بسمه الله الرحمان الرحیم

فهرست مطالب

عنوان صفحه

Contents

[مقدمه 1](#_Toc47862578)

[فصل یکم - اشنایی با آرچ 2](#_Toc47862579)

[1-1- تاریخچه آرچ لینوکس 2](#_Toc47862580)

[1-2- اصول راهبردی آرچ 2](#_Toc47862581)

[3-1- اصل KISS : 4](#_Toc47862582)

[1-4- معماری و ساختار آرچ لینوکس : 4](#_Toc47862583)

[1-5- نصاب آرچ لینوکس : 5](#_Toc47862584)

[1-6- مدیر بسته Pacman : 5](#_Toc47862585)

[7-1- مخازن نرم افزاری یا Repositories : 6](#_Toc47862586)

[8-1- انتشار غلطان یا Rolling Release : 6](#_Toc47862587)

[10-1- مخازن نرم افزاری AUR : 7](#_Toc47862588)

[11-1- پکیج منیجرyaourt : 8](#_Toc47862589)

[فصل دوم – آماده سازی 10](#_Toc47862590)

[2-2- بررسی صحت فایل رسانه : 10](#_Toc47862591)

[1-2-2- بررسی صحت با امضای PGP : 11](#_Toc47862592)

[2-2-2- الگوریتم مجموع مقابله ای : 12](#_Toc47862593)

فهرست شکل ها

فهرست جداول

# مقدمه

شاید تعداد کمی از کاربران گنو/لینوکس ایران با وجود توزیع هایی مثل اوبونتو , فدورا , یا اوپن سوزه حتی نام توزیع آرچ لینوکس را شنیده باشند چه برسد به اینکه بخواهند آنرا تست کنند یا حتی کاربر دائمی آن باشند. این امر بیشتر از اینکه تبلیغی برای آن نشده باشد باید آنرا به حساب دشواری نصب و راه اندازی آن از لحظه نصب تا رسیدن به مرحله کاربردی گذاشت. آرچ لینوکس به دلیل نصاب متنی آن و بعضنا اشتباهات بعضی کاربران در مرحله پیچیده پارتیشن بندی آن و همچنین عملیات طاقت فرسای بعد از نصب از قبیل نصب و پیکربندی سخت افزارها تا میزکار و نرم افزارهای آن , کمتر کاربری رقبت می کند به سراغ آن برود و این ریسک را بکنندکه حتی آنرا تست کند چه برسد به اینکه کاربر دائمی آن باشد. در این مقاله سعی خواهیم کرد به صورت تخصصی به بررسی بخش بخش سیستم عامل آرچ لینوکس و توزیع های بر پایه آن بپردازیم و تا حد امکان از تجربیاتم از آن بنویسم و یک به یک اجزای آنرا بررسی کنیم و ببینیم آیا از آرچ لینوکس میتوان به عنوان یک توزیع کاربر پسند برای کاربران خانگی و استفاده کنندگان تجاری (نه گیگ ها) استفاده کنیم یا که خیر ؟!

# فصل یکم - اشنایی با آرچ

## 1-1- تاریخچه آرچ لینوکس

قبل از آنکه وارد جزئیات شویم همانند سایر اجزای تشکیل دهنده دنیای آزاد نر‌م افزار و GNU لازم میدانم تاریخچه ای از توزیع آرچ لینوکس و چگونگی پیدایش آن و روند توسعه آن مختصری توضیح دهم ولی پیش از آن این پاراگراف را از ویکیپدیا نقل میکنم که جایی از آن نوشته شده : » آرچ لینوکس یک توزیع گنو/لینوکس است که با هدف سبک بودن و ساده بودن شناخته شده است. رویکرد طراحی تیم توسعه بر روی سادگی,ظرافت,صحت کد و کوچک بودن آن گذاشنته شده است. سادگی بر مبنای تعریف آرچ: فاقد افزونه ها , تغیرات و پیچیدگی های غیر ضروری آن است و این سادگی از دیدگاه توسع ‌دهنده وبرنامه نویس تعریف میشود , نه از دیدگاه یک کاربر معمولی « د‌رواقع سادگی آرچ لینوکس از نگاه یک برنامه نویس در بخش Backend و سطح کدنویسی است نه سطح کاربری و کار کردن با آن آغاز توسعه آرچ لینوکس از مارچ ۲۰۰۲توسط Jud Vinet شکل گرفت و این توزیع با الهام از یک توزیع سبک گرای دیگر به نام Crux روند توسعه آن شکل گرفت و آغاز شد. توسعه آرچ لینوکس به مدت ۵سال رهبری آن به عهده جاد وینت بود و زمانی که جاد دیگر وفت کافی برای توسعه آن نداشت ادامه راه را در اول اکتننبر ۲۰۰۷به عهده Aaron Grifin گذاشت که تا به امروز رهبری توسعه آن بر عهده وی می باشد

## 1-2- اصول راهبردی آرچ

قبل از آنکه وارد مباحث تخصصی و آشنایی با زیر و بم ارچ لینوکس شویم نیاز است کنه بنا ۵ اصل راهبردی آرچ لینوکس که به Arch Way معروف است آشنا شویم ۵ اصلی که روند توسعه آرچ وهدف توسعه دهندگان بر آن بنا شده و همواره بر پایه آن کار خود را جلو میبرند

1. سادگی یا : Simplicity همانطور که در ابتدا و مقدمه هم توضیح دادم سادگی آرچ لینوکس بنه صنورت فاقنند افزونه ها و تغیرات پیچیدگی های غیر ضروری است به نحوی که یک سیستم شبه ینونیکس یا Unix Like را بنه معنای واقعی کلمه در اختیار کاربر قرار م ‌یدهد و کاربر با توجه به نیاز خود سیستم و محیط کاری خود را از پایه میسازد و آرچ و توسعه دهندگان آن چیزی را برای استفاده به کاربر القاء یا تحمیل نمی کنند
2. صحت و سادگی کد یا : Code Correctness Over Convenience پیاده سازی, سادگی, ظرافت کد کوچک و مختصر بودن همیشه در اولویت و دستور کار آرچ بوده است. همیشه ساده کردن کار با سیستم از دیدگاه یک کاربر نهایی پیچیدگی هایی را در سطح کدنویسی و Backend ایجاد میکند که به مرور زمان اینن پیچیدگی ها بعضا در مواردی باعث به وجود آمدن مشکلتی در بخش کاربری میشود واین مشکلات میتوانند به صورت Crash یا باگ های امنیتی ظاهر شود
3. کاربر محور بودن یا User Centric : یکی دیگر از اصول ۵گانه آرچ لینوکس خود مختاری کاربر است. آرچ لینوکس با قرار دادن یک بستر قوی و مستحکم در اختیار کاربر بقیه کار وساختن اجزای سیستم را به دست خود او می سپارد تا تک تک اجزای سیستم را از نصب Xorg تا نصب و پیکربندی تک تک درایورهای سخت افزاری , تا نصب مدیریت پنجره و میزکار گرافیکی و تا نصب حتی کوچکترین پکیج ها و نر‌م افزارها را در اختیار خود او باشنند تا از نصب اجزای بی مصرف و غیر قابل استفاده برای کاربر جلوگیری شود و این کار این حسنن را دارد کاربر بنا اجزای سیستم آشنا شود و در صورت بروز مشکل بداند برای حل آن باید به کجای سیستم مراجعه کند . آنرا حل کند و اگر احیانا کاربر دیگری با این مشکل روبه رو شد به او کمک کند و در خدمت جامعه کاربری یا اصطلاحا Community باشد همچنین با این کار از لود ماژول های اضافی در کرنل و پکیج های اضافی جلوگیری میشود ودر نتیجه سرعت بیشتری را در پروسه بوت سیستم شاهد خواهیم بود
4. باز بودن یا Openness : شاید پایبند بودن به اصل آزادی نر ‌مافزار و اوپن سنورس در کمترتوزیع لینوکسی دیده شود. از پچ ها و ماژول هایی که برای کرنل لینوکس نوشته میشود تا درایور سخت افنزاری امکنان دارد درتوزیع های مختلف لینوکس از مشابه سورس بسته آن و نسخه غیر آزاد آن استفاده شود. آرچ با پایبند بودن به این اصل سعی میکند تماما از نرم افزارهای آزاد و اوپن سورس در سیستم خود و مخازن نرم افزاری استفاده کند.
5. آزادی یا freedom : و آخرین اصلی که جز ۵گانه آرچ لینوکس به شمار م ‌یرود اصل آزادی است. آرچ با ساده نگه داشتن سیستم یک محیط خام و به دور از هرگونه نر ‌مافزار و پکیج را در اختیار کاربر قرار م ‌یدهد و ایننن آزادی را به طور کامل به او م ‌یدهد که چه پکیجی را برای استفاده انتخاب و نصب کند. به عنوان مثال کاربری تمایل دارد از میزکار KDEاستفاده کند و کاربر دیگری از Gnome یا XFCE یا کاربری تمایل دارد از مرورگر Firefox استفاده کند و شخص دیگر از مرورگر Chrome آرچ لینوکس این آزادی را به کاربر میدهد آنچه خود می پسندد استفاده کند نه آنچه توسعه دهندگان به صورت پیشفرض روی سیستم نصب می کنند. خوشبختانه با حجم زیاد و تنوع نرم افزاری مخازن آرچ این آزادی بیشنتر قابل لمس است و کاربر میتواند هر آنچنه را که میخواهد ازRepositories پیدا و نصب کند. در همین رابطه Jud Vinet میگوید: آرچ لینوکس همان چیزی است که شما میخواهید بسازید ( Arch Linux is what you make it )

## 3-1- اصل KISS :

با توجه با ۵ اصل Archway که گفته شد حال میتوان خیلی راحتتر در مورد اصل KISS صحبت کرد همانطور که از عبارت keep it simple and stupid/short برمیآید میتوان این عبارت را اینگونه ترجمه کرد »ساده و مختصر نگه ش بدار» (به طوری که یک احمق هم بتواند با آن کار کند) اصل KISS همانطور که در خیلی از توزیعهای دیگر مثل Crux یا Gentoo رعایت میشود در آرچ هم دیده میشود و ارچ همواره این را به کاربر خود متذکر میشود که برای کار با سیستم همیشه آنرا ساده و کوچک نگهدارد و از نصب جزئیات غیر ضروری جلوگیری کند

## 1-4- معماری و ساختار آرچ لینوکس :

آرچ به طور کسترده ای بر پایه ی بسته های باینری یا دودویی است بسته های باینری در این توزیع با هدف معماری های i686و x86\_64ساخته میشوند تا سیستمی مدرن و بهینه ای را بسازند همچنیننن آرچ ازیک سیستم اوتوماتیک برای کامپایل و ساخت بسته ها برای معماری های دیگر به نام ABS یا Arch Build System بهره میبرد که در ادامه به آن اشاره خواهد شد

توجه ارچ به سادگی و اقتصادی بودن سیستم برای توسعه دهندگان در میان مسائل دیگر به این معنی نیست که برای کمک به کاربر برنام ‌ههای گرافیکی برای تنظیمات سیستم شنناخته منی شنود. برای مثال مدیر بسنته آرچ لینوکس هیچ گونه واسط گرافیکی رسمی ندارد ولی بیشتر روی ساختن و آماده کردن فایلهای تنظیمات با حاشیه نویسی خوب و استفاده بهینه از خط فرمان است . این مسأله باعث شد که آرچ به عنوان ینک توزیع برای کاربران متوسط و حرفه ای که از کار با خط فرمان نمی ترسند معروف شود درهمین رابطه Aaron Griffin بر روی وبلگ شخصی خود اینگونه می نویسد : " اعتماد کردن به ابزار پیچیده مدیریت و ساختن سیستم باعث صدمه زدن به کاربر نهایی می شود. اگر شما برای مخفی کردن پیچیدگی سیستم تلاش کنید به یک سیستم پیچید ‌تر خواهید رسید لایه های انتزاعی که باعث مخفی کردن پیچیدگی ها میشود هیچ گاه کاربردی نبوده اند و در مقابل سیستم باید به نحوی طراحی شود که هیچ گونه مخفی سازی نیاز نباشد”

## 1-5- نصاب آرچ لینوکس :

مدیا های موجود برای نصب آرچ لینوکس هم به صورت ) ISO برای رایت بر روی سی.دی( و هم به صننورت IMG برای نصب روی فلش مموری ها از وب سایت ارچ لینوکس قابل دانلود است. همچنین مدیا های آرچ بنه دو صنورت Core Install و Net Installموجود است.حالت Core Installبه این صورت است که از آن می توانید به عنوان یک نصاب Offlineاستفاده کنید به این صورت که تمامی پکیج ها برای نصب سیستم پایه بر روی آن قرار دارد. در مقابل Net Installتنها یک نصاب بوده و برای نصب پکیج ها نیاز به یم خط اینترنت دارد تا پکیج های پایه را نصب کند. گزینه Net install این حسن را دارد که در حین نصب از اینترنت م ‌یتوانید تمامی پکیج های مورد نیاز را که کاربر تمایل دارد نصب کند و همچنین در حین نصب آخرین نسخه پکیج ها در درون مخازن را دانلود و نصب کند و به دلیل التزام آرچ به اصل KISS مبتنی بر کوچک و ساده سازی , نصاب آرچ تنها یک شل اسکریپت ( Shell Script ) ساده است که از یک پوسته گرافیکی ( به صورت Text mode ) برای تعامل با کاربردر هنگام نصب سود می برد ، نصب به صورت پیشفرض تنها یک محیط پایه ی گنو/لینوکس را فراهم می کند که شامل زنجیره ابزار گنو ( GNU Toolchain ) و هسته لینوکس ( Kernel ) ، چند ماژول اضافی و کتابخانه ها و محیط شل است . شخصی سازی و گسترش سیستم (اضافه کردن نرم افزارهایی مثل مدیر پنجره و میزکار و غیره ) باید به صورت دستی و از طریق نصب بسته های دانلود شده از مخازن آنلاین صورت می گیرد . به همین دلیل نصب آرچ عموما در مقایسه با سایر سیستم عامل ها پیچیده انگاشته می شود .

## 1-6- مدیر بسته Pacman :

برخلاف تصور خیلی ها (!) آرچ لینوکس هم مثل سایر توزیع ها مانند fedora ، Ubuntu ، Debian و OpenSUSE و ...... مدیر بسته ای به نام Pacman دارد ( بدون محیط گرافیکی ) . نام Pacman در واقع الهام گرفته از کارکتر بازی Pacman ( یک بازی ویدئویی) است که این نام به نوعی تداعی کننده ی عبارت Package Manager در ذهن کاربر است . Pacman نصب بسته ها ، به روزرسانی ، حذف و بازگردانی بسته ها را انجام می دهد و توانایی تشخیص خودکار وابستگی ها را دارد. بسته های آرچ لینوکس از مخازن بسته های آرچ و برای معماری های i686 و x86\_64 بهینه سازی شده اند و آرچ لینوکس برای رسیدن به سرعت بالا در نصب نرم افزار ها بر اسا پکیج های با فرمت pkg.tar.xz طراحی شده است .

## 7-1- مخازن نرم افزاری یا Repositories :

مخازن نرم افزاری آرچ لینوکس به 6 بخش ، Testing ، Unstable ، Community-Testing ، Core ، Community و Extra تقسیم می شود . مخزن Core تنها کرنل لینوکس و آپدیت های آن و کتابخانه ها و پکیج هایی که برای نصب پایه ی سیستم نیاز هستند درون آن قرار داده می شود. مخزن Extra مخزنی است که نرم افزار های اضافی مثل xorg-server و میزکار ها و ابزاهای کاربردی درون آن قرار داده می شود . در ادامه با مخازن عمومی AUR آشنا خواهیم شد . پکیج های این مخزن قابلیت امتایز گیری از کاربران را دارد . مخزن Community توسط یک سری از کاربران که اصطلاحا « کاربران مورد اعتماد » یا Trusted Users نامیده می شوند به روز می شود . به طوری که هر بسته ای که در AUR از محبوبیت بیشتری برخوردار باشد توسط Trusted User کامپایل و پکیج می شود و درون این مخازن قرار داده می شود . مخزن Testing مخزنی است که قبل از اینکه بسته ای وارد مخازن Core یا Extra شود ابتدا وارد این مخزن شده و تست می شود و بعد از کامپایل و پکیج شده های چند باره و پایداری کامل به Core یا Extra فرستاده می شوند . پکیج های مخزن Testing بسته هایی هستند که توسط توسعه دهندگان آن به پایداری کامل رسیده اند و به اصطلاح نسخه نهایی آن ها منتشر شده اند ولی ابتدا درون این مخزن میایند تا از پایداری کامل آن ها اطمینان حاصل شود . همچنین نرم افزار هایی که هنوز نسخه نهایی آن ها منتشر نشده است و درحالت Beta Stage یا Alpha Stage هستند به صورت ازمایشی در مخزن Unstable قرار میگیرند و استفاده آن جهت مصرف روزانه پیشنهاد نمی شود . و در نهایت مخزن Community-Testing مخزن تست برای مخزن Community است و به شیوه ای که در بالا گفته شد بسته ها بعد از تست در این مخزن به مخزن Community اضافه می شوند.

## 8-1- انتشار غلطان یا Rolling Release :

همانند Gentoo و بر خلاف توزیع های عمده مثل دبیان ، اوبنتو ، فدورا و ..... آرچ لینوکس برای تاریخ انتشارش زمان بندی خاصی را مشخص نمی کند و در عوض از سیستم انتشار غلطان یا Rolling Release استفاده می کند . همراه با بسته های جدیدی که روزانه فراهم می شود . سیستم مدیریت بسته بندی اش به کاربر اجازه می دهد تا به راحتی سیستمشان را به روز نگه دارند . نبست به انکه کاربران را تشویق به حرکت بین انتشار های جدا از هم بکند انتشار های آرچ لینوکس به سادگی فقط یک تصویر لحظه ای از مجموعه بسته های کنونی آن است . بنابراین هیچ فرقی نمی کند که از کدام انتشار ارچ لینوکس برای نصب استفاده کنید ، بعد از یک به روز رسانی کامل شما اخرین نسخه از آرچ لینوکس را خواهید داشت . به عنوان نتیجه آرچ هرگز نیاز به نصب مجدد برای به روز رسانی ها یا داشتن ویزگی های جدید ندارد . ( آرچ را یکبار برای همیشه نصب کنید )

9-1- سیستم اتوماتیک ABS یا ARCH Build System :

همانند توزیع Gentoo آرچ لینوکس هم سیستم اتوماتیکی برای کامپایل سورس برنامه ها و ایجاد فایل باینری دارد . سیستم ABS به این صورت کار می کند که با استفاده از یک فایل شل اسکریپت ( PKGBUILD ) و ابزار makepkg سورس فایل یک برنامه را گرفته ، آنرا کامپایل کرده و فایل pkg.tar.xz که همان پکیج های باینری آرچ لینوکس است ایجاد می کند. در واقع درون فایل شبه شل اسکریپت که با نام PKGBUILD شناخته می شود مشخصاتی از قبلی نام نرم افزار ، ویرایش ( نسخه منتشر شده ) آن ، کامپایل برای نوع معماری ( i686 یا x86\_64 ) ، وابستگی های سورس برنامه برای اجرا ( dependency ) ، وابستگی های آن برای کامپایل ( make depends ) ، برنامه تداخل گر با آن برنامه ( conflict ) ، آدرس سورس برنامه که با توجه به نوع سورس می تواند یک فایل کمپرس شده از سورس برنامه باشد ( یک فایل tar.gz ) یا یک مخزن git یا svn که از آن ها clone کند ، گزینه ای برای md5sum جهت تعیین صحت و اعتبار سورس برنامه و گزینه ها و option های configure و cmake و make و make install برای کامپایل برنامه و سورس آن . بعد از عملیات کامپایل / پکیج سورس برنامه با استفاده از فایل PKGBUILD و ابزار makepkg فایل باینری ایجاد می شود که با استفاده از ابزار مدیریت بسته pacman قابل نصب بروی روی آرچ لینوکس می باشد . از ویژگی های ABS این است که حتی اگر وابستگی یک سورس بر روی سیستم میزبان نصب نباشد به صورت Auto Dependecny از مخازن رسمی Arch واگر حتی در مخازن هم وابستگی موجود نبود از مخازن AUR (در ادامه توضیح داده خواهد شد ) سورس آنرا گرفته و به همراه سورس برنامه اصلی آنرا کامپایل خواهد کرد . به همین دلیل علی رغم عملیات طاقت فرسای کامپایل در سایر توزیع ها مثل Debian یا ubuntu یا ............. در آرچ کامپایل از سورس برنامه با استفاده از PKGBUILD ها و ابزار makepkg تنها با چند دستور ساده از -خط فرمان امکان پذیر خواهد بود .

## 10-1- مخازن نرم افزاری AUR :

مطمئنا سیستم ABS بدون وجود PKGBUILD ها تقریبا ابزاری قابل استفاده نخواهد بود . به همین منظور علاوه بر مخازن رسمی آرچ که قبلا در مورد آن ها صحبت شد ، مخازن دیگری هم وجود دارد به نام AUR [[1]](#footnote-1) که با آرشیو بیش از 30,000 پکیج به جرات می توان از آن به عنوان یکی از کامل ترین منابع و مخازن نرم افزاری گنو/ لینوکس نام برد . تفاوتی که AUR با سایر مخازن نرم افزاری دارد این است که به جای پکیج های باینری یا سورس برنامه ها در آن فایل های PKGBUILD نگه داری می شود و هر کاربر می تواند با دانلود آن ها و استفاده از سیستم ABS و makepkg بسته مور نیاز خود را کامپایل کند . همچنین این مخازن به صورت عمومی مدیریت می شود و هرکس با ایجاد یک حساب کاربری می تواند PKGBUILD هایی که برا هر بسته ای نوشته است در درون این مخازن آپلود کند. به دلیل اینکه بعضی از پکیج های نرمافزاری بنا بر دلایلی که در ادامه گفته خواهد شد نمی توانند در مخازن رسمی قرار گیرند AUR می تواند مکمل خوبی برای مخازن رسمی آرچ باشد ، ولی به دو دلیل نمی توان مخازن AUR را وارد مخازن رسمی آرچ لینوکس کرد :

1. مشکل مجوز ها : نرم افزار هایی که رایگان هستند اما نمی توان آن ها را بازتوزیع[[2]](#footnote-2) کرد می توانند در AUR قرار گیرند . به صورتی که تنها چیزی که در وبسایت آرچ میزبانی می شود. تنها یک فایل شل اسکریپت است که سورس اصلی نرمافزار را از جای دیگری دانلود می کند نمونه هایی از این نرمافزارهای انحصاری Google Earth و Real Player هستند.
2. بسته های رسمی و بهینه شده : AUR همچنین شامل بسیاری از نسخه های ناپایدار مثل نسخه های بتا از نرم افزار هایی است که نسخه پایدار آن ها ( اما کمی قدیمی تر ) در مخازن رسمی وجود دارد.

## 11-1- پکیج منیجرyaourt[[3]](#footnote-3) :

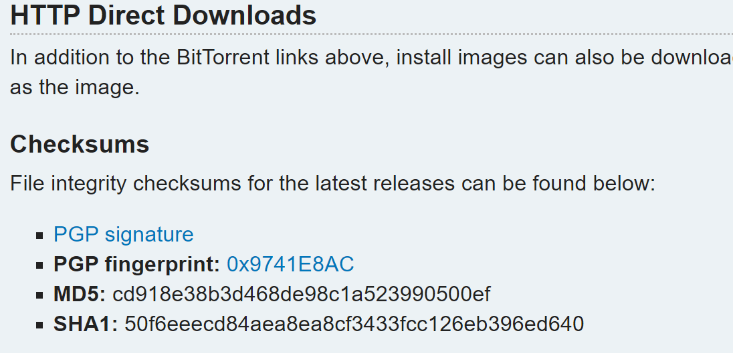
شاید دانلود فایل های PKGBUILD به صورت دستی از مخازن AUR کار چندان جالبی نباشد و مانند سایر task ها در terminal وخط فرمان دانلود آن ها جالب تر به نظر بیاید، همزمان با توسعه pacman به عنوان مدیر بسته پیش فرض آرچ مدیر بسته دیگری به نام yaourt ( شبه پکیج منیجر ) توسعه داده شد که علاوه بر اینکه قابلیت جستجو در مخازن رسمی آرچ را دارا بود در مخازن غیر رسمی AUR هم می تواند جسنجو کند و PKGBUILD های مخازن AUR را از طریق خط فرمان دانلود کند. از ویژگی های yaourt این است که علاوه بر اینکه میتواند PKGBUILD های AUR را از طریق خط فرمان دانلود و fetch کند با سیستم ABS هماهنگ بوده و می تواند بعد از دانلود و ویرایش PKGBUILD ها آن ها را make کند و با کمک pacman آن ها را نصب کند. ولی yaourt با همه این ویژگی ها وبرتری ها هنوز برای نصب پکیج های باینری ساخته شده توسط ABS از pacman استفاده می کند و خود به تنهایی قادر به نصب pkg ها نیست. به همین دلیل pacman را می توان هنوز به عنوان مدیر بسته پیش فرض Arch در نظر گرفت و yaourt جایگزینی برای pacman نمی تواند باشد . به هر حال با توجه به ابزار و امکانات پیشرفته ای که آرچ لینوکس دراختیار کناربر قنرار م ‌یدهند م ‌یتنوان اینگنونه جم ‌عبندی کرد که با توجه با دشواری نصب و راه اندازی آرچ لینوکس)که آن هم با یک بار نصب برای همیشه قابل چشم پوشی است( میتواند گزینه ای مناسب برای کاربرانی باشد که م ‌یخواهند توزیع حنرف ‌های را تجربنه کننند یا تمایل دارند پایداری را به همراه آپدیت بودن و استفاده از آخرین نسخه ی برنام ‌هها تجربه کنند. همچنین آرچ برای برنامه نویسانی که دائم ًا م ‌یخواهند برنام ‌ههای خود و سورس های دیگر توسعه دهنندگان را کامپاینل کننند وتسنت کنند م ‌یتوانند بهترین گزینه باشد. در ضمن رویکرد گیگ ها را به اینن توزینع مثنل توزینع Gentoo را نمی توان نادیده گرفت

# فصل دوم – آماده سازی

1-2- گرفتن اخرین رسانه[[4]](#footnote-4) نصب

آرچ توانایی اجرا بر روی تقریبا تمامی سیستم‌های [x86-64](https://en.wikipedia.org/wiki/X86-64) (پردازنده‌های ۶۴ بیتی) با حد اقل ۵۱۲ مگابایت حافظه دسترسی تصادفی (RAM) را دارا است. نصب پایه‌ای با تمامی بسته‌های مربوط به گروه پایه باید کمتر از ۲ گیگابایت از فضای دیسک را اشغال کند شما می توانید اخرین رسانه آرچ را با توجه به نوعی که می خواهید سیستم بوت شود فایل iso یا netboot image آنرا از صفحه مربوطه[[5]](#footnote-5) با امضای دیجیتالی GnuPG مرتبط به آن را دریافت نمایید

## 2-2- بررسی صحت فایل رسانه :

پیشنهاد می شود قبل از اقدام به نصب ، صحت فایل رسانه را بررسی کنید این کار به چند طریق انجام میشود. روش اول با استفاده از امضای PGP انجام میشود و روش دوم با استفاده از الگوریتم مجموع مقابله ای[[6]](#footnote-6) انجام می شود .

### 1-2-2- بررسی صحت با امضای PGP [[7]](#footnote-7):

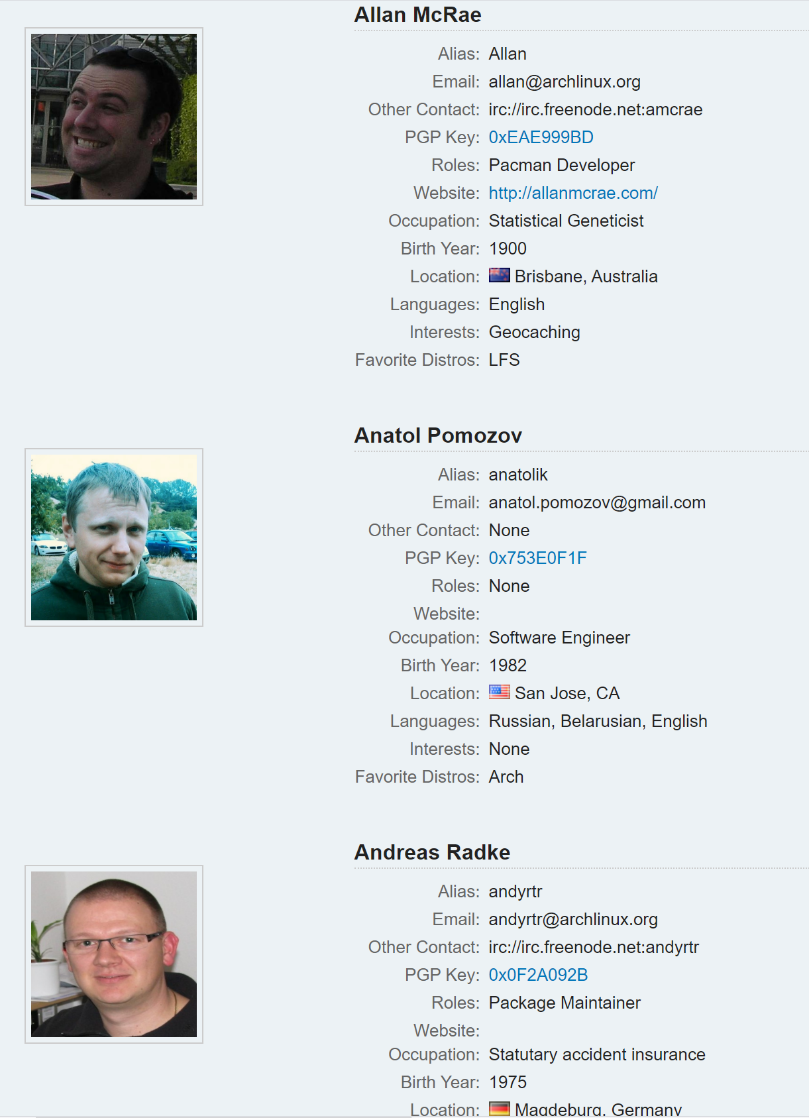
در سیستمی که که GnuPG در آن نصب شده باشد . امضای PGP را از صفحه رسانه آرچ که مطابق شکل بالا می باشد دریافت کنید و با استفاده از برنامه تحت خط فرمان gpg[[8]](#footnote-8) صحت سنجی کنید .

$ gpg --keyserver-options auto-key-retrieve --verify archlinux-version-x86\_64.iso.sig

همچنین اگر اکنون از آرچ استفاده می کنید می توانید با استفاده از مدیر بسته خود آرچ هم صحت سنجی کنید .

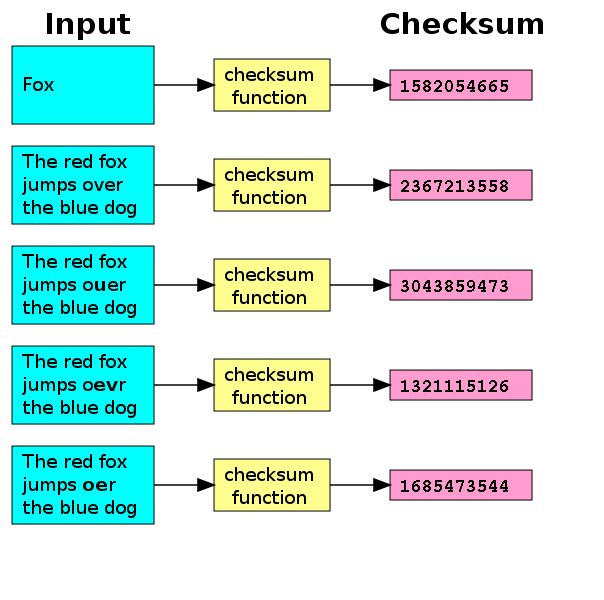
$ pacman-key -v archlinux-version-x86\_64.iso.sig

نکته : خود امضا هم می تواند دستکاری شود اگر از سایت دیگری بجز سایت رسمی[[9]](#footnote-9) اقدام به دریافت نمایید. در این حالت مطمئن شوید کلید عمومی که برای باز کردن امضا استفاده می شود خود توسط کلید قابل اعتماد دیگری امضا شده باشد. برای این منظور هنگام اجرای دستور gpg ( در بالا ) اثر انگشت کلید عمومی[[10]](#footnote-10) که اثر انگشتPGP[[11]](#footnote-11) هم نامیده میشود به شما نمایش داده خواهد شد. برای بررسی صحت اثر انگشت کلید عمومی می توانید از مطابقت دادن آن با اثر انگشت pgp ( همان اثر انگشت کلید عمومی یا کلید PGP ) با اثر انگشتی که در صفحه دانلود رسمی رسانه آرچ درج شده مانند شکل بالا . حالت دیگر برای بررسی صحت امضای PGP مطابقت دادن اثر انگشت کلید عمومی فایل رسانه دریافت شده با اثر انگشت یکی از توسعه دهندگان آرچ لینوکس ، که در صفحه توسعه دهندگان[[12]](#footnote-12) ارچ که به مانند شکل زیر می باشد ( PGP Key )



### 2-2-2- بررسی صحت با استفاده از الگوریتم مجموع مقابله ای یا توابع در هم سازی:

الگوریتم مجموع مقابله ای مانند یک اثر انگشت برای یک فایل عمل می کند به صورت تکنیکالی مجموع مقابله ای ، یک داده ی کوچک از یک داده ی دیجیتالی می باشد که به منظور تشخیص خطا در حین انتقال یا ذخیره سازی از آن مشتق شده است . برای مثال در شکل زیر هر ورودی دیجیتالی که به تابع الگوریتم مجموع مقابله ای داده میشود دارای خروجی مشخصی خواهد بود .



انواع الگوریتم های رمزنگاری یک طرفه را می توان مجموعه ی مقابله ای به حساب اورد . برای مثال md5 یا sha1 را می توان برای اعتبار سنجی رسانه آرچ استفاده کرد .

Sha1sum archlinux-version-x86\_64.iso

خروجی :  50f6eeecd84aea8ea8cf3433fcc126eb396ed640

md5sum archlinux-version-x86\_64.iso

خروجی : cd918e38b3d468de98c1a523990500ef

که با مطابقت دادن خروجی این دستورات با خروجی های صفحه مربوطه می توان فایل رسانه را اعتبار سنجی کرد .

## 3-2- اماده سازی رسانه نصب :

بعد از دانلود رسانه آرچ شما می توانید به صورت های زیر از ان روی سیستم مورد نظر استفاده کنید .

* رایت روی دیسک زنده
* رایت روی فلش
* نصب از طریق

### 1-3-2- رایت روی دیسک زنده :

برای این منظور می توانید از ابزار هایی گرافیکی مانند [Brasero](https://en.wikipedia.org/wiki/Brasero_(software)" \o "wikipedia:Brasero (software)) برای سیستم های لینوکسی و nero برای سیستم ویندوزی استفاده کرد . یکی از ساده ترین راه ها که با فلسفه آرچ نیز در یک راستا می باشد استفاده از ابزار های تحت خط فرمان است . cdrecord برای رایت روی سی دی و growisofs برای رایت روی dvd رو می توان بدین منظور استفاده کرد .

رایت iso روی سی دی :

cdrecord -v -sao dev=/dev/sr0 archlinux.iso

رایت iso روی دی وی دی

growisofs -dvd-compat -Z /dev/sr0=archlinux.iso

### 2-3-2- رایت روی فلش :

برای این منظور میتوانید از ابزار های گرافیکی مانند Rufus برای سیستم های ویندوزی و Multiwriter برای سیستم های لینوکسی استفاده کرد . از ابزار های خط فرمان مانند dd نیز به سادگی می توان استفاده کرد .

# dd bs=4M if=path/to/archlinux.iso of=/dev/sdx status=progress &&flag=sync

### 3-3-2- نصب از طریق شبکه:

PXE بصورت " پیکسی " تلفظ می شود. و به کامپیوتر این امکان را می دهد که از طریق شبکه بصورت Remote بوت شود و این در حالیست که سیستم کلاینت هارد دیسکی ندارد و از طریق یک سرور مجزا در شبکه بوت می شود و سیستم عامل روی آن نصب می شود و آماده به کار می شود. PXE برای اولین بار به عنوان یک فریمورک Wired for Management یا WfM توسط شرکت Intel در سال 1999 معرفی شد. WfM شرکت اینتل امروز توسط Active Management Technology جایگزین شده است اما PXE همچنان ابزاری ارزشمند در نزد متخصصان شبکه در سرتاسر جهان است. بوت کردن سیستم از طریق شبکه معمولا در محیط هایی انجام می شود که سیستم های کلاینت فاقد هارد دیسک هستند و این فرآیند بوت از طریق روتر ها و سرور های مرکزی انجام می شود و سیستم های کلاینت که در اینجا به آنها Thin Client میگوییم بصورت مرکزی مدیریت می شوند. در محیط هایی با پردازش ها و محاسبات متمرکز هزینه های نگهداری کاهش می یابد و امکان افزایش امنیت شبکه و کنترل متمرکز روی سیستم های کلاینت را به ما می دهد. PXE code که توسط سیستم های کلاینت ارائه می شود سیستم های کلاینت را قادر می سازد تا بصورت ریموت Boot شوند و پیکربندی های لازم برای نصب شدن سیستم عامل روی آن انجام شود. لازم به ذکر است که PXE code در تراشه ROM ذخیره می شود. این فرآیند زمینه استفاده از پروتکل های شبکه نظیر UDP ، TFTP ، IP و DHCP را فراهم می کند. مزایای کلیدی استفاده از قابلیت PXE شامل موارد زیر هستند :

* سیستم کلاینت یا همان Workstation نیازی به هارد دیسک یا سایر دستگاه های ذخیره سازی برای بوت شدنش ندارد.
* PXE به Vendor خاصی وابسته نیست. به این معنی که میتوانید براحتی شبکه خود را گسترش دهید و همچنان از PXE استفاده کنید.
* نگهداری از سیستم های کلاینت را آسان می کند زیرا بسیاری از کار ها بصورت Remote انجام می شود.
* ذخیره سازی متمرکز اطلاعات امنیت اطلاعات را برای شبکه فراهم می کند.

در این روش نیاز به یک سرور می باشد .

ابتدا ایمیج را در سرور دانلود کنید :

wget http://hive.ist.unomaha.edu/archlinux/iso/latest/ archlinux-release\_date-x86\_64.iso

در قدم بعدی ایمیج را سوار کنید:

# mkdir -p /mnt/archiso #mount -o loop,ro archlinux-release\_date-x86\_64.iso /mnt/archiso

برای بوت سیستم به این روش پوشه ای به نام archiso در دایرکتوری /mnt/ سرور می سازیم . سپس فایل iso ارچ را در آن مونت می کنیم

2-3-3-2- تنظیمات سرور :

در ابتدا نیاز به تنظیم سرویس DHCP (برای اختصاص آدرس و اعلام پیکربندی) ، سرویس TFTP ( برای انتقال pxelinux و کرنل ) و یک سرویس برای انتقال فایل سیستم مانند ( http server ، NFS یا NBD) می باشد .

شبکه :

Nic را بالا بیاورید و آدرس را به درستی تعیین کنید :

# ip link set eth0 up # ip addr add 192.168.0.1/24 dev eth0

TFTP + DHCP :

شما به هردوی DHCP وTFTP برای پیکر بندی شبکه در مقصد جهت تسهیل در امر رد و بدل کردن اطلاعات در بین سرور و کلاینت PXE نیاز دارید ، dnsmasq هر دو کا را به آسانی انجام می دهد.

dnsmasq را نصب کنید :

# pacman -S dnsmasq

سپس پیکربندی dnsmasq :

# vim /etc/dnsmasq.conf

port=0  
interface=eth0  
bind-interfaces  
dhcp-range=192.168.0.50,192.168.0.150,12h  
dhcp-boot=/arch/boot/syslinux/pxelinux.0  
dhcp-option-force=209,boot/syslinux/archiso.cfg  
dhcp-option-force=210,/arch/  
enable-tftp  
tftp-root=/mnt/archiso

نوبت به اجرای dnsmasq رسیده است:

# systemctl start dnsmasq.service

انتقال فایل سیستم ریشه ی archiso :

با تشکر از (  initcpio hooks archiso\_pxe\_http ، archiso\_pxe\_nfs  initcpio hooks و initcpio hooks archiso\_pxe\_nbd ) در archiso که به کمک این ها می توانید از طریق http ، NFS یا NBD سیستم را بوت کنید ( به ترتیب) ، زمان بوت در هر سه مورد تقریبا یکسان است اما به روش HTTP می توانید وضعیت بارگیری airrootfs.sfs را به صورت درصدی مشاهده نمایید .

HTTP :

از میان همه ی گزینه ها ( مانند apache ) ، darkhttpd  سبک ترین و ساده ترین برای نصب و تنظیم می باشد . ابتدا با استفاده از خط فرمان پکیج مربوطه را نصب نمایید ، سپس با استفاده از darkhttpd ، /mnt/archiso را به عنوان منبع ریشه انتخاب کنید .

# darkhttpd /mnt/archiso

darkhttpd/1.8, copyright (c) 2003-2011 Emil Mikulic.

listening on: http://0.0.0.0:80/

نکته : بسیار مهمه که سرور روی پورت 80 کار کند . اگر darkhttpd بدون دسترسی ریشه اجرا شود ، به صورت پیش فرض به پورت 8080 گوش می دهد ، ولی همچنان کلاینت تلاش می کند که به پورت80 دسترسی پیدا کند ، بدین نتیجه پروسه بوت ناموفق خواهد شد .

:NFS

شما نیاز به تنظیم یک NFS SERVER و پس از مشخص کردن فایل مدیای نصبی[[13]](#footnote-13) که در این مورد خاص /mnt/archiso می باشد به عنوان ریشه ی یا مبدا آن دارید ، بعد از تنظیم سرور به فایل /etc/exports موارد زیر را اضافه کنید .

echo “/mnt/archiso 192.168.0.0/24(ro,no\_subtree\_check)” >> /etc/exports

اگر NFS SERVER در حال اجرا بود می توانید با دستور exportfs -r -a -v فایل سیستم را دوباره به روز رسانی کنید .

انتظار میره که تنظیمات پیش فرض در نصب NFS را در /run/archiso/bootmnt پیدا کنید ، پس شما نیاز به تغییراتی در ویژگی های بوت دارید ، هنگام بوت شدن در کامپیوتر کلاینت روی منوی بوت کلید TAB را فشار داده ، سپس archiso\_nfs\_srv را تغییر دهید

archiso\_nfs\_srv=${pxeserver}:/mnt/archiso

روش جایگزین این است که برای کل فرایند /run/archiso/bootmnt را انتخاب کنید

بعد از لود شدن کرنل ، رسانه راه انداز آرچ فایل سیستم ریشه را از طریق NFS برای بوت شدن کپی خواهد کرد . این مورد ممکن است وقت گیر باشد ، بعد از یکبار لود شدن شما باید یک سیستم در حال اجرا داشته باشید .

نکته : به تنظیمات NFS SERVER را به دلیل حجم در فصل های بعدی خواهیم پرداخت.

NBD :

پکیج nbd را نصب کرده و کانفیگ کنید

pacman -S nbd

/etc/nbd-server/config

[generic]

[archiso]

readonly = true

exportname = /srv/archlinux-release\_date-x86\_64.iso

سپس با استفاده از systemd سرویس nbd را اجرا کنید .

Systemctl start nbd.service

یک PXE SERVER موجود :

اگر در حال حاضر یک PXE SERVER با یک PXELINUX دارید ، می توانید موارد زیر را به /tftpboot/pxelinux.cfg/default برای بوت کردن آرچ با توجه به روش مورد نظر خود استفاده کنید

LABEL archlinux

MENU LABEL Arch Linux x86\_64

LINUX /path/to/extracted/Arch/ISO/arch/boot/x86\_64/vmlinuz-linux

INITRD /path/to/extracted/Arch/ISO/arch/boot/intel-ucode.img,/path/to/extracted/Arch/ISO/arch/boot/amd-ucode.img,/path/to/extracted/Arch/ISO/arch/boot/x86\_64/archiso.img

APPEND archisobasedir=arch archiso\_http\_srv=http://httpserver/path/to/extracted/Arch/ISO/

ip= :::::eth0:dhcp

SYSAPPEND 3

TEXT HELP

Arch Linux 2020.08.01 x86\_64

ENDTEXT

برای NFS و NBD باید به ترتیب پارامتر archiso\_httpd را با archiso\_nfs\_srv ، archiso\_nbd\_srv جایگزین کنید ( مثال کاربردی آن در arch/boot/syslinux/archiso\_pxe.cfg موجود می باشد ) ، هر کدام از روش ها را که انتخاب کنید ، شما باید پارامتر ip= برای یاد دادن به هسته برای ایجاد رابط شبکه قبل از تلاش برای سوار کردن مدیای نصبی از طریق اینترنت وارد کنید ، پارامتر BOOTIF= زمانی الزامی است که چندین رابط سیمی به کامپیوتر کلاینت وصل باشد و یا شما بخواهید resolv.conf در داخل archiso بوت تنظیم شده باشد.

نصب

در این قسمت شما بایستی کشف کنید که چگونه از کلاینت بخواهید راه‌اندازی PXE در الویت قرار بگیرد. در حالت عادی، زمانی که سیستم را راه‌اندازی می‌کنید در گوشهٔ پایین سمت چپ به شما نشان داده خواهد شد که کدام کلید را برای راه‌اندازی PXE با الویت بالاتر فشر دهید. به عنوان مثال در یک IBM x3650 کلید F12 منوی راه‌اندازی را نمایش می‌دهد.

راه‌اندازی http

با نگاهی بر ‎/var/log/messages.log در سرور PXE شما می‌توانید ببینید که در مراحل اولیهٔ راه‌اندازی چه اتفاقی افتاده است

# grep dnsmasq /var/log/messages.log | cut -d' ' -f5-

dnsmasq-dhcp[2544]: DHCPDISCOVER(eth1) 00:1a:64:6a:a2:4d

dnsmasq-dhcp[2544]: DHCPOFFER(eth1) 192.168.0.110 00:1a:64:6a:a2:4d

dnsmasq-dhcp[2544]: DHCPREQUEST(eth1) 192.168.0.110 00:1a:64:6a:a2:4d

dnsmasq-dhcp[2544]: DHCPACK(eth1) 192.168.0.110 00:1a:64:6a:a2:4d

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/pxelinux.0 to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/whichsys.c32 to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_pxe\_choose.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/ifcpu64.c32 to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_pxe\_both\_inc.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_head.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_pxe32.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_pxe64.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/archiso\_tail.cfg to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/vesamenu.c32 to 192.168.0.110

dnsmasq-tftp[2544]: sent /mnt/archiso/arch/boot/syslinux/splash.png to 192.168.0.110

بعد از بارگزاری pxelinux.0 و archiso.cfg از طریق TFTP شما با یک منوی راه‌اندازی syslinux به همراه چندین گزینه مواجه خواهید شد.

نصب بر روی یک ماشین مجازی

نصب بر روی یک ماشین مجازی راه خوبی برای آشنا شدن با آرچ‌لینوکس و پروسهٔ نصب آن بدون خارج شدن از سیستم‌عامل فعلی و پارتیشن‌بندی ذخیره ساز اصلی (به عنوان مثال هارد دیسک شما) است. همچنین با این روش می‌توانید راهنما را در کنار خود داشته باشید. برخی از کاربران هم ترجیح می‌دهند یک سیستم آرچ‌لینوکس به صورت جداگانه و همیشه برای بررسی و آزمایش‌ها بر روی یک ماشین مجازی داشته باشند.

نمونه‌هایی از برنامه‌های شبیه‌ساز عبارتند از VirtualBox, VMware, QEMU, Xen, Varch و  
 Parallels شیوهٔ آماده‌سازی هر ماشین مجازی به نرم‌افزار آن بستگی دارد اما به صورت کلی شامل مراحل زیر می‌شوند:

۱. ساخت دیسک مجازی که میزبانِ سیستم‌عاملِ میهمان می‌شود.

۲. پیکربندی مناسب پارامتر‌های ماشین مجازی.

۳. سوار کردن ایمیج ‎.iso بر روی درایو CD ماشین مجازی.

۴ .به قسمت [راه‌اندازی رسانهٔ نصب](#4.4.راه‌اندازی_رسانهٔ_نصب|outline) بروید

راه‌اندازی رسانهٔ نصب

ابتدا لازم است که الویت راه‌اندازی در بایوس کامپیوتر شما دستکاری شود تا بتوانید رسانهٔ نصب را به درست راه‌اندازی کنید. برای ورود به بایوس به صورت معمول بایستی یکی از کلیدهای Delete, F1, F2, F11 یا F12 را فشار دهید. این کلید معمولاً در صفحهٔ Post که دقیقاً بعد از روشن شدن کامپیوتر نمایش داده می‌شود مشخص است. در پایه‌ای‌ترین حالت و برای نصب از روی لوح فشرده (CD) بایستی الویت راه‌اندازی بر روی CD or DVD قرار بگیرد. در اکثر مادربردهای موجود در ایران قابلیت انتخاب الویت راه‌اندازی با زدن دکمهٔ F12 در هنگام نمایش اطلاعات سیستم (Post) به شما داده می‌شود . بعد از راه‌اندازی رسانهٔ نصب گزینهٔ Boot Arch Linux را برای ورود به نصب انتخاب کرده و کلید Enter را بزنید.

|  |
| --- |
| نکته: شما برای نصب آرچ‌لینوکس به ۶۴ مگابایت رم در پایه‌ای‌ترین حالت نیاز خواهید داشت . است در این مرحله چند ترفند را برای اتصال مستقیم SSH به محیط زندهٔ دیسک انجام دهند |

آزمایش این که آیا رایانهٔ شما در حالت UEFI راه‌اندازی شده؟

در صورتی که شما یک مادربورد UEFI داشته باشید و این قابلیت در آن فعال باشد، CD/USB به صورت خودکار هستهٔ آرچ‌لینوکس را اجرا می‌کند (EFISTUB از طریق Gummiboot Boot Manager). برای بررسی این که آیا رایانهٔ شما در این مود راه‌اندازی شده یا خیر ماژول کرنل efivars را بارگزاری کنید. (قبل از chroot) و سپس بررسی کنید که فایل‌هایی در ‏‎/sys/firmware/efi/vars/‎ وجود دارد:

|  |
| --- |
| # modprobe efivars # before chrooting  # ls -1 /sys/firmware/efi/vars/ |

|  |
| --- |
| نکته: ماژول هسته efivars متغیر‌های UEFI را شناسایی و در ‎/sys/firmware/efi/vars مستقر می‌کنند. این ماژول در طول پروسهٔ راه‌اندازی به صورت خودکار راه‌اندازی نمی‌شود و تا زمانی که این ماژول بارگزاری و هسته در حالت UEFI بدون پارامتر noefi راه‌اندازی شود هیچ فایلی در‎/sys/firmware/efi/vars ساخته نخواهد شد. این متغیرها در ادامه توسط efibootmgr برای اضافه شدن به مدخل بوت‌لودر در منوی راه‌اندازی UEFI تغییر داده می‌شوند. در حالت بایوس، modprobe هیچ‌گونه خطایی در مورد ماژول efivars نخواهد داد. راه صحیح برای شناسایی UEFI چک کردن فایل‌های موجود در این مسیر است: ‎/sys/firmware/efi/vars‎ |

حل مشکلات در حین راه‌اندازی

* اگر شما از چیپست ویدیویی اینتل استفاده می‌کنند و در زمان راه‌اندازی با یک صفحهٔ سیاه خالی مواجه می‌شوید مشکل به احتمال زیاد مربوط به تنظیمات مد هسته (KMS[1](" \l "sdfootnote1sym)) می‌شود. یک راه‌حل راه‌اندازی مجدد سیستم (Restart) و فشار دکمهٔ Tab در حین مشاهدهٔ منوی بوت است. در اینجا شما در آخر رشته‌ای که ظاهر می‌شود عبارت nomodeset را تایپ کنید و Enter را بزنید. راه دوم استفاده از video=SVIDEO-1:d است که اگر کار کند تنظیمات مد هسته را غیرفعال نمی‌کند. برای اطلاعات بیشتر صفحهٔ مروبط به اینتل[2](" \l "sdfootnote2sym) را مشاهده کنید.
* اگر صفحهٔ شما سیاه نشد در عوض سیستم در حین عملیات راه‌اندازی زمانی که در حال بارگزاری هسته بودید قفل کرد باز در حین مشاهدهٔ منوی بوت Tab را فشار دهید و عبارت acpi=off را در انتهای رشتهٔ ظاهر شده تایپ و Enter را بزنید

فصل سوم - نصب

تغییر زبان .

|  |
| --- |
| **ترفند:** این مرحله به صورت اختیاریست و تنها برای آن دسته از کاربرانی که قصد دارند در فایل‌های پیکربندی به زبان خودشان بنویسند استفاده دارد، اگر شما از نشانه‌های خاص برای پسورد وای-فای خود استفاده کرده‌اید یا می‌خواهید پیغام‌های سیستمی (مثلاً خطاها) را به زبان خود دریافت کنید این امکان می‌تواند برای شما مفید واقع شود. |

به صورت پیش‌فرض طرح صفحه‌کلید شما بر روی us تنظیم شده پس اگر شما از کیبورد غیر-US استفاده می‌کنید فرمان زیر را صادر کنید:

|  |
| --- |
| # loadkeys *layout* |

در فرمان بالا *layout* می‌تواند fr, uk, be-latin1 و غیره باشد. برای مشاهده لیست کامل به صفحهٔ مربوط به آن[1](" \l "sdfootnote1sym) مراجعه کنید. فونت‌ها نیز بایستی تغییر کند چرا که بیشتر زبان‌ها بیش از ۲۶ حرف زبان انگلیسی را شامل می‌شوند در نتیجه اگر شما فونت را تغییر ندهید کارکترها به صورت مربع یا نمادهای دیگر در جلوی شما ظاهر خواهند شد پس **دقیقاً** دستور زیر را تایپ کنید:

|  |
| --- |
| # setfont Lat2-Terminus16 |

به صورت پیش‌فرض زبان بر روی انگلیسی-آمریکایی تنظیم شده است. اگر شما قصد عوض کردن زبان را برای پروسهٔ نصب دارید (به عنوان مثال آلمانی) علامت(های) # موجود در جلوی محل مورد نظر در فایل ‎/etc/locale.gen را بردارید. لطفاً ورودی UTF-8 را انتخاب کنید.

از Ctrl + X برای خارج شدن استفاده کنید و زمانی که درخواست ذخیرهٔ تغییرات داده شده کلید Y و سپس Enter را فشار دهید.

|  |
| --- |
| # nano /etc/locale.gen  en\_US.UTF-8 UTF-8  de\_DE.UTF-8 UTF-8 |

|  |
| --- |
| # locale-gen  # export LANG=de\_DE.UTF-8 |

به یاد داشته باشید که Alt+Shift(سمت چپ کیبورد) Keymap را فعال و غیرفعال می‌کند.

**اتصال به اینترنت**

دیمون[1](" \l "sdfootnote1sym) شبکه dhcpcd به صورت خودکار در زمان راه‌اندازی اجرا می‌شود و در صورت وجودِ یک شبکهٔ با سیم تلاش می‌کند تا با دنیای خارج ارتباط برقرار کند. برای اطمینان از تلاش مؤثر این دیمون از ping استفاده کنید:

|  |
| --- |
| # ping -c 3 [www.google.com](http://www.google.com)  PING www.l.google.com (74.125.132.105) 56(84) bytes of data.  64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp\_req=1 ttl=50 time=17.0 ms  64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp\_req=2 ttl=50 time=18.2 ms  64 bytes from wb-in-f105.1e100.net (74.125.132.105): icmp\_req=3 ttl=50 time=16.6 ms  --- www.l.google.com ping statistics ---  3 packets transmitted, 3 received, 0% packet loss, time 2003ms  rtt min/avg/max/mdev = 16.660/17.320/18.254/0.678 ms |

اگر خروجی برخلاف خروجی بالا عبارت ping: unknown host بود متأسفانه تلاش dhcpcd بی‌نتیجه بوده است و شما همانگونه که در ادامه گفته می‌شود بایستی یک اتصال به صورت دستی آماده کنید.

در غیر این صورت به قسمت [آماده سازی درایو ذخیره‌ساز](#5.3.آماده_کردن_درایو_ذخیرساز|outline) بروید.

**با سیم**

این راه را برای تنظیم یک اتصال با سیم از طریق IP address ایستا[1](" \l "sdfootnote1sym) دنبال کنید.

اگر کامیپوتر شما به یک شبکهٔ Ethernet متصل شده، در اغلب مواقع شما یک رابط به اسم eth0 خواهید داشت. اگر شما کارت‌های شبکهٔ بیشتری دارید (سوای کارت شبکه‌ای که به صورت درونی در مین‌بورد شما کار شده است) نام‌های این رابط‌ها به ترتیب eth1 و eth2 و غیره خواهد بود.

شما بایستی این تنظیمات را بدانید:

* Static IP address.
* Subnet mask.
* Gateway's IP address.
* Name servers' (DNS) IP addresses.
* Domain name (مگر این که شما بر روی یک شبکهٔ محلی باشید).

با دستور زیر رابط متصل شده را فعال کنید (به عنوان مثال رابط eth0):

|  |
| --- |
| # ip link set eth0 up |

آدرس را اضافه کنید:

|  |
| --- |
| # ip addr add <ip address>/<subnetmask> dev <interface> |

به عنوان مثال:

|  |
| --- |
| # ip addr add 192.168.1.2/24 dev eth0 |

برای گزینه‌های بیشتر دستور man ip را اجرا کنید.

گِت ‌وِی را به این صورت اضافه کنید:

|  |
| --- |
| # ip route add default via <ip address> |

به عنوان مثال:

|  |
| --- |
| # ip route add default via 192.168.1.1 |

resolv.confرا ویرایش کنید و IP آدرس‌های name serverها و نام دامنهٔ محلی خود را مشخص کنید:

|  |
| --- |
| # nano /etc/resolv.conf  nameserver 61.23.173.5  nameserver 61.95.849.8  search example.com |

|  |
| --- |
| **نکته:** در حال حاضر شما نهایتاً می‌توانید سه nameserver را مشخص کنید. |

اکنون شما بایستی به خوبی به اینترنت متصل شده باشید. اگر نشده‌اید بهتر است جزئیات بیشتری را بررسی کنید

**بی سیم**

این روش را در صورتی که نیاز به اتصال از طریق شبکهٔ بی‌سیم دارید پی بگیرید. راه‌اندازها و ابزارهای وایرلس در محیط زندهٔ مدیای نصب در دسترس قرار دارند. یک شناخت خوب از سخت‌افزار وایرلس شما کلید موفقیت شما در پیکربندی و یک اتصال بی‌دردسر خواهد بود. به خاطر داشته باشید که روشی که در ادامه به آن اشاره می‌کنیم سخت‌افزار وایرلس شما را برای استفاده در محیط زندهٔ مدیای نصب آماده می‌کند. **این مراحل بایستی بعد از نصب در سیستم نصب شده هم اجرا تا اتصال شما برقرار شود.** همچنین توجه داشته باشید که این مراحل اختیاری می‌باشد و چنانچه یک اتصال بی‌سیم در این مرحله مورد نیاز شما نباشد می‌توانید آن را بعداً در سیستم نصب شده انجام دهید.

|  |
| --- |
| **نکته:** در مثال پیش‌ رو، ما از wlan0 به عنوان رابط و linksys برای ESSID استفاده کرده‌ایم. به خاطر داشته باشید که این متغیرها بنابه تنظیمات شما متفاوت می‌باشد. |

راه حل پایه‌ای به این صورت خواهد بود:

* (اختیاری) رابط وایرلس خود را شناسایی کنید:

|  |
| --- |
| # lspci | grep -i net |

* یا اگر از مدل USB استفاده می‌کنید:

|  |
| --- |
| # lsusb |

* مطمئن شوید udev راه‌انداز را بارگیری کرده و راه‌انداز یک رابط هستهٔ وایرلس معمول را با iwconfig ساخته است.

|  |
| --- |
| **نکته:** اگر شما خروجی مشابه را دریافت نمی‌کنید در نتیجه راه‌انداز به درستی بارگیری نشده. در این حالت شما بایستی خودتان راه‌انداز را بارگیری کنید. تنظیمات وایرلس را برای اطلاعات بیشتر مشاهده کنید.[1](" \l "sdfootnote1sym) |

|  |
| --- |
| # iwconfig  lo no wireless extensions.  eth0 no wireless extensions.  wlan0 unassociated ESSID:""  Mode:Managed Channel=0 Access Point: Not-Associated  Bit Rate:0 kb/s Tx-Power=20 dBm Sensitivity=8/0  Retry limit:7 RTS thr:off Fragment thr:off  Power Management:off  Link Quality:0 Signal level:0 Noise level:0  Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0  Tx excessive retries:0 Invalid misc:0 Missed beacon:0 |

* در این نمونه wlan0 در دسترس قرار دارد.
* رابط را بالا بیاورید:

|  |
| --- |
| # ip link set wlan0 up |

* درصد پایینی از چیپست‌های وایرلس علاوه بر راه‌انداز به سفت‌افزار[2](" \l "sdfootnote2sym) نیز احتیاج دارد. اگر چیپست وایرلس شما اینچنین است زمانی که سعی در بالا آوردن رابط دارید با خطایی مشابه این خطا مواجه خواهید شد:

|  |
| --- |
| # ip link set wlan0 up  SIOCSIFFLAGS: No such file or directory |

* اگر مطمئن نیستید از dmesg برای بررسی درخواست سفت‌افزار توسط چیپست وایرلس خود استفاده کنید. مثالی از خروجی یک چیپست اینتل که نیاز به سفت‌افزار دارد و این درخواست را از هسته در زمان راه‌اندازی انجام داده:

|  |
| --- |
| # dmesg | grep firmware  firmware: requesting iwlwifi-5000-1.ucode |

* اگر خروجی مشاهده نمی‌کنید به نظر می‌رسد که چیپست وایرلس شما نیازی به سفت‌افزار ندارد.

|  |
| --- |
| **اخطار:** بستهٔ سفت‌افزار چیپست وایرلس (برای کارت‌هایی که نیاز به آن دارند) در ‎/usr/lib/firmware (محیط زنده) پیش‌نصب شده است **اما بایستی در سیستم اصلی شما برای کارکرد وایرلس بعد از شروع مجدد نصب شود.** نصب بسته در ادامهٔ این کتاب آموزش داده شده. مطمئن شوید که هم ماژول و هم سفت‌افزار قبل از شروع‌مجدد نصب شده باشند. تنظیمات وایرلس را اگر شما از نصب سفت‌افزار مورد نیاز چیپست خاصِ شما اطلاعی ندارید مشاهده کنید.[3](" \l "sdfootnote3sym) |

* در ادامه از منوی وایرلس netcfg برای اتصال استفاده کنید:

|  |
| --- |
| # wifi-menu wlan0 |

* در این مرحله شما بایستی به شبکه متصل شده باشید. اگر نشده‌اید تنظیمات وایرلس را ببینید.

.

1. Arch User Repositories [↑](#footnote-ref-1)
2. Redistribute [↑](#footnote-ref-2)
3. yet another user repository tool [↑](#footnote-ref-3)
4. media [↑](#footnote-ref-4)
5. <https://www.archlinux.org/download/> [↑](#footnote-ref-5)
6. Checksum algorithm [↑](#footnote-ref-6)
7. Pretty Good Privacy [↑](#footnote-ref-7)
8. GNU Privacy Guard [↑](#footnote-ref-8)
9. archlinux.com [↑](#footnote-ref-9)
10. [public key fingerprint](https://en.wikipedia.org/wiki/Public_key_fingerprint) [↑](#footnote-ref-10)
11. PGP fingerprint [↑](#footnote-ref-11)
12. <https://www.archlinux.org/people/developers/> [↑](#footnote-ref-12)
13. archiso [↑](#footnote-ref-13)