

How to become hacker

Eric S.Raymond

چگونه هکر شویم؟

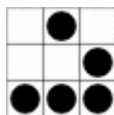
نوشته شده توسط Eric Steven Raymond

فهرست مطالب

- چرا این مقاله
- هکر چیست
- نگرش هکر
- 1. دنیا پر از مشکلات جذابی است که منتظر حل شدن هستند.
- 2. هیچ مسئله ای نباید دو بار حل شود.
- 3. کسالت و سخت گیری زیان آور هستند.
- 4. آزادی خوب است.
- 5. نگرش جایگزین شایستگی نیست.
- مهارت های پایه هک
- 1. برنامه نویسی را یاد بگیرید.
- 2. یکی از یونیکس های متن باز را دریافت کنید و نحوه استفاده و اجرای آن را یاد بگیرید.
- 3. یاد بگیرید از وب جهانی استفاده کنید و HTML بنویسید.
- 4. اگر انگلیسی کاربردی بلد نیستید یاد بگیرید.
- جایگاه در فرهنگ هکرها
- 1. نرم افزار متن باز بنویسید.
- 2. به تست و دیباگ نرم افزارهای متن باز کمک کنید.
- 3. اطلاعات مفید انتشار دهید.
- 4. به کارکرد زیرساخت ها کمک کنید.
- 5. به خود فرهنگ هکر خدمت کنید.
- ارتباط هکر / نرد
- نکاتی برای سبک
- یادداشت تاریخی: هک، متن باز و نرم افزار آزاد

♦ توجه کنید که Free software به صورت نرم افزار آزاد ترجمه می شود نه نرم افزار رایگان.

- [سایر منابع](#)
- [سوالات متداول](#)



چرا این مقاله

به عنوان ویرایش گر **Jargon File** و نویسنده چند مقاله شناخته شده دیگر در زمینه های مشابه، ایمیل های زیادی به من توسط تازه کاران مشتاق در مورد "چطور میتونم یاد بگیرم که هکر جادویی بشم؟" دریافت میکنم. در سال 1996 متوجه شدم که هیچ سوال متداول (FAQs) یا سند وب دیگری وجود ندارد که به این سوال حیاتی پاسخ دهد، پس من این رو شروع کردم. اکنون بسیاری از هکرها آن را قطعی (صریح) می دانند، و من هم همین فکر می کنم. با این حال من ادعا نمی کنم که مرجع انحصاری این موضوع هستم. اگر مطالبی را که در اینجا می خوانید دوست ندارید، مال خودتان را بنویسید.

نمودار پنج نقطه در نه مربع که این مقاله را نزنین میکند گلايدر (glider) نام دارد. این یک الگوی ساده با برخی ویژگی های شگفت انگیز در یک شبیه سازی ریاضی به نام Life است که سال ها هکرها را مجذوب خود کرده است. من فکر می کنم این یک نماد بصری خوب برای هکرها است، در ابتدا کمی مرموز به نظر می رسد، اما دروازه ای به کل جهان با منطق پیچیده خودش است. اطلاعات بیشتر در مورد نشان گلايدر را [اینجا](#) بخوانید.

هکر چیست؟

فایل **Jargon** شامل مجموعه ای از تعاریف از اصطلاح "هکر" است، بیشتر مربوط به مهارت فنی و لذت در حل مشکلات و غلبه بر محدودیت ها است. اگر می خواهید بدانید که چگونه می توانید یک هکر شوید، فقط دو مورد واقعاً مرتبط هستند.

یک جامعه، یک فرهنگ مشترک، از برنامه نویسان و جادوگران شبکه که تاریخچه خود در طول چندین دهه به اولین مینی کامپیوترهای اشتراک گذاری و اولین آزمایش های APRAnet برمی گردانند. اعضای این فرهنگ اصطلاح "هکر" را ابداع کردند. هکرها اینترنت را ساختند، هکرها سیستم عامل یونیکس (Unix) را به آنچه که امروز است تبدیل کردند. هکرها وب جهانی را به کار انداختند. اگر جزئی از این فرهنگ هستید، اگر شما در

آن مشارکت کرده اید و سایر افراد حاضر در آن بدانند که شما کی هستید و شما را هکر خطاب کنند، شما یک هکر هستید.

طرز فکر هکرها به فرهنگ نرم افزار-هکر محدود نمی شود. افرادی هستند که نگرش هکری را به چیزهای دیگری مانند الکترونیک یا موزیک و ... به کار می برند، در واقع شما می توانید آن را در بالاترین سطح هر علم یا هنر بیابید.

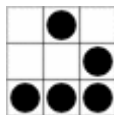
هکرهای نرم افزار این ارواح خویشاوند را در جاهای دیگر می شناسند و ممکن است آن ها را نیز هکر بنامند و برخی ادعا می کنند که ماهیت هکر واقعا مستقل از رسانه خاصی است که هکر در آن کار می کند. اما در ادامه این مقاله به مهارت ها و نگرش های هکرها، نرم افزاری می پردازیم، و سنت های فرهنگ مشترک منشا اصطلاح "هکر" است.

گروه دیگری از افراد هستند که با صدای بلند خود را هکر می نامند، اما اینطور نیست. اینها افرادی هستند (عمدتا پسران نوجوان) که به کامپیوترها نفوذ می کنند سیستم های تلفن را فریب می دهند. هکرهای واقعی این های کرکر (cracker) می نامند و کاری با آن ها ندارند.

هکرهای واقعی بیشتر اوقات فکر می کنند که کرکرها تنبل، بی مسئولیت نه چندان باهوش هستند، و می گویند شکستن امنیت شما را هکر نمی کند همانطور که روشن کردن ماشین بدون سوئیچ شما را مهندس خودرو نمی کند. متأسفانه بسیاری از روزنامه نگاران و نویسندگان فریب خورده اند تا از کلمه "هکر" برای توصیف کرکرها استفاده کنند. این کار هکرهای واقعی را عصبانی میکند.

تفاوت اسای این است: هکرها چیزهایی می سازند، کرکرها آنها را می شکنند.

اگر میخواهید هکر شوید به خواندن این مقاله ادامه دهید. اگر می خواهید کرکر شوید، گروه خبری **alt.2600** را بخوانید و بعد از اینکه فهمیدید آنقدرها هم که فکر می کنید باهوش نیستید، آماده کوبیدن سر خود به در و دیوار باشید. این تمام چیزی است که در مورد کرکرها می گویم.



نگرش هکرها

1. دنیا پر از مشکلات جذابی است که منتظر حل شدن هستند.

2. هیچ مسئله ای نباید دو بار حل شود.
3. کسالت و سخت گیری زیان آور هستند.
4. آزادی خوب است.
5. نگرش جایگزین شایستگی نیست.

هکرها مشکلات را حل می کنند و چیزهایی می سازند و به آزادی و کمک داوطلبانه متقابل اعتقاد دارند. برای پذیرفته شدن به عنوان هکر، شما باید طوری رفتار کنید که انگار خودتان چنین نگرشی دارید. و برای اینکه طوری رفتار کنید که انگار این نگرش دارید، باید واقعاً این نگرش را باور کنید.

اما اگر به پرورش نگرش هکرها به عنوان راهی برای پذیرش در فرهنگ هکرها فکر کنید، این نکته را از دست خواهید داد. تبدیل شدن به فردی که به این ها اعتقاد دارد برای کمک به یادگیری و حفظ انگیزه شما مهم است. مانند تمام هنرهای خلاق، مؤثرترین راه برای استاد شدن، تقلید از طرز فکر استادان است. نه تنها از نظر فکری باید تقلید کرد، بلکه از نظر احساسی نیز باید تقلید کرد.

یا همانطور که در شعر ذن مدرن نیز آمده:

- برای دنبال کردن مسیر:
- به استاد نگاه کن،
- استاد را دنبال کن،
- با استاد قدم بزن،
- از چشم استاد ببین،
- تبدیل شو به استاد.

بنابراین، اگر می خواهید هکر شوید، موارد زیر را تا زمانی که آنها را باور کنید تکرار کنید:

1. دنیا پر از مشکلات جذابی است که منتظر حل شدن هستند

هکر بودن بسیار سرگرم کننده است، اما نوعی سرگرمی است که تلاش زیادی را می طلبد. تلاش انگیزه می خواهد. ورزشکاران موفق انگیزه خود را از نوعی لذت جسمانی در عملکرد بدنشان با عبور از محدودیت های فیزیکی خود می گیرند، به طور مشابه، برای اینکه یک هکر باشید، باید هیجان اصلی را با حل مشکلات، تقویت مهارت های خود و به کار بردن هوش خود داشته باشید.

اگر از آن دسته افرادی نیستید که به طور طبیعی چنین احساسی داشته باشید، برای تبدیل شدن به یک هکر باید این احساس را در خود به وجود آورید. در غیر این صورت

متوجه خواهید شد که انرژی هک شما با عوامل حواس پرتی مانند رابطه جنسی، پول و تایید اجتماعی تضعیف شده است.

(شما همچنین باید نوعی ایمان به ظرفیت یادگیری خود ایجاد کنید، این باور که حتی اگر ممکن است همه چیزهایی را که برای حل یک مشکل نیاز دارید ندانید، اگر فقط به بخشی از آن پردازید و از آن درس بگیرید، خواهید دید. به اندازه کافی برای حل قطعه بعدی یاد خواهیم گرفت و به همین ترتیب، تا زمانی که کارتان تمام شود ادامه دهید.)

2. هیچ مسئله ای نباید دوبار حل شود.

مغز های خلاق ارزشمند و محدود هستند. آنها نباید برای اختراع مجدد چرخ تلف شوند، درحالی که مشکلات جدید و جذابی در انتظار آنها هستند.

برای اینکه مانند یک هکر رفتار کنید باید باور داشته باشید که زمان تفکر سایر هکرها بسیار ارزشمند است به طوری که تقریباً به اشتراک گذاشتن اطلاعات یک وظیفه اخلاقی می باشد، راه حل ها را ارائه دهید تا هکرها دیگر بتوانند به جای اینکه دائماً به مشکلات قدیمی رسیدگی کنند بتوانند موارد جدید را حل کنند.

اما توجه داشته باشید که "هیچ مشکلی نباید دوبار حل شود." به این معنی نیست که شما باید همه راحل های موجود را مقدس بدانید، یا اینکه برای هر مشکلی فقط یک راه حل درست وجود دارد. اغلب با مطالعه اولین بخش راه حل، چیزهای زیادی درمورد مشکلی که قبلاً نمی دانستیم یاد می گیریم. مشکلی ندارد که در اغلب موارد تصمیم بگیریم ما می توانیم بهتر عمل کنیم. چیزی که اشکال دارد فنی مصنوعی (artificial technical)، موانع قانونی یا فنی (مانند کد منبع بسته) است که از استفاده مجدد راه حل خوب جلوگیری میکند و افراد را مجبور می کند چرخ را دوباره اختراع کنند.

(لازم نیست که باور داشته باشید که موظف هستید تمام محصولات خلاقانه خود را ارائه دهید، اگر چه هکریایی که این کار را انجام می دهند آنهایی هستند که بیشترین احترام را از سایر هکرها دریافت میکنند. فروش آن به اندازه ای که برای تامین هزینه های اجاره، خوراک و کامپیوترها با ارزش های هکرها سازگار است. استفاده از مهارت های هک کردن برای حمایت از خوانواده یا حتی ثروتمند شدن خوب است، تا زمانی که وفاداری خود را به هنر خود و هکرها همکار خود در حین انجام آن فراموش نکنید.)

3. کسالت و سخت گیری زیان آور است

هکرها (و به طور کلی افراد خلاق) هرگز نباید خسته شوند یا مجبور به انجام کارهای تکراری احمقانه باشند، زیرا وقتی این اتفاق می افتد به این معنی است که آنها نمی

توانند تنها کاری که می توانند انجام دهند را انجام دهند، حل مشکلات جدید. این اسراف به همه صدمه می زند بنابراین کسالت و سخت گیری نه تنها ناخوشایند بلکه در واقع شر هستند.

برای اینکه مانند یک هکر رفتار کنید، باید این را به اندازه کافی باور کنید تا بخواهید تا جایی که ممکن است بخش های خسته کننده را به صورت خودکار حل کنید، نه فقط برای خودتان بلکه برای دیگران (مخصوصا سایر هکرها).

(یک استثنا آشکار در این مورد وجود دارد. هکرها گاهی اوقات کارهایی را انجام می دهند که ممکن است برای یک ناظر به عنوان تمرینی برای پاکسازی ذهن تکراری یا خسته کننده به نظر برسد، یا به منظور کسب مهارت با تجربه ای خاص که در غیر این صورت نمی توان بدست آورد. اما این با انتخاب است، هرکسی که میتواند فکر کند نباید مجبور به موقعیتی باشد که او را خسته می کند.)

4. آزادی خوب است

هکرها به طور طبیعی ضد استبداد هستند. هرکسی که بتواند برای شما دستور دهد می تواند شما را از حل هر مشکلی که مجرذوب آن هستید باز دارد و باتوجه به روشی که ذهن های مستبدانه کار می کنند، به طور کلی دلایل بسیار احمقانه ای برای این کار پیدا می کنند. بنابراین باید با نگرش اقتدارگرایانه در هر کجا که آن را پیدا کردید مبارزه کرد تا مبادا شما و سایر هکرها را خفه کند.

(این مانند مبارزه با همه اقتدارها نیست. کودکان باید هدایت شوند و مجرمان مهار شوند. یک هکر ممکن است چیزی که می خواهد بدست آورد بیشتر از زمانی است که صرف دستورات می کند، قبول کند که برخی از اقتدارها را بپذیرد. اما این یک معامله محدود و آگاهانه است، تسلیم شخصی که مستبدان میخواهند، پیشنهاد نمی شود.)

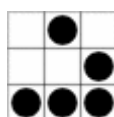
اقتدارگرایان در سانسور و رازداری رشد می کنند. آنها به همکاری داوطلبانه و اشتراک گذاری اطلاعات بی اعتماد هستند فقط " همکاری " را که کنترل می کنند دوست دارند. بنابراین برای اینکه مانند یک هکر رفتار کنید، باید یک خصومت غریزی با سانسور، رازداری و استفاده از زور یا فریب برای وادار کردن بزرگسالان مسئول ایجاد کنید. و شما باید مایل باشید که بر اساس این باور عمل کنید.

5. نگرش جایگزین شایستگی نیست

برای هکر شدن، باید برخی از این نگرش ها را توسعه دهید. اما کنار آمدن با یک نگرش به تنهایی شما را به یک هکر تبدیل نمی کند، بلکه شما را به یک ورزشکار قهرمان یا یک ستاره راک تبدیل می کند. هکر شدن مستلزم هوش، تمرین، فداکاری و کار سخت است.

بنابراین شما باید بیاموزید که نگرش اعتماد نکنید و به هر نوع شایستگی احترام بگذارید. هکرها به افراد توخالی اجازه نخواهند داد زمان خود را هدر دهند، اما آن ها شایستگی را می پرستند به ویژه شایستگی در هک کردن، اما شایستگی در هر چیز ارزش دارد. شایستگی در مهارت هایی که تعداد کمی میتوانند در آن ها تسلط یابند خوب است، شایستگی در مهارت هایی که مستلزم هوشیاری ذهنی، مهارت و تمرکز است، بهترین است.

اگر به شایستگی احترام می گذارید، از پرورش آن در خود لذت خواهید برد، سخت کوشی و فداکاری به جای سخت گیری، به نوعی بازی شدید تبدیل می شود، این نگرش برای تبدیل شدن به یک هکر حیاتی است.



مهارت های پایه هک

1. برنامه نویسی را یاد بگیرید.
2. یکی از یونیکس های متن باز را دریافت کنید و نحوه استفاده و اجرای آن را یاد بگیرید
3. یاد بگیرید از وب جهانی استفاده کنید و HTML بنویسید.
4. اگر انگلیسی کاربردی بلد نیستید یاد بگیرید.

نگرش هکر حیاتی است، اما مهارت ها حیاتی تر هستند. نگرش جایگزینی برای شایستگی نیست، مجموعه ابزار خاصی از مهارت ها وجود دارد که باید قبل از اینکه هر هکری بخواهد شما را یک هکر بخواند، داشته باشید.

این جعبه ابزار در طول زمان به آرامی تغییر می کند زیرا فناوری مهارت های جدیدی ایجاد می کند و مهارت های قدیمی را منسوخ می کند. برای مثال برنامه نویسی شامل زبان ماشین می شد و تا همین اواخر شامل HTML هم نبود. اما در حال حاضر به وضوح شامل موارد زیر است:

1. برنامه نویسی را یاد بگیرید.

البته این مهارت اساسی هک است. اگر هیچ زبان کامپیوتری نمی دانید، توصیه می کنم با پایتون شروع کنید. طراحی تمیز دارد، به خوبی مستند شده و نسبت به مبتدیان مهربان است. با وجود اینکه زبان اول خوبی است، فقط یک اسباب بازی نیست. بسیار قدرتمند و انعطاف پذیر است و برای پروژه های بزرگ مناسب است. من **ارزیابی دقیق تری از پایتون** نوشته ام. آموزش های خوبی در وب سایت **پایتون** موجود است. در **Computer Science Circles** نیز یک آموزش خوب وجود دارد.

من قبلا جاوا را به عنوان یک زبان خوب برای یادگیری زودهنگام توصیه می کردم، اما این انتقاد نظر من را تغییر داد (" The Pitfalls Of Java as a First Programming Language " را جستجو کنید). یک هکر نمی تواند همانطور که آنها، آن را ویرانگر بیان می کنند " مانند لوله کش به مشکل نزدیک شوند". شما باید بدانید که اجزا در واقع چه کاری انجام می دهند. حالا فکر می کنم بهتر است ابتدا C و Lisp و سپس جاوا را یاد بگیرید.

شاید نکته کلی تری در اینجا وجود داشته باشد. اگر زبانی کار زیادی برای شما انجام دهد، ممکن است به طور همزمان ابزار خوبی برای تولید و ابزار بدی برای یادگیری باشد. فقط زبان ها نیستند که این مشکل را دارند. فریم ورک های برنامه های وب مانند، Django، CakePHP، RubyOnRailsT ممکن است رسیدن به یک نوع درک سطحی را بسیار آسان کنند که وقتی مجبور به مقابله با یک مشکل سخت هستید یا حتی فقط راه حل یک مشکل آسان را اشکال زدایی می کنید، شما را بدون منبع می گذارد.

Go جایگزین خوبی برای جاوا می باشد. مهاجرت از پایتون به این زبان نسبتا جدید بسیار آسان است، و یادگیری آن به شما یک قدم جدی در مرحله بعدی احتمالی که یادگیری زبان C است، می دهد. علاوه بر این، یکی از ناشناخته های چند سال آینده این است که Go تا چه حد ممکن است واقعا C را به عنوان یک زبان برنامه نویسی سیستمی جایگزین کند. آینده احتمالی وجود دارد که در آن بیشتر از محدوده سنتی C اتفاق می افتد.

اگر جدی وارد برنامه نویسی شوید، در نهایت باید زبان C، که زبان اصلی یونیکس می باشد را یاد بگیرید. ++C بسیار نزدیک به C است. اگر یکی را بلد باشید، یادگیری دیگری سخت نخواهد بود. با این حال، هیچ یک از این دو زبان برای یادگیری به عنوان اولین زبان خوب نیستند و در واقع، هرچه بیشتر بتوانید از برنامه نویسی در C اجتناب کنید، بهره وری بیشتری خواهید داشت.

C بسیار کارآمد است و از منابع دستگاه شما بسیار کم استفاده می کند. متأسفانه، C با ملزم به انجام بسیاری از مدیریت سطح پایین منابع (مانند حافظه) با دست، این کارایی را به دست می آورد. همه آن کدهای سطح پایین پیچیده و مستعد اشکال هستند و زمان زیادی را برای رفع اشکال صرف می کنند. با ماشین های امروزی که به اندازه کافی

قدرتمند هستند این یک معمولاً این یک انتخاب بد است، استفاده از زبانی که زمان شما را نسبت به زمان کامپیوتر کارآمدتر و هوشمندانه تر استفاده کند، برای مثال پایتون.

زبان های دیگری که برای هکرها اهمیت ویژه ای دارند عبارتند از LISP و Perl. پرل (Perl) به دلایل عملی ارزش یادگیری را دارد. به طور گسترده برای صفحات وب فعال و مدیریت سیستم استفاده می شود، به طوری که حتی اگر هرگز Perl ننویسید، باید خواندن آن را یاد بگیرید. بسیاری از مردم از پرل به روشی استفاده می کنند که من به شما پیشنهاد می کنم از پایتون استفاده کنید تا از برنامه نویسی C در کارهایی که نیازی به کارایی ماشین C ندارند اجتناب کنید. شما باید بتوانید کد آنها را درک کنید.

LISP به دلایل متفاوتی ارزش یادگیری دارد، تجربه روشنگری عمیقی خواهد داشت که در نهایت به دست می آورید. این تجربه شما را به برنامه نویس بهتری برای بقیه روزهایتان تبدیل می کند، حتی اگر از خود LISP زیاد استفاده نکنید. (شما می توانید با نوشتن و تغییر حالت های ویرایش برای ویرایش گر متن Emacs، یا پلاگین های Script-Fu برای GIMP، به راحتی تجربه شروع با LISP را بدست آورید.)

در واقع بهتر است هر پنج زبان Python, Perl, C/C++ و LISP را یاد بگیرید. آنها علاوه بر اینکه مهمترین زبان های هک هستند، رویکرد های بسیار متفاوتی را برای برنامه نویسی نشان می دهند و هرکدام به روش های ارزشمندی به شما آموزش می دهند. Go کاملاً در حدی نیست که بتوان آن را در میان مهم ترین زبان های هک قرار داد، اما به نظر می رسد به سمت این وضعیت پیش می رود.

اما توجه داشته باشید با جمع آوری زبان ها به سطح مهارت یک هکر یا حتی یک برنامه نویس نخواهید رسید، شما باید یاد بگیرید که چگونه در مورد مشکلات برنامه نویسی به طور کلی و مستقل از هر زبانی فکر کنید. برای اینکه یک هکر واقعی باشید، باید به نقطه ای برسید که بتوانید در چند روز یک زبان جدید را با ارتباط دادن درفترچه راهنما با آنچه قبلاً می دانید یاد بگیرید. این بدان معناست که شما باید چندین زبان بسیار متفاوت را یاد بگیرید.

من نمی توانم دستورالعمل های کاملی درباره نحوه یادگیری برنامه نویسی در اینجا ارائه کنم این یک مهارت پیچیده است. اما می توانم به شما بگویم که کتاب ها و دوره ها این کار را نمی کنند. بسیاری، شاید بیشتر بهترین هکرها خود آموز هستند. شما می توانید ویژگی های زبان و بخش هایی از دانش را از کتاب ها بیاموزید، اما ذهنیتی که آن دانش را به مهارت زندگی تبدیل می کند، تنها با تمرین و کارآموزی قابل یادگیری است. کاری که انجام خواهید داد (الف) خواندن کد (ب) نوشتن کد.

پیتر نورویگ (Peter Norvig) که یکی از برترین هکرها گوگل و یکی از نویسندگان پرکاربردترین کتاب درسی در زمینه هوش مصنوعی است، مقاله ای عالی به نام در ده سال برنامه نویسی را به خودتان آموزش دهید (**Teach Yourself Programming in Ten Years**) نوشته است. "دستورالعمل موفقیت برنامه نویسی" او ارزش توجه دقیق را دارد.

یادگیری برنامه نویسی مانند یادگیری نوشتن زبان طبیعی خوب است. بهترین راه برای انجام این کار این است که برخی از مطالب نوشته شده توسط استادان فرم را بخوانید، برخی چیزها را خودتان بنویسید، بیشتر بخوانید، کمی بیشتر بنویسید، بیشتر بخوانید، بیشتر بنویسید... و آنقدر این کار را تکرار کنید تا نوشته شما شروع به توسعه نوعی قدرت و اقتصادی کند که در مدل های خود می بینید.

من در مورد این فرایند یادگیری در " نحوه یادگیری هک " چیزهای بیشتری برای گفتن داشتم. مجموعه ای ساده از دستورالعمل ها است، اما آسان نیست.

یافتن کد خوب برای خواندن قبلا سخت بود، زیرا کد منبع برنامه های بزرگ کمی برای خواندن و سرهم بندی هکهای تازه کار موجود بود. این به طور چشمگیری تغییر کرده است. نرم افزار های متن باز، ابزار های برنامه نویسی و سیستم عامل ها (همه توسط هکرها ساخته شده اند) اکنون به طور گسترده در دسترس هستند. که مرا به موضع بعدی ما می برد ...

2. یکی از یونیکس های منبع باز را دریافت کنید و نحوه استفاده و اجرای آن را یاد بگیرید.

من فرض می کنم شما یک کامپیوتر شخصی دارید یا می توانید به آن دسترسی داشته باشید. (یک لحظه وقت بگذارید و از معنای آن قدردانی کنید. فرهنگ هکر در ابتدا زمانی شکل گرفت که رایانه ها آنقدر گران بودند که افراد نمی توانستند آنها را داشته باشند.) مهم ترین قدمی که هر تازه کار می تواند در جهت کسب مهارت های هکر بردارد، دریافت یک نسخه از لینوکس یا یکی از BSD-UNIX، نصب آن بر روی یک ماشین شخصی و اجرای آن است.

بله، سیستم عامل های دیگری به جز یونیکس در دنیا وجود دارد. اما آنها به صورت باینری توزیع می شوند، شما نمی توانید کد را بخوانید و نمی توانید آن را تغییر دهید. تلاش برای یادگیری هک کردن در مایکروسافت ویندوز یا هر سیستم منبع بسته دیگری مانند تلاش برای یادگیری رقصیدن در حالی که کل بدنتان را گچ گرفته اند است.

در سیستم عامل Mac OS X این امکان وجود دارد، اما تنها بخشی از سیستم منبع باز است، احتمالاً به بن بست های زیادی برخورد خواهید کرد و باید مراقب باشید که عادت بد وابسته بودن به کد اختصاصی اپل را ایجاد نکنید. اگر روی یونیکس تمرکز کنید، زیر کاپوت می توانید چیزهای مفیدی یاد بگیرید.

یونیکس سیستم عامل اینترنت است. درحالی که می توانید بدون دانستن یونیکس استفاده از اینترنت را یاد بگیرید، بدون درک یونیکس نمی توانید هکر اینترنت باشید. به همین دلیل، امروزه فرهنگ هکر ها به شدت یونیکس محور است. (همیشه درست نبوده و برخی از هکر های قدیمی هنوز از این موضوع خوشحال نیستند، اما همزیستی بین یونیکس و اینترنت به اندازه ای قوی شده است که حتی غول مایکروسافت نیز نمی تواند به طو جدی آن را تخریب کند.)

بنابراین یک یونیکس بیاورید، من خودم لینوکس را دوست دارم، اما راه های دیگری هم وجود دارد(و بله، می توانید هم لینوکس و هم ویندوز مایکروسافت را روی یک دستگاه اجرا کنید). یاد بگیرید. آن را اجرا کنید. آن را سرهم کنید. با آن با اینترنت صحبت کنید. کد را بخوانید. کد را اصلاح کنید، شما ابزار برنامه نویسی بهتری (از جمله C , Python, Lisp و Perl) نسبت به هر سیستم عامل مایکروسافتی که رویای میزبانی را در سر می پروراند، دریافت خواهید کرد، شما سرگرم خواهید شد، تا زمانی که به عنوان یک هکر استاد به آن نگاه نکنید، بیش از آنچه که فکر می کنید در حال یادگیری هستید، دانش را جذب خواهید کرد.

برای اطلاعات بیشتر در مورد یادگیری یونیکس، به **Loginataka** مراجعه کنید. همچنین ممکن است بخواهید نگاهی به **هنر برنامه نویسی یونیکس** داشته باشید.

وبلاگ **Let's Go Larval** درچه ای از فرآیند یادگیری یک کاربر جدید لینوکس است که به نظر من به طور غیرعادی شفاف و مفید است. پست **چگونه لینوکس را یادگرفتم** نقطه شروع خوبی است.

برای دستیابی به لینوکس، به Linux Online مراجعه کنید! می توانید از آنجا لینوکس دانلود کنید یا (ایده بهتر) یک گروه کاربری محلی لینوکس پیدا کنید تا در نصب به شما کمک کند.

در طول ده سال اول زندگی این مقاله، من گزارش دادم که از دیدگاه یک کاربر جدید، تمام توزیع های لینوکس تقریباً معادل هستند. اما در سال 2006-2007، بهترین انتخاب واقعی ظاهر شد: اوبونتو. در حالی که سایر توزیع ها حوزه های قدرت خاص خود را دارند، اوبونتو در دسترس ترین توزیع برای مبتدیان لینوکس است. اما مراقب رابط دسکتاپ "Unity" که وحشتناک و تقریباً غیرقابل استفاده است باشید که چند سال بعد اوبونتو به

عنوان رابط دسکتاپ پیش فرض معرفی کرد. نوع Xubuntu یا Kubuntu انتخاب بهتری هستند.

می توانید راهنما و منابع BSD Unix را در www.bsd.org بیابید.

یک راه خوب برای شروع کردن، بوت کردن چیزی است که طرفداران لینوکس آن را سی دی زنده (live CD) می نامند، توزیعی که بدون نیاز به تغییر هارد دیسک شما به طور کامل از روی سی دی یا USB اجرا می شود. این ممکن است کند باشد، زیرا سی دی ها کند هستند، اما راهی برای بررسی احتمالات بدون نیاز به انجام کار شدید است.

من یک مقدمه (پرایمر) در مورد **اصول اولیه یونیکس و اینترنت** نوشته ام.

من توصیه می کردم اگر تازه کار هستید، لینوکس یا BSD را به عنوان پروژه انفرادی نصب نکنید. امروزه نصب کننده ها (installers) آنقدر خوب شده اند که انجام آن به طور کامل به تنهایی امکان پذیر است، حتی برای یک تازه کار. با این وجود، من همچنان توصیه می کنم با گروه کاربر لینوکس محلی خود تماس بگیرید و درخواست کمک کنید. نمی تواند صدمه بزند، و ممکن است روند را هموار کند.

3. یاد بگیرید از وب جهانی استفاده کنید و HTML بنویسید.

بسیاری از چیزهایی که فرهنگ هکر ساخته است کار خود را دور از چشم انجام می دهند و به اداره کارخانه ها، دفاتر و دانشگاه ها کمک می کنند بدون اینکه تأثیر آشکاری بر نحوه زندگی غیر هکرها داشته باشند. وب تنها استثنای بزرگ است، اسباب بازی براق و عظیم هکری که حتی سیاستمداران نیز اذعان دارند که دنیا را تغییر داده است. تنها به همین دلیل (و همچنین بسیاری از موارد خوب دیگر) باید نحوه کار با وب را بیاموزید.

این فقط به معنای یادگیری نحوه هدایت مرورگر نیست (هرکسی می تواند این کار را انجام دهد)، بلکه به معنای یادگیری نحوه نوشتن HTML، زبان نشانه گذاری وب است. اگر برنامه نویسی بلد نیستید، نوشتن HTML برخی عادات ذهنی را به شما یاد می دهد که به یادگیری شما کمک می کند. بنابراین یک Home page بسازید. اما داشتن یک Home page به اندازه کافی خوب نیست که شما را به یک هکر تبدیل کند. وب پر از صفحات اصلی است. بسیاری از آنها لجن بی معنی و با محتوای صفر هستند - لجن بسیار جذاب به نظر می رسد، اما توجه داشته باشید که لجن به لجن است (برای اطلاعات بیشتر در مورد این به **صفحه جهنم HTML** مراجعه کنید).

برای ارزشمند بودن، صفحه شما باید دارای محتوا باشد، باید برای سایر هکرها جالب و یا مفید باشد. و این ما را به موضوع بعدی می برد ...

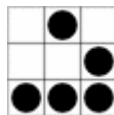
4. اگر انگلیسی کاربردی بلد نیستید یاد بگیرید

من به عنوان یک آمریکایی و بومی انگلیسی زبان، قبلاً تمایلی به پیشنهاد این موضوع نداشتم، مبادا به عنوان نوعی امپریالیسم فرهنگی تلقی شود. اما چندین نفر با زبان مادری زبان‌های دیگر از من خواسته‌اند که به این نکته اشاره کنم که انگلیسی زبان کاری فرهنگ‌ها و اینترنت است و برای عملکرد در جامعه‌ها باید آن را بدانید.

در حوالی سال 1991 متوجه شدم که بسیاری از هکرهایی که زبان انگلیسی را به عنوان زبان دوم دارند، از آن در بحث‌های فنی استفاده می‌کنند، حتی زمانی که یک زبان مادرزادی مشترک دارند. در آن زمان به من گزارش شد که انگلیسی واژگان فنی غنی‌تری نسبت به هر زبان دیگری دارد و بنابراین ابزار بهتری برای این کار است. به دلایل مشابه، ترجمه کتاب‌های فنی که به زبان انگلیسی نوشته شده‌اند، اغلب رضایت بخش نیستند (در صورتی که اصلاً انجام شوند).

لینوس توروالدز، که یک فنلاندی است، کد خود را به انگلیسی کامنت گذاری کرد (ظاهراً هرگز به ذهنش خطور نکرده بود که خلاف آن عمل کند). تسلط او به زبان انگلیسی عامل مهمی در توانایی او برای استخدام یک جامعه جهانی از توسعه دهندگان برای لینوکس بوده است. مثالی است که ارزش دنبال کردن را دارد.

انگلیسی زبان بومی بودن، تضمین نمی‌کند که مهارت‌های زبانی شما به اندازه کافی خوب باشد تا به عنوان یک هکر عمل کنید. اگر نوشته‌های شما نیمه با سواد، غیر دستوری و مملو از غلط املایی باشد، بسیاری از هکرها (از جمله خود من) تمایل دارند شما را نادیده بگیرند. در حالی که شلخته نویسی همیشه به معنای تفکر شلخته نیست، ما به طور کلی همبستگی را قوی می‌دانیم و ما هیچ فایده‌ای برای متفکران شلخته نداریم. اگر هنوز نمی‌توانید با مهارت بنویسید، یاد بگیرید.



وضعیت در فرهنگ هکرها

1. نرم افزار متن باز بنویسید.
2. به تست و دیباگ نرم افزار های متن باز کمک کنید.
3. به کارکرد زیرساخت ها کمک کنید.
4. به خود فرهنگ هکر خدمت کنید.

مانند بسیاری از فرهنگ‌های بدون اقتصاد پولی، فرهنگ هکر بر اساس شهرت است. شما در حال تلاش برای حل مشکلات جالب هستید، اما اینکه چقدر آنها جالب هستند، و اینکه آیا راه حل‌های شما واقعاً خوب هستند، چیزی است که فقط همتایان فنی یا مافوق شما معمولاً توانایی قضاوت دارند.

بر این اساس، وقتی بازی هکر را انجام می‌دهید، یاد می‌گیرید که امتیاز خود را در درجه اول بر اساس آنچه که سایر هکرها در مورد مهارت شما فکر می‌کنند حفظ کنید (به همین دلیل است که شما واقعاً یک هکر نیستید تا زمانی که سایر هکرها به طور مداوم به شما هکر بگویند). واقعیت این است که هک با تصویری به عنوان کار انفرادی پوشیده شده است. همچنین توسط یک تابو فرهنگی هکری (که به تدریج از اواخر دهه 1990 رو به زوال است، اما هنوز هم قوی است) مخالف اعتراف به دخیل بودن نفس یا اعتبار خارجی در انگیزه فرد است.

به طور خاص، فرهنگ هکر چیزی است که مردم شناسان آن را فرهنگ هدیه می‌نامند. شما در آن موقعیت و شهرت را با بخشش چیزها بدست می‌آورید نه با تسلط به دیگران، نه با زیبا بودن و نه با داشتن چیزهایی که دیگران می‌خواهند. به طور مشخص، با صرف زمان، خلاقیت و نتایج مهارت خود.

اساساً پنج نوع کار وجود دارد که می‌توانید انجام دهید تا مورد احترام هکرها قرار بگیرید:

1. نرم افزار متن باز بنویسید.

اولین کار (مرکزی ترین و سنتی ترین) این است که برنامه‌هایی بنویسید که سایر هکرها فکر می‌کنند سرگرم کننده یا مفید هستند و منابع برنامه را در اختیار کل فرهنگ هکر قرار دهید تا از آنها استفاده کنند.

(ما قبلاً این آثار را «نرم افزار آزاد (free software)» می‌نامیدیم، اما بسیاری از افراد که دقیقاً معنی «آزاد» را نمی‌دانستند گیج می‌شدند. اکثر ما اکنون اصطلاح «نرم افزار متن باز (open source)» را ترجیح می‌دهیم).

مورد احترام ترین نیمه خدایان Hackerdom افرادی هستند که برنامه‌های بزرگ و توانمندی نوشته‌اند که نیازهای گسترده‌ای را برآورده کرده و آنها را ارائه کرده‌اند، به طوری که اکنون همه از آنها استفاده می‌کنند.

اما یک نکته تاریخی خوب در اینجا وجود دارد. در حالی که هکرها همیشه به توسعه دهندگان منبع باز در میان آنها به عنوان سخت ترین هسته جامعه ما نگاه می‌کردند، قبل از اواسط دهه 1990 اکثر هکرها بیشتر اوقات روی منبع بسته کار می‌کردند. این

هنوز درست بود زمانی که من اولین نسخه این مقاله را در سال 1996 نوشتم. جریان اصلی نرم افزار منبع باز پس از سال 1997 برای تغییر شرایط لازم بود. امروزه، «جامعه هکرها» و «توسعه دهندگان منبع باز» دو توصیف برای فرهنگ و جمعیت یکسانی هستند، اما شایان ذکر است که همیشه اینطور نبوده است. (برای اطلاعات بیشتر در این مورد، به بخش "[یادداشت تاریخی: هک، منبع باز و نرم افزار رایگان](#)" مراجعه کنید).

2. به تست و دیباگ نرم افزار های متن باز کمک کنید.

آنها همچنین به کسانی می ایستند و نرم افزار متن باز را دیباگ می کنند خدمات می دهند. در این دنیای ناقص، ما به ناچار بیشتر زمان توسعه نرم افزار خود را در مرحله اشکال زدایی صرف خواهیم کرد. به همین دلیل است که هر نویسنده منبع باز که فکر می کند به شما خواهد گفت که آزمایش کننده های بتا خوب (که می دانند چگونه علائم را به وضوح توصیف کنند، مشکلات را به خوبی بومی سازی کنند، می توانند اشکالات را در انتشار سریع تحمل کنند و مایل به اعمال چند روال تشخیصی ساده هستند) ارزش آنها برابر یاقوت هم وزن آن ها است. حتی یکی از اینها می تواند بین مرحله اشکال زدایی که یک کابوس طولانی و طاقت فرسا است و مرحله ای که صرفاً یک مزاحمت مفید است تفاوت ایجاد کند.

اگر تازه کار هستید، سعی کنید برنامه در دست توسعه ای را پیدا کنید که به آن علاقه مند هستید و یک آزمایش کننده بتا خوبی باشید. یک روند طبیعی از کمک به برنامه های آزمایشی تا کمک به اشکال زدایی آنها تا کمک به اصلاح آنها وجود دارد. از این طریق چیزهای زیادی یاد خواهید گرفت و با افرادی که بعداً به شما کمک خواهند کرد، کارما خوبی ایجاد خواهید کرد.

3. اطلاعات مفید انتشار دهید.

یک چیز خوب دیگر این است که اطلاعات مفید و جالب را در صفحات وب یا اسنادی مانند لیست سوالات متداول (FAQ) جمع آوری و فیلتر کنید و آنها را به طور کلی در دسترس قرار دهید.

نگهدارنده های سوالات متداول فنی اصلی تقریباً به اندازه نویسندگان منبع باز مورد احترام هستند.

4. به کارکرد زیر ساخت ها کمک کنید.

فرهنگ هکر (و مهندسی توسعه اینترنت، برای آن موضوع) توسط داوطلبان اداره می شود. کارهای ضروری اما غیر جذاب زیادی وجود دارد که باید انجام شود تا ادامه یابند از

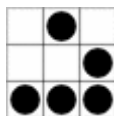
جمله مدیریت لیست های پستی، اداره گروه های خبری، نگهداری سایت های آرشیو نرم افزاری بزرگ، توسعه RFC ها و سایر استانداردهای فنی.

افرادی که این جور کارها را به خوبی انجام می دهند، احترام زیادی دریافت می کنند، زیرا همه می دانند که این شغلها زمان زیادی می گیرند و به اندازه بازی با کد سرگرم کننده نیستند. انجام آنها نشان از فداکاری دارد.

5. به خود فرهنگ هکر خدمت کنید.

در نهایت، می توانید به خود فرهنگ هک خدمت کرده و آن را رواج دهید (مثلاً با نوشتن یک مقاله دقیق در مورد نحوه هکر شدن :-). این کاری نیست که در اوایل انجام دهید شما باید به خاطر یکی از چهار چیز اول به خوبی شناخته شوید و مدتی در این اطراف حضور داشته باشید.

فرهنگ هکر دقیقاً رهبر ندارد، اما قهرمانان فرهنگی و بزرگان قبیله و مورخان و سخنگویان دارد. وقتی به اندازه کافی در سنگر بوده اید، ممکن است به یکی از اینها تبدیل شوید. مراقب باشید: هکرها به منیت آشکار در بزرگان قبیله خود بی اعتماد هستند، بنابراین به وضوح، رسیدن به این نوع شهرت خطرناک است. به جای تلاش برای آن، باید خود را طوری قرار دهید که در دامن شما بیفتد، و سپس نسبت به موقعیت خود متواضع و مهربان باشید.



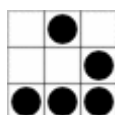
ارتباط هکر/نرد

برخلاف افسانه رایج، برای هکر شدن نیازی نیست که یک نرد باشید. با این حال، این کمک می کند و بسیاری از هکرها در واقع نرد هستند. چیزی شبیه یک طرد شده اجتماعی بودن به شما کمک می کند تا روی چیزهای بسیار مهم مانند فکر کردن و هک کردن تمرکز کنید.

به همین دلیل، بسیاری از هکرها برچسب "گیک" را به عنوان نشان افتخار انتخاب کرده اند، این راهی برای اعلام استقلال آنها از انتظارات اجتماعی عادی است (و همچنین علاقه مند بودن به چیزهای دیگری مانند بازی های علمی تخیلی و استراتژی که اغلب با هکر بودن همراه است). در دهه 1990 از اصطلاح "nerd" استفاده می شد. زمانی که "nerd" یک تحقیر کننده خفیف و "گیک" نسبتاً خشن تر بود. مدتی پس از سال 2000 آنها مکان خود را تغییر دادند، حداقل در ایالات متحده. فرهنگ عامه پسند، و در حال حاضر حتی یک فرهنگ غرور گیک در میان افرادی که اهل فن نیستند وجود دارد.

اگر بتوانید به اندازه کافی روی هک تمرکز کنید تا در آن خوب باشید و همچنان یک زندگی داشته باشید، اگر بتوانید به اندازه کافی روی هک تمرکز کنید تا در آن خوب باشید و همچنان یک زندگی داشته باشید، خوب است. این امروز بسیار ساده تر از زمانی است که من در دهه 1970 شروع کرده بودم، فرهنگ جریان اصلی اکنون برای افراد تکنولوژیک بسیار دوستانه تر است. حتی تعداد فزاینده‌ای از افرادی وجود دارند که متوجه می‌شوند هکرها اغلب همسرانی خوب و افراد با عشق هستند.

اگر به دلیل نداشتن زندگی مجذوب هک شدید، این نیز اشکالی ندارد حداقل برای تمرکز مشکلی نخواهید داشت. شاید بعداً یک زندگی بدست آورید.



نکاتی برای سبک

باز هم برای هک شدن باید وارد ذهنیت هکر شوید. وقتی پشت کامپیوتر نیستید می‌توانید کارهایی را انجام دهید که به نظر می‌رسد کمک کننده باشد. آنها جایگزین هک نیستند (هیچ چیز نیست) اما بسیاری از هکرها آنها را انجام می‌دهند، و احساس می‌کنند که آنها به نوعی با ماهیت هک ارتباط برقرار می‌کنند.

- یاد بگیرید که زبان مادری خود را به خوبی بنویسید. اگرچه این یک تصور کلیشه ای رایج است که برنامه نویسان نمی‌توانند بنویسند، تعداد شگفت انگیزی از هکرها (از جمله تمام هکرهایی که من می‌شناسم) نویسندگان بسیار توانا هستند.
- مطالب علمی تخیلی بخوانید. به کنوانسیون های علمی تخیلی بروید (راه خوبی برای ملاقات با هکرها و هکرهای اولیه است).
- به یک فضای هکر بپیوندید و چیزهایی بسازید (یک راه خوب دیگر برای ملاقات با هکرها و هکرهای اولیه).
- هنرهای رزمی تمرین کنید. به نظر می‌رسد نوع انضباط ذهنی مورد نیاز برای هنرهای رزمی از جنبه های مهمی شبیه به کاری است که هکرها انجام می‌دهند. محبوب ترین هنرهای رزمی در بین هکرها قطعاً هنرهای دست خالی آسیایی مانند تائه کوان دو، اشکال مختلف کاراته، کونگ فو، آیکیدو یا جو جیتسو هستند. هنرهای شمشیربازی غربی و هنرهای شمشیر آسیایی نیز پیروان قابل مشاهده ای دارند. در جاهایی که قانونی است، تیراندازی با تپانچه از اواخر دهه 1990 در حال محبوب شدن است. هکری ترین هنرهای رزمی آنهایی هستند که بر نظم ذهنی، آگاهی آرام، و کنترل دقیق تأکید دارند تا قدرت خام، ورزشکاری یا سرسختی بدنی.

- یک رشته مدیتیشن واقعی را مطالعه کنید. مورد علاقه همیشگی در میان هکرها زن است (نکته مهم این است که می توان از زن بدون کسب دین یا دور انداختن دینی که قبلاً دارید بهره برد). سبک های دیگر نیز ممکن است کارساز باشند، اما مراقب باشید یکی را انتخاب کنید که نیازی به باور چیزهای دیوانه کننده نداشته باشد.
- یک گوش تحلیلی برای موسیقی ایجاد کنید. یاد بگیرید که از انواع خاص موسیقی قدردانی کنید. یاد بگیرید که برخی از آلات موسیقی را به خوبی بنوازند یا چگونه آواز بخوانید.
- درک خود را از جناس و بازی با کلمات توسعه دهید.

هرچه تعداد بیشتری از این کارها را انجام می دهید، به احتمال زیاد شما یک هکر طبیعی هستید. چرایی این چیزها به طور خاص کاملاً روشن نیست، اما آنها با ترکیبی از مهارت های چپ و راست مغز مرتبط هستند که به نظر می رسد مهم است. هکرها باید بتوانند هم منطقی استدلال کنند و هم در یک لحظه از منطق ظاهری یک مشکل خارج شوند.

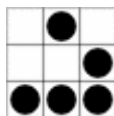
به همان شدتی که بازی می کنید کار کنید و به همان شدتی که کار می کنید بازی کنید. برای هکرها واقعی، مرزهای بین "بازی"، "کار"، "علم" و "هنر" همگی تمایل به ناپدید شدن یا ادغام شدن در یک بازی خلاقانه سطح بالا دارند. همچنین، به طیف محدودی از مهارت ها اکتفا نکنید. اگرچه اکثر هکرها خود را به عنوان برنامه نویس معرفی می کنند، اما به احتمال زیاد در چندین مهارت مرتبط مهارت زیادی دارند، مدیریت سیستم، طراحی وب و عیب یابی سخت افزار رایانه شخصی از موارد رایج هستند. از طرف دیگر، هکری که یک مدیر سیستم است، احتمالاً در برنامه نویسی اسکریپت و طراحی وب کاملاً ماهر است. هکرها کارها را نصفه انجام نمی دهند. اگر روی یک مهارت سرمایه گذاری کنند، تمایل دارند در آن مهارت بسیار خوبی کسب کنند.

در نهایت، چند کار که نباید انجام دهید:

- از شناسه کاربری یا نام کاربری احمقانه و بزرگ استفاده نکنید.
- وارد جنگ های شعله ای در Unsenet (یا هر جای دیگر) نشوید.
- خودتان را "cyberpunk" خطاب نکنید و وقت خود را برای کسی که این کار را انجام می کند تلف نکنید.
- نامه یا ایمیلی پر از غلط املائی و گرامر بد ارسال نکنید.

تنها شهرتی که با انجام هر یک از این کارها به دست خواهید آورد، به عنوان احمق است. هکرها خاطرات طولانی دارند، ممکن است سالها طول بکشد تا با اشتباهات اولیه خود به اندازه ای زندگی کنید تا پذیرفته شوید.

مشکل با نام های کاربری مستحق توضیح است. پنهان کردن هویت خود در پشت اسم مستعار یک رفتار نوجوانانه و احمقانه از ویژگی های کرکرها است. مانند warez d00dz و سایر اسم های مستعار. هکرها این کار را نمی کنند. آن ها به کاری که انجام می دهند افتخار می کنند و می خواهند که با نام واقعی شان مرتبط باشد. بنابراین اگر نام مستعار دارید، آن را رها کنید. در فرهنگ هکرها فقط شما را به عنوان یک بازنده نشان می دهد.



یادداشت تاریخی: هک، متن باز، و نرم افزار آزاد

زمانی که در اواخر سال 1996 این روش را نوشتم، برخی از شرایط پیرامون آن با شکل امروزی آن بسیار متفاوت بود. چند کلمه در مورد این تغییرات ممکن است به روشن شدن مسائل برای افرادی که در مورد رابطه متن باز، نرم افزار آزاد و لینوکس با جامعه هکرها سردرگم هستند کمک کند. اگر در این مورد کنجکاو نیستید، می توانید مستقیماً از اینجا به سؤالات متداول و کتاب شناسی بروید. اخلاق و جامعه هکرها همانطور که در اینجا توضیح دادم مدتها قبل از ظهور لینوکس پس از سال 1990 است؛ من برای اولین بار در حدود سال 1976 با آن درگیر شدم، و ریشه های آن به اوایل دهه 1960 برمی گردد. اما قبل از لینوکس، بیشتر هک ها بر روی سیستم عامل های اختصاصی یا تعداد معدودی از سیستم های شبه تجربی خانگی مانند MIT ITS انجام می شد که هرگز خارج از جایگاه های آکادمیک اصلی خود خارج نشدند. در حالی که قبلاً (قبل از لینوکس) تلاش هایی برای تغییر این وضعیت وجود داشت، تأثیر آنها در بهترین حالت بسیار حاشیه ای بود و محدود به جوامعی از مؤمنان واقعی بود که اقلیت های کوچکی بودند حتی در جامعه هکرها، چه رسد به دنیای کلی نرم افزار های عمومی.

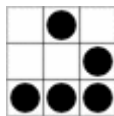
چیزی که اکنون "متن باز" نامیده می شود، به جامعه هکرها برمی گردد، اما تا سال 1985 این یک عمل عامیانه بی نام بود تا یک جنبش آگاهانه با تئوری ها و مانیفست ها. این ماقبل تاریخ زمانی به پایان رسید که در سال 1985، ریچارد استالمن ("RMS") هکر قدیمی سعی کرد نامی به آن بدهد "نرم افزار آزاد". اما نامگذاری او نیز یک عمل ادعایی بود، او بار ایدئولوژیک را به برچسب "نرم افزار آزاد" وصل کرد که بسیاری از جامعه هکرها موجود هرگز آن را نپذیرفتند. در نتیجه، برچسب "نرم افزار رایگان" با صدای بلند توسط اقلیت قابل توجهی از جامعه هکرها (به ویژه در میان کسانی که با BSD Unix مرتبط هستند) رد شد و توسط اکثریت بقیه (از جمله خود من) با رزرو جدی اما بی صدا استفاده شد.

با وجود این ملاحظات، ادعای RMS برای تعریف و رهبری جامعه هکرها تحت عنوان "نرم افزار آزاد" تا اواسط دهه 1990 به طور گسترده برقرار بود. تنها با ظهور لینوکس به طور جدی به چالش کشیده شد. لینوکس به توسعه منبع باز یک خانه طبیعی داد. بسیاری از پروژه‌هایی که تحت شرایطی منتشر می‌شوند که اکنون آن را متن باز می‌نامیم و از یونیکس‌های اختصاصی به لینوکس مهاجرت کرده‌اند. جامعه پیرامون لینوکس به طور انفجاری رشد کرد و بسیار بزرگتر و ناهمگون تر از فرهنگ هکرها قبل از لینوکس شد. RMS قاطعانه تلاش کرد تا همه این فعالیت ها را در جنبش "نرم افزار آزاد" خود قرار دهد، اما هم به دلیل تنوع در حال انفجار جامعه لینوکس و هم بدبینی عمومی بنیانگذار آن، لینوس توروالدز، مانع شد. توروالدز به دلیل نداشتن هر جایگزینی به استفاده از اصطلاح "نرم افزار آزاد" ادامه داد، اما علنا توشه ایدئولوژیک RMS را رد کرد. بسیاری از هکرها جوان از این روش پیروی کردند.

در سال 1996، زمانی که من برای اولین بار این مقاله را منتشر کردم، جامعه هکرها به سرعت در حال سازماندهی مجدد در اطراف لینوکس و تعداد انگشت شماری از سیستم عامل های منبع باز دیگر (به ویژه سیستم عامل های BSD Unix) بودند. خاطره جامعه از این واقعیت که بیشتر ما دهه ها را صرف توسعه نرم افزارهای متن بسته بر روی سیستم عامل های متن بسته کرده بودیم، هنوز محو نشده بود، اما این واقعیت از قبل به نظر بخشی از گذشته مرده بود. هکرها به طور فزاینده ای خود را به عنوان هکر با پیوست های خود به پروژه های منبع باز مانند لینوکس یا آپاچی تعریف می کردند.

با این حال، اصطلاح "متن باز" هنوز ظهور نکرده بود. تا اوایل سال 1998 این کار را انجام نمی داد. هنگامی که این کار انجام شد، اکثر جامعه هکرها آن را طی شش ماه بعد پذیرفتند. استثنای اقلیتی بودند که از نظر ایدئولوژیک به اصطلاح "نرم افزار آزاد" متصل بودند. از سال 1998 و به ویژه پس از حدود سال 2003، شناسایی «هک» با «توسعه متن باز» (و نرم افزار رایگان) بسیار نزدیک شده است. امروزه تلاش برای تمایز بین این دسته‌ها فایده چندانی ندارد و بعید به نظر می‌رسد که در آینده تغییر کند.

با این حال، شایان ذکر است که همیشه اینطور نبود.



سایر منابع

پل گراهام مقاله ای به نام **هکهای بزرگ** و مقاله ای دیگر در مورد دوره **کارشناسی** نوشته است که در آن بسیار خردمندانه صحبت می کند.

هکرهای جواتر ممکن است چیزهایی را که هر هکر زمانی می دانست را جالب و مفید ببابند.

من همچنین تاریخچه مختصری از هکردام (Hackerdom) را نوشته ام.

من مقاله ای به نام **کلیسای جامع و بازار** نوشته ام که در مورد نحوه عملکرد لینوکس و فرهنگ های منبع باز توضیح می دهد. من به این موضوع حتی در ادامه **خانه نشینی در نووسفر** به طور مستقیم تری پرداخته ام.

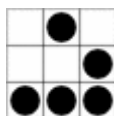
Rick Moen یک سند عالی در مورد **نحوه اجرای یک گروه کاربری لینوکس** نوشته است.

من و ریک مون در سند دیگری در مورد **نحوه پرسیدن سوالات هوشمندانه** همکاری کرده ایم. این به شما کمک می کند به گونه ای به دنبال کمک باشید که احتمال اینکه آن را دریافت کنید بیشتر شود.

اگر به آموزش اصول اولیه نحوه کار کامپیوترهای شخصی، یونیکس و اینترنت نیاز دارید، به **The Unix and Internet Fundamentals HOWTO** مراجعه کنید.

وقتی نرم افزار را منتشر می کنید یا افزونه ای را برای نرم افزار می نویسید، سعی کنید دستورالعمل های موجود در **دستورالعمل انتشار نرم افزار** را دنبال کنید.

اگر از شعر ذن لذت بردید، ممکن است **Rootless Root: The Unix Koans of Master Foo** را نیز دوست داشته باشید.



ترجمه شده توسط : آتیلا اصغری 2022

asghariatilla@gmail.com

<https://atillaasghari.ir>

Translated by Atilla Asghari 2022

Email : asghariatilla@gmail.com

Website: <https://atillaasghari.ir>

Eric S.Raymond : esr@thyrsus.com , <http://www.catb.org/~esr>