

تاریخچه و تولد

نیترا ف شگفت انگیز

هوشمندانه ترین برنامه ی حرفه چینی دنیا

نیترا ف شگفت انگیز

من در هر ترم مستندهای زیادی را آماده می‌کنم و چون افراد زیادی آنها را مطالعه می‌کنند، کیفیت بالای آنها برای من بسیار اهمیت دارد. برخی از من می‌پرسند چگونه این مستندها را تولید می‌کنم. پاسخ من به این سؤال معمولا موجب تعجب شنوندگان می‌شود؛ چه طور امکان دارد برنامه‌ی کوچکی به نام «نیترا ف» که حجم آن چند صد کیلو بایت است چنین مستندهایی را تولید کند. نیترا ف پیاده‌سازی جدیدی از «تیرا ف» است و تیرا ف برنامه‌ای است که طراحی اولیه‌ی آن توسط برنامه‌نویسانی انجام شده است که با یونیکس، دنیای سیستم‌های عامل را متحول کرده‌اند. اصل مهمی که در تیرا ف به خوبی دیده می‌شود سادگی است و قدرت این سادگی بارها من را نیز شگفت زده کرده است.

در ادامه سعی می‌کنم نشان دهم چرا از نیترا ف برای تولید مستندهایم استفاده می‌کنم، چه جنبه‌هایی از نیترا ف برای من بسیار جالب هستند و چه ویژگی‌هایی در نیترا ف سبب می‌شود نوشتن مستندهای فارسی برای من تا این حد لذت‌بخش باشد. مستند جاری به چگونگی استفاده از «تیرا ف» و «نیترا ف» نمی‌پردازد (برای این منظور مستندهای دیگری موجود هستند). این مستند کامل نیست و سعی می‌کنم در صورتی که زمان اجازه دهد جنبه‌های جالب دیگری از نیترا ف را به آن بیفزایم.

در جستجو

گزارش نتایجی که بدست آورده‌ایم یا مطالعاتی که انجام داده‌ایم بخش بسیار مهمی از زندگی دانشگاهی یا حرفه‌ای محسوب می‌شود. اما در دوران حروفچینی دیجیتال، علاوه بر متنی که می‌نویسیم معمولاً لازم است آن را حروفچینی کنیم (دلایل زیادی برای حروفچینی متن وجود دارند، از جمله قوانین سازمانی و افزایش خوانایی). اما برای حروفچینی باید ابزار مناسبی را انتخاب کنیم. من هم بارها چنین مستنداتی را آماده کرده‌ام و برای انتخاب ابزار مناسب، روزها جستجو کرده‌ام.

هنگامی که دانشجوی کارشناسی بودم برای تولید مستندهای فارسی ابزارهای محدودی وجود داشتند. از بین نرم‌افزارهایی که به صورت گسترده استفاده می‌شدند، سه انتخاب اصلی داشتم: الف) استفاده از نرم‌افزارهای شرکت مایکروسافت؛ چون از ویندوز استفاده نمی‌کردم، تمایلی به استفاده از آنها نداشتم (خروجی آنها هم چندان خوب نبود)، ب) استفاده از نرم‌افزارهای گرافیکی لینوکس برای حروفچینی؛ انتخاب خوبی نبودند چون به ندرت از برنامه‌های گرافیکی استفاده می‌کردم (و البته در آن زمان خروجی خوبی نداشتند)، ب) استفاده از زیتک و فارسی‌تک؛ هر دوی آنها در آن زمان محدودیت‌هایی داشتند (برای مثال فارسی‌تک وابسته به سیستم عامل‌های شرکت مایکروسافت بود و زیتک خروجی خوبی نداشت). در کنار این مشکلات، اندازه‌ی زیاد برنامه‌های تک و بسته‌های آن موجب می‌شد آنها را به عنوان یک راه حل جدی در نظر نگیرم. من ترجیح می‌دادم از ابزاری استفاده کنم که ساده، قدرتمند و منعطف باشد و هر قسمتی از آن را که نیاز داشتم به سادگی بفهمم و تغییر دهم. من ترکیب این ویژگی‌ها را در تک نیافتم.

از این رو تصمیم گرفتم برای ابزار دیگری برای حروفچینی فارسی جستجو کنم. نتیجه‌ی این جستجو، ابزاری قدیمی و قدرتمند به نام تیراف بود. اما یک مشکل بزرگ وجود داشت: هیچ یک از نسخه‌های تیراف موجود، فارسی را پشتیبانی نمی‌کردند. سه نسخه‌ی اصلی تیراف به صورت آزاد در دسترس بودند: Groff که به صورت پیش‌فرض در سیستم‌عامل‌هایی مثل لینوکس نصب می‌شد، نسخه‌ی تیراف همراه سیستم عامل Plan 9 و نسخه‌ی جدیدتری به نام تیراف Heirloom. دو مورد آخر از تغییر نسخه‌ای اولیه‌ی تیراف که همراه سیستم عامل یونیکس نوشته شده بود حاصل می‌شدند. Groff امکانات بسیار خوبی داشت ولی به خوبی از Unicode پشتیبانی نمی‌کرد. دو نسخه‌ی دیگر Unicode را پشتیبانی می‌کردند اما تغییر شکل حروف فارسی و تغییر جهت حروف در متن‌های راست به چپ را انجام نمی‌دادند.

پس از بررسی این نسخه‌ها در نهایت تصمیم گرفتم تیراف Plan 9 را برای پشتیبانی از فارسی تغییر دهم. برای این کار، ابتدا آن را به سیستم عامل لینوکس انتقال دادم و سپس آن را برای متن‌های راست به چپ و تصحیح شکل حروف فارسی تغییر دادم^۱. انجام این تغییرات، با توجه به قدمت کد این نسخه از تیراف، خوانایی کم آن و برخی از محدودیت‌های تبدیل خروجی تیراف به فرمت PostScript، خیلی ساده نبود.

تولّد نیتراف

از این نسخه‌ی تغییر داده شده‌ی تیراف، بیش از دو سال استفاده کردم و پایان‌نامه‌ی کارشناسی ارشد خود را با آن نوشتم. در این بازه، تجربه‌ی خوبی در استفاده از تیراف بدست آوردم و منابع خوب و پراکنده‌ای در زمینه‌ی حروفچینی دیجیتال مطالعه کردم و در نتیجه، انتظار من از یک ابزار حروفچینی به تدریج گسترش یافت. با وجود اینکه خروجی این نسخه‌ی تغییر داده شده قابل قبول بود، آن را به عنوان ابزار حروفچینی ایده‌آل نمی‌پنداشتم. برخی از محدودیت‌های موجود در تیراف Plan 9 و پشتیبانی نکردن از برخی از ویژگی‌ها و الگوریتم‌های مهم حروفچینی، من را به این فکر انداخت که نسخه‌ی جدیدی از تیراف را پیاده‌سازی کنم. نسخه‌ای که محدودیت‌های موجود در سایر نسخه‌های تیراف را رفع کند و از طرف دیگر از زبان‌های غیر غربی و راست به چپ به خوبی پشتیبانی کند. نام «نیتراف» را برای این برنامه انتخاب کردم.

تیراف به تعدادی برنامه‌ی کوچک‌تر مستقل شکسته شده است و این مسئله یکی از جالب‌ترین جنبه‌های معماری تیراف است که در یونیکس به فراوانی مشاهده می‌شود. برنامه‌ی اصلی تیراف، کد تیراف را می‌خواند و کد قابل تبدیل به فرمت‌های خروجی را تولید می‌کند (مشابه کد میانی در کامپایلرها). تعدادی پیش‌پردازشگر (مثلاً برای کشیدن شکل) قبل از تیراف ورودی را پردازش می‌کنند و قسمت‌های لازم را به کد تیراف تبدیل می‌کنند. کد خروجی تیراف نیز توسط یکی از پس‌پردازشگرها به یک فرمت فایل خروجی مثل PDF تبدیل می‌شود. این معماری کار پیاده‌سازی نیتراف را برای من بسیار ساده‌تر ساخت و در هر گام تمرکز خود را روی یکی از این برنامه‌ها قرار دادم و سعی کردم سازگاری را با برنامه‌های قدیمی تیراف حفظ کنم. ابتدا برنامه‌ی اصلی نیتراف^۲ را نوشتم. سپس، برنامه‌هایی برای

۱ <https://repo.or.cz/w/troff.git>

۲ <https://github.com/aligrudi/neatroff.git>

۳ <https://github.com/aligrudi/neatpost.git>

تبدیل خروجی نیتراف به PostScript^۲ و برای تولید توصیف فونت‌های رایج برای نیتراف^۱ پیاده‌سازی کردم.

یکی از مشکلاتی که باقی مانده بود، نوشتن عبارت‌های ریاضی بود. پیش‌پردازشگر eqn تیراف Plan 9 عبارت‌های ریاضی را حروفچینی می‌کرد ولی خروجی آن با خروجی تک قابل مقایسه نبود. از این روی یک پیش‌پردازشگر برای عبارت‌های ریاضی^۲ نوشتم که مطابق با الگوریتم تک آنها را حروفچینی کند. نوشتن چنین پیش‌پردازشگرهایی بسیاری از چالش‌های موجود در تولید کد در کامپایلرها را داشت؛ کدی که توسط این پیش‌پردازشگرها تولید می‌شود باید در زمان ترجمه‌ی کد تیراف اجرا شود و در آن زمان باید اجزاء عبارت‌های ریاضی به شکل مناسبی را در کنار هم بچیند.

به تدریج تعدادی بسته و پیش‌پردازشگر برای کارهای دیگر نیز برای نیتراف نوشتم^۳. اخیراً یک بسته برای نوشتن پایان‌نامه‌های فارسی آماده کردم^۴.

نیتراف بی‌نظیر

ممکن است این سؤال برای خوانندگان ایجاد شده باشد که یک ابزار حروفچینی خوب چه ویژگی‌هایی دارد و نیتراف از این دید در چه جایگاهی قرار دارد. در ادامه سعی می‌کنم به برخی از ویژگی‌های خوب نیتراف اشاره کنم. نیتراف بسیاری از الگوریتم‌های حروفچینی رایج را پیاده‌سازی می‌کند، بسیاری از ویژگی‌های فونت‌های OpenType را پشتیبانی می‌کند و برای نوشتن زبان‌های راست به چپ به خوبی کار می‌کند. همه‌ی این موارد برای یک برنامه‌ی حروفچینی حرفه‌ای لازم هستند. اما جنبه‌های دیگری از نیتراف (و تیراف) آن را متمایز می‌کنند (البته این موارد سلیقه‌ای هستند و ممکن است خوانندگان گرامی در برخی از آنها کاملاً با نویسنده‌ی این متن هم‌عقیده نباشند): الف) در تیراف بسیاری برای برخی کاربردها راه‌حل‌های بسیار جالب و کوتاهی وجود دارند، ب) نوشتن بسته‌ی ماکرو برای تیراف ساده است و ج) ماکروها در تیراف انتزاع بسیار خوبی برای پوشاندن جزئیات ارائه می‌دهند.

۱ <https://github.com/aligrudi/neatmkfn.git>

۲ <https://github.com/aligrudi/neateqn.git>

۳ https://github.com/aligrudi/neatroff_make.git

۴ https://github.com/nit-ce/neatroff_ths.git

اطلاعات بیشتر

پیوند	توضیح
http://litcave.rudi.ir/neatfbeg.pdf	راهنمای راه‌اندازی نیتراف (فارسی)
http://litcave.rudi.ir/neatfarsi.pdf	معرفی خلاصه‌ی نیتراف (فارسی)
https://github.com/nit-ce/neatroff_ths.git	بسته‌ی پایان‌نامه‌ی فارسی
http://litcave.rudi.ir/neatroff.pdf	معرفی نیتراف
http://litcave.rudi.ir/neatstart.pdf	شروع استفاده از نیتراف
http://litcave.rudi.ir/neateqn.pdf	عبارت‌های ریاضی در نیتراف
https://github.com/aligrudi/neatroff.git	کد نیتراف
https://github.com/aligrudi/neatpost.git	کد پس‌پردازشگر نیتپست
https://github.com/aligrudi/neateqn.git	کد پیش‌پردازشگر نیت‌ایکیوان
https://github.com/aligrudi/neatrefer.git	کد پیش‌پردازشگر نیت‌ریفر
https://github.com/aligrudi/neatmkfn.git	کد برنامه‌ی تولید توصیف فونت نیتراف
https://github.com/aligrudi/neatroff_make.git	بسته‌های ماکروی همراه نیتراف
https://repo.or.cz/troff.git	نسخه‌ی تیراف Plan 9 انتقال داده شده به لینوکس