

Co je to software?

Autor textu: Florian Cramer

Autorky překladu: Jana Horáková a Lucie Semecká

Kulturní definice

Předchozí kapitoly [míněno knihy *Slova se stávají tělem (Words made flesh)*, pozn. red.] se zabývaly „komputacemi“, tedy výpočty a algoritmy prováděnými buďto přímo uvnitř jazyka samotného, nebo s pomocí mechanických či elektronických zařízení. Jak je vidět u mnoha spekulativních komputací, například *Sefer Jecira* s jejím těžko proniknutelným mechanickým systémem, je obtížné, ne-li přímo nemožné, vést přesnou hranici mezi symbolicko-imaginárním a materiálním zpracováním, mezi softwarem a hardwarem.

Co je software? Kdybychom ho definovali pouze jako algoritmus, tak se nám v první řadě nepodaří obsáhnout tímto pojmem nesmírnou spekulativní imaginaci od magie až po kódová díla, u nichž je výpočetní kód jen zřídkakdy čistým algoritmem, ale spíše metaforickým hybridem. Striktně formální definice softwaru také selže, budeme-li chtít její pomocí popsat repliku z filmu Stevena Seagala „*300,000 stran kódu. Nebo 60 minut sadomasochistického porna z produkce triple-X.*“ [1] Kulturně podmíněné chápání softwaru tedy není nijak přitažené za vlasy, ale v podstatě se již stalo součástí obecného povědomí. Proč vlastně zavedl matematik John W. Tukey v roce 1957 termín „software“, když víme, že pojem algoritmus, foneticky odvozený od jména perského matematika z 9. století Muhammada Ibn Músá al-Chórezmi, existoval již staletí předtím? Je zjevné, že pojmem software zde Tukey odkazoval k algoritmům jako k řídicí logice, abstrahované od stroje. Operační systém *Unix*, který běží na téměř jakémkoli hardwaru, [1] je hlavním příkladem softwaru abstrahovaného od hardwaru. Kulturní historie komputace je bohatá na metafory a významy vpisované do formálních procesů a to samé platí i pro software. V roce 1970, pouze třináct let po ustanovení termínu Tukeyem, si ho metaforicky přivlastnila výstava Jacka Burnhama *Software*. V eseji *Systems Esthetics* z roku 1968 Burnham poznamenal, že psané dílo od konceptuálního umělce Donalda Judda „[...] připomíná něco, co by počítačový programátor zřejmě nazval entitou se strukturou seznamu.“ [2] Jak je vidět na těchto přirovnáních, Burnham nejenže převzal termín software jako metaforu konceptuálního umění, ale také proměnil a estetizoval tvorbu počítačového softwaru v umění koncepční.

Software jako praxe

Zrovna tak jako literaturou není jen to, co je psané, literatura zahrnuje i další kulturní činnosti – tradici orálního vyprávění, básnické přednesy, kulturní politiku – tak i software je jak materiálem, tak praxí. Jak ukazuje sloveso „googlovat“, které se užívá ve smyslu hledání pomocí vyhledávače *Google*, ale i v počítačném smyslu pro „brouzdání“, „povídání“, „stahování“, přímo z používání softwaru se rodí lidské činnosti. „Googlování“ je pouhou zkratkou pro užívání webového klient-server softwaru napsaného programátory společnosti *Google*. V tomto smyslu nejsou softwarem už jen algoritmy, které vykonává stroj, ale zahrnuje i interakci či kulturní apropiaci prostřednictvím uživatele. Toto přivlastňování je více než jen kybernetická interakce mezi člověkem a strojem a něco, co počítačová věda a mediální teorie často redukuje na ukazování, klikání a další pavlovovské reakce v rámci naprogramovaného

systému. Stejně zúžené chápání interakce zapříčinilo, že „interaktivní umění“ v obvyklé podobě behaviorálních videoinstalací umělecky směřovalo do slepé uličky. Pravá interakce s technickými systémy zahrnuje kreativní užití i zneužití vně černé skříňky, metaforizaci, psaní a přepisování, konfigurování a diskonfigurování, mazání. Všechny tyto činnosti také vykonávají počítačové programy.

Nebyly to ale jen umělecké apropiace, které se vepsaly do softwaru metafory. Vysokoúrovňové, na stroji nezávislé programovací jazyky a operační systémy jako *C* a *Unix* daly vzniknout zhruba ve stejném čase kultuře, která pozvolna oddělila software od konceptu kódu běžícího ve stroji. Skrze výpisy programového kódu v knihách a počítačových časopisech, skrze útržky zdrojového kódu a patche neboli „záplaty“ kolující v elektronických sítích nebo i ústní konverzaci, software začal žít svým životem. Výsledkem byla politicko-filozofická hnutí jako *Free software*, programátorské hříčky jako třeba rekurzivní akronymy, hackerský slang mísící angličtinu s konstrukty počítačového jazyka, anebo poezie v počítačových jazycích, například první báseň v jazyce *Perl* od Larryho Walla z roku 1990. Svobodný software – chápáný dle *GNU* jako zhmotněná hodnota, která je nejen inženýrskou svobodou, ale i ontologickou svobodou – je snad nejsilnějším příkladem kulturního a filozofického pojetí softwaru. Také v počítačové vědě kvete umělecké chápání softwaru, počínaje *Uměním počítačového programování* Donalda Knutha [3] až k současné knize *Hackerři a malíři* [4] od Paula Grahama, ačkoli může být založené na úzkém chápání umění jako vysoké řemeslné zručnosti. Upuštění od definice softwaru jako algoritmů běžících na hardwarovém zařízení nám pomáhá vyvarovat se všeobecného nepochopení softwarového umění jen jako jakési formy umění hrstky geniálních programátorů. Pokud je software širokou kulturní praxí, může softwarové umění tvořit téměř kterýkoli umělec.

Software versus hardware

Vedle mlhavosti odlišení softwaru jako strojového procesu od softwaru jako lidské kulturní činnosti je stejně tak nejasný i technický rozdíl mezi softwarem a hardwarem. Je příkazový kód hardwarem proto, že je vypálený na EPROM, jde snad o software, když je uložen na vymazatelném flash ROM? A co mikrokód, počítačové programy uchovávané přímo uvnitř čipů v každém moderním CPU (Central Processing Unit)? Nebo čipy jako Transmeta Crusoe, který má pouze minimální hardwarové zapojení a implementuje svůj CPU příkazový set – jako Intel-kompatibilní x86 – výhradně skrze integrovaný emulační software (napsaný původně tvůrcem Linuxu, Torvaldsem). A co BIOS nebo firmware počítačových základních desek, grafických karet a síťových adaptérů, bez kterého by tento hardware jednoduše nebyl provozuschopný? Není to nakonec jen úplně nahodilé rozlišování, jde-li o obvod propojený v uspořádání čipových tranzistorů, nebo je-li ta samá logika uchována uvnitř paměťového čipu? Ani samotná definice hardwaru není o nic méně diskutabilní. První moderní výpočetní hardware, Turingův stroj, materiálně neexistoval, ale byl jen teoretický a imaginární. To samé platí i pro počítač Mix Donalda Knutha. Kulturní historie komputace ukazuje, že i hardware může být metaforický, pokud algoritmy běží na jakémkoli představitelném materiálu, třeba i celém kosmu jako ve spekulaci Quirinia Kuhlmana. Přesto nakonec rozdíl mezi softwarem a hardwarem spočívá na karteziánských kategoriích: Je například lidský mozek, který vykonává výpočty, kusem hardwaru?

Pokud je třeba, aby dualita softwaru a hardwaru byla zrušena, pak pojetí softwaru jako nehmotného a hardwaru jako materiálního musí být opuštěno též. Rozdíl mezi materialitou a nehmotností existuje uvnitř softwaru samotného: algoritmus je hmotný ve své uchovávané, kódované podobě i ve většině

kulturních projevů s ním spojených. Nehmotným komponentem mohou být fantazie a falešné představy vkládané do softwaru, takže zasunutí CD-ROM, když navážeme na první příklad (2), může vyhodit Zemi do povětří anebo sexuálně rozpálit (nebo možná naopak). Některý software existuje především nebo i výhradně jen v imaginární formě:

- **vaporware**, mlžný produkt, ten stále slibovaný „trvalý upgrade“ padajícího počítačového programu, nebo stále odkládaná nová verze softwarového díla, jako je počítačová hra *Duke Nukem Forever*;
- virálně šířené poplašné zprávy (**hoaxy**), tj. e-mailové „memy“, které přikazují naivním čtenářům na svých počítačích mazat klíčové systémové soubory;
- **fiktivní virové infekce**; hardwarová selhání nebo lidské chyby špatně interpretované jako infekce počítačovým virem.

V případě, že umístění nebo dokonce sama existence nějakého softwaru není zcela jasná, pokud software není kódem běžícím na stroji, protože se objevuje jako pseudo kód v knize, nebo není ani funkčním algoritmem, jako například většina poezie v *Perlu*; je technická definice softwaru příliš limitovaná. Koneckonců, „software“ a „komputaci“ není možné od sebe jen tak striktně oddělit. Kulturní historie softwaru je i kulturní historií komputace.

Závěr

Software je kulturní praxe tvořená

- (a) z algoritmů;
- (b) v některých případech (ne však nutně) ve spojení s imaginárními nebo skutečnými stroji;
- (c) lidskou interakcí v širším smyslu jakékoli kulturní aropriace a užití;
- (d) spekulativní imaginací.

Historii softwaru lze proto vykládat jako historii intelektuální, a to v opozici proti mediálním teoriím, které považují kulturní imaginaci za sekundární produkt materiální technologie. Tato má knížka dávala za příklad v prvé řadě jazykové komputace, a to zejména proto, že jazyk sám o sobě může mít výpočetní charakter. Díky své abstraktnosti a gramatické struktuře také vyjadřuje výpočetní proces mnohem lépe než jakákoli jiná symbolická forma. Programovací jazyky se svou modifikovanou angličtinou jsou toho důkazem. Jazyky, ale stejně tak i architektura, mohou sloužit jako hlavní příklady kulturní historie komputace, protože obojí kombinuje formální instrukce s imaginací.

Kulturní dějiny komputace dokazují, že jsou stejně tak bohaté a plné protikladů jako dějiny kterékoli jiné symbolické formy. Zahrnují protiklady, algoritmy jako nástroj versus algoritmy jako materiál pro estetickou hru a spekulaci; komputaci jako vnitřní činnost přírody (např. v pythagorejském myšlení) nebo Boha (jako v kabale a magii) oproti komputaci jako kultuře a prostředku kulturní reflexe (počínaje hnutím Oulipo a hackerskými subkulturami šedesátých let 20. století); komputaci jako prostředek zrušení sémantiky (Bense) versus komputaci jako možnost strukturovat a vytvářet sémantiku (Lullismus a umělá inteligence), komputaci jako prostředek ke tvorbě totality (Quirinus Kuhlmann) versus komputaci jako prostředek k rozkládání věcí (Tristan Tzara, aleatorická literatura), software jako ontologická svoboda (GNU) proti softwaru jako ontologickému zotročení (Netochka Nezvanova),

extatická komputace (Kuhlmann, *Kabala*, Burroughs) versus racionalistická komputace (od Leibnize k Turingovi) versus patafyzická komputace jako parodie komputace racionalistické i iracionalistické (*Oulipo* a generativní psychogeografie), algoritmus jako expanze (Lullismus, generativní umění) oproti algoritmu jako omezení (*Oulipo*, net.art), kód jako chaotická imaginace (Jodi, kódová díla) versus kód jako strukturovaná strukturovaný popis chaosu (Tzara, John Cage).

Komputace a její imaginace jsou plné rozporů, nabitě metafyzickou a ontologickou spekulací. Za těmito protiklady a spekulacemi se skrývá posedlost kódem, který uskutečňuje přelud, že slova se stávají tělem. Ten však zůstává pouze přeludem, protože provedení kódu vždy zas a znovu selhává ve schopnosti naplňovat bezmezná spekulativní očekávání v něj vložená. Kulturní a politická sémantika vyplývá pouze z jeho tupých formalismů a jejich zásahů do každodenního života, od výpisů zůstatků na účtu až k „softwaru konečného uživatele“. Formalismy vytváří sémantiku zcela jinak, než lidé očekávají od domněle „inteligentního stroje“. Počítače tu tedy jsou od toho – jak praví hackerská moudrost – aby řešily problémy, o kterých bychom nevěděli, že existují, kdyby nebylo počítačů samotných.

Co je software? Právě tímto názvem pojmenoval Florian Cramer pátou, závěrečnou, kapitolu své knihy Words Made Flesh (2005), jejíž název odkazuje k biblickému výroku „slovo tělem učiněno bylo“. Co je software? Tak zní otázka, na kterou Cramer, jehož zajímá především kulturně podmíněné chápání programů, hledá dílčí odpovědi v celé knize a shrnuje je právě v závěrečné kapitole. Recenzi knihy Words made flesh, která je významným příspěvkem k mapování kulturní historie softwaru, jsme proto doplnili překladem této závěrečné, rekapitulující kapitoly Cramerovy knihy. Kniha vyšla v roce 2005 v Rotterdamu, Piet Zwart Institutu. Je dostupná též [online] z: <http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf>. Pozn. red.

Poznámky:

[1] Volný klon *Unixu NetBSD* je schopen provozu na téměř všech existujících hardwarových platformách včetně IBM-kompatibilních PC, kapesních počítačů, herních konzolí, starých domácích počítačů Amiga a Atari i proprietárních unixových serverů.

[2] BURNHAM, Jack, 1968. *Systems Esthetics*. *Artforum*, 9, 1968.

[3] KNUTH, Donald E, 1973–1998. *The Art of Computer Programming*. Reading, MA: Addison–Wesley.

[4] GRAHAM, Paul, 2004. *Hackers and Painters*. Sebastopol: O'Reilly.

Vysvětlivky k překladu:

(1) Tento výrok z filmu *Přepadení 2 – Temné území* Cramer blíže rozebírá v 1. kapitole.

(2) Viz již výše zmíněný výrok ze Seagalova filmu.