

فلسفه یونیکس



محمد رضا قربانی

در یونیکس، کدها تنها برای کار راه انداختن، یا فروختن چیزی نیستند. در یونیکس قرار است همه چیز متعالی باشد. فلسفه‌ای پشت هر کار و هر اقدام وجود داشته باشد و تمام برنامه‌نویسان و توسعه‌دهندگان آن‌هم، به این نکته معتقدند. از مکملی، خالق پایپ‌ها^۱ و یکی از بنیانگذاران سنت یونیکس، فلسفه یونیکس را این طور خلاصه می‌کند: «فلسفه یونیکس این است، برنامه‌هایی بنویسید که یک کار بکنند و آن را خوب انجام دهد. برنامه‌هایی بنویسید که با هم کار کنند. برنامه‌هایی بنویسید که جریان‌های متن را مدیریت کنند، چون متن یک رابط بین‌المللی است.» این فلسفه معمولاً به این صورت خلاصه می‌شود: «یک کار بکن، اما آن یک کار را خوب انجام بده.»

راب پایک، یکی از اعضای تیم یونیکس درباره برنامه‌نویسی به زبان C در یونیکس، این گونه می‌نویسد:

۱. نمی‌توانید بفهمید برنامه کجا و کجا وقتش را صرف می‌کند. اشتباهات درست در جاهایی که انتظار ندارید رخ می‌دهند، بنابراین تا زمانی که نفهمیدید ایراد از کجاست، دست به کد خود نزنید.

۲. اندازه بگیرد. تا زمانی که ابزارهای تحلیل‌تان به شما آماری از بازدهی نرم‌افزار ندهاند، دست به سریع‌تر کردن آن نزنید.

۳. الگوریتم‌های عجیب و غریب در داده‌های کوچک از الگوریتم‌های ساده، کندتر عمل می‌کنند، درجه این الگوریتم‌ها $O(n)$ است و معمولاً این n کوچک است. بنابراین تا زمانی که مطمئن نشدید که n شما بزرگ نیست، سراغ الگوریتم‌های خاص نروید.

۴. تا آن جاکه می‌توانید ساختار داده و الگوریتم‌های خود را ساده کنید. پیاده‌سازی ساختارهای پیچیده با خطاهای عجیب و غریب همراه خواهد بود. ساختمان داده را در بیشتر برنامه‌ها می‌توان با آرایه‌ها، لیست‌های پیوندی، جداول هش و درخت‌های باینری پیاده کرد.

۵. داده‌محوری. یادتان باشد که تنها زمانی یک نرم‌افزار خوب خواهید داشت که ساختار داده‌های آن درست طراحی شده باشد.

مایک گانکارز، یکی از اعضای گروه مخترع سیستم پنجره اکس^۲ بر اساس تجربیات شخصی‌شان و همچنین بحث‌هایی که با دیگر برنامه‌نویسان می‌کردند، فلسفه یونیکس را به ۹ پارامتر اساسی

منتهی کردند که بدین شرح است:

۱. کوچک زیباست.
۲. هر برنامه را طوری بنویسید که یک کار را خوب انجام بدهد.

۳. سریع برنامه را مستند کنید.
۴. قابلیت حمل را برکاری ترجیح دهید.
۵. داده‌ها را در فایل‌های متنی ذخیره کنید.
۶. از اهرم نرم‌افزاری برای پیش‌برد خود استفاده کنید.

۷. از اسکریپت‌های پوسته برای افزایش تاثیر و قابلیت حمل استفاده کنید.

۸. از رابط‌های کاربری محدودکننده بپرهیزید.
۹. هر برنامه، خود یک فیلتر است.

البته، ده پارامتر دیگر هست که وارد فلسفه یونیکس بشود. اما بحث در مورد آن بالاست و در برخی جاها مناظره‌هایی تند انجام می‌شود. این ده پارامتر، بدین شرح هستند:

۱. به کاربران اجازه دهید خودشان محیط را تغییر دهند.

۲. هسته سیستم کوچک و سبک باشد.

۳. از حروف کوچک و کلمات کوتاه استفاده کنید.

۴. چیزی را چاپ نکنید! درختان را نجات دهید.

۵. سکوت طلاست.

۶. موازی بیندیشید.

۷. جمع تمام بخش‌ها همواره از کل بزرگتر است.

۸. بدتر بهتر است.

۹. به دنبال ۹۰ درصد راه حل باشید.

۱۰. سلسله مراتبی بیندیشید.

بدتر، بهتر است؟

مورد هشتم به شدت جلب توجه می‌کند. ریچارد پ. گابریل پیشنهاد می‌کند که مزیت کلیدی یونیکس در این است که فلسفه‌ای از طراحی را پیاده می‌کند که او آن را «بدتر بهتر است» نامیده است. در سبک طراحی «بدتر بهتر است»، سادگی رابط کاربری و پیاده‌سازی مهم‌تر از دیگر ویژگی‌های سیستم، از جمله درستی، پایداری و کامل بودن برنامه است. گابریل می‌گوید که این شیوه طراحی مزیت‌های بهتری دارد و عموماً کیفیت نتیجه بهتر درک می‌شود.

برای مثال، در روزهای آغازین یونیکس هسته‌ای مونولیتیک^۳ داشت که در آن تمام پروسس‌های کاربر از طریق هسته سیستم عامل اجرا می‌شد. اگر سیگنال به پروسسی می‌رسید که منتظر عملیات طولانی ورودی/خروجی بود، در این صورت باید چه می‌شد؟ سیگنال به تاخیر می‌افتاد؟ کنترل‌کننده سیگنال وقتی

پروسس در حالت هسته است، نباید اجرا شود؟ آیا هسته باید فراخوان سیستمی را پس‌بزند و آن را ذخیره کند تا بعداً استفاده کند؟

در این موارد، آقایان کن تامپسن و دنیس ریچی سادگی را به کمال ترجیح می‌دهند. سیستم یونیکسی باید فراخوان سیستمی را سریعاً با خطایی برگرداند که کاری انجام نداده است (فراخوان سیستمی دچار وقفه شده) با شماره خطای ۴ در سیستم‌های امروزی.

آقای اریک ریمنود در کتاب هنر برنامه‌نویسی یونیکس، فلسفه یونیکس را در «اصل بوسه ۴» (ساده و احمقانه نگاه‌اش دار) خلاصه می‌کند. همچنین وی فهرستی از قوانین نوشتن برنامه، به سبک یونیکسی را تنظیم کرده است:

قانون ماجولار بودن: برنامه‌ها جدا جدا باشند که رابطی تر و تمیز آن‌ها را به هم پیوند بدهد.

قانون شفافیت: شفافیت از هوشمندی بهتر است.

قانون تجمعی: برنامه‌ها را طوری طراحی کنید که به برنامه‌های دیگر وصل شوند.

قانون جداسازی: سیاست از مکانیزم جداسازی؛ رابط کاربری از موتور یک برنامه.

قانون سادگی: ساده‌سازی کنید؛ تنها وقتی لازم است، سراغ پیچیدگی‌ها بروید.

قانون کم‌خرجی: برنامه بزرگ را وقتی بنویسید که با هیچ روش کوتاه‌تری نشود کار کرد.

قانون سکوت: اگر برنامه‌ای چیزی برای گفتن ندارد که کاربر را شگفت‌زده کند، بهتر است چیزی نگویید. پس از اوج گرفتن بنیاد نرم‌افزارهای آزاد، گنو

روی استانداردهای برنامه‌نویسی‌ای شبیه به یونیکس فعالیت می‌کند. در واقع، فلسفه یونیکس آنقدر متعالی است که رسیدن به آن، همواره جز چالش‌های برنامه‌نویس‌ها بوده است. برنامه‌نویسان قدیمی یونیکس معتقدند برنامه‌های لینوکسی دیگر پرآب و تاب‌تر شده‌اند. بهتر است فلسفه یونیکس را با این جمله از دنیس ریچی، خالق زبان برنامه‌نویسی C به پایان برسانیم: «یونیکس ساده است. اما به یک نابغه نیاز است تا سادگی آن را درک کند.»

پی‌نوشت‌ها

1. Pipe
2. X Window System
3. Monolithic
4. KISS Principle: Keep It Simple Stupid

منابع

- http://www.linfo.org/unix_philosophy.html
<http://www.faqs.org/docs/artu/ch01s06.html>

clickhelp@jamejamonline.ir

مهدی رشنو

پرسش و پاسخ

من فایلی با پسوند ts دارم و موضوع فایل کلیپ تصویری ضبط شده است. ولی در کامپیوتر نمی‌توانم آن را باز کنم. لطفاً مرا راهنمایی کنید.

فایل مورد نظر شما دارای فرمت تصویری دریافتی از گیرنده‌های ماهواره است. برای آنکه سیستم شما بتواند آن را پخش کند، می‌توانید آن را به یکی از قالب‌های قابل شناسایی توسط ویندوز تبدیل کنید. برای این کار می‌توانید از نرم‌افزارهای میدلی مانند Tconverter یا TS to MPG Converter استفاده کنید. در بعضی موارد با تغییر پسوند فایل از ts به mpeg می‌توان به نتیجه دلخواه رسید.

چطور می‌توان از طریق اینترنت، پیامک رایگان فرستاد؟ (بدون استفاده از موبایل)

سایت‌های فراوانی هستند که با ثبت نام در آنها می‌توانید به صورت

تصویری در اینترنت وجود دارد و می‌خواهید آن را در وبلاگ خود قرار دهید، می‌توانید روی آن کلیک راست کرده و گزینه Properties را انتخاب کنید. در قسمت Address می‌توانید نشانی عکس را ببینید و در ویرایشگر وبلاگ خود وارد کنید.

پروانه زوروش، اندیمشک - من مدتی است که یک وبلاگ ایجاد کرده‌ام برای ایجاد یک ساعت آنالوگ که به صورت زنده در وبلاگ نمایش داده شود، چه کاری باید انجام دهم؟

یک ساعت آنالوگ، یک برنامه کوچک جاوا است، که ساعت سیستمی که آن را می‌خواند به صورت آنالوگ نمایش می‌دهد. کافیست کد جاوای این ساعت را (که به راحتی در اینترنت قابل دستیابی است) در ویرایشگر قالب وبلاگ خود قرار دهید.

سامان شجاعی راد، اندیمشک - می‌خواستم تعدادی عکس را در وبلاگ بگذارم. چگونه باید این کار را انجام دهم؟

ابتدا باید عکس‌های مورد نظر خود را، چنانچه در اینترنت نیستند، در فضای اینترنت قرار دهید. برای این کار می‌توانید از سایت‌هایی که فضای رایگان برای آپلود تصاویر ارائه می‌دهند مانند (Tinypic) استفاده کنید. پس از ورود به این سایت‌ها امکانی برای آپلود تصویر در اختیار کاربر قرار می‌دهند تا تصویر مورد نظر خود را در فضای اینترنت قرار دهد. پس از آپلود کردن تصاویر، سایت مورد نظر چند نشانی از عکس خودتان به شما می‌دهد (یعنی با رفتن به آن نشانی می‌توانید عکس خود را ببینید). در ویرایشگر وبلاگ خود، بخشی به عنوان درج تصویر وجود دارد. پس از انتخاب آن، آدرسی که از سایت مربوطه گرفته‌اید، وارد کنید. چنانچه