

Автоматическое монтирование USB-накопителей

- [Перехват события udev](#)
- [Вызов сценария обработки события как системного сервиса](#)
- [Сценарий обработки события](#)
- [Приёмы отладки](#)

В современных версиях Astra Linux для управления подключаемыми устройствами используется механизм [udev](#)

Источник: <https://serverfault.com/questions/766506/automount-usb-drives-with-systemd>



Данная статья применима к:

- ОС ОН Орёл 2.12;
- ОС СН Смоленск 1.6;
- ОС СН Ленинград 8.1.

Перехват события udev

События udev возникают при изменении статуса подключенных устройств. Наиболее употребительные события:

- подключение устройства (событие "add");
- отключение устройства (событие "remove").

Перехват событий осуществляется с помощью сценариев обработки. Файлы со сценариями - обработчиками событий udev располагаются в каталогах:

- `/lib/udev/rules.d/`
- `/run/udev/rules.d/`
- `/etc/udev/rules.d/`

Каталоги обрабатываются в том порядке, в котором они перечислены. Перед выполнением файлы упорядочиваются по алфавиту. Файлы с одинаковыми именами - переписываются последним найденным файлом, т.е. файл, найденный в последнем каталоге (`/etc/udev/rules.d/`) заменит собой ранее найденный файл.

Стандартно имя каждого файла - сценария начинается с двух цифр, и имеет расширение `.rules`.

Пример файла перехвата события `/etc/udev/rules.d/99-local.rules`:



`/etc/udev/rules.d/99-local.rules`

```
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="add", RUN+="/bin/systemctl start usb-mount@%k.service"
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="remove", RUN+="/bin/systemctl stop usb-mount@%k.service"
```

Этот перехватчик обрабатывает события подключения/отключения дисковых устройств с именами начинающимися с букв "sd", после которых следует одна любая строчная буква ([a-z]), после которой следует одна цифра ([0-9]).

Однако сам перехватчик не выполняет при этом прямых действий, а вызывает для выполнения этих действия системную службу `usb-mount@%k.service`, то есть вызывает сценарий обработки события как системную службу.

Обратите внимание, на имя службы `usb-mount@%k`, которое имеет особый смысл:

- при выполнении правила обработки события служба udev вместо переменной `%k` автоматически подставит имя устройства (полный список переменных имеется в документации).
Т.е. при подключении устройства, например, `/dev/sdb1` будет выполняться команда `/bin/systemctl start usb-mount@sdb1.service`;
- при вызове службы, в имени которой содержится символ "@" системная служба вызова служб разберёт это имя на части, и передаст часть, находящуюся после символ а "@" как параметр вызываемой службе.
Т.е. вызов `systemctl start usb-mount@sdb1.service` превратится в вызов службы `usb-mount` с параметрами `start` и `sdb1`.

Как организовать обработку этого вызова и саму службу описано ниже.

Для того, чтобы новые правила обработки были зарегистрированы системой и начал работать, необходимо перезагрузить правила udev командой

```
udevadm control --reload-rules
```

Вызов сценария обработки события как системного сервиса

Итак, сценарий обработки вызывается как системная служба. Для вызова системных служб используются так называемые "юниты", специальные сценарии запуска служб, расположенные в каталоге `/etc/systemd/system/`.

Пример обработчика вызова службы для вышеуказанного правила перехвата события udev разместим в файле `/etc/systemd/system/usb-mount@.service`:

```
i /etc/systemd/system/usb-mount@.service
[Unit]
Description=Mount USB Drive on %i
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=/usr/local/bin/usb-mount.sh add %i
ExecStop=/usr/local/bin/usb-mount.sh remove %i
```

Этот сценарий умеет обрабатывать две команды - start и stop, но сам опять ничего не делает, а вызывает исполнимый файл сценария обработки события `/usr/local/bin/usb-mount.sh`.

Обратите внимание на параметр `%i` - вместо него будет автоматически подставлена часть имени вызова службы, находящаяся после символа `"@"`.

Сценарий обработки события

Сценарий обработки события в принципе может размещаться где угодно, для примера используем файл `/usr/local/bin/usb-mount.sh`

```
i /usr/local/bin/usb-mount.sh
#!/bin/bash
# / .
usage() {
    echo ": $0 {add|remove} device_name (, sdb1)"
    exit 1
}

if [[ $# -ne 2 ]]; then
    usage
fi

ACTION=$1
DEVBASE=$2
DEVICE="/dev/${DEVBASE}"

# ,
MOUNT_POINT=$(/bin/mount | /bin/grep ${DEVICE} | /usr/bin/awk '{ print $3 }')

do_mount() {
    if [[ -n ${MOUNT_POINT} ]]; then
        echo ": ${DEVICE}    ${MOUNT_POINT}"
        exit 1
    fi
}

# : $ID_FS_LABEL, $ID_FS_UUID, $ID_FS_TYPE
eval $(/sbin/blkid -o udev ${DEVICE})

# :
LABEL=${ID_FS_LABEL}
if [[ -z "${LABEL}" ]]; then
    LABEL=${DEVBASE}
elif /bin/grep -q " /media/${LABEL} " /etc/mtab; then
# :
    LABEL+="-${DEVBASE}"
fi
MOUNT_POINT="/media/${LABEL}"
echo " : ${MOUNT_POINT}"
/bin/mkdir -p ${MOUNT_POINT}

#
OPTS="rw,relatime"

# :
if [[ ${ID_FS_TYPE} == "vfat" ]]; then
    OPTS+=",users,gid=100,umask=000,shortname=mixed,utf8=1,flush"
fi
```

```

if ! /bin/mount -o ${OPTS} ${DEVICE} ${MOUNT_POINT}; then
    echo " ${DEVICE} ( = $?)"
    /bin/rmdir ${MOUNT_POINT}
    exit 1
fi

echo "**** ${DEVICE} ${MOUNT_POINT} ****"
}

do_unmount() {
    if [[ -z ${MOUNT_POINT} ]]; then
        echo ": ${DEVICE} "
    else
        /bin/umount -l ${DEVICE}
        echo "**** ${DEVICE}"
    fi
}

#
for f in /media/* ; do
    if [[ -n $(/usr/bin/find "$f" -maxdepth 0 -type d -empty) ]]; then
        if ! /bin/grep -q " $f " /etc/mtab; then
            echo "**** $f"
            /bin/rmdir "$f"
        fi
    fi
done

case "${ACTION}" in
    add) do_mount ;;
    remove) do_unmount ;;
    *) usage ;;
esac


```

После создания файла сценария не забыть сделать его исполнимым:

```
chmod +x /usr/local/bin/usb-mount.sh
```

Приёмы отладки

Включение вывода отладочных сообщений в файл /var/log/syslog

```
 udevadm control -l debug
```

Тестовая отработка правил udev без их загрузки:

```
udevadm test /dev/sdb1
```

Мониторинг событий udev:

```
udevadm monitor -k -u -p
```

Путь к устройству:

```
udevadm info -q path -n /dev/sdd1
```

Полная информация об устройстве:

```
udevadm info -a -p $(udevadm info -q path -n /dev/sdd1)
```