**فهرست مطالب**

**عنوان صفحه**

Contents

[مقدمه 1](#_Toc47635421)

[فصل یکم - اشنایی با آرچ 2](#_Toc47635422)

[1-2- تاریخچه آرچ لینوکس 2](#_Toc47635423)

[1-2- اصول راهبردی آرچ 2](#_Toc47635424)

[3-1- اصل KISS : 4](#_Toc47635425)

[1-4- معماری و ساختار آرچ لینوکس : 4](#_Toc47635426)

[**1-6- نصاب آرچ لینوکس :** 5](#_Toc47635427)

[7-1- مدیر بسته Pacman : 6](#_Toc47635428)

**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

## 1-2- 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 بودنیا 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 لینوکس کامپایل از سورس برنامه با استفاده از PKGBUILD ها و ابزار makepkg تها با چند دستور ساده از خط فرمان امکان پذیر خواهد بود .