

Installing Arch Linux



Presented by yours truly
Cameron Calpin

About Arch Linux

- Arch Linux is an independently developed, i686/x86-64 general purpose GNU/Linux distribution
- Development focuses on simplicity, minimalism, and code elegance. Arch is installed as a minimal base system, configured by the user upon which their own ideal environment is assembled by installing only what is required or desired for their unique purposes.
- GUI configuration utilities are not officially provided, and most system configuration is performed from the shell by editing simple text files.
- Arch is backed by **pacman**, an easy-to-use binary **package manager** that allows you to upgrade your entire system with one command. Pacman is coded in C and designed from the ground up to be lightweight, simple and very fast. Arch also provides the **Arch Build System**, a ports-like system to make it easy to build and install packages from source, which can also be synchronized with one command. You can even rebuild your entire system with one command.

About Arch Linux Cont.

- Judd Vinet, a Canadian programmer and occasional guitarist, began developing Arch Linux in early 2001. Its first formal release, Arch Linux 0.1, was on March 11, 2002. Inspired by the elegant simplicity of Slackware, BSD, PLD Linux, and CRUX, and yet disappointed with their lack of package management at the time; Vinet built his own distribution on similar principles as those distros. But, he also wrote a package management program called pacman, to automatically handle package installation, removal, and upgrades.
- In late 2007, Judd Vinet retired from active participation as an Arch developer, and smoothly transferred the reins over to American programmer Aaron Griffin, aka Phrakture, who remains the lead Arch developer to this day.

Why Arch Linux?

- As a distro it is probably the closest one can get to really building their own custom version of Linux without actually building their own custom version of Linux. Everything is up to the user from how the devices are partitioned, what video driver you want to use (proprietary vs open-source) what shell you want to use, you can chose your DE, choose the boot-loader, etc. It also gives you an intimate knowledge of what goes on when you install an operating system beyond just clicking "Next" a few times, and if you like a challenge its fun too!

- Make sure you have vmware(workstation/player) or virtualbox. If not, go to:

https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0
<https://www.virtualbox.org/wiki/Downloads/>

- Vmware/virtualbox are virtual machines that allow you to run iso files (CD/DVD images).
- The iso file we'll be downloading is Arch Linux. The link to download Arch linux: <https://www.archlinux.org/download/>
- Go down to United States Section (because United states sites have better open source distro)
- Click on kernel.org, and you'll be brought to an index. Click on: archlinux-2016.02.01-dual.iso

- If on vmware workstation/player, click on Create a new virtual machine.
- Click on Typical (Recommended).
- Click on the Installer disc image file (iso) and search for the arch linux iso file in your downloads folder (ignore the warning it gives you when selecting the arch linux, don't worry about it...seriously)
- Click Next, then click on the Linux button and click on the scroll down menu from the version.
- Select Other Linux 3.x kernel. Click on Next then give your Arch Linux a name.
- Leave the Location as it is. Allocate Maximum Disk Size (GB) to at least 10-15 GB.

- Boot Arch Linux. Choose the **Boot Arch Linux (i686)**.
- In the command line, type **fdisk -l**. You will see the allocated memory you chose in the **/dev/sda**.
- INFO: /dev/ is the part in the unix directory tree that contains all "device" files -- unix traditionally treats just about everything you can access as a file to read from or write to. The sd is originally identified a SCSI device, but since the wildgrowth of USB (and other removable) data carriers it became a catch-all for any block device (another unix term; in this context, anything capable of carrying data) that wasn't already accessible via IDE. The letter immediately after **sd** signifies the order in which it was first found -- a,b,c...z, Aa...Az... etc.
- Now type in **cfdisk /dev/sda**. This is a linux partition editor that we're going to use to create 3 partitions on the disk.
- choose the **dos** as a label type. Choose **[new]** to create a partition. Set partition size to **2G** and choose **[primary]**. Select **[Bootable]** for the **/dev/sda1**.
- Again, select **[new]**, set partition size to **2G**, choose **primary**, then go to **[type]** and select **82 Linux swap / Solaris**.
- INFO: Swapping is the process whereby a page of memory is copied to the preconfigured space on the hard disk, called swap space, to free up that page of memory. When the system requires more memory than is physically available, the kernel swaps out less used pages and gives memory to the current application (process) that needs the memory immediately.

- Go to **Free Space**, select **[new]** and select **primary**, and partition the rest of your allocated memory here. For **/dev/sda3**, go to **[write]**, enter **yes**, and press **Enter**, and go to **[quit]**.
- If you enter **fdisk -l** in the command line, you can see the three partitioned disks that you created.
- Now type **mkfs.ext4 /dev/sda1**, then **mkswap /dev/sda2**, then **mkfs.ext4 /dev/sda3**.
- Now we're going to mount the partitions. Type **mount /dev/sda3 /mnt**. Next we make directories for the following mount devices. Type **mkdir /mnt/boot /mnt/var /mnt/home**.
- We're going to mount our first dev to the boot. Type **mount /dev/sda1 /mnt/boot**.
- Next we will install the base operating system. Type **pacstrap /mnt base**. Next is the base development. Type **pacstrap /mnt base-devel** (this part requires lots of patience).

- Now we'll update our system. Type **pacman -Syu**.
- Type **genfstab -p /mnt >> /mnt/etc/fstab**. Then type **arch-chroot /mnt**, and type **bash**.
- Setting the clock, type **hwclock --systohc --utc**.
- Type **mkinitcpio -p linux**. Once that's done, we'll set a root password. Type **passwd root**.
- Now we're going to make an additional user. Type **useradd -m -g users -G wheel -s /bin/bash sysadmin**. Type **passwd sysadmin** to set a password for sysadmin.
- Now we're going to install grub. Type **pacman -S grub-bios**. Now we'll install the Grub configuration to sda disk. Type **grub-install /dev/sda**. Next, type **grub-mkconfig -o /boot/grub/grub.cfg**.
- Now we're going to enable the DHCP client service. Type **systemctl enable dhcpcd@.service**. Type **exit** twice to exit out of bash and root.

- Now we're going to reboot the system. Type reboot. Once at the boot menu, select ***Arch Linux**.
- Side Note: Make sure to disable the CD-ROM device in workstation. Go to Removable devices > CD/DVD (IDE) > Disconnect.
- Log in as root and enter your password.
- Now we're going to create a hostname. Type nano /etc/hostname.
- Now we'll check our ip address configuration. Type ip addr. We should have a valid IPv4 address.
- Again, we'll enable the DHCP client configuration. Type systemctl enable dhcpcd@ens32.service. Then type dhcpcd to get a new ip address. Type ip addr and check for a new valid IPv4 address. To test if the connection is working, type ping 8.8.8.8.

- Now we're going to install the sudo packages. Type **pacman –S sudo**.
- Now type **nano /etc/sudoers**. We're going to add sysadmin to sudoers. Scroll down until you see root ALL=(ALL) ALL and under it, type **sysadmin ALL=(ALL) ALL**. press **Ctrl X** and type **y**.
- Now we'll install the X server packages. Type **pacman –S xorg** and press enter twice. Then type **pacman –S xterm** to install xterm.
- To install the xorg-clock, type **pacman –S xorg-xclock**.
- To install other xorg packages, type **pacman –S xorg-init**, then **pacman –S xorg-server-utils**, theeeeeeeeen type **pacman –S mesa**.
- Now to install the LXDM desktop manager. Type **pacman –S lxdm**.
- Now to install the LXDE desktop environment. Type **pacman –S lxde**.
- Now to enable LXDM service. Type **systemctl enable lxdm.service**.
- Now reboot system.

- If you have vmware, you can install some additional packages.
- Also, if you don't have a connection, type **systemctl enable dhpcd@.service**, then type **ip addr** to check for a valid connection. Remember, use the ping (**ping 8.8.8.8**) command to test your connection.
- Go to **Ixterminal**, and type **su -**, then type **pacman -S net-tools**, **pacman -S gtkmm**, **pacman -S open-vm-tools**.
- For post installation configuration, type **cat /proc/version > /etc/arch-release**.
- Configuration for open-vm-tools, type **nano /usr/lib/systemd/system/vmtoolsd.service**. Under ExecStart=/usr/bin/vmtoolsd, type **KillSignal=SIGKILL**, then press **Ctrl x** and exit.
- Next, type **systemctl enable vmtoolsd.service** to enable vmtools.
- Reboot

Sources

- <https://www.archlinux.org/about/>
- https://wiki.archlinux.org/index.php/Arch_Linux
- <https://www.linux.com/news/software/applications/8208-all-about-linux-swap-space>
- <https://www.youtube.com/watch?v=JguJMRu1riA>

`# rm -rf /`



fixes all linux issues