

Własny Debian LiveUSB

(aktualizacja: 2019-05-19)

Live USB możemy bardzo prosto utworzyć instalując system z użyciem debootstrapa i uzupełniając go w chrootcie o jądro, grubę i potrzebne nam narzędzia.

Artykuł ten prezentuje krok po kroku proces budowy własnego live usb. Zaprezentowane w nim zostały polecenia i konfiguracje m.in. związane z następującymi zagadnieniami:

- gpt; efi / uefi; grub;
- montowanie obrazu dysku z pliku (mount disk image from file);
- chroot; montowanie i odmontowywanie specjalnych systemów plików dla chroot; odmontowywanie systemów montowanych z opcją rbind (rbind umount);
- debootstrap;
- optymalizacja dla systemów na pamięciach flash: dane zmienne na tmpfs (ramdysk).

Nie jest to skrypt, który można uruchomić i czekać na gotowy produkt w postaci live USB. Jest to natomiast poradnik, który należy czytać i świadomie wykonywać kolejne podane w nim polecenia. Wymagane jest co najmniej ustawienie zmiennych DEV (wskazującej na urządzenie blokowe na którym prowadzimy instalację) oraz MNT wskazujące na punkt montowania tworzonego rootfs'a. Część prezentowanych ustawień jest opcjonalnych (np. konfiguracja apt'a) i odzwierciedla preferencje autora.

Obraz w pliku

Utworzenie obrazu dysku w pliku, a następnie przekopiowanie go przy pomocy dd na nośnik, jest opcjonalne, ale niekiedy może znacząco przyspieszyć proces instalacji.

```
dd if=/dev/zero of=usb.img bs=1M count=1913
losetup --partscan --find --show usb.img
```

Przygotowanie tablicy partycji

```
# tworzymy tablicę partycji typu gpt
# zasadniczo wystarczył by zwykły MBR z partycją EFI, ale zrobimy hybrydę GPT + MBR
parted $DEV "mklabel gpt"
```

```
# tworzy partycję która posłuży do wgrania grubę (musi mieć co najmniej 128 kiB)
# należy zaznaczyć iż nie jest to partycja /boot
# jest to surowe (bez filesystemu) miejsce na dysku gdzie zostanie wgrany
# fragment grubę normalnie wgrywany do MBR
parted $DEV "mkpart grub 0 2MB";
```

```
# ustawiamy dla tej partycji flagę "GRUB BIOS partition"
parted $DEV "set 1 bios_grub on";
```

```
parted $DEV "mkpart efi 2MB 200MB";
parted $DEV "set 2 boot on";
```

```
# resztę dysku możemy podzielić wg uznania,
# tutaj robimy jedną dużą partycję o nazwie rootfs
parted $DEV "mkpart rootfs 200MB 100%";
```

```
# przy pomocy gdisk utworzenie hybrid MBR
gdisk $DEV
# r ->
#   o
#   h ->
#       1 2 3
#       n 83 n
#       ef y
#       83 n
#   o
#   w ->
#       y
```

```

if [ -e ${DEV}1 ]; then
    DEVp=${DEV}
elif [ -e ${DEV}p1 ]; then
    DEVp=${DEV}p
else
    print "ERROR cant find device for partitions on ${DEV}"
fi

mkfs.vfat -F 32 ${DEVp}2
mkfs.xfs ${DEVp}3

```

Instalacja systemu

```

# zamontowanie systemu plików
mount ${DEVp}3 ${MNT}

# ustawiamy adres mirrora i instalowaną gałąź Debiana
# (chyba że był wcześniej były ustawione)
MIRROR=${MIRROR:="ftp://ftp.icm.edu.pl/pub/Linux/debian/"}
SUITE="stable"

# instalujemy system debootstrapem
debootstrap ${SUITE} ${MNT} ${MIRROR}

```

Chroot

```

# zamontowanie /proc, /sys i /dev na potrzeby chroot'a
mount -t proc proc $MNT/proc
mount -t sysfs sysfs $MNT/sys
mount -o rbind /dev $MNT/dev
mount -o rbind /run $MNT/run/

# ustawienie zmiennych środowiskowych
export DEV DEVp MIRROR SUITE
export LC_ALL="C.UTF-8"

# chroot
chroot $MNT

```

Uwaga: wszystkie następane kroki wykonywane są wewnątrz chroot'a

Podstawowa konfiguracja systemu

```

# używamy UTF-8
echo 'LANG="C.UTF-8"' > /etc/default/locale;
echo 'set -a; . /etc/default/locale; set +a' > /etc/profile.d/locale.sh

# w /etc/fstab ustawiamy rootfs na read-only, a /tmp na tmpfs
# /var/run i /var/lock są współcześnie linkami do /run na tmpfs
cat << EOF > /etc/fstab
/dev/disk/by-uuid/$UUID / xfs ro 0 0
tmpfs /tmp tmpfs defaults 0 0
EOF

# katalogi często pisane na ramdysku
# /tmp będzie punktem montowania więc go czyścimy
rm -fr /var/tmp; ln -s /run/tmp /var/tmp
rm -fr /var/log; ln -s /run/log /var/log

cat > etc/tmpfiles.d/on_tmpfs.conf <<EOF
d /run/tmp 1777 root root -
L+ /var/tmp - - - - /run/tmp
d /run/log 0755 root root -
L+ /var/log - - - - /run/log
EOF

# linkujemy /proc/mounts do /etc/mtab
ln -s /proc/mounts /etc/mtab

```

Chroot - dodajemy ramfs'y

```
mount /tmp
mount -t tmpfs tmpfs /run
mkdir -p /run/{lock,log,tmp}
```

Podstawowa konfiguracja APT

```
cat << EOF > /etc/apt/apt.conf.d/03misc
# nie trzyma polecanych i sugerowanych jak zależności
APT::AutoRemove::RecommendsImportant "false";
APT::AutoRemove::SuggestsImportant "false";

# nie instaluje automatycznie polecanych i sugerowanych
APT::Install-Recommends "false";
APT::Install-Suggests "false";

# wyłączenie wyszukiwania przyrostowego w aptitude
aptitude::UI::Incremental-Search "false";

# wyłączenie pobierania aktualizacji listy pakietów jako pdiff
# (polecam jeżeli stosunkowo rzadko aktualizujemy listy pakietów np. na testingu)
Acquire::PDiffs "false";
EOF

echo "deb $MIRROR $SUITE main" > /etc/apt/sources.list
apt-get update

apt-get install aptitude
```

Instalacja przydatnych narzędzi i programów

```
SYSBASE="aptitude bsdmainutils cron less rsync vim xxd
procps psmisc lsof htop iotop
file screen tmux sudo bash-completion
konwert gpg gpgv gpgsm bzip2 xz-utils p7zip-full
python3 whiptail gawk bc man-db manpages"

HWTTOOLS="sysstat lm-sensors hddtemp
pciutils usbutils lsscsi dmidecode
hdparm sdparm smartmontools
picocom jpneulator
ipmitool pcmciautils"

FSTOOLS="lvm2 mdadm dmsetup multipath-tools
parted gdisk
e2fsprogs xfsprogs btrfs-progs dosfstools
nfs-kernel-server"

NETBASE="iproute2 iptables iputils-ping isc-dhcp-client
net-tools iputils-tracepath dnsutils mtr-tiny
ethtool vlan bridge-utils arptables ebtables
netcat-openbsd socat telnet
wget curl htrack ntpdate debootstrap
openssh-client openssh-server sshfs"

NETDIAG="inetutils-traceroute traceroute paris-traceroute tcptraceroute
tcpdump nmap
sipcalc ipv6calc
arping arp-scan ndisc6
vnstat iftop netdiag tcpflow ngrep
dnstop dnstracer dnswalk"

NETWIFI=" wireless-tools wpa_supplicant hostapd rfkill"

X11TOOL="xorg openbox iceweasel xpdf"

aptitude install $SYSBASE $HWTTOOLS $FSTOOLS $NETBASE $NETDIAG $NETWIFI $X11TOOL
aptitude clean
```

Grub i jądro

```
# włączamy non-free
echo "deb $MIRROR $SUITE main contrib non-free" > /etc/apt/sources.list
aptitude update

# instalacja pakietów gruba, jądra oraz narzędzi efi
aptitude install grub2 grub-efi-amd64-bin efivar efibootmgr \
    linux-image-amd64 firmware-linux
aptitude clean

# plik konfiguracyjny GRUBa
UUID=`grub-probe -d ${DEVp}3 -t fs_uuid`
cat << EOF > /boot/grub/grub.cfg
# korzystanie z konsoli na porcie szeregowym
#serial --speed=115200 --unit=1 --word=8 --parity=no --stop=1
#terminal_input serial console
#terminal_output serial console
# za to na którym RS jest grub odpowiada --unit w serial
#
# korespondujące opcje uruchamiania dla Linuxa:
# module /boot/vmlinuz [...] console=tty0 console=ttyS1,115200n8
# korespondujące opcje uruchamiania dla XENa:
# multiboot /boot/xen.gz [...] com2=115200,8n1 vga=mode-0x0319 console=com2,vga
# module /boot/vmlinuz [...] console=tty0 console=hvc0
#
# w opcjach kernela szczególną rolę odgrywa ostatni podany terminal (ostatnia opcja console)
# jest on związany z /dev/console i na nim pokazywany jest init, ponadto należy uruchamiać getty
# na odpowiednim urządzeniu poprzez wpis w inittab np.:
# T0:23:respawn:/sbin/getty -L /dev/ttyS1 115200 vt100
# lub (dla wariantu XENowego, UWAGA: w przykładzie tylko runlevel 4 i 5):
# T1:45:respawn:/sbin/getty 38400 hvc0

# timeout w sekundach dla domyślnej pozycji menu
set timeout=60
set default="0"

# ładowanie modułów - typ tablicy partycji
insmod part_gpt
insmod part_msdos

# ładowanie modułów - RAID
insmod mdraid1x # mdraid metadata 1.x
insmod mdraid09 # mdraid metadata 0.9
#insmod raid5rec # faulty RAID4/5
#insmod raid6rec # faulty RAID6

# ładowanie modułów - LVM
insmod lvm

# ładowanie modułów - File System
insmod xfs

# ustawienie root'a
search --no-floppy --fs-uuid --set=root $UUID

# ustawienia wizualne
if [ x${feature_default_font_path} = xy ] ; then
    font=unicode
else
    font="/usr/share/grub/unicode.pf2"
fi

if loadfont $font ; then
    if [ x${feature_all_video_module} = xy ] ; then
        insmod all_video
    else
        insmod efi_gop
        insmod efi_uga
        insmod ieee1275_fb
        insmod vbe
```

```

    insmod vga
    insmod video_bochs
    insmod video_cirrus
fi
insmod gfxterm
insmod gettext
fi

set menu_color_normal=white/red
set menu_color_highlight=yellow/black
set pager=1
terminal_output gfxterm

# pozycje menu
menuentry 'Debian GNU/Linux USB-LIVE by rrp@opcode.eu.org' {
    echo 'Loading Linux ...'
    linux /vmlinuz root=UUID=$UUID rootdelay=5 net.ifnames=0 ro
    echo 'Loading initial ramdisk ...'
    initrd /initrd.img
}

menuentry 'Debian GNU/Linux USB-LIVE by rrp@opcode.eu.org init=/bin/bash' {
    echo 'Loading Linux ...'
    linux /vmlinuz root=UUID=$UUID rootdelay=5 net.ifnames=0 ro init=/bin/bash
    echo 'Loading initial ramdisk ...'
    initrd /initrd.img
}
EOF

# instalacja GRUBa dla UEFI
mkdir /boot/efi
mount ${DEV}2 /boot/efi
grub-install --efi-directory=/boot/efi --target=x86_64-efi
mkdir /boot/efi/EFI/BOOT
cp /boot/efi/EFI/debian/grubx64.efi /boot/efi/EFI/BOOT/bootx64.efi

# instalacja GRUBa w MBR
grub-install --boot-directory=/boot/ --target=i386-pc $DEV

```

Konfiguracja końcowa

```

# automatyczne podnoszenie sieci
echo "allow-hotplug eth0" > /etc/network/interfaces.d/eth0
echo "iface eth0 inet dhcp" >> /etc/network/interfaces.d/eth0

# ssh pozwala na logowanie na root'a i nie jest automatycznie uruchamiane
sed -e 's|^[# ]*PermitRootLogin.*$|PermitRootLogin yes|' -i /etc/ssh/sshd_config
systemctl disable ssh.service

# różne ustawienia
systemctl mask hwclock-save.service systemd-random-seed.service udev-finish.service \
    systemd-update-utmp-runlevel.service systemd-update-utmp.service
rm /usr/lib/modules-load.d/ipmievd.conf
echo "liveUSB" > /etc/hostname

# hasło root'a
passwd

# konfiguracja użytkownika - skrypty do montowania i odmontowywania chroot'ów
# oraz fałszywa historia linii poleceń (z tym co może nam się przydać w przyszłości)
cat << EOF > /root/prepareChroot.sh
#!/bin/bash

if [ $# -ne 2 ]; then
    echo USAGE: \$0 device mountpoint
    exit
fi

DEV=\$1
MNT=\$2

mount \$DEV \$MNT

```

```

mount -t proc proc \${MNT}/proc
mount -t sysfs sysfs \${MNT}/sys
mount -o rbind /dev \${MNT}/dev
mount -o rbind /run \${MNT}/run/

chroot \${MNT}
EOF
chmod +x /root/prepareChroot.sh

cat << EOF > /root/umountChroot.sh
if [ \${#} -ne 2 ]; then
    echo USAGE: \${0} mountpoint
    exit
fi

MNT=\${1}

umount \${MNT}/proc
umount \${MNT}/sys
mount --make-rslave \${MNT}/dev && umount -R \${MNT}/dev
mount --make-rslave \${MNT}/run && umount -R \${MNT}/run
umount \${MNT}
EOF
chmod +x /root/umountChroot.sh

cat << EOF > /root/.bash_history
mount -o remount,rw /
systemctl start ssh.service
/root/prepareChroot.sh /dev/sdb1 /mnt/1
chroot /mnt/1
/root/umountChroot.sh /mnt/1
EOF
ln -s /root/.bash_history /.bash_history

# samobójstwo bash'a ... zapobiega zapisaniu historii linii poleceń
# skutkuje też wyjściem z chroot'a
kill -9 $$

```

Odmontowanie chroot'a

Uwaga: wszystkie następne kroki wykonywane poza chroot-em

```

umount \${MNT}/boot/efi/
umount \${MNT}/run
umount \${MNT}/tmp
umount \${MNT}/proc
umount \${MNT}/sys
mount --make-rslave \${MNT}/dev
umount -R \${MNT}/dev
mount --make-rslave \${MNT}/run
umount -R \${MNT}/run
umount \${MNT}
losetup -d \${DEV}

```

Licencja

Copyright (c) 2015-2019, Robert Ryszard Paciorek <rrp@opcode.eu.org>

To jest wolny i otwarty dokument/oprogramowanie. Redystrybucja, użytkowanie i/lub modyfikacja SĄ DOZWOLONE na warunkach licencji MIT.

This is free and open document/software. Redistribution, use and/or modify ARE PERMITTED under the terms of the MIT license.

The MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.