

آرچ لینوکس

آزادی ، اوپن سورس ، انعطاف

ارائه دهنده :

روزبه شفیعی

Roozbeh@ArchLinux.ir

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 بعضا 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 بکنده 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 می توان به عنوان یک توزیع کاربرپسند برای کاربران خانگی و استفاده کنندگان تجاری (نه گیگ ها) استفاده کنیم یا که خیر!؟

تاریخچه آرچ لینوکس:

قبل از آنکه وارد جزئیات شویم همانند سایر اجزای تشکیل دهنده دنیای آزاد نرم افزار و GNU لازم می دانم 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 می کنم که جایی از آن نوشته شده :

« آرچ لینوکس یک توزیع گنو/لینوکس است که با هدف سبک بودن و ساده بودن ساخته شده است. رویکرد طراحی تیم توسعه بر روی سادگی, ظرافت, صحت کد و کوچک بودن آن گذاشته شده است. سادگی بر مبنای تعریف آرچ: فاقد افزونه ها , تغییرات و پیچیدگی های غیر ضروری آن است و این سادگی از دیدگاه توسعه دهنده و برنامه نویس تعریف می شود , 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 ۲۰۰۲ توسط 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 ۵ سال رهبری آن به عهده جاد وینت بود و زمانی که جاد دیگر وقت کافی برای توسعه آن نداشت ادامه راه را در اول اکتبر ۲۰۰۷ به عهده Aaron Griffin گذاشت که تا به امروز رهبری توسعه آن بر عهده وی می باشد.

Arch way و ۵ اصل راهبردی آرچ لینوکس :

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

(۱) سادگی یا **Simplicity**: همانطور که در ابتدا و مقدمه هم توضیح دادم سادگی آرچ لینوکس به صورت فاقد افزونه ها و تغییرات پیچیدگی های غیر ضروری است به نحوی که یک سیستم شبه یونیکس یا **unix like** را به معنای واقعی کلمه در اختیار کاربر قرار می دهد و کاربر با توجه به نیاز خود سیستم و محیط کاری خود را از پایه (base) می سازد و آرچ و توسعه دهندگان آن چیزی را برای استفاده به کاربر القاء یا تحمیل نمی کنند.

(۲) 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 می کند که به مرور زمان این پیچیدگی ها بعضاً در مواردی باعث به وجود آمدن مشکلاتی در بخش کاربری می شود و این مشکلات می تواند به صورت **Crash** یا باگ های امنیتی ظاهر شود.

(۳) کاربر محور بودن یا **User Centric**: یکی دیگر از اصول ۵ گانه آرچ لینوکس خود مختاری کاربر است. آرچ لینوکس با قرار دادن یک بستر قوی و مستحکم در اختیار کاربر بقیه کار و ساختن 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 کوچک ترین پکیج ها و نرم افزارها را در اختیار خود او باشد تا از نصب اجزای بی مصرف و غیر قابل استفاده برای کاربر جلوگیری شود و این کار این حسن را دارد کاربر با اجزای سیستم آشنا شود و در صورت بروز مشکل بداند برای حل آن باید به کجای سیستم مراجعه کند. آنرا حل کند و اگر احیاناً کاربر دیگری با این مشکل روبه رو شد به او کمک کند و در خدمت جامعه کاربری یا اصطلاحاً **community** باشد. همچنین با این کار از لود ماژول های اضافی در کرنل و پکیج های اضافی جلوگیری می شود و در نتیجه سرعت بیشتری را در پروسه بوت سیستم شاهد خواهیم بود.

(۴) باز بودن یا **Openness**: شاید پایبند بودن به اصل آزادی نرم افزار و اوپن سورس در کمتر توزیع لینوکسی دیده شود. از پچ ها و ماژول هایی که برای کرنل لینوکس نوشته می شود تا درایور سخت افزاری امکان دارد در توزیع های مختلف لینوکس از مشابه سورس بسته آن و نسخه غیر آزاد آن استفاده شود. آرچ با پایبند بودن به این اصل سعی می کند تماماً از نرم افزارهای آزاد و اوپن سورس در سیستم خود و مخازن نرم افزاری استفاده کند.

(۵) آزادی یا **Freedom**: و آخرین اصلی که جز ۵ گانه آرچ لینوکس به شمار می رود اصل آزادی است. آرچ با ساده نگه داشتن سیستم یک محیط خام و به دور از هرگونه نرم افزار و پکیج را در اختیار کاربر قرار می دهد و این آزادی را به طور کامل به او می دهد که چه پکیجی را برای استفاده انتخاب و نصب کند. به عنوان مثال کاربری تمایل دارد از میز کار **KDE** استفاده 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 می دهد آنچه خود می پسندد

استفاده کند نه آنچه توسعه دهندگان به صورت پیش فرض روی سیستم نصب می کنند. خوشبختانه با حجم زیاد و تنوع نرم افزاری مخازن آرچ این آزادی بیشتر قابل لمس است و کاربر می تواند هر آنچه را که می خواهد از Repositories نصب پیدا و نصب کند. در همین رابطه Jud Vinet می گوید :

Arch Linux is what you make it : آرچ لینوکس همان چیزی است که شما می خواهید بسازید

اصل KISS یا Keep it simple and stupid/short :

با توجه با ۵ اصل Archway که گفته شد حالا می توان خیلی راحت تر در مورد اصل KISS صحبت کرد. همانطور که از عبارت keep it simple and stupid/short بر می آید می توان 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 توزیع های دیگر مثل Crux یا Gentoo رعایت می شود در آرچ هم دیده می شود و آرچ همواره این را به کاربر خود متذکر می شود که برای کار با سیستم همیشه آنرا ساده و کوچک نگه دارد و از نصب جزئیات غیر ضروری جلوگیری کند.

معماری و ساختار آرچ لینوکس :

آرچ به طور گسترده ای بر پایه ی بسته های باینری (Binary) یا دودویی است. بسته های باینری در این توزیع با هدف معماری های i686 و x86_64 ساخته می شوند تا سیستمی مدرن و بهینه ای را بسازند. همچنین آرچ از یک سیستم اوتوماتیک برای کامپایل و ساخت بسته ها برای معماری های دیگر به نام ABS یا Arch Build System بهره می برد که در ادامه به آن اشاره خواهد شد.

توجه آرچ به سادگی و اقتصادی بودن سیستم برای توسعه دهندگان در میان مسائل دیگر به این معنی نیست که برای کمک به کاربر برنامه های گرافیکی برای تنظیمات سیستم شناخته می شود. برای مثال مدیر بسته آرچ لینوکس هیچ گونه واسط گرافیکی رسمی ندارد ولی بیشتر روی ساختن و آماده کردن فایل های تنظیمات با حاشیه نویسی خوب و استفاده بهینه از خط فرمان است. این مسأله باعث شد که آرچ به عنوان یک توزیع برای کاربران متوسط و حرفه ای که از کار با خط فرمان نمی ترسند معروف شود. در همین رابطه Aaron Griffin بر روی وبلاگ 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 پیچیده تر خواهید رسید. لایه های انتزاعی که باعث مخفی کردن پیچیدگی ها می شود هیچ گاه کاربردی نبوده اند و در مقابل سیستم باید به نحوی طراحی شود که هیچ گونه مخفی سازی نیاز نباشد.

نصاب آرچ لینوکس :

مدیا های موجود برای نصب آرچ لینوکس هم به صورت ISO (برای رایت بر روی سی دی) و هم به صورت IMG برای نصب روی فلش مموری ها از وب سایت آرچ لینوکس قابل دانلود است. همچنین مدیا های آرچ به دو صورت

Core Install و Net Install موجود است. حالت Core Install به این صورت است که از آن می‌توانید به عنوان یک نصاب Offline استفاده کنید. به این صورت که تمامی پکیج‌ها برای نصب سیستم پایه بر روی آن قرار دارد. در مقابل Net Install تنها یک نصاب بوده و برای نصب پکیج‌ها نیاز به یم خط اینترنت دارد تا پکیج‌های پایه را نصب کند. گزینه Net install این حسن را دارد که در حین نصب از اینترنت می‌توانید تمامی پکیج‌های مورد نیاز را که کاربر تمایل دارد نصب کند و همچنین در حین نصب آخرین نسخه پکیج‌ها 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 کاربر در هنگام نصب سود می‌برد. نصب به صورت پیشفرض تنها یک محیط پایه ی گنو/لینوکس را فراهم می‌کند که شامل GNU Toolchain و هسته لینوکس (Kernel)، چند ماژول اضافی و کتابخانه‌ها 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 می‌گیرد. به همین دلیل نصب آرچ عموماً در مقایسه با سایر سیستم عامل‌ها پیچیده انگاشته می‌شود.

مدیر بسته Pacman :

بر خلاف تصور خیلی‌ها (!) آرچ لینوکس هم مثل سایر توزیع‌ها مانند Fedora ، Ubuntu ، Debian و OpenSuse و... مدیر بسته ای با نام pacman دارد (بدون محیط گرافیکی). نام pacman در واقع الهام گرفته از کاراکتر بازی pacman (کنسول‌های بازیهای ویدئویی) است که این نام به نوعی تداعی کننده ی عبارت Package Manager در ذهن کاربر است. Pacman نصب بسته‌ها، به روزرسانی، حذف و بازگردانی بسته‌ها را انجام می‌دهد و توانایی تشخیص خودکار وابستگی‌ها را دارد. بسته‌های آرچ لینوکس از مخازن بسته‌های آرچ و برای معماری‌های i686 و x86_64 بهینه سازی شده اند و آرچ لینوکس برای رسیدن به سرعت بالا در نصب نرم‌افزارها بر اساس پکیج‌های با فرمت pkg.tar.xz طراحی شده‌اند.

مخازن نرم افزاری یا Repositories :

مخازن نرم افزاری آرچ لینوکس به ۶ بخش، Testing ، Unstable ، Community-Testing ، Core ، Community و Extra تقسیم می‌شود. مخزن Core تنها کرنل لینوکس و آپدیت‌های آن و لایبرری‌ها و پکیج‌هایی که برای نصب پایه ی سیستم نیاز هستند درون آن قرار داده می‌شود. مخزن Extra مخزنی است که نرم افزارهای اضافی مثل xorg-server و میزکارها و ابزارهای کاربردی درون آن قرار داده می‌شود.

در ادامه با مخازن عمومی AUR آشنا خواهیم شد. پکیج‌های این مخازن قابلیت امتیاز گیری از کاربران را دارد. مخزن Community توسط یک سری از کاربران که اصطلاحاً «کاربران مورد اعتماد» یا Trusted Users نامیده می‌شوند به روز می‌شود. به طوری که هر بسته ای که در AUR از محبوبیت بیشتری برخوردار باشد توسط

trusted users کامپایل و پکیج می‌شود و در درون این مخازن قرار داده می‌شود.

مخزن Testing مخزنی است که قبل از اینکه بسته ای وارد مخازن Core یا Extra شود ابتدا وارد این مخزن شده و تست می‌شود و بعد از کامپایل ها و پکیج شدن‌های چند باره و پایداری کامل به Core یا Extra فرستاده می‌شوند. پکیج‌های مخزن Testing بسته‌هایی هستند که توسط توسعه دهندگان آن به پایداری کامل رسیده‌اند و به اصطلاح نسخه نهایی آن‌ها منتشر شده‌اند ولی ابتدا به درون این مخزن می‌آیند تا از پایداری کامل آن‌ها اطمینان حاصل شود.

همچنین نرم افزارهایی که هنوز نسخه نهایی آن‌ها منتشر نشده است و در حالت Beta Stage یا Alpha Stage هستند به صورت آزمایشی در مخزن Unstable قرار می‌گیرد و استفاده آن جهت مصرف روزانه پیشنهاد نمی‌شود.

و در نهایت مخزن Community-Testing مخزن تست برای مخزن Community است و به شیوه ای که در بالا گفته شد بسته‌ها بعد از تست در این مخزن به مخزن Community اضافه می‌شود.

انتشار غلطان یا Rolling Release :

همانند Gentoo و بر خلاف بقیه توزیع‌های عمده مثل دبیان ، اوبونتو ، فدورا و.... آرچ لینوکس برای تاریخ انتشارش زمان بندی خاصی را مشخص نمی‌کند و در عوض از سیستم انتشار غلطان یا Rolling Release استفاده می‌کند. همراه با بسته‌های جدیدی که روزانه فراهم می‌شود ، سیستم مدیریت بسته بندی اش به کاربر اجازه می‌دهد تا به راحتی سیستمشان را به روز نگه‌دارند . نسبت به آنکه کاربران را تشویق به حرکت بین انتشارهای جدا از هم بکند انتشارهای آرچ لینوکس به سادگی فقط یک تصویر لحظه‌ای از مجموعه بسته‌های کنونی آن است. بنابراین هیچ فرقی نمی‌کند که از کدام انتشار آرچ لینوکس برای نصب استفاده کنید، بعد از یک به روز رسانی کامل شما آخرین نسخه از آرچ لینوکس را خواهید داشت. به عنوان نتیجه آرچ هرگز نیاز به نصب مجدد برای به روز رسانی‌ها یا داشتن ویژگی‌های جدید ندارد. (آرچ را یکبار برای همیشه نصب کنید)

سیستم اوتوماتیک ABS یا Arch Build System :

همانند توزیع Gentoo آرچ لینوکس هم سیستم اوتوماتیکی برای کامپایل سورس برنامه‌ها و ایجاد فایل Binary دارد. سیستم ABS به این صورت کار می‌کند که با استفاده از یک فایل شبه شل اسکریپت (PKGBUILD) و ابزار makepkg سورس فایل یک برنامه را گرفته ، آنرا کامپایل کرده و یک فایل pkg.tar.xz که همان پکیج‌های باینری آرچ لینوکس است ایجاد می‌کند.

درواقع درون فایل شبه شل اسکریپت که با نام PKGBUILD شناخته می‌شود مشخصاتی از قبیل نام نرم‌افزار ، ویرایش (نسخه منتشر شده) آن ، کامپایل برای نوع معماری (i686 یا x86_64) ، وابستگی‌های سورس برنامه برای اجرا (dependency) ، وابستگی‌های آن برای کامپایل (make depends) ، برنامه تداخل گر با آن برنامه (conflict) ، آدرس سورس برنامه که با توجه به نوع سورس می‌تواند یک فایل کمپرس شده از سورس برنامه باشد (یک فایل tar.gz) یا یک مخزن git یا svn که از آن‌ها clone کند ، گزینه ای برای md5sum جهت تعیین صحت و اعتبار سورس برنامه و گزینه‌ها و option‌های /configure یا 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 می‌شود که با استفاده از ابزار مدیریت بسته pacman قابل نصب بر روی آرچ لینوکس می‌باشد. از ویژگی‌های ABS این است که حتی اگر وابستگی یک سورس بر روی سیستم میزبان نصب نباشد به صورت Auto Dependency از مخازن رسمی Arch و اگر حتی در مخازن هم بسته 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 علی‌رغم عملیات طاقت فرسای کامپایل در سایر توزیع‌ها مثل debian یا ubuntu یا... در آرچ لینوکس کامپایل از سورس برنامه‌ها با استفاده از PKGBUILD ها و ابزار makepkg تنها با چند دستور ساده در خط فرمان امکان‌پذیر خواهد بود.

مخازن نرم افزاری AUR :

مطمئناً سیستم ABS بدون وجود PKGBUILD ها تقریباً ابزاری قابل استفاده نخواهد بود. به همین منظور علاوه بر مخازن رسمی آرچ که قبلاً در مورد آن‌ها صحبت شد مخازن دیگری هم وجود دارد با نام AUR یا Arch Users Repositories که با آرشیو بیش از ۳۰,۰۰۰ پکیج به جرأت می‌توان از آن به عنوان یکی از کامل‌ترین منابع و مخازن نرم افزاری گنو/لینوکس نام برد. تفاوتی که AUR با سایر مخازن نرم افزاری دارد این است که به جای پکیج‌های باینری یا سورس برنامه‌ها در آن فایل‌های PKGBUILD در آن نگهداری می‌شود و هر کاربر می‌تواند با دانلود آن‌ها و استفاده از سیستم ABS و makepkg بسته مورد نیاز خود را compile کند. همچنین این مخازن به صورت عمومی مدیریت می‌شود و هرکس با ایجاد یک حساب کاربری می‌تواند PKGBUILD هایی که برای هر بسته ای نوشته است در درون این مخازن آپلود کند.

به دلیل اینکه بعضی از پکیج‌های نرم افزاری بنا بر دلایلی که در ادامه گفته خواهد شد نمی‌توانند در مخازن رسمی قرار گیرند AUR می‌تواند مکمل خوبی برای مخازن رسمی آرچ باشد. ولی به دو دلیل نمی‌توان مخازن AUR را وارد مخازن رسمی آرچ لینوکس کرد :

۱) مشکل مجوزها : نرم افزارهایی که رایگان هستند اما نمی‌توان آن‌ها را بازتوزیع (Redistributable) کرد می‌توانند در AUR قرار گیرند. به صورتی که تنها چیزی که در وب سایت آرچ میزبانی می‌شود تنها یک فایل شل اسکریپت است که سورس اصلی نرم‌افزار را از جای دیگر دانلود می‌کند. نمونه‌هایی از این نرم‌افزارهای انحصاری Google Earth و Real Player هستند.

۲) بسته‌های رسمی بهینه شده : AUR همچنین شامل بسیاری از نسخه‌های ناپایدار مثل نسخه‌های بتا از نرم افزارهایی است که نسخه پایدار آن‌ها (اما کمی قدیمی تر) در مخازن رسمی وجود دارد.

پکیج منیجر yaourt یا yet another user repository tool :

شاید دانلود فایل‌های PKGBUILD به صورت دستی از مخازن AUR کار چندان جالبی نباشد و مانند سایر task ها در terminal و خط فرمان دانلود آن‌ها جالب‌تر به نظر بیاید. همزمان با توسعه pacman به عنوان مدیر

بسته پیش فرض آرچ لینوکس مدیر بسته ی 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 جستجو کند و PKGBUILD های مخازن AUR را از طریق خط فرمان دانلود کند. از ویژگی های yaourt این است که علاوه بر اینکه می تواند PKGBUILD های AUR را 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 و برتری 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 Linux در نظر گرفت و تا جایگزینی yaourt با pacman می تواند راه درازی در پیش باشد. (البته شاید)

به هر حال با توجه به ابزار و امکانات پیشرفته ای که آرچ لینوکس در اختیار کاربر قرار می دهد می توان اینگونه جمع بندی کرد که با توجه به دشواری نصب و راه اندازی آرچ لینوکس (که آن هم با یک بار نصب برای همیشه قابل چشم پوشی است) می تواند گزینه ای مناسب برای کاربرانی باشد که می خواهند توزیع حرفه ای را تجربه کنند یا تمایل دارند پایداری را به همراه آپدیت بودن و استفاده از آخرین نسخه ی برنامه ها تجربه کنند. همچنین آرچ برای برنامه نویسانی که دائماً می خواهند برنامه های خود و سورس های دیگر توسعه دهندگان را کامپایل کنند و تست کنند می توانند بهترین گزینه باشد. در ضمن رویکرد گیگ ها را به این توزیع مثل توزیع Gentoo را نمی توان نادیده گرفت.

مرداد ماه ۱۳۸۹. کنفرانس سراسری *Iran LAMP*

نوشته شده توسط روزبه شفیعی, از اعضای تیم توسعه چاکرا گنو/لینوکس و مدیر بخش فارسی زبان آرچ لینوکس

<http://PHPdevelopers.ir>

<http://ArchLinux.ir>

<http://RoozbehOnline.org>