

چگونه سورس کد برنامه ها را روی لینوکس به یک فایل اجرایی تبدیل کنیم؟



گیت هاب بزرگترین مخزن سورس کدهای جهان است. سایتی که تمامی شرکت های نرم افزاری بزرگ و کوچک و توسعه دهندگان حرفه ای و آماتور سورس کد نرم افزارهای منبع باز خود را روی آن قرار می دهند. بیشتر نرم افزارهای منبع بازی که روی گیت هاب در اختیار کاربران قرار دارند، فاقد فایل اجرایی و کامپایل شده اند و کاربر باید سورس کدها را دانلود کند و پس از نصب ملزومات مورد نیاز فایل اجرایی برنامه را ایجاد کند. دو واژه Clone و Build نیز به همین موضوع اشاره دارند. در این مقاله به شما نشان می دهیم که چگونه برنامه ای از سایت گیت هاب را دانلود کنید و روی لینوکس اجرا کنید. اگر برنامه نویس هستید، اما شناختی در مورد نحوه انجام این کار ندارید، این مقاله برای شما است.

آشنایی با مفهوم منبع و منبع باز بودن برنامه ها

یک برنامه کاربردی، مجموعه ای از دستورات متنی است که درون یک محیط توسعه یکپارچه نوشته، ویرایش و ذخیره سازی می شود. برنامه ای که فایل های متنی حاوی دستورات را دریافت و پردازش می کند و یک فایل اجرایی از برنامه در اختیارتان قرار می دهد کامپایلر نام دارد. در برخی موارد نیز سورس کد درون یک مفسر اجرا می شوند و نیازی به کامپایل نیست. فایل های متنی که دستورات برنامه درون آن ها قرار می گیرند سورس برنامه (Source Code) نامیده می شوند. متن باز (Open Source) که این روزها زیاد در مورد آن مطلب یا مقاله می خوانید به نرم افزاری اطلاق می شود که دستورات آن در اختیار همه مردم قرار دارد و کاربران (افراد) که برنامه نویسی می دانند) می توانند بخشی از برنامه را تغییر دهند و آن را کامپایل کنند. در این حالت نسخه ای از برنامه که سفارشی شده و متناسب با نیازهای کاربران است ساخته می شود. بیشتر برنامه های سیستم عامل ویندوز و مک متن باز نیستند. به عبارت دقیق تر برخی از شرکت ها و توسعه دهندگان کدهای منبع محصولات خود را به شکل عمومی منتشر نمی کنند و در نتیجه امکان تغییر نرم افزار وجود ندارد یا اگر امکان پذیر باشد فرآیند پیچیده و مشکلی است. در دنیای متن باز و به ویژه لینوکس داستان کاملاً فرق می کند. نرم افزارها به شکل متن باز و رایگان در اختیار کاربران قرار دارند. البته در برخی موارد شما باید مبلغ ناچیزی برای ویرایش کدها به توسعه دهنده اصلی پرداخت کنید یا موافقت او در ویرایش کدها را کسب کنید.

برنامه های کنترل نسخه

مجموعه ای متشکل از فایل های حاوی سورس کدهای یک پروژه را پایگاه کد (Codebase) می گویند. در پروژه های بزرگ برنامه نویسان مختلفی روی پایگاه داده کار می کنند. هر کدی که تغییر پیدا می کند باید بررسی شود، جزئیات ثبت شود و نسخه های قبلی نگه داری شوند، زیرا در برخی موارد باید اشتباهات برنامه نویسی اصلاح شوند یا کدهای قبلی به دلیل عملکرد بهتر یا ایمن بودن جایگزین کدهای جدید شوند. اگر برنامه نویسان مختلفی در حال ویرایش سورس یک فایل هستند، تغییرات باید به شکل بلادرنگ ثبت شده و ترکیب شوند. برای پاسخ گویی به تمامی این

نیازها، تیم‌های برنامه‌نویسی به نرم‌افزاری که نسخه‌های مختلف را کنترل می‌کند نیاز دارند. یکی از نرم‌افزارهای کنترل نسخه معروف گیت (Git) است. لینوس توروالدز خالق هسته سیستم‌عامل لینوکس، سال‌ها پیش نرم‌افزاری به نام گیت را برای کنترل نسخه و پایگاه داده هسته لینوکس طراحی کرد. امروزه گیت یکی از محبوب‌ترین نرم‌افزارهای کنترل نسخه است که میلیون‌ها توسعه‌دهنده نه فقط برای توسعه هسته لینوکس بلکه برای انجام کارهای خود از آن استفاده می‌کنند. با استفاده از گیت، پایگاه داده یک پروژه در مخازن نرم‌افزاری ذخیره‌سازی می‌شود. مخزن (Repository) سروری است که میزبان فایل‌ها است. توسعه‌دهندگان می‌توانند نسخه‌ای از فایل‌های یک پروژه را روی کامپیوتر خود ذخیره‌سازی کنند و برای ویرایش و ساخت فایل اجرایی از آن استفاده کنند.

مطلب پیشنهادی



راهنمای جامع کار با فایل‌های آرشیو در سیستم‌عامل لینوکس (بخش اول)

گیت‌هاب

گیت‌هاب پس از محبوبیت گیت ساخته شد و به مرور زمان تبدیل به یک مخزن بزرگ شد. در حقیقت، توسعه‌دهندگان نرم‌افزار به سادگی نیاز داشتند تا سورس کدها و فایل‌های پروژه‌های خود را روی آن میزبانی کنند و همچنین امکان استفاده عمومی از سورس کد پروژه‌ها به شکل ساده‌ای فراهم باشد. در فروردین‌ماه 1398 اعلام شد که گیت‌هاب بیش از 100 میلیون مخزن نرم‌افزاری را میزبانی می‌کند. اگر نرم‌افزار متن باز دریافت کردید، به احتمال بسیار زیاد سورس کد آن روی گیت‌هاب قرار دارد، زیرا توسعه‌دهندگان ترجیح می‌دهند از گیت‌هاب برای میزبانی مخزن نرم‌افزار خود استفاده کنند، این کار باعث می‌شود سایر توسعه‌دهندگان کدهای آن‌ها را مشاهده کنند، قابلیت‌های جدیدی به آن‌ها اضافه کنند یا اگر مشکلی در سورس کدها پیدا کردند آن‌را برطرف کنند. در کنار گیت‌هاب، گزینه‌های دیگری شبیه به BitBucket و GitLab هم وجود دارند که البته به اندازه گیت‌هاب بزرگ و محبوب نیستند.

کالبد شکافی یک مخزن

در گیت‌هاب یک مخزن از پوشه‌هایی ساخته می‌شود که درون آن‌ها فایل‌های مختلفی به ویژه سورس کدهای برنامه‌ها قرار دارد. در یک مخزن فایل‌های مختلفی را مشاهده می‌کنید که برخی از آن‌ها مستندات برنامه‌ها، راهنمای برنامه، مجوزهای فایل‌ها، اسکریپت‌های شل و سورس کدهای اصلی یک برنامه هستند. در ارتباط با مخازن و فایل‌ها قاعده خاصی وجود ندارد و ممکن است مخازنی با تنوع مختلف را مشاهده کنید.

مطلب پیشنهادی



هشت توزیع لینوکسی سبک قابل اجرا روی همه سیستم‌ها

فایل مرا بخوان (Readme)

در گیت‌هاب یک فایل سنتی و قدیمی به نام مرا بخوان در اکثر مخازن وجود دارد. این فایل به شکل‌های مختلفی شبیه به readme، ReadMe یا README نوشته می‌شود که در اغلب موارد فرمت فایلی آن md است یا هیچ‌گونه فرمت فایلی ندارد. به طور مثال، اگر به مخزن ویرایشگر Atom مراجعه کنید، فهرستی از فایل‌ها و پوشه‌ها را مشاهده می‌کنید که یکی از فایل‌های درون این مخزن README.md نام دارد. گیت‌هاب به شکل خودکار محتویات فایل readme را در صفحه آغازین یک مخزن قرار می‌دهد. اگر فایل readme فرمت فایلی md داشته باشد از زبان مارک‌دان (Markdown) در ساخت آن استفاده می‌شود. زبان مارک‌دان به توسعه‌دهندگان اجازه می‌دهد از عناصر سبک‌بندی همچون فونت‌ها، تصاویر و پاراگراف‌ها استفاده کنند. شکل 1 نمونه‌ای از یک فایل readme مبتنی بر زبان مارک‌دان را نشان می‌دهد.

Building
<ul style="list-style-type: none"> • Linux • macOS • Windows
Discussion
<ul style="list-style-type: none"> • Discuss Atom on our forums • Chat about Atom on our Slack team => instructions for joining
License
MIT

به طور معمول، فایل‌های readme به عناوینی شبیه به موضوع پروژه، مسئول پروژه، نوع مجوز، نحوه مشارکت در پروژه و روش ساخت و اجرای نرم‌افزار اشاره دارند. ممکن است دستورالعمل ساخت برنامه درون این فایل قید نشده باشد، اما اعلام می‌دارد در چه مکانی این اطلاعات وجود دارند. اطلاعات دیگری که درون این فایل قرار دارند، در رابطه با ابزارهای ساخت و وابستگی‌های برنامه (کتابخانه‌ها و چارچوب‌های موردنیاز) هستند

نحوه کپی کردن فایل

چگونه می‌توانیم یک کپی از مخزن نرم‌افزاری گیت‌هاب (Clone) تهیه کنیم و آن را کامپایل کنیم تا یک فایل اجرایی داشته باشیم؟ در این مقاله نحوه کپی و ساخت نرم‌افزار متن باز Atom Editor که یک محیط توسعه یکپارچه قدرتمند است و برای کامپایل برنامه‌های پایتون از آن استفاده می‌شود را شرح می‌دهیم. در این پروژه از فایل‌ها و پوشه‌های کمی استفاده شده است. فایل ReadMe کوتاه و مختصر است و اطلاعات کافی در اختیارمان قرار می‌دهد. در فایل readme بخشی به نام Development قرار دارد که لینکی به صفحه building from source در آن قرار گرفته است. در این لینک نحوه کامپایل نرم‌افزار Atom Editor تشریح شده است. در لینک فوق به آدرس <https://boxes.thomasjensen.com/build.html> اطلاعات لازم برای کامپایل وجود دارد (شکل 2).

License
Boxes is free software under the GNU General Public License, version 2 (GPLv2).
Development
The website has information on building from source , testing, and contributing to the source code. Boxes has received support from many contributors over the years, and continues to be supported. Many thanks for that!
Running it
Try if boxes works by typing <code>echo foo boxes</code> on the command line. It should give you
<pre> /*****/ /* foo */ /*****/ </pre>

وابستگی‌ها

در فایل توضیح این برنامه بخشی تحت عنوان Building on Linux وجود دارد که نحوه ساخت برنامه را شرح می‌دهد و اعلام می‌دارد Flex و C compiler، Bison، make برای ساخت نرم‌افزار باید روی کامپیوتر نصب شده باشند. به برنامه‌ها و کتابخانه‌هایی که برای ساخت یک برنامه کاربردی به آن‌ها نیاز است وابستگی‌ها (Dependencies) گفته می‌شود (شکل 3).

Building on Linux / UNIX
Just unzip the source archive or clone the GitHub repo. Make sure you have your C compiler installed, including Flex and Bison.
The location of the system-wide config file is compiled in, so if you don't like it's default location, you can edit the top level <i>Makefile</i> and change the value of

در کنار نرم‌افزارها ی یاد شده، به نرم‌افزار گیت هم نیاز است

تا فرآیند کپی مخزن انجام شود. پس از نصب ملزومات، در آخرین مرحله باید دستوراتی که برای کامپایل Atom

Editor در توزیع‌های اوبونتو، فدورا و منجارو قرار دارند را اجرا کنیم. اگر هر یک از نرم‌افزارهای موردنیاز به شکل پیش‌فرض روی این توزیع‌ها قرار ندارد، ابتدا باید نرم‌افزارهای مربوطه را دانلود و نصب کنید و سپس به سراغ کامپایل برنامه بروید. در این مقاله فرض کرده‌ایم یکی از توزیع‌های فوق را روی کامپیوتر خود نصب کرده‌اید.

نصب گیت، فلکس، بیسون و GCC

برای نصب وابستگی‌ها از دستورات لینوکس در محیط ترمینال استفاده می‌کنیم که پیچیدگی خاصی ندارد. ابتدا برنامه ترمینال را اجرا کنید. برای نصب گیت از فرمان زیر استفاده کنید:

```
sudo apt-get install git
```

برای نصب فلکس فرمان زیر را اجرا کنید:

```
sudo apt-get install flex
```

برای نصب bison فرمان زیر را اجرا کنید:

```
sudo apt-get install bison
```

برای نصب make فرمان زیر را اجرا کنید:

```
sudo apt-get install make
```

در توزیع فدورا برای نصب ملزومات موردنیاز برای ساخت Atom Editor از دستورات زیر استفاده کنید:

```
sudo dnf install flex
```

```
sudo dnf install bison
```

```
sudo dnf install make
```

در توزیع Manjaro باید کامپایلر سی (GCC) را نصب کنید. برای نصب ملزومات روی این توزیع دستورات زیر را اجرا کنید:

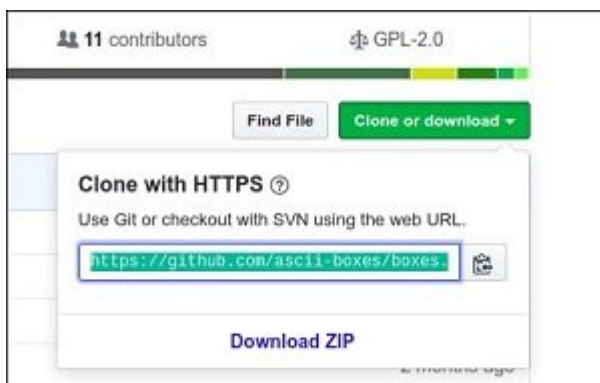
```
sudo pacman -Syu gcc
```

```
sudo pacman -Syu flex
```

```
sudo pacman -Syu bison
```

کپی کردن یک مخزن

هر مخزن گیت‌هاب یک آدرس اینترنتی مخصوص به خود دارد که برای کپی مخزن از طریق نرم‌افزار گیت روی کامپیوتر استفاده می‌شود. برای دسترسی به این آدرس خاص، دکمه سبز رنگ Clone or download که بالای صفحه مخزن برنامه قرار دارد را کلیک کنید. اگر تمایل دارید فایل آرشیوی که تمامی فایل‌های مخزن درون آن قرار دارد را بارگیری کنید، باید از گزینه Download ZIP استفاده کنید، اما برای بهره‌مندی از ابزار گیت، باید لینک کادر فوق را کپی کنید. روی دکمه کپی آدرس در کنار فیلد آدرس کلیک کنید (شکل 4).



در مرحله بعد باید از دستور CD برای رفتن به پوشه‌ای که قرار است محتویات مخزن درون آن کپی شود، استفاده کنیم. اکنون دستور git clone را تایپ کنید و سپس کلیدهای Ctrl+Shift+V را برای کپی آدرس در پنجره ترمینال گنوم فشار دهید (شکل 5).

```
dave@howtogeek:~/work$ git clone https://github.com/ascii-boxes/boxes.  
git
```

در همه
توزیع‌ها
امکان
کپی

آدرس به این شکل وجود ندارد. کپی فایل‌ها روی کامپیوتر آغاز شده و پس از اتمام فرآیند کپی، پیغامی مبنی بر

کامل شدن کار نشان داده می‌شود (شکل 6).

```
dave@howtogeek:~/work$ git clone https://github.com/ascii-boxes/boxes.git
Cloning into 'boxes'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 2405 (delta 15), reused 21 (delta 9), pack-reused 2369
Receiving objects: 100% (2405/2405), 1.28 MiB | 805.00 KiB/s, done.
Resolving deltas: 100% (1467/1467), done.
dave@howtogeek:~/work$
```

فایل‌ها در پوشه‌ای به نام boxes ذخیره می‌شوند. اگر پوشه فوق را باز کنید فهرستی از فایل‌ها و پوشه‌هایی که در صفحه گیت‌هاب قرار دارد را درون پوشه فوق مشاهده می‌کنید (شکل 7).

```
dave@howtogeek:~/work/boxes$ ls
boxes-config      boxes.spec      LICENSE         README.Win32.md  tools
boxes-pkg.openwrt boxes.vim       Makefile       src
boxes.portable.nuspec doc             README.md      test
dave@howtogeek:~/work/boxes$
```

تا این
بخش از
مقاله
موفق
شدیم،
سورس‌کد

ها و سایر فایل‌ها را روی کامپیوتر کپی کنیم و وابستگی‌ها را روی کامپیوتر نصب کنیم. اکنون زمان ساخت برنامه است.

ساخت برنامه کاربردی

برای ساخت برنامه کاربردی باید با توجه به دستورالعملی که درون مخزن گیت‌هاب قرار گرفته کار را ادامه دهید. گاهی اوقات، تنها یک فایل شل ویژه که دستورات درون آن قرار دارد را اجرا می‌کنیم و گاهی اوقات کار بیشتری را باید انجام دهیم. در ارتباط با Atom Editor باید از ابزار make استفاده کنیم. ابزار فوق دستورات درون فایل makefile را خوانده و اجرا می‌کند. دستورات درون این فایل به make اعلام می‌دارند که چگونه برنامه را کامپایل کند. از فرمان make دوبار استفاده می‌کنیم. در اولین اجرا از فرمان فوق برای ساخت برنامه کاربردی استفاده می‌کنیم و در اجرای دوم برای انجام برخی آزمایش‌ها از آن استفاده می‌کنیم. فرمان زیر برنامه Atom Editor را ایجاد و آزمایش می‌کند. دو فرمان از طریق دو کاراکتر && به صورت پشت سر هم در یک خط قرار گرفته و اجرا می‌شوند.

```
make && make test
```

در مدت زمان کامپایل دستورات زیادی در پنجره ترمینال اجرا می‌شوند، اما پس از کامل شدن کار به خط فرمان باز خواهید گشت.

استقرار و اجرای برنامه

برنامه ساخته شده و باید از طریق فایل باینری ایجاد شده آنرا اجرا کرد. مسیری که فایل اجرایی در آن ذخیره شده، همان پوشه‌ای است که تعیین کرده‌اید، اما باید فایل ساخته شده را درون پوشه `usr/bin/` کپی کنید. این کار به شل در پیدا کردن فایل زمانی که قصد استفاده از آنرا داریم کمک می‌کند. اگر از فرآیند کپی کردن صرفنظر کنید، در هر بار اجرا باید از دستور CD برای رفتن به پوشه‌ای که فایل اجرایی درون آن قرار دارد استفاده کنید. در ارتباط با برخی از برنامه‌ها باید ضمن کپی فایل اجرایی، سایر فایل‌ها همچون فایل پیکربندی، راهنما و... را درون filesystem کپی کنید. این نکات در فایل راهنمای برنامه‌ها قید شده است. در ارتباط با Atom Editor مجبور هستیم چنین کاری را انجام دهیم، زیرا در دستورالعمل ساخت برنامه به این موضوع اشاره شده است. برای انتقال فایل‌ها و فایل

اجرای، دستورات زیر باید با سطح دسترسی روت و از طریق فرمان `sudo` اجرا شوند.
فرمان زیر فایل راهنما (man page) را درون پوشه `man1` کپی می‌کند:
`sudo cp doc/boxes.1 /usr/share/man/man1`
فرمان زیر فایل پیکربندی کلی را درون پوشه `/usr/share/` کپی می‌کند:
`sudo cp boxes-config /usr/share/boxes`
دستور زیر فایل اجرایی را درون مسیر `usr/bin/` کپی می‌کند:
`sudo cp src/boxes /usr/bin`

آزمایش برنامه

اکنون برنامه `Boxes` که کامپایل و انتقال داده‌اید را آزمایش کنید. برای باز کردن راهنمای برنامه فرمان زیر را اجرا کنید:

Man boxes

تمامی سوییچ‌های قابل استفاده با فرمان `boxes` همراه با توضیح کوتاهی در پنجره ترمینال نشان داده می‌شوند (شکل 8).

```
boxes(1)                General Commands Manual                boxes(1)

NAME
    boxes - text mode box and comment drawing filter

SYNOPSIS
    boxes [-hlmrv] [-a format] [-d design] [-f file] [-i indent]
    [-k bool] [-p pad] [-s size] [-t tabopts] [infile [outfile]]

DESCRIPTION
    Boxes is a text filter which can draw any kind of box around
    its input text. Box design choices range from simple boxes to
    complex ASCII art. A box can also be removed and repaired,
    even if it has been badly damaged by editing of the text
    inside. Since boxes may be open on any side, boxes can also
    be used to create regional comments in any programming lan-
    guage. New box designs of all sorts can easily be added and
    shared by appending to a configuration file.

    boxes is a command line tool, but also integrates with any
    Manual page boxes(1) line 1 (press h for help or q to quit)
```

برای خارج شدن از صفحه `Manual` کلید `Q` را فشار دهید. اکنون فرمان زیر را اجرا کنید:
`echo How-To Geek | boxes`
با اجرای فرمان فوق خروجی نشان داده شده در شکل 9 را مشاهده می‌کنید.

```
dave@howtogeek:~$ echo How-To Geek | boxes
/*****/
/* How-To Geek */
/*****/
dave@howtogeek:~$ █
```

دستور `boxes` چیست؟

فرمان boxes برای نوشتن متن در کادرها و همچنین محیط ترمینال استفاده می‌شود. شما می‌توانید پس از درج سویچ -d نام فریم موردنظر را وارد کنید تا سبک‌بندی دور متن تغییر کند. به‌طور مثال اگر دو فرمان زیر را اجرا کنید، خروجی شکل 10 را مشاهده می‌کنید.

[echo How-To Geek | boxes -d whirly](#)

[echo How-To Geek | boxes -d c-cmt2](#)

```
dave@howtogeek:~$ echo How-To Geek | boxes -d whirly
.+".+."+."+."+
( How-To Geek )
)          (
(          )
"+."+."+."+."+
dave@howtogeek:~$ echo How-To Geek | boxes -d c-cmt2
/*
* How-To Geek
*/
dave@howtogeek:~$ █
```

کلام آخر
در این
مقاله با
نحوه
کامپایل
سورس
کد
برنامه‌های
لینوکسی
که به
شکل

عمومی درون مخازن گیت‌هاب قرار دارند آشنا شدیم. مراحل ساخت برنامه‌های اجرایی از سورس کدها بدون پیچیدگی خاصی به شرح زیر است:

1. مشاهده دستورالعملی که درون مخزن قرار گرفته است.
 2. بررسی ابزارهای موردنیاز برای نصب و نصب ابزارهایی که روی سیستم نیستند.
 3. کپی کردن مخزن درون کامپیوتر
 4. دنبال کردن دستورالعمل‌های ساخت که اغلب با تایپ ساده فرمان make انجام می‌شود.
 6. کپی فایل‌ها به مکان‌های موردنیازی که درون دستورالعمل به آن‌ها اشاره شده است.
- اگر مراحل که برای ساخت یک برنامه تشریح شده‌اند، مبهم هستند، به سراغ فرم‌ها و انجمن‌های نرم‌افزار بروید و پرسش خود را درون آن فرم‌ها وارد کنید. اگر برنامه وب‌سایتی دارد از بخش Contact Us سایت کمک بگیرید.

تاریخ انتشار:

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/16250/%DA%86%DA%AF%D9%88%D9%86%D9%87-%D8%B3%D9%88%D8%B1%D8%B3%E2%80%8C%DA%A9%D8%AF-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%87%D8%A7-%D8%B1%D8%A7-%D8%B1%D9%88%DB%8C-%D9%84%DB%8C%D9%86%D9%88%DA%A9%D8%B3-%D8%A8%D9%87-%DB%8C%DA%A9-%D9%81%D8%A7%DB%8C%D9%84-%D8%A7%D8%AC%D8%B1%D8%A7%DB%8C%DB%8C-%D8%AA%D8%A8%D8%AF%DB%8C%D9%84-%DA%A9%D9%86%DB%8C%D9%85%D8%9F>