Оглавление

1 Общие сведения о программном обеспечении

Linux Terminal Service Manager (LTSM) это набор программ для организации доступа к рабочему столу (сервер Linux) на основе терминальных сессий (с использованием протокола VNC или RDP).

Удаленный терминальный доступ решает следующие задачи:

- Улучшение работы клиент-серверных приложений за счет их запуска на сервере. Для этого есть специальный термин — «близость к данным», чем лучше связь между клиентской частью программы и серверной, тем быстрее выполняются задачи.
- Перевод пользователей с ПК на тонкие клиенты. Вместо компьютера с данными пользователю устанавливается миниатюрное устройство, которое позволяет подключаться к терминальной сессии. Тонкий клиент не требует обслуживания, не шумит, не греется, потребляет мало электричества. Позволяет свести к минимуму техническую поддержку на рабочих местах.
- Экономия трафика в сетях, и как следствие, уменьшение ширины и стоимости канала. В случае с терминальным доступом трафик, который раньше проходил между клиентскими станциями и серверами заменяется на трафик передачи изображения удаленного экрана.
- Централизованное управление программным обеспечением, позволяет привести все категории рабочих мест к унифицированному виду. Один администратор может управлять сотней рабочих мест. Ферма терминальных серверов позволяет оперативно доставлять необходимые корпоративные приложения, централизованно устанавливать обновления, управление данными сотрудников.

Для пользователя: увеличивается скорость работы с корпоративными программами, повышается стабильность работы, уменьшаются случаи обращения в службу технической поддержки.

Для системного администратора: переход к системе терминального доступа позволяет автоматизировать множество рутинных задач системного администратора, связанных с разворачиванием, обновлением и обслуживанием рабочих мест пользователей.

2 Описание процессов жизненного цикла ПО

Поставка программного обеспечения включает в себя дистрибутив, содержащий:

- исходный код проекта **LTSM**
- электронные документы по установке, эксплуатации и описанию процессов, обеспечивающих поддержание жизненного цикла ПО

Дистрибутив и документация закачиваются с сайта разработчика.

Использование требует выполнения следующих видов работ:

- настройка серверной части программного продукта
- настройка пользовательского окружения интерфейса
- настройка сетевой инфраструктуры
- настройка полномочий доступа к данным

Требования к уровню квалификации специалистов для работы с ПО:

- базовые знания администрирования ОС **Linux**
- базовые знания сетевого администрирования

Обучение специалистов по установке, настройке и работе с ПО **LTSM**, может выполняться:

- самостоятельно с использование прилагаемой документации
- путем консультаций согласно договору на техническую поддержку

Техническая поддержка оказывается только зарегистрированным пользователям. Под зарегистрированным пользователем понимается пользователь с которым заключен договор на настройку и сопровождение сервера. В техническую поддержку входят консультации с ответами на вопросы по функционалу, по установке, по возникающим ошибкам, исправление ошибок в работе ПО, а также обновление программного обеспечения **LTSM** и документации в случае выхода новой версии в период техподдержки.

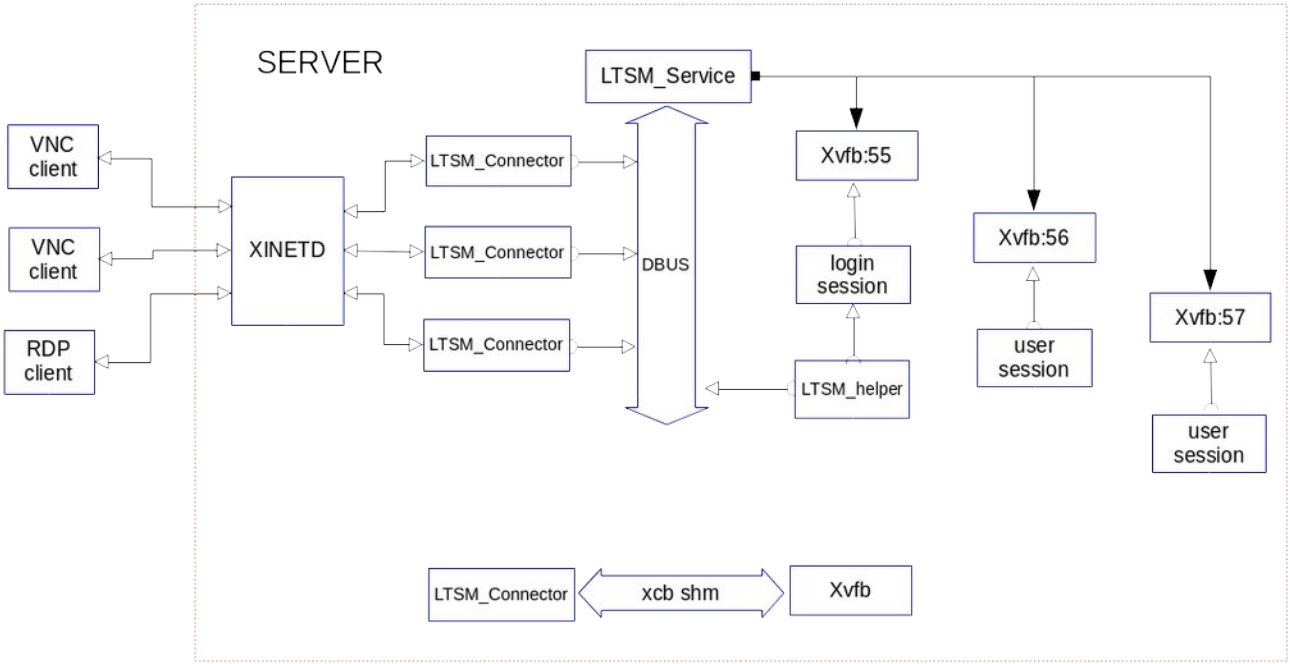
Вся доработка программного обеспечения связанная с добавлением новых функциональных возможностей не включена в техническую поддержку, и происходит отдельным договором, и стоимостью в зависимости от объема дополнительных работ.

3 Функциональные возможности системы

- Организация удаленного доступа на виртуальные рабочие сеансы сервера через протокол RFB или RDP с любого рабочего места на основе операционных систем Windows, MacOS, Linux, или FreeBSD и с любой точки сети Internet.
- Процедура проверки подлинности пользователя путем сравнения введенного им кодовой фразы с паролем на сервере.
- Сохранение рабочего сеанса пользователя при его отключении и восстановление рабочего сеанса пользователя при его повторном подключении.
- Шифрование трафика от конечного пользователя до сервера, на базе библиотеки gnutls, так же и с поддержкой ГОСТ алгоритмов.

4 Краткое описание взаимодействия сервисных программ

Схема взаимодействия:

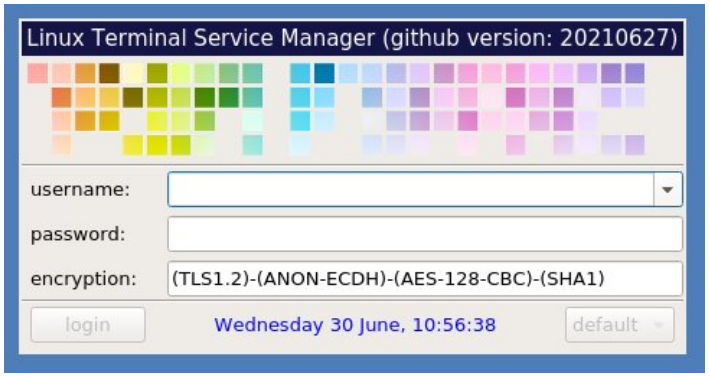


Основным механизмом обмена данными между процессами является шина **dbus**. В данной системе эта реализация осуществлена через API **sd-bus** поставляемым с **systemd** (подсистемой инициализации и управления службами в Linux).

Клиенты подключаются к системе через протокол **RFB** или **RDP** на служебный сервис **xinetd**/(**systemd sockets**), а **LTSM_connector** является обработчиком этого протокола.

При подключении клиента и согласовании протокола взаимодействия (**handshake**) коннектор через шину **dbus** отправляет запрос на запуск **login** сессии. При ограничении на максимальное количество сессий, коннектор может отказать в обслуживании клиентам.

Login сессия осуществляется через графическую утилиту **LTSM_helper**, задача которого только в удобном формате предоставить пользователю возможность авторизации в систему.



Всю авторизацию и аутентификацию осуществляет непосредственно сервис **LTSM_service** используя системные возможности **PAM** *(Pluggable Authentication Modules)*.

После успешной авторизации запускается сессия пользователя, либо если она уже была запущена, идет процесс непосредственного подключения к уже запущенной сессии пользователя.

Дальнейший весь механизм обмен данными происходит только между **LTSM_connector** и **Xvfb** через **xcb** протокол и **SHM** *(shared memory)*.

При отключении удаленной сессии пользователя, запущенный сеанс сохраняется, и при последующем подключении он восстанавливается.

Так же корректно обрабатывается весь аудит входа/выхода и несанкционированные попытки доступа в систему, все записи отражаются в системных журналах с тегом службы **LTSM_service**.

5 Установка программного обеспечения

Минимальные требования к оборудованию:

- 64-разрядный процессор с тактовой частотой 1.4ГГц, совместимый с набором инструкций для архитектуры x64
- 4 Гбайта оперативной памяти
- жесткий диск для установки базовой операционной системы не менее 40Гб.

Быстрая установка также возможна с использованием автоматизированной системы **Docker**:

```
docker pull docker.io/ltsm/devel:latest
docker run -i docker.io/ltsm/devel:latest
```

Установка пакетным менеджером (операционная система **CentOS7** x64)
Для установки в пакетном режиме необходимы следующие компоненты:

- `ltsm-latest.el7.x86_64.rpm`
- `sdbus-cpp-1.2.0-1.el7.x86_64.rpm`
- `freerdp-libs-2.4.0-1.el7.x86_64.rpm`
- `libwinpr-2.4.0-1.el7.x86_64.rpm`

Эти пакеты дополнительно выложены на ресурсе:
https://ltsm-soft.ru/el7

Команда пакетной установки:

```
yum install \
    http://ltsm-soft.ru/el7/ext/sdbus-cpp-1.2.0-1.el7.x86_64.rpm \
    http://ltsm-soft.ru/el7/ext/freerdp-libs-2.4.0-1.el7.x86_64.rpm \
    http://ltsm-soft.ru/el7/ext/libwinpr-2.4.0-1.el7.x86_64.rpm \
    http://ltsm-soft.ru/el7/ltsm-latest.el7.x86_64.rpm
```

Все примеры конфигурационных файлов идут вместе с установочным пакетом **ltsm-latest**, и расположены в **/usr/share/doc/ltsm/etc**

Дальнейшая настройка заключается в создании служебного пользователя и копировании сервисных файлов в соответствующие разделы, все эти действия описаны ниже.

6 Самостоятельная сборка программного обеспечения

Для самостоятельной сборки необходим минимум программ разработки, это компилятор **gcc** либо **clang** с поддержкой стандарта c++17, сборочная утилита **cmake** версии 3.14 и выше, системные библиотеки **pam-devel, gnutls-devel, zlib-devel**, набор графических библиотек **SDL2-devel, SDL2_image-devel, libxcb-devel, xcb-util-keysyms-devel, qt5-devel** (в составе модулей **Qt5::Core Qt5::Gui Qt5::DBus Qt5::Widgets**), так же дополнительно необходима библиотека **sdbus-cpp** из проекта https://github.com/Kistler-Group/sdbus-cpp

Процесс сборки проекта LTSM по командам:

```
git clone https://github.com/AndreyBarmaley/linux-terminal-service-manager.git
mkdir linux-terminal-service-manager/build && cd linux-terminal-service-manager/build
cmake .. -DCMAKE_BUILD_TYPE=Release && make
```

При наличии всех компонентов, результат сборки будет положительным, в виде готовых файлов:

- LTSM_connector
- LTSM_service
- LTSM_helper
- LTSM_sessions
- LTSM_sdl2x11

Пример сборки в операционной системе **Linux Debian 11**:

```
apt-get update && apt-get install -y cmake g++ git libpam0g-dev \
    libsdbus-c++-dev libgnutls28-dev freerdp2-dev libxcb1-dev libxcb-xtest0-dev \
    libxcb-xfixes0-dev libxcb-shm0-dev libxcb-randr0-dev libxcb-damage0-dev \
    libxcb-keysyms1-dev libsdl2-image-dev qtbase5-dev

git clone https://github.com/AndreyBarmaley/linux-terminal-service-manager.git
mkdir linux-terminal-service-manager/build && cd linux-terminal-service-manager/build
cmake .. -DCMAKE_BUILD_TYPE=Release && make
```

7 Добавление служебного пользователя и групп

Ограничение прав реализуется через группы, рекомендуемая схема:

```
groupadd -r ltsm-shm
groupadd -r ltsm-dbus
groupadd -r ltsm-auth
```

Группа **ltsm-dbus** служит для ограничения доступа на шину **dbus**: *ltsm.manager.service*
Группа **ltsm-auth** служит для ограничения на доступ к файлам **Xvfb** *xauthfile*
Группа **ltsm-shm** служит для ограничения доступа на **shm** канал взаимодействия между **LTSM_connector** и **Xvfb**

Служба коннектора работает под служебным пользователем, команда на создание пользователя:

```
useradd -c "LTSM xvfb user" -d /var/lib/ltsm -g ltsm-shm -G ltsm-auth,ltsm-dbus \
        -l -m -N -r -s /sbin/nologin ltsm-xvfb
```

8 Настройка сервиса LTSM_service

Служба запускается с правами **root**, является менеджером **dbus** *ltsm.manager.service*. Регистрация в **dbus** осуществляется через создание файла */etc/dbus-1/system.d/ltsm.manager.service.conf*:

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus Configuration 1.0//EN"
        "http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
    <policy user="root">
        <allow own="ltsm.manager.service" />
        <allow send_destination="ltsm.manager.service" />
        <allow send_interface="ltsm.manager.service" />
    </policy>

    <policy group="ltsm-dbus">
        <allow send_destination="ltsm.manager.service" />
    </policy>
</busconfig>
```

Регистрация службы как **unit** для **systemd**, осуществляется через создание файла конфигурации */etc/system.d/ltsm_service.service:*

```
[Unit]
Description=LTSM service

[Service]
ExecStart=/usr/sbin/LTSM_service

[Install]
WantedBy=multi-user.target
```

Команда на регистрацию и запуск сервиса:

```
systemctl enable ltsm_service.service
systemctl start ltsm_service.service
```

Факт запуска сервиса отражен в системном журнале, для просмотра воспользуйтесь командой:

```
journalctl -t ltsm_service
```

Файл конфигурации сервиса: /etc/ltsm/config.json

9 Настройка коннектора LTSM_connector

Служба запускается с правами пользователя **ltxm-xvfb**, является клиентом **dbus** *ltsm.manager.service.*

Регистрация службы как **unit** для **systemd**, осуществляется через создание файла конфигурации */etc/system.d/ltsm_connector@.service:*

```
[Socket]
ListenStream=3389
ListenStream=5900
Accept=yes

[Install]
WantedBy=sockets.target
```

*Регистрация службы как **socket** для **systemd**, осуществляется через создание файла конфигурации /etc/system.d/ltsm_connector.socket:*

```
[Unit]
Description=LTSM connector
After=network.target ltsm_service.service

[Service]
Type=simple
ExecStart=/usr/local/sbin/LTSM_connector --type auto
User=ltsm-xvfb
Group=ltsm-shm
StandardInput=socket
StandardOutput=socket

[Install]
WantedBy=multi-user.target
```

Команда на регистрацию и запуск сервиса:

```
systemctl enable ltsm_connector.socket
systemctl start ltsm_connector.socket
```

На каждое клиентское подключение запускается отдельный **LTSM_connector**
Коннектор поддерживает шифрование трафика через системную библиотеку **gnutls**, по умолчанию используется протокол шифрования **TLS1.2**. С версии **gnutls-3.6.3** поддерживает работу с сертификатами и ключами **ГОСТ**.

Факт запуска коннектора отражен в системном журнале, для просмотра воспользуйтесь командой:

```
journalctl -t ltsm_connector
```

Файл конфигурации сервиса: /etc/ltsm/config.json

10 Настройка дополнительных параметров безопасности

Аутентификация и авторизация пользователей осуществляется через системный механизм **PAM**, рекомендуемый сервисный конфигурационный файл */etc/pam.d/ltsm (пример для **CentOS7**)*:

```
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       substack     system-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth

# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_console.so

# pam_selinux.so open should only be followed by sessions to be executed in the user
context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    include      system-auth
session    include      postlogin
```

Ограничение ресурсов для сессий пользователей реализуется через системные возможности **pam_limits**.
Пример ограничения системных ресурсов на группу пользователей:

```
@users hard   nproc         30
@users hard   priority      10
@users hard   as            262144
@users hard   nofile        128
```

Ограничение на количество процессов 30, приоритет nice 10, максимальное адресное пространство 256Мб, максимальное количество открытых файлов 128

11 Настройка дополнительных ресурсов

Информация о подключении пользователя отражается в файле *$HOME/.ltsm/conninfo*:

```
LTSM_REMOTEADDR=192.168.100.101
LTSM_TYPECONN=vnc
```

На основании этой информации, можно автоматически организовать подключение дополнительных ресурсов, принтеров через **CUPS**, звук через **pulse-audio**, достаточно воспользоваться механизмом *autostart* для выбранного Window Manager.

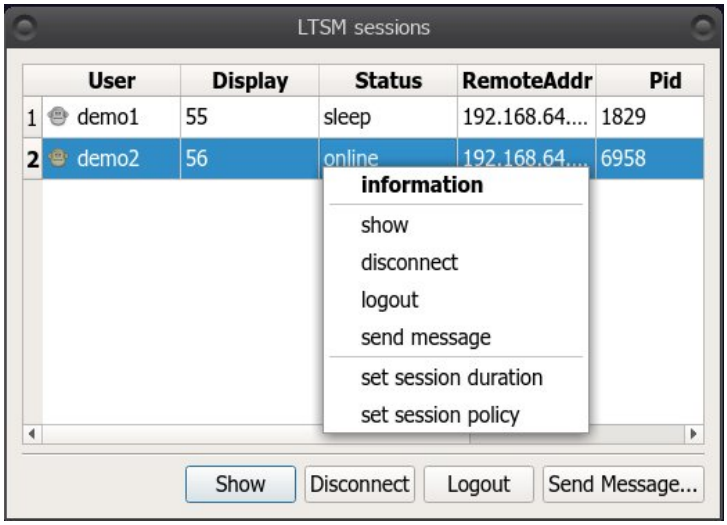
Так же при **реконнекте** в рабочую сессию необходимо отслеживать эти изменения, для этого можно воспользоваться утилитами **inotify-tools**.

При отправке администратором системного сообщения создается файл в каталоге *$HOME/.ltsm/messages/*. Используя штатные утилиты **zenity** и **inotify-tools** можно организовать вывод сообщений в любом графическом окружении.

12 Утилита администрирования LTSM_sessions

Для корректной работы администратору необходим доступ в группы **ltsm-dbus, ltsm-auth**.
Реализован следующий функционал:

- Отключение коннекта, пользовательская сессия переводится в статус **offline**
- Выключение сеанса, пользовательская сессия удаляется
- Отправка системного сообщения
- Подключение к сессиям пользователей через дополнительную программу **LTSM_sdl2x11**
- Изменение параметров выбранной сессии (политику подключения и время жизни сессии)



Политики авторизации

authlock - при успешной авторизации, если сессия используется уже с другого адреса, вы получите ошибку: **Session busy, from ipaddress X.X.X.X**
authtake - при успешной авторизации предыдущий коннект закрывается
authshare - разрешен доступ к экрану для множественных коннектов

13 Описание всех параметров конфигурационного файла

параметр	тип	описание
service:debug	*string*	уровень отладки служебных сообщений программы service
connector:debug	*string*	уровень отладки служебных сообщений программы connector
logging:facility	*string*	служебный подраздел для службы журналирования
vnc:encoding:debug	*integer*	уровень отладки VNC encodings
vnc:encoding:threads	*integer*	количество используемых ядер процессора для кодировки VNC
vnc:encoding:blacklist	*list*	черный список неиспользуемых типов кодировки VNC
vnc:gnutls:disable	*boolean*	запретить использование шифрования
vnc:gnutls:priority	*string*	настройка для специфичного типа кодирования
vnc:gnutls:debug	*integer*	уровень отладки gnutls
vnc:gnutls:anonmode	*boolean*	режим работы TLS1.2 анонимный туннель (true) или x509
vnc:gnults:cafile	*string*	параметр для x509 режима
vnc:gnutls:crlfile	*string*	параметр для x509 режима
vnc:gnutls:certfile	*string*	параметр для x509 режима
vnc:gnutls:keyfile	*string*	параметр для x509 режима
rdp:wlog:level	*string*	уровень отладочных сообщений для протокола rdp trace,debug,info,warn,error,fatal,off (по умолчанию "error")
rdp:security:rdp	*boolean*	поддержка уровня безопасности RDP (по умолчанию true)
rdp:security:tls	*boolean*	поддержка уровня безопасности TLS (по умолчанию true)
rdp:security:nla	*boolean*	поддержка уровня безопасности NLA (по умолчанию false)
rdp:tls:level	*integer*	1,2,3,4,5 (see SSL_get_security_level) (по умолчанию 1)
rdp:encryption:level	*string*	compatible, higth, low, fips (по умолчанию "compatible")
rdp:server:certfile	*string*	параметры x509 для коннектора rdp
rdp:server:keyfile	*string*	параметры x509 для коннектора rdp
group:shm	*string*	служебная группа (ltsm-shm)
group:auth	*string*	служебная группа (ltsm-auth)
user:xvfb	*string*	служебный пользователь (ltsm-xvfb)
pam:service	*string*	наименование сервиса PAM (ltsm)
access:group	*string*	фильтр списка логинов по группе
access:uid:min	*int*	фильтр по нижней границе uid для списка логинов
access:uid:max	*int*	фильтр по верхней границе uid для списка логинов
access:users	*list*	фильтр по списку разрешенных логинов
helper:autocomplete	*boolean*	использовать автодополнение ввода при логине
helper:idletimeout	*integer*	количество секунд бездействия экрана авторизации
helper:path	*string*	путь до программы (/usr/libexec/ltsm/LTSM_helper)
helper:args	*string*	аргументы для программы
helper:dateformat	*string*	формат даты в стиле Qt

helper:title	*string*	заголовок окна
login:failure_count	*integer*	количество неудачных попыток авторизации
session:duration_max_sec	*integer*	длительность основной сессии пользователя (в сек)
session:path	*string*	путь скрипта на запуск предпочитаемой сессии пользователя (/etc/ltsm/xsessions)
session:policy	*string*	базовая политика авторизации в сессию пользователя
informer:path	*string*	утилита диалога (/usr/bin/zenity)
informer:args	*string*	дополнительные параметры запуска: "--info --no-wrap --text %{msg}"
default:width	*integer*	ширина экрана по умолчанию
default:height	*integer*	высота экрана по умолчанию
display:min	*integer*	нижняя граница разрешенных displays
display:max	*integer*	верхняя граница разрешенных displays
display:solid	*integer*	предпочитаемый цвет фона в формате RGB
xauth:path	*string*	путь до программы (/usr/bin/xauth)
xauth:args	*string*	дополнительные аргументы запуска: "-f %{authfile} add :%{display} \".\" %{mcookie}"
xauth:file	*string*	формат файла авторизации X11 сессии ("/var/run/ltsm/auth_%{display}") возможные переменные подстановки: %{pid}, %{remoteaddr}, %{display}
mcookie:path	*string*	путь до программы (/usr/bin/mcookie)
xvfb:path	*string*	путь до программы (/usr/bin/Xvfb)
xvfb:args	*string*	дополнительные аргументы запуска
xvfb:socket	*string*	формат файла сокета ("/tmp/.X11-unix/X%{display}")
xvfb:lock	*string*	формат файла блокировки Xvfb ("/tmp/.X%{display}-lock")
session:connect	*string*	скрипт выполняется с правами пользователя при коннекте в сессию, (возможные переменные подстановки: %{display}, %{user})
session:disconnect	*string*	скрипт выполняется с правами пользователя при отключении сеанса, (возможные переменные подстановки: %{display}, %{user})
system:logon	*string*	скрипт выполняется с системными правами, при входе пользователя в новую сессию, (рекомендуемое значение: "/usr/bin/sessreg -a -l :%{display} %{user}")
system:logoff	*string*	скрипт выполняется с системными правами, при выходе пользователя из рабочей сессии, (рекомендуемое значение: "/usr/bin/sessreg -d -l :%{display} %{user}")
system:connect	*string*	скрипт выполняется с системными правами, при коннекте в сессию, (переменные подстановки %{display}, %{user})
system:disconnect	*string*	скрипт выполняется с системными правами, при отключении сеанса, (переменные подстановки %{display}, %{user})