

1. ВВЕДЕНИЕ

1.1. Назначение	2
1.2. Условные обозначения.....	3
1.3. Принципы построения системы коммутации.....	4
1.3.1. Обобщённый коммутатор.....	4
1.3.2. Виды соединений.....	6
1.3.3. Пульт оператора.....	6
1.3.4. Рабочее место оператора.....	6
1.4. Обзор внутренней операционной системы	7
1.4.1. Свойства системы.....	7
1.4.2. Подготовка к работе	7
1.4.3. Ввод строк	9
1.4.4. Переключение виртуальных консолей.....	10
1.4.5. Прекращение процесса и перезагрузка	11
1.4.6. Зачем нужно монтировать дискеты	11
1.4.7. Установка времени и даты.....	12
1.5. Размещение и назначение файлов.....	13
1.6. Основные команды.....	16
1.6.1. Команда cd	16
1.6.2. Команда mkdir.....	17
1.6.3. Команда ls.....	17
1.6.4. Команда cp	18
1.6.5. Команда mv	19
1.6.6. Команда rm.....	20
1.6.7. Команда chmod	20
1.6.8. Команда mount.....	21
1.6.9. Команда umount.....	22
1.6.10. Команда Fatfsys.....	23
1.6.11. Команда fdformat	23
1.6.12. Команда dinit.....	23
1.6.13. Команда slay.....	23
1.6.14. Команда tar	24
1.6.15. Команда gzip	24
1.6.16. Команда on	25
1.7. Репликации БД.....	26
1.7.1. Модификации ДВО	27
1.7.2. Особенности работы на единственной машине	27

1.1. Назначение

Руководство оператора (в дальнейшем руководство) предназначено для изучения и эксплуатации программного обеспечения (ПО) коммутатора цифровых каналов (КЦК), а также Бокса Групповой Ступени Коммутации (БГСК). Руководство содержит подробные сведения, необходимые оператору для конфигурирования аппаратуры с целью адаптации рабочей программы к конкретным условиям местной сети связи и текущего контроля за работой аппаратуры. Руководство предназначено для специалистов узлов связи, осуществляющих непосредственную эксплуатацию цифровых систем коммутации "Омега".

Руководство распространяется как на одиночный КЦК или БГСК, так и на несколько КЦК и/или БГСК, объединенных между собой, а так же с некоторыми дополнительными устройствами. С этой целью в руководстве введено понятие обобщенного коммутатора, под которым понимается совокупность устройств, предназначенных для коммутации цифровых каналов.

Замечание. Данное описание распространяется также и на БКУ (БКУ192 и БКУ400), однако с некоторыми ограничениями. Ограничения обусловлены прежде всего серьезными отличиями в аппаратной части. Так, например, к БКУ192 можно подключить только один интерфейсный блок. Технические характеристики БКУ192 можно найти в документе "Руководство по технической эксплуатации КАН192". Технические характеристики БКУ400 можно найти в документе "Руководство по технической эксплуатации КАН400".

1.2. Условные обозначения

В данном руководстве приняты следующие условные обозначения:

АМТС - автоматическая междугородная телефонная станция;
АОН - автоматическое определение номера;
АТС - автоматическая телефонная станция;
БГСК - бокс групповой ступени коммутации;
ВСК - выделенные сигнальные каналы;
ИКМ - импульсно-кодовая модуляция (PCM);
КАН - концентратор абонентской нагрузки;
КИ - канальный интервал (time slot);
КЦК - коммутатор цифровых каналов;
ОЗУ - оперативное запоминающее устройство (RAM);
ОС - операционная система;
ПО - программное обеспечение;
СИАС - сигнал извещения аварийного состояния;
СЦС - сверхцикловая синхронизация (multiframe synchronization);
СУБД - система управления базами данных;
Ц - цикл (frame);
ЦАТС - цифровая автоматическая телефонная станция;
ЦКУ - центральное коммутационное устройство;
ЦОС - цифровая обработка сигналов;
ЦП - центральный процессор (CPU);
ЦС - цикловая синхронизация (frame synchronization).

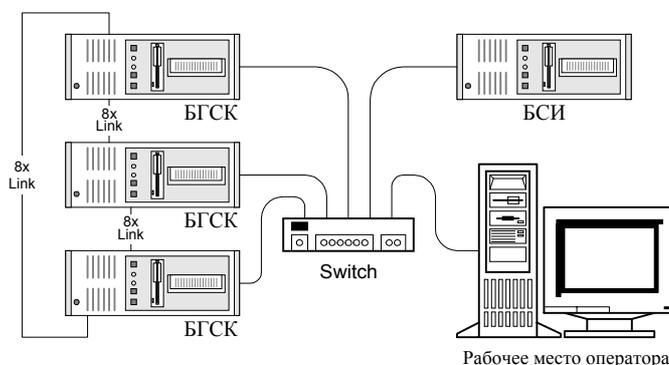
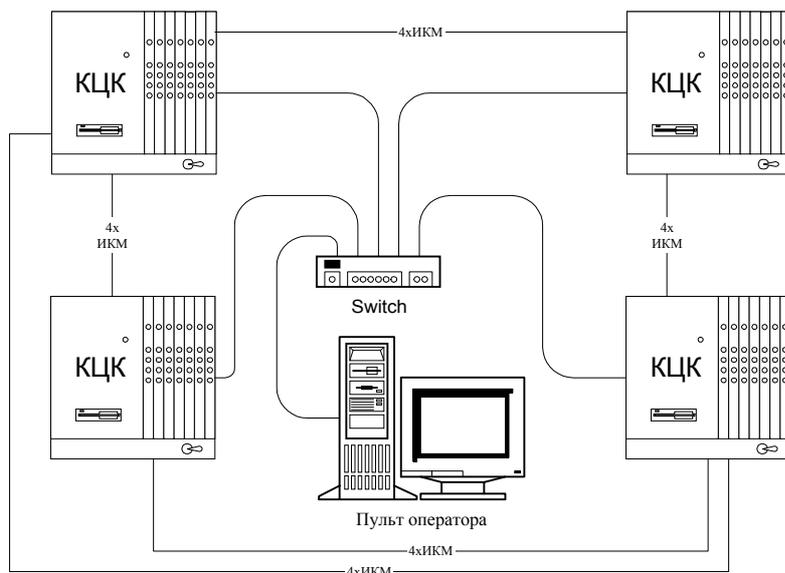
1.3. Принципы построения системы коммутации

1.3.1. Обобщённый коммутатор

Обобщённым коммутатором называется совокупность устройств, предназначенных для коммутации цифровых каналов. Обобщённый коммутатор состоит из отдельных боксов, число которых в системе коммутации может колебаться от 1 до 16.

В настоящее время в обобщённый коммутатор входят только активные боксы, то есть те, которые занимаются непосредственно маршрутизацией и коммутацией вызовов. Остальные боксы, хотя и являются частью ЦАТС “Омега” не входят в состав обобщённого коммутатора.

Бокс с номером 1 обычно, если система коммутации содержит выносной пульт оператора или рабочее место оператора, является персональным компьютером пульта оператора или рабочим местом оператора и служит для настройки, управления и диагностики всей системы. Ниже приведены некоторые варианты построения обобщённого коммутатора в составе ЦАТС “Омега”.



На каждом боксе установлена Внутренняя Операционная Система, позволяющая так организовать взаимодействие процессов, находящихся в отдельных боксах, как

если бы они находились в одном боксе. Это дает возможность, кроме всего прочего, на рабочем месте оператора легко получать информацию о состоянии каждого бокса обобщенного коммутатора. Например, если необходимо с помощью программы конфигуратора, исполняемый файл которой находится на жестком диске бокса номер 1, сконфигурировать бокс с номером 4, можно с бокса номер 1 осуществить запуск программы конфигуратора на боксе номер 4, и при этом результаты будут отображаться на дисплее бокса номер 1.

Рабочее место оператора присутствует в системе на правах дополнительного оборудования и может в течение любого промежутка времени находиться в выключенном состоянии. На работу системы это не влияет.

В системе может присутствовать любое количество рабочих мест оператора. Первое из них обычно является боксом с номером один.

Задачи, связанные со сбором и хранением информации в системе “Омега” функционально возложены на Бокс Служебной Информации (БСИ).

Если в системе присутствует ОДИН активный бокс, то он может выполнять одновременно и функции БСИ. Для этого на него устанавливается жёсткий диск достаточно большого объёма. Однако, в таком варианте, во первых, допустимо использование не всех функций БСИ (например нельзя снимать статистику и обеспечивать быстрый просмотр трафика), а, во вторых, нельзя обслуживать через Базу Данных более полутора тысяч абонентов.

Если в системе присутствует БОЛЕЕ ОДНОГО активного бокса, тогда с целью увеличения надежности обобщенного коммутатора только Бокс Служебной Информации (БСИ) имеет жёсткий диск большого объема, а активные боксы имеют только флэш-диски, более надежные, но менее вместительные. На флэш-дисках находится информация, необходимая для работы самого бокса.

В системе возможно наличие Бокса Интеллектуальной Платформы, на который возлагаются функции предоставления пользователям дополнительных сервисов.

Возможно совмещение функций Бокса Интеллектуальной Платформы и Бокса Служебной Информации в том случае, когда объём выполняемой обоими функциональными устройствами работы невелик..

Ранее было возможно совмещение функций Рабочего Места Оператора и БСИ. В настоящее время подобное не допускается.

Возможно подключение монитора и клавиатуры к любому боксу, однако для любого бокса кроме РМО это аварийный режим.

1.3.2. Виды соединений

Для обмена информацией между боксами обобщенного коммутатора существуют четыре вида соединений:

- внешние соединения – ИКМ-тракты, с помощью которых боксы обобщенного коммутатора соединяются с внешней средой (другими АТС, специализированными устройствами, СОРМ, ...);
- внутренние соединения – ИКМ-тракты, с помощью которых осуществляется передача сигнализации и разговоров между боксами внутри системы;
- вспомогательные – для обмена информацией боксы обобщенного коммутатора соединяются между собой по сети Ethernet;
- объединительные – только для матрицы 64x64, БГСК внутри обобщенного коммутатора объединяются между собой с помощью последовательной шины - ST-bus, что позволяет не использовать для наращивания ёмкости дорогостоящие ИКМ порты. Однако, протяжённость такого соединения ограничена (менее пяти метров) .

1.3.3. Пульт оператора

(только для ранних поставок)

В общем случае, пульт оператора ЦАТС "Омега" представляет собой специальное устройство, выполненное на базе ЭВМ промышленного применения. Его надёжностные характеристики соответствуют станционным. Пульт оператора является частью станционного оборудования.

Пульт оператора совмещает функции РМО и БСИ.

1.3.4. Рабочее место оператора

В отличие от Пульта Оператора, Рабочее Место Оператора не является неотъемлемой частью станционного оборудования. РМО не несёт никакой нагрузки, за исключением обеспечения доступа оператору к системе. Он может быть включён только на время работы. Тем не менее, желательно не пренебрегать качеством данного устройства.

<p>ВНИМАНИЕ. В любом случае установка программного обеспечения на Рабочее Место Оператора должна осуществляться сотрудниками предприятия - изготовителя коммутационного оборудования.</p>
--

1.4. Обзор внутренней операционной системы

1.4.1. Свойства системы

Во многих случаях данная Внутренняя Операционная Система (ВОС) подобна MS-DOS, UNIX и многим другим традиционным операционным системам. Она реализует больше возможностей, чем одни, и меньше, чем другие. Конкретно в ней реализованы:

- многозадачность, возможность выполнять несколько различных копий кода, каждый из которых имеет свои собственные данные, на одной и той же машине в одно и то же время;
- реальное время, подразумевается, что система может отвечать на данное событие за определенный промежуток времени, то есть время может быть переменным, но всегда будет иметь определенный верхний предел;
- файловая система, иерархический набор файлов, содержащих программы и данные, и каталогов, содержащих информацию о файлах и других каталогах;
- устройства, доступ к внешней памяти, например, к дискам, и внешнему миру, например, к терминалам, через хорошо определенные независимые от устройств структуры;
- распределенность, то есть приложение может быть распространено на несколько физических боксов; различные части приложения могут взаимодействовать достаточно спокойно, не сознавая, что они находятся на различных машинах;
- взаимодействие между процессами, межпроцессорное взаимодействие позволяет любой задаче взаимодействовать и обмениваться данными с любой другой задачей.

1.4.2. Подготовка к работе

После включения питания и автоматического запуска операционной системы начинает свою работу утилита `login`. Утилита `login` запрашивает имя пользователя и пароль, которые проверяются в базе данных. Если пользователь введет правильную комбинацию, `login` запустит командный интерпретатор и загрузит среду пользователя.

Обычно (по крайней мере, так конфигурируется при поставке, если не оговорено другое) на приглашение ввести имя пользователя нужно ответить `root`

```
login: root
```

и нажать клавишу `<Enter>`. В состоянии поставки пароль для этого пользователя не запрашивается.

Далее ВОС готова к диалогу с пользователем. На экран обычно выдается приглашение (`#`). Диалог пользователя с операционной системой осуществляется в форме "команд" (утилит). Команда состоит из имени и, возможно, параметров. Ввод команды заканчивается нажатием клавиши `<Enter>`. Когда пользователь вводит команду, операционная система осуществляет поиск соответствующей утилиты или программы и ее выполняет.

Для получения краткой информации о назначении и формате запуска какой-либо утилиты или системной программы используйте специальную утилиту `use`.

Например:

`use cd` информация о запуске утилиты `cd`.

Замечание 1. Утилита `use` используется для получения информации только о программах и утилитах операционной системы.

Замечание 2. Если Вы работаете без пульта оператора, т.е. монитор и клавиатура подключены непосредственно к ЦКУ или БГСК, Вы не сможете воспользоваться утилитой `use`, т.к. ее нет на флэш-диске.

После ввода `login`'а вы можете запустить `MiShell Commander` - менеджер файлов, аналогичный по функциям `Norton Commander`'у. Для этого в командной строке введите:

`mqc` и нажмите клавишу `<Enter>`.

В данном руководстве не описывается работа с менеджером файлов. Если Вы недавно сели за компьютер, прочтите руководство по `Norton Commander`-у.

<p>ВНИМАНИЕ: <code>MQC</code> не является частью ПО ЦАТС “Омега” и не поддерживается в рамках гарантийного обслуживания.</p>

1.4.3. Ввод строк

Многие программы используют режим редактирования. Если программа выполняется в этом режиме, клавиши и их назначение, которые Вы можете использовать для ввода данных в виде строк приведены в Табл. I-1

Табл. I-1

Если Вы хотите:	Нажмите эту клавишу:
Перемещение курсора	
Переместить курсор влево на одну позицию	[←] (стрелка влево)
Переместить курсор вправо на одну позицию	[⇒] (стрелка вправо)
Переместить курсор в начало строки	[Home]
Переместить курсор в конец строки	[End]
Удаление символов	
Удалить символ слева от позиции курсора	[←] (Back space)
Удалить символ в позиции курсора	[Del]
Удалить все символы в текущей строке	[Ctrl]+[U]
Вставка/стирание символов	
Переключить режим вставки на режим наложения и обратно (значение по умолчанию –вставка)	[Ins]
Ввод строки	
Ввести строки	[Enter]

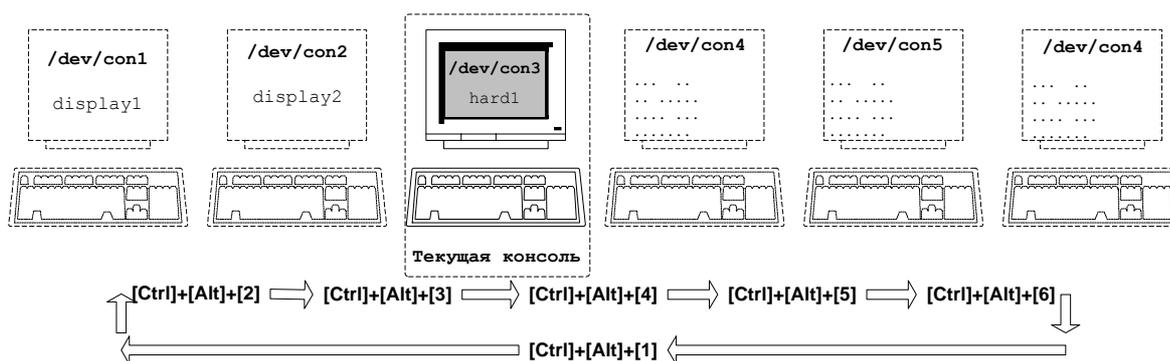
Командный интерпретатор позволяет Вам восстановить командную строку, которую Вы вводили раньше, затем вновь выполнить ее. Эти команды сохраняются командным интерпретатором в буфере. Клавиши для перемещения в буфере указаны в Табл. I-2.

Табл. I-2

Если Вы хотите переместиться в буфере:	нажмите эту клавишу:
Назад	[↑] (клавиша со стрелкой вверх)
Вперед	[↓] (клавиша со стрелкой вниз)

Когда Вы повторяете команду, Вы можете изменить её, используя стандартные клавиши редактирования строк.

1.4.4. Переключение виртуальных консолей



Экран дисплея и системная клавиатура в целом представляют собой консоль. Чтобы взаимодействовать с несколькими программами одновременно, операционная система позволяет работать с так называемых виртуальных консолей. Эти виртуальные консоли операционная система знает под именами `/dev/con1`, `/dev/con2` и т.д. Ввод/вывод информации той или иной программой осуществляется с/на консоль, с которой программа была запущена, или с/на консоль, которая была указана при запуске. При этом безразлично, работает оператор с данной консолью или нет. Тем не менее, реальный экран и клавиатура могут быть в один и тот же момент времени определены только как одна консоль, ведь на экране не помещается две картинки сразу. Таким образом, оператор всегда одновременно видит только одну виртуальную консоль. В системе поддерживается возможность одновременного существования до шести виртуальных консолей. Вы можете переключиться с одной виртуальной консоли на другую и, таким образом, из одной программы на другую, вводя комбинации клавиш в соответствии с Табл. I-3.

Табл. I-3

Если Вы хотите увидеть:	нажмите:
Следующую активную консоль	<code><Ctrl>-<Alt>-<Enter></code> или <code><Ctrl>-<Alt>-<+></code>
Предшествующую активную консоль	<code><Ctrl>-<Alt>-<-></code>
<code>/dev/con1</code>	<code><Ctrl>-<Alt>-<1></code>
<code>/dev/con2</code>	<code><Ctrl>-<Alt>-<2></code>
...	...
<code>/dev/con6</code>	<code><Ctrl>-<Alt>-<6></code>

Примечание. Упомянутые в таблице клавиши `<+>` и `<->` располагаются на цифровой клавиатуре с правой стороны.

Чтобы запустить `login` на неиспользованной консоли, сделайте следующее:

- переключитесь на другую консоль через комбинацию клавиш `<Ctrl>-<Alt>-<n>`;
- нажмите любую клавишу, `login` будет запущен, Вы можете теперь управлять консолями через любую из комбинаций клавиш переключения консоли, указанных в Табл. I-3 например, `<Ctrl>-<Alt>-<+>`.

- Когда Вы завершаете сеанс, вводя `logout` или `exit` или нажимая комбинацию клавиш `<Ctrl>-<D>`, консоль будет снова незанятой. Она не появится при использовании любой из циклических комбинаций клавиш переключения консоли. Исключением является консоль 1, на которой система обычно автоматически перезапускает утилиту `login`.

1.4.5. Прекращение процесса и перезагрузка

Если Вы должны прекратить процесс, запущенный с текущей консоли, нажмите комбинацию клавиш `<Ctrl>-<C>` или `<Ctrl>-<Break>`. Система сделает попытку прекратить процесс.

ВНИМАНИЕ!: Система попытается снять все запущенные с данной консоли утилиты.

Для перезагрузки Вашего компьютера, используйте комбинацию клавиш `<Ctrl>-<Alt>-<Shift>-`.

Для перезагрузки удаленного компьютера запустите на нем утилиту `shutdown`.

Например:

```
shutdown -n3 -f
```

или

```
on -n3 shutdown -f
```

перезагружает бокс номер три.

ВНИМАНИЕ! Перед вводом этих команд постарайтесь завершить работу всех приложений. Если Вы производите перезагрузку в момент, когда проводится критическая коррекция данных на диске, возможно, файловой системе потребуется восстановление.

1.4.6. Зачем нужно монтировать дискеты

Работая в операционной системе MS-DOS (MS Windows 95/98/XP), Вам, наверное, приходилось пользоваться дискетами. И это было очень просто – достаточно вставить дискету в дисковод и начать работать с ней так же, как с жёстким диском, отличие лишь в букве диска и скорости работы. Во внутренней операционной системе, как и во всех unix-подобных ОС, любое сменное устройство, будь то дискета, CD диск или дополнительный винчестер, перед началом работы с ним нужно смонтировать. Т.е. после того, как Вы вставили дискету в дисковод, нужно объяснить операционной системе, что у Вас есть носитель информации с известной ей файловой системой, и Вы хотите видеть его под каким-то именем (называемым точкой монтирования) на экране своего компьютера.

Для монтирования дискет с файловой системой ВОС используется утилита `mount`, дискет с файловой системой FAT (DOS-дискета) – `Fatfsys`. После монтирования дискета будет выглядеть как директория в корне Вашего диска, хотя она не является настоящей директорией и не расположена на жестком диске или диск-он-чипе. Реально утилитой монтирования будет создано местоположение в

пространстве путей, которое выглядит как директория и работать с ним можно так же, как с директорией.

ВНИМАНИЕ: Перед тем, как удалить дискету из дисковода, ее необходимо размонтировать.

1.4.7. Установка времени и даты

В системе “Омега” существует специальная утилита.

```
set_time
```

На активных боксах эта утилита находится в директории
`/tmp/tools`

На остальных боксах она находится в директории
`/omega/tools` (или `/home/tools` для версий 505 и младше)

Утилита запускается без параметров.

После запуска утилиты должен происходить примерно следующий диалог с пользователем (в примере ответы пользователя выделены жирным шрифтом, в реальности шрифт обычный):

```
Введите год[1997..2059] : 2006  
Введите месяц[1..12] : 11  
Введите день[1..31] : 2  
Введите час[0..23] : 12  
Введите минуту[0..59] : 12  
Введите секунду[0..59] : 33  
Введите Зима/Лето [0/1] : 1
```

После ввода последнего значения программа раздаёт введённые параметры всем боксам системы.

Внимание!
Введение правильных значений ВСЕХ запрашиваемых параметров **ОБЯЗАТЕЛЬНО!**

1.5. Размещение и назначение файлов

На каждом боксе обобщенного коммутатора расположено необходимое для его работы программное обеспечение: Типы файлов, которые Вы найдете в основных каталогах, представлены в Табл. I-4.

ВНИМАНИЕ: Оператору может понадобиться содержимое только тех каталогов, которые в Табл. I-4 выделены жирным шрифтом.

Замечание. Во всех версиях по 505 включительно основной директорией для хранения коммутационного ПО являлась директория home. Начиная с версии 507 (506-й версии не существует) основной директорией для хранения коммутационного ПО стала директория omega.

Табл. I-4

Каталог:	Описание:
Системные файлы	
/bin	Выполняемые системные
/dev	Блочные файлы устройств
/etc	Файлы инициализации и конфигурации
/etc/config	Sysinit и файлы конфигурации
/etc/licenses	Лицензии
/usr/bin	Выполняемые не системные
/usr/lib/terminfo	Файлы описания терминала
Коммутационная программа	
/home/bin или /omega/bin	Выполняемые
/home/configurations или /omega/configurations	Файлы, в которых содержится информация о конфигурации системы.
/home/files или /omega/files	Программы, загружаемые в интерфейсные блоки текущего бокса и блоки сигнальной обработки.
/home/log_files или /omega/log_files	Файлы отчётов основных программ.
/home/source или /omega/source	Архивы коммутационных программ
/home/version или /omega/version	Здесь находится идентификатор версии коммутационного ПО
/home/version/arc_log или /omega/version/arc_log	Архив данных аварийного завершения работы бокса
/home/tools или /omega/tools	Архивы утилит конфигурирования и обзора коммутатора (на активных боксах) или те же утилиты, готовые к выполнению (на пассивных).

Каталог:	Описание:
/tmp	Принятое по умолчанию место для временных файлов. Эта директория находится на ram-диске (в оперативной памяти). После загрузки операционной системы в неё распаковываются архивы утилит конфигурирования и обзора коммутатора, а также создаются файлы, содержащие диагностическую информацию
/hard/phonebook	Абонентская база данных (по 505 включительно)
/hard/phonebook10	Абонентская база данных (507)
/hard/abonents	Локальная база данных (по 505 включительно)
/hard/abonents10	Локальная база данных (507)
/hard/phones	Базы данных [ДВО, абонентов, биллинга] (508)
/hard/traffic	Трафик (можно размещать только на жестком диске) (по 505)
/hard/traffic10	Трафик (можно размещать только на жестком диске) (с 507)
/hard/statistics	Файлы статистического контроля объектов
/hard/protocols	В эту директорию дублируется информация, выводимая в окно сообщений, если разрешено протоколирование
/hard/basic_stat	Файлы базовой статистики

Конфигурационные файлы, хранящиеся в директории /home/configurations (или /omega/configurations), имеют имена

current_diag – конфигурация диагностики;
current_info – конфигурация системы;
current_hard – конфигурация оборудования;
current_prot – конфигурация протоколов;
current_soft – конфигурация линий;
omega.cfg – общие параметры данного бокса в системе (с 508 версии).

Замечание.

Начиная с версии 505аа на боксы не участвующие в коммутации (пассивные боксы) устанавливается сокращённая версия ПО. Для такого бокса актуальными являются только два вида конфигураций и соответственно два конфигуратора:

config_diag – конфигурация диагностики;
config_hard – конфигурация оборудования;

Внимание! Тем не менее ВСЕ конфигурационные файлы должны быть в наличии.

Программы, загружаемые в интерфейсные блоки, хранятся в директории /home/files (или /omega/files) и называются:

rcm30x4_0.503 для блока 0
rcm30x4_1.503 для блока 1
rcm30x4_2.503 для блока 2

pcm30x4_3.503 для блока 3
pcm30x4_4.503 для блока 4
pcm30x4_5.503 для блока 5
pcm30x4_6.503 для блока 6

Программа, загружаемая в блок DSPU, хранится в директории `/home/files` (или `/omega/files`) и называется

`dspu_0.out`

Программы, загружаемые в блок ADSP, хранятся в директории `/home/files` (или `/omega/files`) и называются:

`dsp133.axf`
`loader.axf`

1.6. Основные команды

Внутренняя Операционная Система – одна из многих юникс-подобных систем, отличающаяся от большинства своей простотой и компактностью. При этом основные команды и утилиты, присутствующие во всех юникс-подобных ОС, есть и в ВОС. В этой главе будут рассмотрены некоторые из них, самые необходимые.

Внимание! При наборе команд следует внимательно следить за установкой пробелов в тех позициях, где это необходимо. Параметры отделяются от команды и от других параметров пробелами.

Учтите, что в юникс-системах регистр имеет значение, например `Fatfsys` нужно обязательно писать с большой буквы, а `cp` с маленькой.

Табл. I-5

Работа с каталогами	
<code>cd</code>	Изменить Ваш текущий рабочий каталог
<code>mkdir</code>	Создать каталог
<code>pwd</code>	Отобразить Ваш текущий рабочий каталог
<code>ls</code>	Распечатать содержимое каталога
Работа с файлами	
<code>cp</code>	Копировать файлы
<code>mv</code>	Переименовать или переместить файлы
<code>rm</code>	Удалить файлы
<code>chmod</code>	Изменить атрибуты файла
Другие команды	
<code>ps, sin</code>	Отобразить процессы, выполняемые в данный момент
<code>mount</code>	Монтировать ВОС дискету
<code>umount</code>	Размонтировать ВОС дискету
<code>Fatfsys</code>	Монтировать DOS дискету
<code>tar</code>	Архивировать файлы
<code>gzip</code>	Упаковывать 1 файл
<code>slay</code>	Снять процесс
<code>on</code>	Выполнить процесс на другом боксе

1.6.1. Команда `cd`

`cd` - изменение текущего рабочего каталога

Синтаксис

```
cd [directory]
```

Параметры

`directory` - Имя пути каталога, который станет новым рабочим каталогом.

Примеры

```
cd..      Перейти к следующему более высокому каталогу в
          структуре каталогов.
```

<code>cd /</code>	Перейти к корневому каталогу.
<code>cd /path/dir</code>	Перейти к каталогу <code>/path/dir</code> .
<code>cd //1/home/tools</code>	Перейти к каталогу <code>/home/tools</code> на 1-ом узле.

Замечание. Вы можете использовать команду `pwd`, чтобы отображать текущий рабочий каталог.

1.6.2. Команда `mkdir`

`mkdir` - создание каталогов

Синтаксис

```
mkdir [-m mode] [-p] dir...
```

Параметры

<code>-m mode</code>	После создания каталога устанавливает биты разрешений (допусков) файла нового каталога к указанному значению режима. Параметр <code>mode</code> – строка символьного режима, как определено для утилиты <code>chmod</code> .
<code>-p</code>	Создает любые недостающие промежуточные компоненты имени пути.
<code>dir</code>	Спецификация (имя пути) создаваемого каталога.

Примеры:

<code>mkdir /Sasha</code>	Создать каталога <code>Sasha</code> в корневом каталоге текущего бокса.
---------------------------	---

1.6.3. Команда `ls`

`ls` - вывод содержимого каталога

Синтаксис

```
ls [-CFRacdilqrstul] [-DSbfgnopvx] [file...]
```

Параметры

<code>-a</code>	Перечисляет все файлы, включая те, которые начинаются с точки (<code>.</code>).
-----------------	---

По умолчанию они не перечисляются;

<code>-d</code>	Обрабатывает каталоги подобно файлам - дает информацию о самих каталогах, а не о файлах или подкаталогах, которые он содержит.
-----------------	--

<code>-l</code>	(буква нижнего регистра). Выводит в длинном формате. Эта опция предоставляет большинство относящейся к файлам информации, такой как разрешения доступа, число связей (псевдонимов), владелец/группа файла, наряду с размером, датой и именем файла.
-----------------	---

`File` Имя пути отображаемого файла. Если данный файл не найден, выводится диагностическое сообщение на вывод стандартной ошибки.
 Файлы отображаются в последовательности, указанной в командной строке.

Расширения Операционной Системы

`-D` Показывает только каталоги;
`-v` Выводит сначала каталоги;

Примеры:

`ls -a` Отобразить содержимое текущего каталога;
`ls -a -l` Отобразить содержимое текущего каталога подробно;
`ls -a //1/home/tools` Отобразить содержимое каталога `/home/tools` первого бокса.

1.6.4. Команда `cp`

`cp` - копирование файлов

Синтаксис

`cp [-f|-i] [-Rrp] [extensions] source_file target_file`
`cp [-f|-i] [-Rrp] [extensions] source_file... target_dir`

Параметры

`-i` Работа в интерактивном режиме.
`-p` После копирования пытается дублировать время модификации и файловый режим каждого входного файла в соответствующий выходной файл.
`-r` рекурсивное копирование каталогов.
`-R` Если `source_file` - каталог, производит рекурсивное копирование этого каталога со всеми содержащимися в нем файлами и подкаталогами.
`source_file` Имя пути файла, подлежащего копированию. Если вы хотите, чтобы файл-источник называл каталог, вы должны также указывать опцию `-R`;
`target_file` Имя пути файла, в который копируется одиночный файл;
`target_dir` Имя пути существующего каталога, который должен содержать выходной файл(ы).

Расширения Операционной Системы

`-c` Создает любые необходимые каталоги, чтобы открывать маршрут приемника.
`-f` Вызывает развязывание файла адресата до копирования.
`-v` Подробная работа.
`-V` Более подробная работа.

- x Копирует только, если файл-приемник уже существует.
- X Копирует только, если файл-приемник не существует.

Примеры:

Копирование файла `a430vsks.502` из ВОС дискеты, смонтированной под именем `fd0`, в файл `pcm30x4_0.503` в директории `/home/files/` на текущем боксе.

```
cp /fd0/a430vsks.502 /home/files/pcm30x4_0.503
```

Копирование каталога `/etc` со всем его содержимым из корневого каталога бокса 1 в директорию `/tmp` бокса 4.

```
cp -frv //1/etc //4/tmp/etc
```

1.6.5. Команда mv

`mv` - перемещение файлов

Синтаксис

```
mv [-f| -i] [-V|v] source_file target_file
mv [-f| -i] [-V|v] source_file target_dir
```

Параметры

- f Принудительная перезапись; не запрашивает подтверждения, если приемник существует. перезапись даже файлов только для чтения;
- i Работа в интерактивном режиме
- source_file Имя пути переносимого файла или каталога;
- target_file Новое имя пути переносимого файла или каталога;
- target_dir Имя пути существующего каталога, в который файл-источник должен перемещаться.

расширения Операционной Системы

- v Подробный вывод.
- V Очень подробный вывод;

Примеры

В текущем каталоге переименовать файл `pcm30x4_1.503` в `pcm30x4_2.503`:

```
mv pcm30x4_1.503 pcm30x4_2.503
```

В каталоге `/home/files` второго бокса переименовать файл `pcm30x4_1.503` в `pcm30x4_2.503`:

```
mv //2/home/files/pcm30x4_1.503 //2/home/files/pcm30x4_2.503
```

1.6.6. Команда `rm`

`rm` - удаление файлов

Синтаксис

```
rm [-Rfir] [-d] [-l n] [-v] file...
```

Параметры

<code>-R</code>	рекурсивное удаление файлов и подкаталогов в каталогах, заданных как аргументы. Этот процесс удаляет данный каталог и все дерево файлов и каталогов, находящееся в нем;
<code>-f</code>	Вызывает удаление каждого заданного файла без запроса на подтверждение;
<code>-i</code>	Интерактивная работа; вывод запроса на подтверждение перед удалением каждого существующего файла;
<code>file</code>	Полный путь удаляемого файла.

Расширения Операционной Системы

<code>-d</code>	Если параметр <code>-R</code> задан, удаляет файлы, но оставляет дерево каталогов нетронутым (то есть <code>rmdir</code> не выполняется);
<code>-v</code>	Подробный вывод; распечатайте файлы и каталоги по мере их удаления.

Примеры

<code>rm /tmp/log_files/*</code>	Удалить все файлы из директории <code>/tmp/log_files/</code>
<code>rm -Rf Sasha</code>	Удалить каталог <code>Sasha</code> и все его содержимое без запросов.
<code>rm //3/tmp/old_info</code>	Удалить файл <code>old_info</code> из каталога <code>tmp</code> бокса 3.

1.6.7. Команда `chmod`

`chmod` - изменение режимов файла

Синтаксис

```
chmod [-R] mode file...
```

Параметры

<code>-R</code>	Рекурсивное изменение режимов файла. Для каждого параметра <code>file</code> , который именуется каталог, <code>chmod</code> изменяет биты режима файла каталога и всех файлов, лежащих в файловой иерархии ниже его;
<code>mode</code>	Представляет изменения, которые необходимо сделать в режиме файла для каждого аргумента <code>file</code> (смотрите описание ниже);

`file` Имя пути файла, чьи биты режима файла должны быть изменены.

Описание

У каждого файла имеются атрибуты, присвоенные ему при создании. Атрибуты определяются с помощью битов режима доступа, имеющих следующие значения:

400	разрешение чтения владельцу файла
200	разрешение записи владельцу файла
100	разрешение выполнения владельцу файла
040	разрешение чтения членам группы
020	разрешение записи членам группы
010	разрешение выполнения членам группы
004	разрешение чтения прочим пользователям
002	разрешение записи прочим пользователям
001	разрешение выполнения прочим пользователям

Атрибуты могут отображаться символами:

```
-rwxrwxr-x 1 root root 11070 Sep 27 1999 chmod
-rw-rw-r-- 1 root root 23 Oct 10 1996 net
```

или числами, тогда `rwxrwxr-x` будет выглядеть как `775`, `rw-rw-r-` как `664`. `775` означает, что разрешено всё, кроме записи прочим пользователям, `664` – всё кроме выполнения всем категориям пользователей и записи прочим пользователям. Текстовые файлы обычно имеют биты режима доступа `664`, запускаемые `777`. После переноса с DOS дискеты или по другим причинам (но не самопроизвольно) биты режима доступа могут измениться, и исполняемый файл перестанет запускаться.

Утилита `chmod` позволяет изменять любые или все биты режима доступа файла для одного или более файлов. Для каждого указанного вами файла `chmod` изменяет биты режима доступа файла согласно операнду `mode`.

Чтобы изменить биты режима доступа файла, пользователем `chmod` должен быть владелец файла или суперпользователь (`root`).

Опция `mode` может быть указана либо в символьной форме, либо в виде неотрицательного восьмиричного целого числа.

Пример:

```
chmod 777 /tmp/tools/config_hard       Сделать исполняемым файл
config_hard
```

1.6.8. Команда `mount`

`mount` - монтирование разделов и файловых систем

Синтаксис

```
mount [-p block_special_file]
```

```
block_special_file directory [-r]]...
```

Параметры

`-p block_special_file` Читает таблицу разделов с указанного блочного специального файла и монтирует все найденные разделы. Разделы будут выглядеть, как блочные специальные файлы.

Блочные специальные файлы – это файлы типа `/dev/fd0` (блочный специальный файл дисковод), `/dev/hd0` (блочный специальный файл жесткого диска) и другие файлы в директории `/dev`.

`block_special_file directory [-r] -` монтирует блочный специальный файл, как файловую систему по имени `directory`. Если указывается опция `-r`, монтировать ее, как файловую систему только для чтения.

Утилита `mount` используется, чтобы устанавливать дисковые разделы, как блочные специальные файлы и чтобы монтировать блочные специальные файлы, как файловые системы ВОС. Когда вы монтируете блочный специальный файл, как файловую систему, местоположение в пространстве путей, где монтируется файловая система, называется точкой монтирования.

Пример:

```
mount /dev/fd0 /fd0    Смонтировать ВОС дискету под именем /fd0.
```

После монтирования в корневом каталоге появится директория `/fd0`, войдя в которую вы получите доступ к дискете с файловой системой ВОС.

1.6.9. Команда `umount`

`umount` - размонтирование файловых систем или разделов

Синтаксис

```
umount block_special_file
umount mount_point
```

<code>block_special_file</code>	Имя блочного специального файла;
<code>mount_point</code>	Точка монтирования файловой системы.

Примеры

```
umount /fd0    Размонтировать ВОС дискету:
```

1.6.10. Команда `Fatfsys`

`Fatfsys` – монтирование DOS дискеты.

Чтобы получить доступ к дискетам с файловой системой FAT, нужно запустить утилиту `Fatfsys`. Через несколько секунд в корневом каталоге появится директория `/dos`. Войдя в директорию `/dos` и поддиректорию `/a`, вы получите доступ к дискете с файловой системой FAT. Если у Вас старая версия ВОС, то вместо `Fatfsys` используйте `Dosfsys`.

ВНИМАНИЕ. Если что-то не получается, не пытайтесь создать директорию `/dos` с помощью команды `mkdir` или клавиши [F7] в MQC. Создание директории никак не поможет в деле монтирования дискеты. `/dos` – только выглядит как директория, на самом деле это точка монтирования, создаваемая утилитой `Fatfsys` и не являющаяся частью файловой системы жёсткого диска.

1.6.11. Команда `fdformat`

Форматируют дискету командой:

```
fdformat /dev/fd0.
```

После форматирования дискета не будет ни DOS, ни ВОС, поскольку файловая система после форматирования не создается. Чтобы создать на дискете файловую систему ВОС, используйте команду `dinit`.

ВНИМАНИЕ! При форматировании все данные, хранившиеся на дискете, будут уничтожены.

1.6.12. Команда `dinit`

После форматирования создать на дискете файловую систему ВОС можно командой

```
dinit /dev/fd0
```

Теперь можно рассматривать флоппи-дискету, как ВОС дискету, смонтировав ее под именем `/fd0`.

ВНИМАНИЕ! При инициализации все данные, хранившиеся на дискете, будут уничтожены.

1.6.13. Команда `slay`

Снятие процесса осуществляется командой

```
slay process_name
```

Например команда

`slay Fatfsys` – приведёт к снятию процесса обработки DOS'овской файловой системы и удалению из пространства путей директории (точки монтирования) `/dos`. При этом желательно выйти из просмотра этой дискеты с помощью команды `cd /` или перехода в другую директорию при помощи `mcsc`.

1.6.14. Команда `tar`

Создание архива производится командой

```
tar -cf file filename ...
```

`file` имя архива,

`filename` имена файлов или директорий, из которых создается архив.

Примеры:

```
tar -cf archive.tar //1/usr //1/bin
```

создаст в текущей директории файл `archive.tar`, содержащий директории `//1/usr` и `//1/bin`. При этом при разархивировании содержимое архива `tar` положит именно на первый бокс и именно в директории `//1/usr` и `//1/bin`.

```
tar -cf /tmp/archive.tar *
```

создаст в директории `/tmp` файл `archive.tar`, содержащий текущую директорию. При разархивировании содержимое архива `tar` положит в текущую на момент разархивирования директорию.

Извлечение файлов из архива выполняется командой

```
tar -xf file, где file - имя архива.
```

Внимание! `tar` не упаковывает файлы, а лишь создает архив. Архив будет не меньше, чем суммарный размер всех входящих в него файлов.

Тем не менее, если нужно упаковать, например, все содержимое директории, придется сначала создать архив, т.к. программа упаковки `gzip` умеет сжимать только файлы, но не директории. А программа архивирования умеет создавать файл из директории.

1.6.15. Команда `gzip`

Упаковка (сжатие без потери информации) файла производится утилитой

```
gzip file            где file – имя упаковываемого файла.
```

Для распаковки используйте

```
gzip -d file
```

1.6.16. Команда `on`

В ВОС Вы можете выполнять команды на другой машине (кроме Вашей собственной), если обе машины находятся в одной сети. Это называется удаленным выполнением. Каждая машина в сети называется боксом. Когда команда вызвана на другом боксе, стандартный ввод, стандартный вывод и вывод стандартной ошибки команды отображаются на экране Вашей консоли или терминале, если Вы явно не переназначите их на другое устройство. Для того, чтобы вызвать команду на другом боксе, используйте команду `on`.

Примеры:

Запустить утилиту `display_details` на 4-ом боксе, вывод будет отображаться на экране вызова:

```
on -n4 //4/tmp/tools/display_details
```

Загрузить команду `sin` из `/bin` бокса 2 и выполнить ее на боксе 4 (вывод команды `sin` все еще отображается на экране вызова):

```
on -n4 //2/bin/sin
```

Выполнить команду `sin` на боксе 4 и переадресовать ее выход на консоль 1 бокса 2 (поток входных данных команды `sin` и выходные ошибки все еще подключаются к экрану вызова):

```
on -n4 sin >//2/dev/con1
```

1.7. Репликации БД

Начиная с ПО версии 508 все «телефонные» базы данных (абоненты, ДВО, доп. флажки ДВО, расписания) обрабатываются (создаются и модифицируются) централизованно на БСИ и при необходимости автоматически реплицируются в БГСК.

Осуществляется это с помощью следующих средств:

- в raw-транспорте программа `mar demon` поддерживает т.н. индексные рассылки;
- каждая программа на БСИ, модифицирующая какую-либо базу данных, заносит запись об изменении в журнал изменений («кольцевой» журнал) и передает `mar demon`'у сведения для индексной рассылки. Одновременно об изменении извещается программа драйвера баз данных (`dbdriver`);
- драйвер базы данных, работающий как в режиме сервера (БСИ), так и в режиме клиента (БГСК), который и обеспечивает репликации;
- служебных программ `actualnames` (узнать актуальные имена БД на БГСК) и `glogtest` (вывести информацию журнала изменений).

Размещаются файлы БД в директории `/hard/phones`.

После запуска, драйвер БД передает в `mar demon` данные индексных рассылок БД, которые для каждой БД содержат сведения о самом последнем изменении каждой БД.

При получении этих данных `mar demon` распространяет их во все узлы сети, указывая в качестве процесса-получателя драйвер БД-агент. Рассылки повторяются каждые 30 сек, что обеспечивает их доставку во вновь подключённые боксы.

Агент, получив рассылку, сравнивает ее со своей последней обработанной и, если пришедшая оказывается более свежей, запрашивает сведения обо всех изменениях, начиная со следующего за последним обработанным (режим инкрементальной обработки – запрос только тех записей, которые были изменены). Если изменений очень много (больше, чем может одновременно хранить `glog`, который при заполнении затирает самую старую запись) или бокс агента-получателя был отключен в течение времени, за которое было выполнено много изменений, выполняется загрузка всей БД.

Во время полного обновления БД необходимо сохранить работоспособность стационарного ПО, поэтому загрузка всей базы ведется в отдельный файл, начало имени которого совпадает с именем файла на БСИ, а окончание формируется из даты и времени рассылки, вызвавшей полное обновление.

После окончания загрузки БД выполняется переключение на новую БД, а старая БД уничтожается. Сведения о последнем проведенном изменении записываются в специальный `replica`-файл, хранящий в первых 4-х байтах время последней полной загрузки (формат `time_t`) и данные последнего проведенного изменения (специальная структура. Вне драйвера БД интереса не представляет). Имя `replica`-файла `bcast_NN.replica`, где NN есть индекс БД в рассылке (с лидирующим нулем). Размещается файл, как и все остальные, в директории `/hard/phones`.

В случае БД абонентов функции переключения на новую БД и уничтожение старой БД возлагаются на процесс `line_switch`.

Так как в процессе закачки БД может произойти авария бокса, в том числе и перезагрузка, драйвер-агент при старте закачки создает специальный файл-индикатор и, если он обнаруживается при старте драйвера, недокачанная БД уничтожается (авария может привести к повреждению файловой системы, поэтому использовать такую БД опасно).

Кроме того драйвер обрабатывает ряд нештатных ситуаций, когда файлы поддержки репликаций уничтожены или разрушены. В этом случае делается попытка взять в качестве актуальной самую свежую БД и по ней восстановить герліса-файл. При отсутствии БД они создаются вновь и в них вносятся минимально необходимая служебная информация.

При нарушении связи (авария сети) драйвер прекращает закачку новой БД и уничтожает недокачанные файлы.

Потенциально допустима ситуация, когда одновременно присутствует 2 экземпляра БД абонентов (с разными именами, естественно). Это возможно, если БД абонентов первоначально отсутствует на БГСК, создается драйвером БД при старте и настолько мала, что успевает полностью закачаться до старта `line_switch`. В этом случае ненужный файл будет удален при первой же перезагрузке БГСК.

1.7.1. Модификации ДВО

Так как любые изменения БД выполняются только на БСИ, `line_switch` все запросы, приводящие к вызову функции изменения БД адресовать к драйверу БД БСИ. Прочие запросы (например, запросы профилей) могут выполняться локально, что приводит как к сокращению времени обработки запроса, так и к повышению надежности функционирования станции в целом.

Планировщик (`schexes`) загружается только на БСИ и модифицирует БД расписаний, которая не реплицируется на БГСК, так как ее единственным потребителем является сам планировщик, а модифицируется она драйвером БД (см. предыдущий абзац) и от имени диалоговой программы работы с БД (`panel`).

1.7.2. Особенности работы на единственной машине

Когда все станционное ПО размещено на одной машине, т.е. БСИ и БГСК совмещены, необходимость в репликациях отпадает, драйвер БД обслуживает все запросы локально, а имена БД фиксированы и совпадают с базовыми именами. Тем не менее `glog`'и ведутся и поддерживаются всеми участниками процесса.