Instalacja Arch Linux

Maciej Tracz

Technikum Mechatroniczne nr 1 w Warszawie

Rok 2020

1 Instalacja

Proces instalowania warto podzielić na 3 etapy:

1. Tworzenie nośnika ISO
2. Instalacja systemu na docelowej maszynie
3. Konfiguracja poinstalacyjna

Teraz kolejno przejdziemy przez wszystkie 3 na przykładzie Linux Manjaro XFCE 20.0.3.

1.1 Tworzenie nośnika

Aby móc zainstalować system operacyjny na komputerze potrzebujemy urządzenia będącego nośnikiem obrazu takiego systemu. Do tego potrzebujemy pendriva conajmniej 8GB oraz obrazu systemu w postaci ISO. Taki można pobrać na stronach twórców w zakładce Downloads. Gdy mamy już te elementy potrzebujemy narzędzia, które przygotuje pendriva jako nośnik. Polecam Rufusa dla użytkowników Windowsa oraz UUByte Software na MacOS. Nie są to jednak jedyne opcje i warto poszukać czy aktualnie nie wyszły nowsze, lepsze narzędzia.

Windows

Rufus: https://rufus.ie/
YUMI: https://www.pendrivelinux.com/yumi-multiboot-usb-creator/

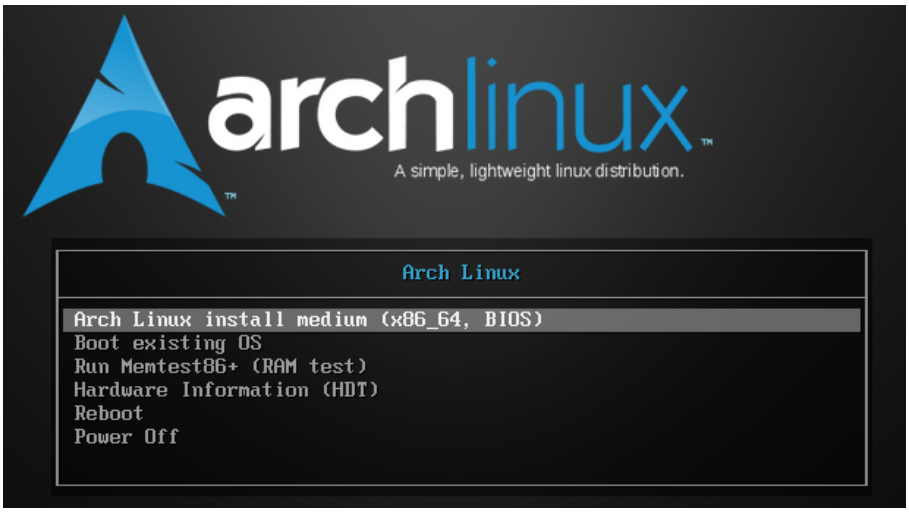
MacOS

UUByte Software: https://www.uubyte.com/download/uubyte-iso-editor.dmg
Disk Utility - The Default ISO Buner (narzędzie wbudowane, polecane na starych urządzeniach.)

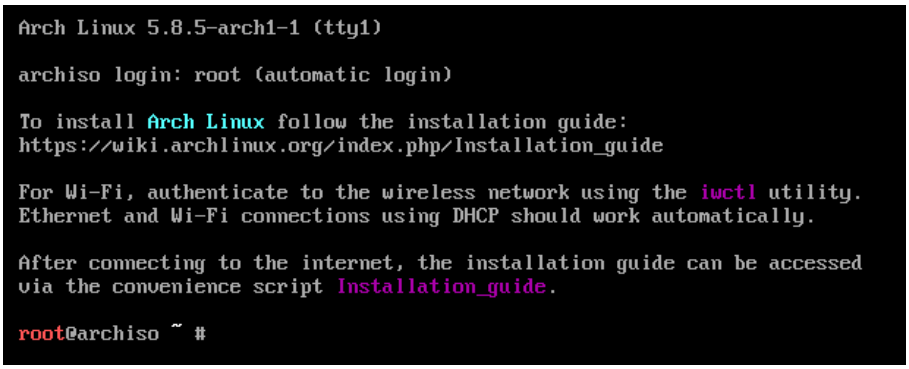
Teraz gdy masz masz wszytsko co potrzebne, postępuj zgodnie z instrukcją na stronie lub po prostu dodaj ISO i wybierz domyślne ustawienia. Jeśli posiadasz już odpowiednio przygotowanego pendriva przejdzmy do następnego etapu.

2 Instalacja

1. Wybór urządzenia rozruchowego



2. Uruchomienie środwiska instalacyjnego Live



3. Partycjonowanie dysku Root

```
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop0: 556.19 MiB, 583208960 bytes, 1139080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ # fdisk /dev/sda

Welcome to fdisk (util-linux 2.36).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x37109c98.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-16777215, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-16777215, default 16777215): +7G

Created a new partition 1 of type 'Linux' and of size 7 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@archiso ~ # _
```

4. Partycjonownanie przestrzeni wymiany

```
root@archiso ~ # fdisk /dev/sda

Welcome to fdisk (util-linux 2.36).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (14682112-16777215, default 14682112):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (14682112-16777215, default 16777215):

Created a new partition 2 of type 'Linux' and of size 1023 MiB.

Command (m for help): t
Partition number (1,2, default 2): 2
Hex code or alias (type L to list all): 82

Changed type of partition 'Linux' to 'Linux swap / Solaris'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@archiso ~ #
```

5. Sprawdzanie wykonania operacji na dyskach

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x37109c98

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1           2048 14682111 14680064    7G 83 Linux
/dev/sda2       14682112 16777215  2095104 1023M 82 Linux swap / Solaris


Disk /dev/loop0: 556.19 MiB, 583208960 bytes, 1139080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```

6. Formatowanie partycji Root systemem plików ext4

```
root@archiso ~ # mkfs.ext4 /dev/sda1
mke2fs 1.45.6 (20-Mar-2020)
Creating filesystem with 1835008 4k blocks and 458752 inodes
Filesystem UUID: e3c42478-e3ba-4a6e-a903-15fbe66dcaae
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

7. Nawiązanie połączenia z internetem (w przypadku wifi użyj komendy wifi-menu)

```
root@archiso ~ # ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group defa
ult qlen 1000
    link/ether 00:0c:29:5d:e9:e2 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
root@archiso ~ # ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.2.130  netmask 255.255.255.0  broadcast 172.16.2.255
        inet6 fe80::20c:29ff:fe5d:e9e2  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5d:e9:e2  txqueuelen 1000  (Ethernet)
        RX packets 7959  bytes 10996454 (10.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3854  bytes 292994 (286.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 284  bytes 25488 (24.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 284  bytes 25488 (24.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@archiso ~ # _
```

8. Pozyskanie list źródeł pakietów

```
root@archiso ~ # pacman -Syy && pacman -S reflector
:: Synchronizing package databases...
 core                          131.2 KiB  1312 KiB/s 00:00 [##################################] 100%
 extra                        1660.9 KiB  2.55 MiB/s 00:01 [##################################] 100%
 community                       5.2 MiB  2.05 MiB/s 00:03 [##################################] 100%
resolving dependencies...
looking for conflicting packages...

Packages (1) reflector-2020.9-1

Total Download Size:   0.02 MiB
Total Installed Size:  0.08 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n] _
```

```
root@archiso ~ # cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
root@archiso ~ # reflector --country Poland  -f 12 -n 12 --save /etc/pacman.d/mirrorlist
root@archiso ~ #
```

9. Zainstaluj Archa

```
[root@yzere ~]# mount /dev/sda1 /mnt
[root@yzere ~]#
```

```
  -> Running build hook: [block]
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
  -> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodet
ect
==> Starting build: 5.8.9-arch2-1
  -> Running build hook: [base]
  -> Running build hook: [udev]
  -> Running build hook: [modconf]
  -> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: wd719x
==> WARNING: Possibly missing firmware for module: aic94xx
==> WARNING: Possibly missing firmware for module: xhci_pci
  -> Running build hook: [filesystems]
  -> Running build hook: [keyboard]
  -> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
(12/13) Reloading system bus configuration...
Running in chroot, ignoring command 'try-reload-or-restart'
(13/13) Warn about old perl modules
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LC_MESSAGES = "",
        LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
pacstrap /mnt base linux linux-firmware vim nano  30.93s user 24.73s system 34% cpu 2:39.74 total
root@archiso ~ # pacstrap /mnt base linux linux-firmware vim nano_
```

10. Konfiguracja

- Wygenerowanie fstab i dodanie strefy czasowej

```
root@archiso ~ # genfstab -U /mnt >> /mnt/etc/fstab
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# timedatectl set-timezone Europe/Warsaw
[root@archiso /]#
```

- Dodanie naszego konta do hostów i stworzenie pliku locale

```
[root@archiso /]# echo yzere > /etc/hostname
[root@archiso /]# touch /etc/hosts
[root@archiso /]# echo LANG=pl.UTF-8 >> /etc/locale.conf
[root@archiso /]# export LANG=pl.UTF-8
[root@archiso /]# echo yzere > /etc/hostname
[root@archiso /]# touch /etc/hosts
[root@archiso /]# vim /etc/hosts
```

- Ustawienie adresów sieciowych na maszynie

```
127.0.0.1       localhost
::1             localhost
127.0.1.1       yzere_
~
~
~
~
~
~
```

- Ustawianie hasła do konta root

```
[root@archiso ~]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso ~]# _
```

11. Instalacja bootloadera Grub

```
[root@archiso ~]# pacman -S grub && grub-install /dev/sda
resolving dependencies...
looking for conflicting packages...

Packages (1) grub-2:2.04-8

Total Download Size:    6.74 MiB
Total Installed Size:  32.91 MiB

:: Proceed with installation? [Y/n] _
```

12. Generowanie pliku konfiguracyjnego Grub-a

```
[root@archiso ~]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
done
[root@archiso ~]# _
```

13. Teraz wpisz 'exit' (może 2 razy z rzędu), a po tym 'shutdown now' / 'reboot now'. Twój system jest gotowy do używania. Naciesz się terminalem :)