библиотекой.

Библиотека «libconvert.so» реализует подготовку двоичных данных для пересылки их абоненту, который использует другой порядок следования байтов.

Задачи библиотеки «libconvert.so»:

- реализует функции для перестановки местами каждой пары смежных байтов базовых типов разной размерности.

Входные данные:

- данное, описанное определенным базовым типом.

Выходные данные:

- результат перестановки.

### **Библиотека «libdelay.so»**

Библиотека «libdelay.so» является динамически подключаемой библиотекой.

Библиотека «libdelay.so» реализует функции для организации задержек выполнения процессов и потоков.

Задачи библиотеки «libconvert.so»:

- организация задержек от миллисекунд до секунд с использованием системной функции ОС «nanosleep».

Входные данные:

- время задержки в секундах, в миллисекундах или в секундах с миллисекундами.

### **Библиотека «liblog.so»**

Библиотека «liblog.so» является динамически подключаемой библиотекой.

Библиотека «liblog.so» предоставляет объект для записи информации в журнал логирования («./log/server.log»), циклической архивации журнала с перезаписью устаревших архивов, при превышении допустимого объема файла журнала.

Задачи библиотеки «liblog.so»:

- запись сообщений в файл журнала логирования с текущей датой и временем;
- контроль объема файла журнала;
- циклическая архивация файла журнала при превышении объема файла журнала при записи.

Входные данные:

- сообщение для записи, имя объекта или модуля.

Выходные данные:

- записанные строки сообщения в файл журнала в определенном формате (дата, время, имя объекта или модуля, который, который сделал запись).

### **Библиотека «libc\_eventgl.so»**

Библиотека «libc\_eventgl.so» является динамически подключаемой библиотекой.

Библиотека «libc\_eventgl.so» организует выделение общего защищенного ресурса (разделяемая память) для обмена данными между двумя процессами в по принципу синхронного обмена сообщениями (процесс, который выполняет передачу сообщения «Set» другому процессу, блокируется до тех пор, пока процесс-получатель не выполнит прием сообщения «Wait» и его обработку, а также не отправит ответное сообщение «Reset»).

Задачи библиотеки «libc\_eventgl.so»:

- организует выделение общего ресурса (разделяемая память) для обмена данными между двумя процессами;

- организует и обеспечивает атомарный доступ к выделенному ресурсу для двух процессов;

- предоставляет возможности обмена синхронными сообщениями с данными между двумя процессами.

Входные данные:

- передаваемые данные другому процессу.

Выходные данные:

- ответ с результатом приема данных другим процессом;
- данные от другого процесса в ответе по результатам приема.

### **Библиотека «libcrc16.so»**

Библиотека «libcrc16.so» является динамически подключаемой библиотекой.

Библиотека «libcrc16.so» предоставляет объекты для вычисления контрольной суммы предоставленных данных.

Задачи библиотеки «libc\_eventgl.so»:

- производит расчет контрольной суммы данных, переданных для расчета методу объекта библиотеки;
- расчета контрольной суммы CRC16 по алгоритму CRC-16/CCITT.

Входные данные:

- передаваемые данные для расчета контрольной суммы.

Выходные данные:

- рассчитанная контрольная сумма переданных данных.

### **Библиотеки «libsystimer.so», «libtimervirt.so» и «libtimerSelect.so»**

Библиотеки «libsystimer.so», «libtimervirt.so» и «libtimerSelect.so» являются динамически подключаемыми библиотеками.

Библиотеки предоставляют объекты для работы с системными таймерами API Linux и API POSIX:

- библиотека «libsystem\_timer.so» - организация работы с расширенными таймерами системных часов семейства POSIX;
- библиотека «libtimer\_virt.so» - организует работу с интервальным таймером API Linux (стандартизирован в POSIX);
- библиотека «libtimer\_select.so» - организация работы таймера на основе системного вызова «select».

Задача библиотек:

- предоставляет упрощенный интерфейс для работы с системными таймерами.

Входные данные:

- интервал работы таймера (секунды, миллисекунды, микросекунды, наносекунды).

## **Задачи загружаемых модулей ядра ОС Linux для ПО БСС**

### **Модуль ядра «mod\_chsthr\_multi\_fops.ko»**

Модуль служит для организации таймера широкого разрешения в ядре ОС.

Задача модуля ядра «mod\_chsthr\_multi\_fops.ko»:

- организация получения события процессом (поток) о истечении периода работы таймера с помощью блокируемой функции системного ввода-вывода «read» или «poll»;

Входные данные:

- настройки и команды управления таймером;
- параметры командной строки: режим регистрации модуля, период срабатывания таймера.

### **Модуль ядра «mod\_event\_fops\_once.ko»**

Модуль служит для организации взаимодействия двух процессов по принципу обработки прерываний через очередь ожидания события.

Задача модуля ядра «mod\_event\_fops\_once.ko»:

- организация получения события процессом-клиентом от процесса-генератора события с помощью системных функций ввода-вывода «write», «read» или «poll»;

Входные данные:

- настройки и команды управления очередью ожидания события;
- параметры командной строки: режим регистрации модуля, период срабатывания сторожевого таймера для разблокирования очереди ожидания события.

### **Бинарные файлы внутри-машинной базы данных**

Файлы ВМБД — бинарные файлы в определенном формате с характеристиками и описанием объектов управления и контроля.

### **Задачи скриптов загрузки и настройки ПО БСС**

Скрипты выполняют загрузочную функцию для ПО БСС при старте ОС, а так же содержат параметры командной строки для запускаемых процессов ПО БСС и загружаемых модулей ядра ОС.

#### **Скрипт «killserv.sh»**

Задачи скрипта «killserv.sh»:

- останавливает процессы ПО БСС;
- выгружает модули ядра ПО БСС;
- очищает (удаляет) используемые ресурсы ОС ПО БСС;

#### **Скрипт «startserv.sh»**

Задачи скрипта «startserv.sh»:

- запускает на выполнение скрипт «startservAssembly.sh»;
- передает скрипту «startservAssembly.sh» режим работы комплекта БСС по умолчанию.

Выходные данные:

- режим работы комплекта по умолчанию.

## **Скрипт «startservAssembly.sh»**

Задачи скрипта «startservAssembly.sh»:

- запуск на исполнение скрипта «killserv.sh»;
- анализ аргументов командной строки при запуске скрипта;
- если не существует, то создание файла журнала логирования для ПО БСС в каталоге «./log»;
- удаления узлов символьных устройств модулей ядра ПО БСС;
- создание узлов символьных устройств модулей ядра ПО БСС;
- загрузка модуля ядра ПО БСС с параметрами;
- в зависимости от установленного режима комплекта — запуск процессов ПО БСС с соответствующими режиму параметрами командной строки.

Входные данные:

- режим работы комплекта по умолчанию.

## **Скрипты «S90start\_serv» или «rc.local»**

Скрипты «S90start\_serv» или «rc.local» запускаются на исполнение ОС при старте.

Задачи скриптов «S90start\_serv» или «rc.local»:

- запуск стартового скрипта «startserv.sh».