

1. Časť wekových weekendových zamyslení

Čím ďalej tým viac zisťujem, že množstvo javov v počítačoch a programovaní, ktoré mi boli doteraz záhadou, majú korene, niekedy až prekvapujúco hlboké, v dávno zabudnutej histórii.



Ale chápem, ako to vzniká: na začiatku bola skupina úzko špecializovaných odborníkov, ktorí sa viacmenej navzájom poznali a vytvárali terminológiu a postupy atď. podľa dennej potreby. Potom bola prvá generácia žiakov, ktorí tie postupy a terminológiu prevzali akosi prirodzene, spolu aj s tým podhubím. Potom prišla druhá generácia, ktorým už to podhubie nebolo dožičené a tak bola terminológia a postupy odtrhnuté od ich pôvodného koreňa. A keďže nie tá priekopnícka, nultá 😊, ale až prvá a častokrát až tá druhá generácia pocítila potrebu tie znalosti spísať, ono sa to nejako zakonzervovalo a ostalo to tak.

V poslednom čase som sa stretol s dvomi konkrétnymi príkladmi:

1. Keď som sa prvýkrát stretol blízšie s hĺbkami postupu prekladu C, narazil som aj na skratku **.bss** (trocha mi to vtedy pripomínalo BSE, chorobu "sialených krav" 😊), čo je oblasť v pamäti vyhradená pre statické (globálne) neinicializované dáta, ktoré sú pri spustení programu automaticky vynulované. Tá skratka sa mi zdala nelogická - a kolega mi aj povedal, že je to "Block Started by Symbol" - zdalo sa mi to blbosť, prečo sa to nevolá `uninit.data` alebo `data.uninit` alebo tak nejak?

Historické vysvetlenie sa dá nájsť (spolu s niekoľkými ďalšími podobnými) napríklad [tu](#), a pochádza z pravekeho assembleru doprevádzajúceho FORTRAN (prvý vyšší programovací jazyk, mimochodom), kde to predstavovalo pseudoinštrukciu, ktorá vyhradilo miesto (blok) pre premenné a priradilo na jeho začiatok navštívie (t.j. symbol).

2. UNIXovsky príkaz **ls** slúžiaci na vypísanie zoznamu suborov v adresári - vždy som si myslel, že to je z **list** - ono tie UNIXovské príkazy sú všetky až chorobne skratené, nebudeme teraz rozoberať, či to bol následok setrenia miestom a prenosovými linkami, alebo skor neschopnosti UNIXakov písať dlhšie slova 😊. Lenže ono je to inak, a názov **ls** má oveľa hlbšie korene, presne tak hlboko ako samotný UNIX... Pôvodný autor UNIXu Ken Thompson (spolu aj s autorom jazyka C, Denisom Ritchiem a množstvom iných ľudí, čo sa okolo C/UNIXu tmolili) pracovali na projekte MULTICS, seriózneho a rozsiahleho operačného systému, ktorý mal v sebe zahrnúť všetky dovtedajšie poznatky a požiadavky na OS (bavíme sa o strede 60. rokov). MULTICS špecifikoval aj požiadavky na hardware, z čoho jedna bola taká, že z pohľadu programu bol jednotný prístup k operačnej pamäti aj disku - navonok to vyzeralo ako jedna veľká pamäť (pocitac s MULTICSom sa nikdy nevypínal, naopak, OS mal zabudovanú podporu na pridávanie a odoberanie hardwaru - pamäte, disku, procesorov(!) - za jazdy). No a v tej "jednotnej" pamäti boli jednotlivé "kusy" dát, programov atď. umiestnené v logických nazvaných "segment". Program, ktorý vypísal zoznam segmentov v určitom adresári (ano, aj hierarchický strom adresárov pochádza z MULTICSu), sa volal "list segments", skratene "ls".

Keď Thompsonov zamestnávateľ, AT&T/Bell Labs, odskočili od projektu MULTICS (ktorý bol síce dokončený ale veľkej slávy sa nedožil, napriek obrovskému vplyvu na rozvoj počítačov a programovania - je to tiež veľmi poučná historka o ambiciózných projektoch), Thompson jedným konkrétnym cieľom práce, ale aj o prístupe k modernému a vykonávaniu, i keď pomerne jednocelovému a unikátnemu, počítaču, na ktorom MULTICS súčasne vyvíjali aj testovali (nie, vtedy nemal každý na stole písíčko 😊). A tak, plný inspirácie, na omnoho jednoduchšom a menšom počítači, ktorý mal k dispozícii, zbastlil akúsi paródiu na MULTICS (ono aj meno UNIX je vlastne paródia na meno MULTICS). No a keďže jedným z potrebných nástrojov bola utilitka na vypísanie suborov v adresári, použil meno, ktoré predtým tak často používal na tento účel: **ls**.

Pekný víkend!

wek

[Diskusia k článku](#)

[2. časť ->](#)