

Для начала нам нужен стенд на котором мы будем собирать образ системы. Для стенда подойдет любой дистрибутив. В нашем случае это виртуальная машина с минимальным набором ПО к которой есть доступ по ssh.

Создаем основной скрипт создания iso образа.

```
cat /root/make_auto_iso.sh
```

```
#!/bin/bash
isobuilder='/mnt/isobuilder/'
isodist='/exports/repo/iso/alse/alse-1.8.1.6-27.06.2024_14.12-di.iso'
mountpoint='/mnt/isobuilder/isomntpoint/'
irmod='/mnt/isobuilder/irmod'
cddir='/mnt/isobuilder/cd/'
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[0;33m'
#### Проверки наличия ####

if [ ! -f ${isodist} ]; then
    echo "ISO образ не найден по указанному пути"
else
    echo "ISO образ ${isodist} найден"
fi

if [ ! -d ${mountpoint} ]; then
    echo "Точка монтирования не найдена по указанному пути"
    echo "Создаю точку монтирования ${mountpoint}"
    mkdir ${mountpoint} && echo " Точка монтирования ${mountpoint} создана"
else
    echo "Точка монтирования ${mountpoint} уже существует"
fi

echo " Монтирую ISO образ ${isodist} в точку монтирования  ${mountpoint}"
sudo mount -o loop ${isodist} ${mountpoint} && echo "ISO образ ${isodist}
успешно смонтирован в точку монтирования  ${mountpoint}"

if [ ! -d ${irmod} ]; then
    echo "Папка irmod не найдена по указанному пути"
    echo "Создаю папку ${irmod}"
    mkdir ${irmod} && echo " Папка ${irmod} создана"
else
    echo "Папка ${irmod} уже существует"
fi

if [ ! -d ${cddir} ]; then
    echo "Папка cd не найдена по указанному пути"
    echo "Создаю папку ${cddir}"
    mkdir ${cddir} && echo " Папка ${cddir} создана"
else
    echo "Папка ${cddir} уже существует"
```

fi

```
echo -e "Синхронизирую содержимое смонтированного диска в папку ${cddir}
${YELLOW}"
rsync -a -v -H --exclude=TRANS.TBL /mnt/isobuilder/isomntpoint/
/mnt/isobuilder/cd
tput sgr0

echo -e "* Отмонтирую ${mountpoint} $GREEN"
umount ${mountpoint} && echo "* Отмонтирован ${mountpoint} успешно"
echo "* Удаляю ${mountpoint}"
rm -rf ${mountpoint} && echo "* Папка ${mountpoint} удалена успешно"

cd ${irmod} && echo "* Перехожу в папку ${irmod} - успешно"
gzip -d < ../cd/install.amd/initrd.gz | cpio --extract --make-directories --
no-absolute-filenames && echo "* Распаковываю initrd.gz - успешно"
cp -f ../preseed.cfg preseed.cfg && echo "* Копирую файл preseed.cfg -
успешно"

find . | sudo cpio -H newc --create | sudo gzip -9 >
../cd/install.amd/initrd.gz && echo "* Запаковываю образ initrd.gz -
успешно"
cd ${cddir} && echo "* Перехожу в директорию ${cddir} - успешно"
pwd
cp -f ../txt.cfg ./isolinux/ && echo "* Копирую кастомное текстовое меню в
папку ./isolinux/txt.cfg - успешно"
cp -f ../menu.cfg ./isolinux/ && echo "* Копирую кастомное текстовое меню в
папку ./isolinux/txt.cfg - успешно"
mkdir ./extra/ && echo "* Создаю папку extra - успешно"
cp -f ../postinst.sh ./extra && echo "* Копирую файл postinst.sh extra
- успешно"
cp -f ../first_start.sh ./extra && echo "* Копирую файл first_start.sh
extra - успешно"
cp -f ../firstinit.sh ./extra && echo "* Копирую файл first_init.sh extra
- успешно"
md5sum `find -follow -type f` > ./md5sum.txt && echo "* Пересчитываю MD5sum
- успешно"
cd ${isobuilder} && echo "* Перехожу в директорию ${isobuilder} - успешно"
echo "* Начинаю схлопывать образ в output.iso "
genisoimage -o ./output.iso -r -J -no-emul-boot -boot-load-size 4 -boot-
info-table -b isolinux/isolinux.bin -c isolinux/boot.cat ./cd && echo -e "*
Схлопывание образа завершено - успешно"

if [ ! -f ./output.iso ]; then
    echo "ISO образ не найден по указанному пути"
else
    echo -e "${GREEN}ISO образ output.iso найден - успешно"
fi

mv -f /mnt/isobuilder/output.iso /exports/repo/iso/alse/ && echo "* Переношу
```

```
файл образа в папку /exports/repo/iso/alse/ - успешно"
```

```
rm -rf ${cddir}
rm -rf ${irmod}
tput sgr0
```

2.Создадим директорию где будем собирать iso образ

```
mkdir /mnt/isobuilder
```

3. Содержимое директории:

```
ls mnt/iso/builder/
firstinit.sh first_start.sh menu.cfg postinst.sh preseed.cfg
preseed.cfg.save preseed.cfg.save.1 txt.cfg
```

```
[username@computername ~]# cat /mnt/isobuilder/preseed.cfg
```

```
# настройка зеркала с пакетами
d-i mirror/protocol string http
d-i mirror/country string manual
d-i mirror/http/hostname string repo.yourdomain.ru
d-i mirror/http/directory string /repo/dist/alse-1.8/
d-i mirror/http/proxy string

# Preseeding only locale sets language, country and locale.
d-i debian-installer/locale string en_US

d-i netcfg/get_hostname string ws-01
d-i netcfg/get_domain string yourdomain.ru

astra-license astra-license/license boolean true
krb5-config krb5-config/kerberos_servers string
libnss-ldapd libnss-ldapd/ldap-base string
libnss-ldapd libnss-ldapd/ldap-uris string
libnss-ldapd libnss-ldapd/nsswitch multiselect services
ald-client ald-client/make_config boolean false
ald-client ald-client/manual_configure note
tasksel tasksel/astra-feat-setup multiselect

# Настройки клавиатуры
d-i console-tools/archs select at
d-i console-keymaps-at/keymap select ru
d-i keyboard-configuration/toggle select Alt+Shift
d-i keyboard-configuration/layout select Russian

d-i keyboard-configuration/variantcode string
d-i keyboard-configuration/variant select Russian
d-i keyboard-configuration/layoutcode string ru
d-i keyboard-configuration/xkb-keymap select ru
```

```
d-i languagechooser/language-name-fb select Russian
d-i countrychooser/country-name select Russia
d-i console-setup/fontface select Terminus
d-i console-setup/ask_detect boolean false
d-i console-setup/layoutcode string ru

# подключение репозиторийев
d-i apt-setup/non-free boolean true
d-i apt-setup/contrib boolean true
apt-mirror-setup          apt-setup/use_mirror      boolean false
unknown apt-setup/use_mirror      boolean false

# выключить показ диалога с WEP ключом.
d-i netcfg/wireless_wep string

# Network interface.
# Automatic network configuration is the default.
# If you prefer to configure the network manually, uncomment this line and
# the static network configuration below.
d-i netcfg/disable_autoconfig boolean true
d-i netcfg/disable_dhcp boolean true

# Static network configuration.
# IPv4
# Pick a particular interface instead (eth0):
# d-i netcfg/choose_interface select eth0

# netcfg выберет активный сетевой интерфейс, если такой имеется. Если
# есть более одного интерфейса, то при данной настройке список
# интерфейсов выводится не будет.
d-i netcfg/choose_interface select auto
d-i netcfg/get_ipaddress string 10.88.121.82
d-i netcfg/get_netmask string 255.255.255.0
d-i netcfg/get_gateway string 10.88.121.32
d-i netcfg/get_nameservers string 10.88.40.141 10.88.40.142
d-i netcfg/confirm_static boolean true

# задаёт, установлены или нет аппаратные часы по Гринвичу.
d-i clock-setup/utc boolean true

# настройка timezone
d-i time/zone string Europe/Moscow

# определяет, нужно ли использовать NTP для установки часов во время установки
d-i clock-setup/ntp boolean false

# разбиение диска создание разделов

#d-i partman-auto/disk string /dev/vda
# Auto select first disk.
d-i partman/early_command string debconf-set partman-auto/disk "$(list-
```

```

devices disk | head -n1)"
d-i partman-auto/method string lvm
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-lvm/device_remove_lvm_span boolean true
d-i partman-lvm/confirm boolean true
d-i partman-auto/purge_lvm_from_device boolean true
d-i partman-auto-lvm/new_vg_name string vgsys
d-i partman-auto-lvm/guided_size string max
d-i partman-basicfilesystems/choose_label string gpt
d-i partman-basicfilesystems/default_label string gpt
d-i partman-partitioning/choose_label string gpt
d-i partman-partitioning/default_label string gpt
d-i partman/choose_label string gpt
d-i partman/default_label string gpt
# d-i partman-auto/choose_recipe select          custom-lvm
d-i partman-auto/expert_recipe string custom-lvm :: \

```

```

    1 8 1 free \
        $gptonly{ } \
        $primary{ } \
        $bios_boot{ } \
        method{ biosgrub } \
    . \
    512 1 512 ext4 \
        $gptonly{ } \
        $primary{ } \
        $bootable{ } \
        method{ format } format{ } \
        use_filesystem{ } filesystem{ ext4 } \
        label{ BOOT } \
        mountpoint{ /boot } \
    . \
    9096 2 15148 xfs \
        $lvmok{ } \
        lv_name{ root } \
        method{ format } format{ } \
        use_filesystem{ } filesystem{ xfs } \
        label{ R00T } \
        mountpoint{ / } \
    . \
    4096 3 4096 xfs \
        $lvmok{ } \
        lv_name{ var } \
        method{ format } format{ } \
        use_filesystem{ } filesystem{ xfs } \
        label{ VAR } \
        mountpoint{ /var } \
    . \
    2048 4 2048 xfs \
        $lvmok{ } \
        lv_name{ journal } \

```

```

        method{ format } format{ } \
        use_filesystem{ } filesystem{ xfs } \
        label{ JOURNAL } \
        mountpoint{ /var/log/journal } \
    . \
512 5 512 xfs \
    $lvmok{ } \
    lv_name{ ossec } \
    method{ format } format{ } \
    use_filesystem{ } filesystem{ xfs } \
    label{ OSSEC } \
    mountpoint{ /var/ossec } \
    . \
    2048 6 4096 xfs \
    $lvmok{ } \
    lv_name{ tmp } \
    method{ format } format{ } \
    use_filesystem{ } filesystem{ xfs } \
    label{ TMP } \
    mountpoint{ /tmp } \
    . \
    2048 7 2048 linux-swap \
    $lvmok{ } \
    lv_name{ swap } \
    method{ swap } format{ } \
    label{ SWAP1 } . \
    2048 70 2048 xfs $lvmok{ } lv_name{ home } method{ format } format{ } \
    use_filesystem{ } filesystem{ xfs } label{ HOME } mountpoint{
/home } . \
    1 100 -1 xfs $lvmok{ } lv_name{ deleteme } \
    method{ format } format{ } \
    use_filesystem{ } filesystem{ xfs } \
    label{ DELETEME } \
    mountpoint{ /tmp/deleteme } \
    .

d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman/confirm_write_new_label boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# устанавливаемый пакет (мета) с образом ядра; можно указать «none»,
# если ядро устанавливать не нужно.
d-i base-installer/kernel/image string linux-image-generic
#bootstrap-base base-installer/kernel/image select linux-6.1-generic
bootstrap-base base-installer/kernel/image select linux-6.6-generic
d-i passwd/make-user boolean true

```

```
# пароль суперпользователя, любой открытым текстом
d-i passwd/root-password password 1Jd3x6tyBc
d-i passwd/root-password-again password 1Jd3x6tyBc

# создать учётную запись обычного пользователя.
d-i passwd/user-fullname string ladmin
d-i passwd/username string ladmin

# пароль обычного пользователя, или открытым текстом
d-i passwd/user-password password XpLL17YP9
d-i passwd/user-password-again password XpLL17YP9

console-setup console-setup/fontface select Terminus

# Вы можете указать нужно ли устанавливать non-free и contrib ПО.
d-i apt-setup/non-free boolean true
d-i apt-setup/contrib boolean true
d-i apt-setup/services-select multi-select
d-i apt-setup/security_host string
d-i apt-setup/volatile_host string

# По умолчанию программа установки требует, чтобы репозитории
# аутентифицировались с помощью известного ключа gpg.
# Этот параметр выключает данную аутентификацию.
# Предупреждение: это небезопасно, рекомендуется не делать этого.
d-i debian-installer/allow_unauthenticated string true

## установка базовой системы + рабочий стол Fly
tasksel tasksel/first multiselect Base, Fly

# Установка только базовой системы, без Fly.
#tasksel tasksel/first multiselect Base
#tasksel tasksel/first multiselect Base, Fly
# отдельные дополнительные пакеты для установки
#d-i pkgsel/include string openssh-server
d-i pkgsel/include string openssh-server network-manager autofs keyutils
libnfsidmap2 nfs-common rpcbind

# специфичные настройки для Astra Linux
astra-license astra-license/license boolean true
krb5-config krb5-config/kerberos_servers string
libnss-ldapd libnss-ldapd/ldap-base string
libnss-ldapd libnss-ldapd/ldap-uris string
libnss-ldapd libnss-ldapd/nsswitch multiselect services
ald-client ald-client/make_config boolean false
ald-client ald-client/manual_configure note
# НЕ настраивать киоск

#Выберите уровень защищенности
d-i astra-additional-setup/os-check select Maximum security level
Smolensk
```

```
# Дополнительные настройки ОС
d-i astra-additional-setup/additional-settings-voronezh multiselect
Enable Mandatory Integrity Control, Disable ptrace capability, Request
password for sudo command
d-i astra-additional-setup/additional-settings-smolensk multiselect
Enable Mandatory Integrity Control, Enable Mandatory Access Control, Disable
ptrace capability, Request password for sudo command
# Программный RAID недоступен

#astra-feat-setup astra-feat-setup/feat multiselect kiosk mode
tasksel tasksel/astra-feat-setup multiselect

d-i console-cyrillic/toggle select Caps Lock

# некоторые версии программы установки могут отсылать отчёт
# об установленных пакетах. По умолчанию данная возможность
# выключена, но отправка отчёта помогает проекту
# определить популярность программ и какие из них включать на CD.
popularity-contest popularity-contest/participate boolean false
apt-cdrom-setup apt-setup/cdrom/set-first boolean false
unknown apt-setup/cdrom/set-first boolean false

# эту переменную можно безопасно устанавливать, она указывает, что grub
# должен устанавливаться в MBR,
# если на машине не найдено другой операционной системы.
d-i grub-installer/only_debian boolean true

# это заставляет grub-installer устанавливать в MBR, даже если на машине
# есть другая ОС, что менее безопасно, так как может привести к отказу
# загрузки этой другой ОС.
d-i grub-installer/with_other_os boolean true
d-i grub-installer/password password 1Jd3x6t3yBc
d-i grub-installer/password-again password 1Jd3x6t3yBc
grub-installer grub-installer/password-mismatch error

# остановить систему после завершения установки, а
# не перегружаться в установленную систему.
#d-i preseed/late_command "init 6"
#d-i debian-installer/exit/halt boolean true
d-i finish-install/reboot_in_progress note
#d-i finish-install/exit/poweroff boolean true

d-i preseed/late_command string cp -R /cdrom/extra/ /target/root/; \
cp /var/log/syslog /target/root/; \
cd /target; \
chmod +x /target/root/extra/postinst.sh; \
cp /target/root/extra/first_start.sh /target/usr/local/bin/first_start.sh; \
chmod +x /target/usr/local/bin/first_start.sh; \
in-target /bin/bash /root/extra/postinst.sh;
```

```
#d-i preseed/late_command string \  
# in-target gawk -i inplace '!/deleteme/' /etc/fstab; \  
# in-target mkdir '/etc/ssh/authkeys'; \  
# in-target mkdir -p /root/.ssh; \  
# in-target chown -R root:root /root/.ssh; \  
# in-target chmod 700 /root/.ssh; \  
# in-target systemctl enable ssh.service; \  
# in-target cp -R /cdrom/extra/ /root/; \  
# in-target apt-get install tree; \  
# in-target apt-get install atop; \  
# in-target apt-get install htop; \  
  
# in-target lvremove -f -y /dev/vgsys/deleteme; \  
# in-target umount /tmp/deleteme ; \  

```

Содержиоме файла menu.cfg

cat /mnt/isobuilder/menu.cfg

```
menu hshift 13  
menu width 49  
  
menu title Installer boot menu  
include stdmenu.cfg  
include txt.cfg  
include adtxt.cfg  
menu end  
config prompt.cfg
```

cat /mnt/isobuilder/txt.cfg

```
default install  
label install  
    menu label ^Installation  
    menu default  
    kernel /install.amd/vmlinuz  
    append initrd=/install.amd/initrd.gz vga=788 quiet  
modprobe.blacklist=evbug file=/preseed.cfg astra_install=1 nomodeset  
netcfg/disable_autoconfig=false net.ifnames=0 hostname=installation-  
stage.somedomain.ru console-setup/ask_detect=false nomodeset auto=true  
priority=critical debian-installer/allow-unauthenticated=true debian-  
installer/locale=en_US console-keymaps-at/keymap=ru astra-  
license/license=true localechooser/translation/warn-light=true  
localechooser/translation/warn-severe=true domain=service.ru locale=en_US
```

cat /mnt/isobuilder/postinst.sh

```
#!/bin/bash  
echo "deb [arch=amd64]  
https://repo.yourdomain.ru/repo/astra/stable/1.8_x86-64/repository-main/
```

```
1.8_x86-64 main contrib non-free non-free-firmware" > /etc/apt/sources.list
echo "deb [arch=amd64]
https://repo.yourdomain.ru/repo/astra/stable/1.8_x86-64/repository-extended/
1.8_x86-64 main contrib non-free non-free-firmware" >> /etc/apt/sources.list
echo "deb [trusted=yes]
https://repo.yourdomain.ru/repo/alldpro/frozen/01/2.3.0/ 1.7_x86-64 main
base" >> /etc/apt/sources.list
echo "deb [trusted=yes]
https://repo.yourdomain.ru/repo/alldpro/stable/repository-extended/ generic
main" >> /etc/apt/sources.list
echo "# deb [arch=amd64 trusted=yes signed-by=/etc/apt/keyrings/GPG-KEY-
POSTGRESPRO.gpg ]
https://repo.yourdomain.ru/repo/postgrespro/std-16/astra-smolensk/1.8
1.8_x86-64 main" >> /etc/apt/sources.list
echo "# deb [trusted=yes arch=amd64 signed-by=/etc/apt/keyrings/zabbix-
official-repo-apr2024.gpg] https://repo.yourdomain.ru/repo/zabbix/7.0/debian
bookworm main" >> /etc/apt/sources.list
echo "# deb [arch=amd64 signed-by=/etc/apt/keyrings/debian-archive-
key-12.gpg] http://deb.debian.org/debian bookworm main contrib non-free non-
free-firmware" >> /etc/apt/sources.list
apt update
apt install -y mc telnet
touch /etc/rc.local
echo "#!/bin/sh -e" >>/etc/rc.local
echo "/usr/local/bin/first_start.sh" >>/etc/rc.local
echo "exit 0" >>/etc/rc.local
chmod +x /etc/rc.local
chmod +x /usr/local/bin/first_start.sh
systemctl enable rc-local.service
systemctl enable ssh.service
```

cat /mnt/isobuilder/first_start.sh

```
#!/bin/bash
touch /root/first_start.log
exec &> /root/first_start.log
exec > >(tee /dev/tty2 /var/log/my_logs)
DEBIAN_FRONTEND='noninteractive' apt-get -y -o Dpkg::Options::='--force-
confdef' -o Dpkg::Options::='--force-confold' dist-upgrade -y && echo "*"
Обновление системы прошло - успешно"
hostname="pobeda-ws$(hostnamectl | grep Machine | awk '{print $3}'|
hostnamectl | grep Machine | awk '{print $3}'| tr -cd [:digit:]| cut -b
1,3,5,7)"
hostnamectl set-hostname ${hostname} --static && echo "*" Назначаю имя
хоста ${hostname} - успешно"
cat /var/log/syslog >> /root/syslog && echo "*" Записываю лог загрузки к логу
инсталляции в файл /root/syslog - успешно"
cd /root && echo "*" Перехожу в каталог /root - успешно"
echo "Скачиваю архив jc из репозитория"
echo "wget https://repo.yordomain.ru/repo/local/app/pip3_jc_alse18.tar.gz"
wget "https://repo.yourdomain.ru/repo/local/app/pip3_jc_alse18.tar.gz"
```

```
if [ ! -f ./pip3_jc_alse18.tar.gz ]; then
    echo "*"      Файл pip3_jc_alse18.tar.gz не найден по указанному пути"

else
    echo -e "*"      Файл pip3_jc_alse18.tar.gz скачан и найден - успешно"
fi
echo "Распаковываю скачанный архив."
tar zxv -f /root/pip3_jc_alse18.tar.gz && echo "*"      Распаковка скачанного
архива прошла - успешно "
echo "Начинаю установку пакетов jq python3-pip python3-venv"
apt install jq python3-pip python3-venv -y && echo "*"      Установка пакетов
прошла - успешно"
python3 -m venv .venv && echo "*"      Создано виртуальное окружение для python -
успешно"
source .venv/bin/activate && echo "*"      Загружено виртуальное окружение python
- успешно "
echo "Устанавливаю jc"
python3 -m pip install --no-index --find-links=/root/pip3_jc_alse18/ jc &&
echo "*"      Установка jc прошла - успешно "
echo "<<<Готов приступить к сбору диагностических данных.>>>"
echo "Выгружаю данные Hostnamectl в формат json."
hostnamectl | jq -Rn '[inputs | capture("(?<key>[^:]*): *(?<value>.*)")]' |
from_entries' > hostnamectl.json
echo "Выгружаю данные lsusb в формат json."
lsusb | jc -p --lsusb > lsusb.json
echo "Выгружаю данные lsblk в формат json."
lsblk | jc -p --lsblk > lsblk.json
echo "Выгружаю данные df в формат json."
df -h | jc -p --df > df.json
echo "Выгружаю данные df в формат json."
netstat -tunlp | jc -p --netstat > netstat_tunlp.json
chmod +x /root/extra/firstinit.sh
source /root/extra/firstinit.sh && echo "*"      Выгружаю данные для firstinit
в формат json."
cd /root
cat /root/firstinit.json | curl --connect-timeout 2 --max-time 3 -X POST -H
'Content-type: application/json'
"https://repo.yourdomain.ru/svc/reg.php?${identity4}&${identity}&dt=firstini
t&dfmt=json" --data "$(</dev/stdin)"
ip --json address show | curl --connect-timeout 2 --max-time 3 -X POST -H
'Content-type: application/json'
"https://repo.yourdomain.ru/svc/reg.php?${identity4}&${identity}&dt=ipaddr&
fmt=json" --data "$(</dev/stdin)"
cat lsblk.json | curl --connect-timeout 2 --max-time 3 -X POST -H 'Content-
type: application/json'
"https://alxtrepo.sirena-travel.ru/svc/reg.php?${identity4}&${identity}&dt=l
sblk&dfmt=json" --data "$(</dev/stdin)"
cat hostnamectl.json | curl --connect-timeout 2 --max-time 3 -X POST -H
'Content-type: application/json'
"https://repo.yourdomain.ru/svc/reg.php?${identity4}&${identity}&dt=hostname
ctl&dfmt=json" --data "$(</dev/stdin)"
```

```
cat df.json |jq --raw-output | curl --connect-timeout 2 --max-time 3 -X POST
-H 'Content-type: application/json'
"https://repo.yourdomain.ru/svc/reg.php?${identity4}&${identity}&dt=df&dfmt=
json" --data "$(</dev/stdin)"
#rm -rf /usr/local/bin/first_start.sh && echo "*"      Удаляю файл
/usr/local/bin/first_start.sh - успешно"
systemctl disable rc-local.service && echo "*"      Отключаю сервис rc-local -
успешно"
rm -rf /etc/rc.local && echo"* Удаляю файл /etc/rc.local"
echo "*"      Вызываю init 6 - успешно"
init 6
```

From: <https://wiki.it-dream.ru/> - A&V Wiki

Permanent link: https://wiki.it-dream.ru/doku.php%D1%83%D1%81%D1%82%D0%80%D0%8D%D0%8E%D0%82%D0%8A%D0%80_%D1%81_%D1%84%D0%88%D0%85%D1%88%D0%8A%D0%88_%D1%87%D0%85%D1%80%D0%85%D0%87_iso_%D0%8E%D0%81%D1%80%D0%80%D0%87rev=1736425501

Last update: 2025/01/09 12:25

