Installing Arch Linux on Windows by treewolf

~ Purpose of This Guide

The objective of this tutorial is to help users set up a working, secured distribution of Arch Linux.

~ Materials

A portable medium on which Arch has been mounted or burned. (Will not be covered on how to do this).

A computer to install Arch Linux on. I will assume the computer is dual-booting with Windows preinstalled running on an 86_64 architecture.

Knowledge of your RAM size.

~ Step 1

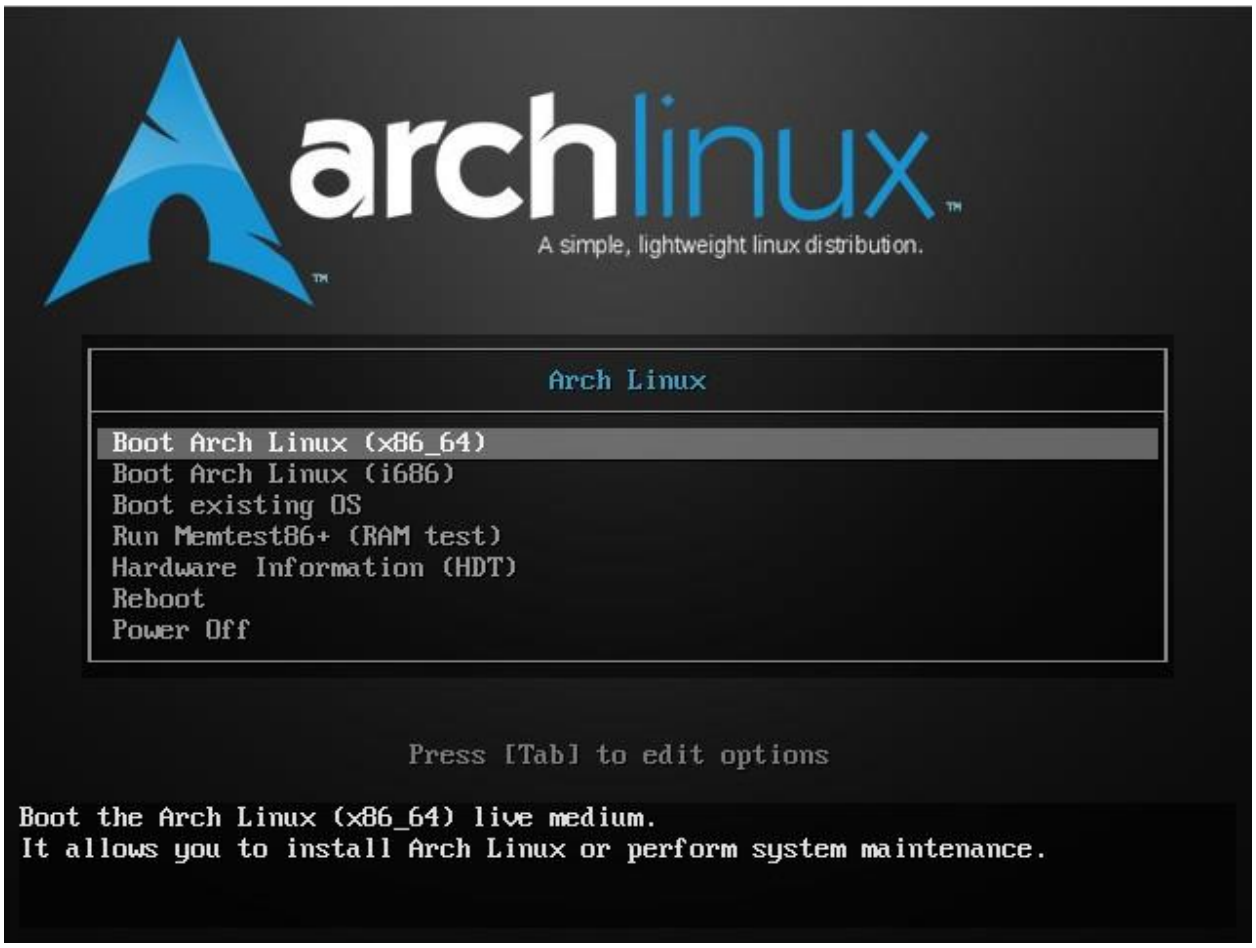
If you don't already have the latest Arch Linux distribution, visit https://www.archlinux.org/download/ and choose a mirror located close to you.

To mount to a pen drive, use this tool https://rufus.akeo.ie/.

~ Step 2

Turn on your machine and repeatedly press F12 to manually load your pen drive. Use Google if you run into any problems.

~ Step 3


archlinux
A simple, lightweight linux distribution.
Arch Linux
Boot Arch Linux (x86_64)
Boot Arch Linux (i686)
Boot existing OS
Run Memtest86+ (RAM test)
Hardware Information (HDT)
Reboot
Power Off
Press [Tab] to edit options

```
Boot the Arch Linux (x86_64) live medium.
It allows you to install Arch Linux or perform system maintenance.
```

Choose the option for x86_64 architecture. This should boot into the iso and into a prompt.

~ Step 4 ~ Pre-Installation

We will use the commonly styled partition scheme of /root, /swap, and /home. In the prompt, type:

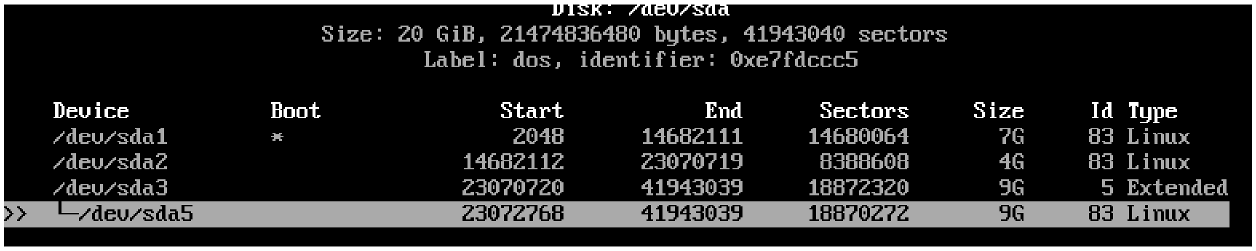
cfdisk

Highlight the option for **dos** and hit enter. Then type **n** and enter the size you want your root partition to be. I like to do one-third of my total free space. Then select **primary** and press enter. You want your root to be bootable, so highlight your newly made partition and press **b**.

Highlight **Free space** and press **n**. This is for the swap partition, and good practice dictates to make it 2x or equal your physical RAM size. This too will be a **primary** partition. Highlight your swap partition and press **t**. Find **Linux swap / Solaris** and press enter.

Now for the last partition, the home partition. Since you are dual-booting, you still have a windows partition[s], so you must create an extended partition first. To do this, select

Free space and press **n**. It should automatically have the remaining size, so just press enter. Select **extended**. Now highlight your extended free space and press **n** and enter. This is my screenshot and yours should look the same or similar:



```
                              Disk: /dev/sda
              Size: 20 GiB, 21474836480 bytes, 41943040 sectors
                     Label: dos, identifier: 0xe7fdccc5

   Device           Boot            Start        End     Sectors    Size    Id Type
   /dev/sda1        *                2048   14682111    14680064      7G    83 Linux
   /dev/sda2                     14682112   23070719     8388608      4G    83 Linux
   /dev/sda3                     23070720   41943039    18872320      9G     5 Extended
>>  └─/dev/sda5                  23072768   41943039    18870272      9G    83 Linux
```

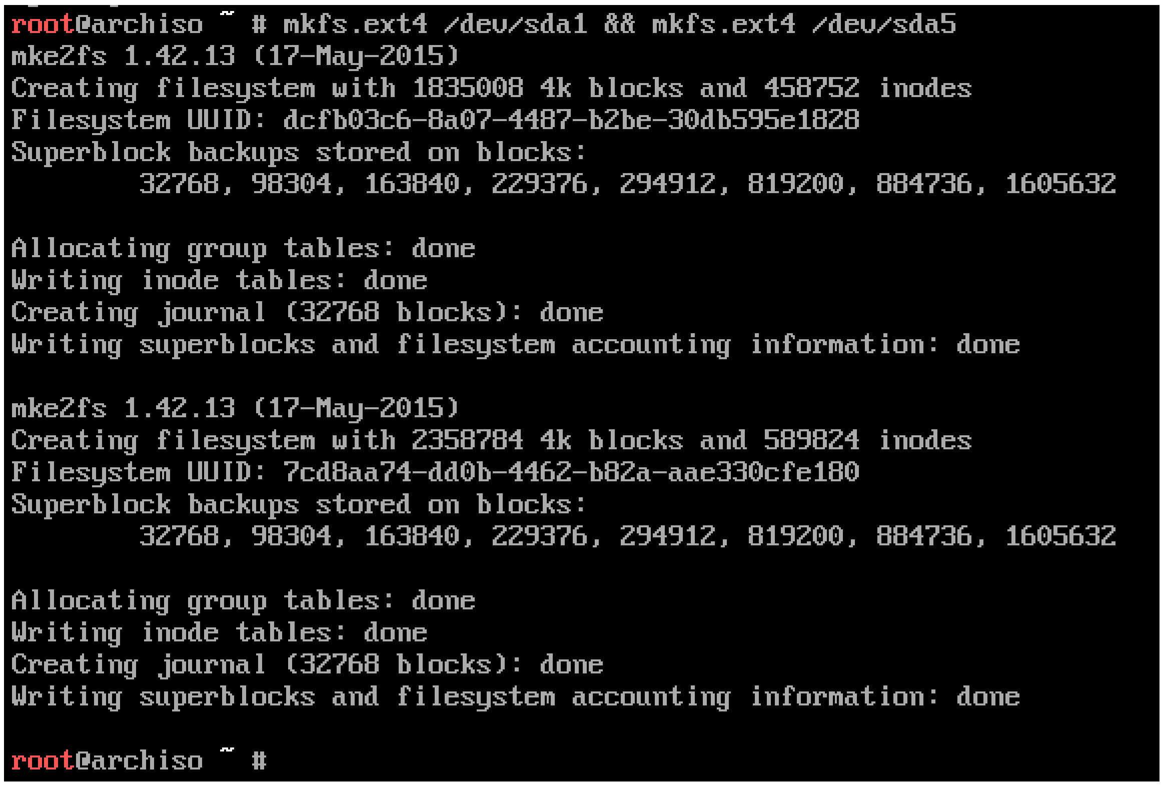
You are almost done with your partitions. Now you need to save and exit cfdisk. Press **Shift + w** and type **yes** and hit enter. Now press **q** to exit.

From this point on, I will use /dev/sda1 for my root, /dev/sda2 for my swap, and /dev/sda5 for my home partitions.

We have to format the partitions to use them so type (remember, substitute your own partition names):

```
mkfs.ext4 /dev/sda1 && mkfds.ext4 /dev/sda5
mkswap /dev/sda2
swapon /dev/sda2
```

You should get something like this:



```
root@archiso ~ # mkfs.ext4 /dev/sda1 && mkfs.ext4 /dev/sda5
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 1835008 4k blocks and 458752 inodes
Filesystem UUID: dcfb03c6-8a07-4487-b2be-30db595e1828
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 2358784 4k blocks and 589824 inodes
Filesystem UUID: 7cd8aa74-dd0b-4462-b82a-aae330cfe180
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ #
```

To work on your partition, we need to mount them:

```
mount /dev/sda1 /mnt
mkdir /mnt/home
mount /dev/sda5 /mnt/home
```

~ Step 5 ~ Installation

Install your base components. Use default selections and press y when prompted; this may take around 5 - 10 minutes:

```
pacstrap –i /mnt base base-devel
genfstab –U –p /mnt >> /mnt/etc/fstab
```

then chroot:

```
arch-chroot /mnt /bin/bash
```

We need to configure our system, so I will use my own locales. If yours are different, change them, respectively.

```
echo <YOUR_OWN_HOSTNAME> >> /etc/hostname
```

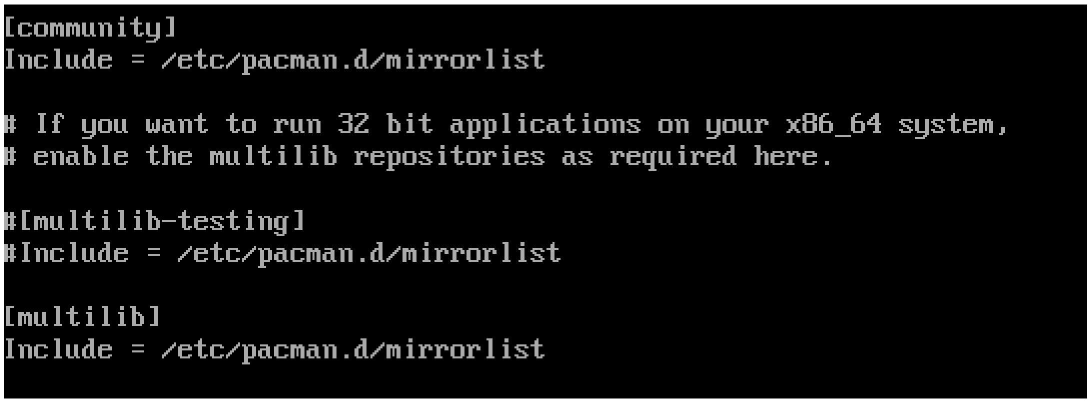
Check if hostname was set. If it wasn't then edit /etc/hosts.

Edit and find your locale. Mine is **en_US.UTF-8** and local time is **Los_Angeles**:

nano /etc/locale.gen
locale-gen
echo LANG=en_US.UTF-8 > /etc/locale.conf
export LANG=en_US.UTF-8
ln –s /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
hwclock --systohc --utc
mkinitcpio –p linux

uncomment your multilib repo:

nano /etc/pacman.conf



```
[community]
Include = /etc/pacman.d/mirrorlist

# If you want to run 32 bit applications on your x86_64 system,
# enable the multilib repositories as required here.

#[multilib-testing]
#Include = /etc/pacman.d/mirrorlist

[multilib]
Include = /etc/pacman.d/mirrorlist
```

pacman –Syu

Time to create a user account. First, we need to secure the root account:

passwd

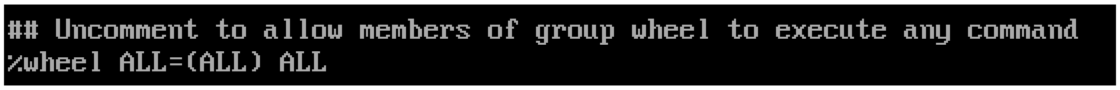
Then we need to add a regular user with Administrator privileges and set its password.

useradd –mg users –G wheel,storage,power –s /bin/bash <YOUR_USERNAME>
passwd <YOUR_USERNAME>

Now let's install sudo, which is like the pop up box you get in windows when you want to do something that requires Administrative privileges.

pacman –S sudo
sudo visudo

Scroll all the way down and uncomment to allow members of group wheel to execute any command by pressing **i** and deleting the ‘ # ‘ sign:



```
## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL) ALL
```

Then add to the bottom of the file by scrolling to the end of the last line, press **a** then enter on a new line:

```
Defaults:ALL timestamp_timeout=0
```

Now press **Esc** and then type:

```
:wq
```

and press enter.

~ Step 6 ~ Installing the bootloader

Without the bootloader, how will you find your linux set-up? This is a crucial step so don’t forget it! Type:

```
pacman –S grub
grub-install /dev/sda
pacman –S os-prober
grub-mkconfig –o /boot/grub/grub.cfg
```

You want this tool so you can access the internet later:

```
pacman –S iw wpa_supplicant networkmanager
systemctl enable NetworkManager
```

This is for a desktop environment. To later start your desktop, type **startx**:

```
pacman –S xorg-xinit xorg-server xorg-utils xorg-server-utils mesa
pacman –S lxde
echo exec startlxde > ~/.xinitrc
```

This is to change your volume settings:

```
pacman –S alsa-utils
```

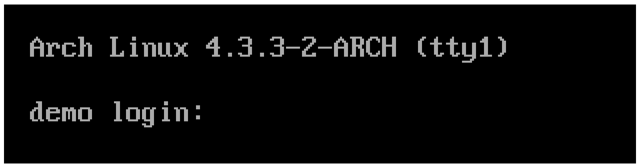
You are probably using a portable computer, so you may need to use the touchpad:

```
pacman –S xf86-input-synaptics
```

We are done with the basic installation. Let’s exit and reboot:

```
exit
umount –R /mnt
reboot
```

Select the first option, and you get your new login shell!



```
Arch Linux 4.3.3-2-ARCH (tty1)

demo login:
```

After you boot, you enable network:

nmcli n on

if you have wifi, do:

ip addr
nmcli device <WIFI DEVICE NAME> connect <NETWORK SSID> password <PASSWoRD>

It is also good to have wget:

sudo pacman –S wget

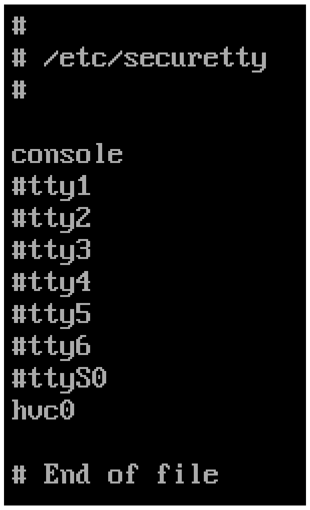
Now we need to turn on NumLock when booting:

sudo mkdir /etc/system/system/getty@.service.d
echo [Service] > activate-numlock.conf
echo “ExecStartPre=/bin/sh –c ‘setleds +num < /dev/%I’ ” >> activate_numlock.conf

~ Step 7 ~ Secure the Computer

First you want to disable root login. You should only use root when a valid user is already logged in, so comment out all the tty’s:

sudo nano /etc/securetty



```
#
# /etc/securetty
#

console
#tty1
#tty2
#tty3
#tty4
#tty5
#tty6
#ttyS0
hvc0

# End of file
```

Then do:

```
sudo nano /etc/ssh/ssh_config
```

and change the **Cipher cbc** to **Cipher aes256**

You also want to make sure you require a password to use **wget**, which is a downloader:

```
sudo chmod 750 /usr/bin/wget
```

You want to be able to lock all your screens from physical tampering, regardless if you are using xterm or cli. I like to use this pretty nifty tool. To use this tool just enter

```
sudo physlock
sudo wget https://github.com/muennich/physlock/archive/v0.5.tar.gz
tar –xvf *.gz
cd physlock*
sudo make install
```

Arch Linux comes with a pre-installed firewall, but by default it isn't enabled nor configured. You can download this bash file that will set up your firewall for you at https://github.com/treewolf/Misc/blob/master/ArchLinux/iptables_setup by using

```
sudo wget
https://raw.githubusercontent.com/treewolf/Misc/master/ArchLinux/iptables_setu
p -O iptables_setup
sudo chmod 700 iptables_setup
sudo ./iptables_setup
```

or look at the documentation. You also want to make sure clamav is up. This is your anti-virus.

```
sudo pacman –S clamav
sudo freshclam
sudo systemctl enable clamd.service
sudo systemctl start clamd.service
sudo systemctl enable freshclamd.service
sudo systemctl start freshclamd.service
```

Then you want a sandbox for your browser and other processes. You can use it with firefox by typing **firejail firefox**:

```
sudo wget https://github.com/netblue30/firejail/archive/0.9.36.tar.gz
tar –xvf 0.9.36.tar.gz
cd firejail*
./configure
sudo make install
```

Now we will install firefox browser:

sudo pacman –S firefox

Last, we need to change our finger settings in case it was set by default:

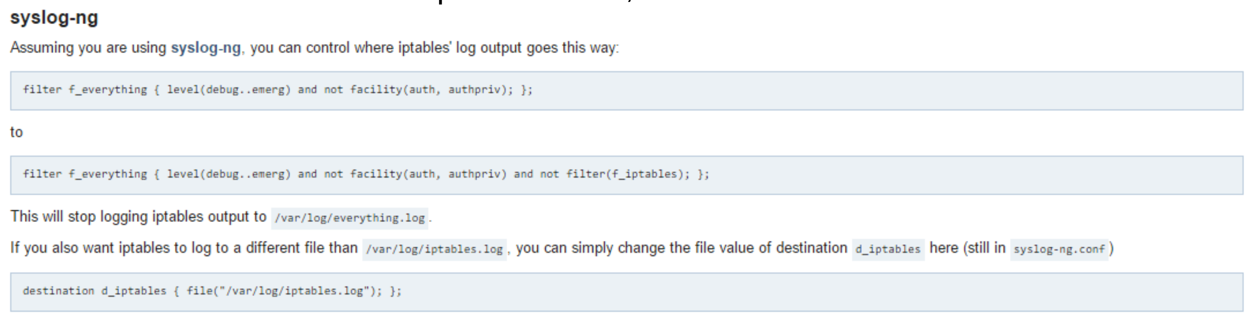
sudo chfn root

Enter **none** for all. Do the same for any user as you see fit.

We also want to see the logs for the firewall, in case we need to analyze suspicious packets. Download syslog-ng using

sudo pacman –S syslog-ng
sudo systemctl enable syslog-ng
sudo systemctl start syslog-ng

and follow the directions in the picture below, taken from the Arch Linux website.



syslog-ng

Assuming you are using **syslog-ng**, you can control where iptables' log output goes this way:

```
filter f_everything { level(debug..emerg) and not facility(auth, authpriv); };
```

to

```
filter f_everything { level(debug..emerg) and not facility(auth, authpriv) and not filter(f_iptables); };
```

This will stop logging iptables output to `/var/log/everything.log`.

If you also want iptables to log to a different file than `/var/log/iptables.log`, you can simply change the file value of destination `d_iptables` here (still in `syslog-ng.conf`)

```
destination d_iptables { file("/var/log/iptables.log"); };
```

Last, erase the **quiet** in the file **/etc/default/grub** to see verbose, so you can tell if your computer has any base problems while booting.

You are done. You now have a basically-secured Linux box. Well Done!