

# Zpravodaj Československého sdružení uživatelů TeXu

---

CSTUG editorial board

Donald Knuth – Computer Literacy Bookshops Interview

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 4 (1994), No. 4, 141–154

Persistent URL: <http://dml.cz/dmlcz/149723>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1994

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

---

---

DONALD KNUTH — **Computer Literacy**  
**Bookshops Interview**

---

*Interview vedený firmou Computer Literacy Bookshop, Inc., dne 7. prosince 1993\**

Donald E. Knuth patří podle mnohých do světové špičky v *computer science*. První tři svazky díla *The Art of Computer Programming*, které ani za 30 let neztratily na významu, mu přinesly Turingovu cenu ACM v roce 1974 a národní medaili za vědu v roce 1979. Knuth také vyvinul převratné aplikace v počítačové sazbě (T<sub>E</sub>X a METAFONT) a vývoji software (WEB) a má na svém kontě více než 100 publikací.

Nyní, kdy je emeritním profesorem Stanfordovy univerzity, vrhl Knuth opět veškerou svou energii do psaní. Dan Doernberg se jej v tomto rozhovoru z prosince 1993 ptal na to, co v poslední době dělal a na co se můžeme těšit.

---

**CLB:** Právě jste publikoval knihy jak o systému CWEB, tak i o *Stanford GraphBase*, dvou oblastech vašeho vlastního výzkumu. Začneme CWEBem, který je spojením jazyka C a T<sub>E</sub>Xu a slouží k dokumentování programů.

**Knuth:** Systém CWEB je nadstavbou C, a je to zdaleka nejlepší na světě známá programovací metoda. Musím být prostě upřímný a říci, že je to senzační věc. Kniha *CWEB System of Structured Documentation* je definitivní uživatelskou příručkou a vysvětluje toho víc, než může kdokoli potřebovat.

**CLB:** Řekl jste, že CWEB zvyšuje řádově produktivitu programátora — jak to?

**Knuth:** No, možná ne řádově, možná jen dvojnásobně. Lidé, kteří používali CWEB, zjistili, že píší lepší programy, které jsou přenositelnější, lze je snáze ladit, udržovat... a dají se také rychleji napsat.

**CLB:** Používá se CWEB jen ve Stanfordu, anebo také v průmyslu?

---

\* Originální verzi rozhovoru lze získat z WWW  
<http://www.clbooks.com/nbb/knuth.html>.

**Knuth:** Používá se na celém světě. Na řadě systémů jsme měli WEB, původní verzi pro Pascal, a řada lidí, kteří se jím infikovali, se pak stále rozrůstala. T<sub>E</sub>X byl napsán ve WEBu. Silvio Levy v roce 1987 provedl konverzi na CWEB. Ten byl dlouho experimentální, ale teď mohu říci „Experiment vyšel!“ CWEB je mnohem lepší než WEB, protože C je mnohem hezčí jazyk pro systémové programování a řadu jiných věcí. Nenapadá mne, proč by měl někdo, kdo se opravdu zajímá o programování, tímto pohrdnout ve prospěch jiného systému.

**CLB:** Snadno použitelné, rychlé, všechno bezvadné?

**Knuth:** Správně, a taky cítíte po napsání programu ohromné uspokojení!

**CLB:** I když napíšete špatný program?

**Knuth:** Skoro... no... jo! Zeptejte se Jill, jak párkrát za týden vyjdu ze své pracovny a říkám „To programování v CWEBu je taková zábava!“ Je to fakt, nemůžu se toho nabažit.

Duševní rozpoložení, v němž se ocitáte, když píšete program v CWEBu, je o hodně lepší než starý přístup. Představujete si, že píšete pro lidskou bytost, lidské bytosti vysvětlujete, co by měl počítač udělat, a ne že jste jen v roli toho, kdo má počítači nějak sdělit, co má dělat. Máte-li to ale vysvětlit jiné osobě, tak si to musíte sám daleko lépe sesumírovat. Tento postup pomáhá i v případě programu, který za hodinu zahodíte. CWEB je prostředek, který doporučuji používat, i když třeba píšete program jen pro sebe, pro své oči.

**CLB:** CWEB se zdá být hodně blízko modelům strukturovaného programování v 70. letech...

**Knuth:** Správně, jde o další krok. U strukturovaného programování někteří radili vytvářet programy shora dolů, a jiní zase zdola nahoru. S WEBem nebo CWEBem můžete kus udělat zdola nahoru a kus shora dolů, podle toho, co vám přijde vhodné z hlediska programu nebo té části, v níž zrovna jste.

Metodologie strukturovaného programování byla výborná... jenže k opravdovému pochopení ji nesmíte chápat jako kuchařku s předpisy, ale jako způsob porozumění vztahu mezi vyšší a nižší perspektivou pohledu na program. Uděláte to tak, že na celý program nahlížíte jako na síť, spoustu malých kousků, které jsou samy jednoduché a mají jednoduchá propojení s jinými malými kousky. Schéma WEBu právě podporuje tento způsob chápání složitého celku pomocí jednoduchých malých částí a propojení mezi nimi.

Ty části můžete vytvářet v takovém pořadí, které vám psychologicky nejvíce vyhovuje. Někdy je můžete udělat zdola nahoru. Zdola nahoru znamená, že nějak tušíte, že se asi bude pro něco hodit určitý podprogram, takže ho napíšete hned, když jste na to připraven nebo dokud máte náladu. Při tomto programování zdola nahoru se vaše tužka stává stránku od stránky mocnější, protože na straně devět vyvinete nástroje, které můžete použít na straně deset. . . vaše tužka je silnější.

Pokud jde o programování shora dolů, tam začnete zkraje a řeknete si „Tohle napíšu napřed a pak toto, a potom toto. . .“, ale později je na vás, abyste podrobně zorepsal, oč v každé části běží — může být, že po sto stránkách budete lapat po dechu, když konečně přijdete na to, jak to vlastně máte všechno udělat!

Na několika prvních stranách vypadá programování shora dolů obyčejně velmi pěkně, ale pak je čím dál těžší udržet všechny nitky v chodu. Programování zdola nahoru také často vypadá nějakou dobu pěkně, vaše tužka sílí, ale to právě zas svádí k různým trikům. Pokud oba přístupy nějak psychologicky vhodně zkombinujete, pak to funguje, dokonce i když končíte.

Takhle jsem to dělal s  $\text{T}_{\text{E}}\text{X}$ em, což je hodně velký program: víc než 500 stran zdrojového kódu v knize. V celém tom programu, u každé jeho řádky byla vždy jen jediná věc, kterou jsem musel udělat jako příští. Neměl jsem žádný velký výběr; každý krok se zakládal na tom, co už jsem do té doby udělal. Žádná metodologie by mne nenaučila, jak psát software tohoto druhu, pokud bych se jí přísně držel. Ale když jsem si představil, že ten program vysvětluji dobrému, schopnému programátorovi, tedy všechno, co v něm je, našel jsem jen jednu přirozenou cestu, jak to udělat. Uspořádání, v němž se kód nachází v knize, odpovídá tomu, jak jsem jej psal.

**CLB:** Do jaké míry jste sledoval nebo sledujete ony „svaté války“ — debaty o metodologiích softwareového inženýrství?

**Knuth:** Nesledoval jsem všechny nuance těchto věcí, ale dominantních myšlenek jsem si všiml. Až za mnoho let po jiných lidech jsem vlastně zjistil, co jsou to CASE nástroje. Myslím si, že nebylo dobře dělat z toho takové náboženství. V oněch dobách se způsob programování řídil „politickou korektností“.

Podobná věc se stala v matematickém společenství ve dvacátých letech, kdy se říkalo, že dobrý matematik musí dokazovat věty určitým způsobem. Nehodilo se používat jisté prostředky důkazu, o nichž někteří soudili, že vás mohou zavést do paradoxů. Bylo to něco jako pěstování

matematiky s jednou rukou uvázanou za zády. Politicky korektní strukturované programování podobným způsobem překázelo lidem v napsání dobrých programů, i když přesně věděli, co dělají, a to jen proto, že jejich postup zrovna nezapadal do současného chápání správnosti. *Computer science* je jako každý jiný obor; postupuje v módních vlnách. Některé z trendů jsou dobré, jenže skoro s každou dobrou myšlenkou se tak nějak zachází jinak, než jak by to mělo být.

Například si vezměte generátory náhodných čísel. Patnáct let jsme neměli žádnou teorii o tom, jak generovat náhodná čísla. Pak někdo dokázal takovou drobnost o jedné z metod: když se vzal průměr sériových korelací přes celou periodu čítající miliardu čísel, pak ten průměr byl nula, což bylo dobré. Z ničeho nic všichni všeho nechali, vyhodili své staré rutiny a přešli k této nové metodě, poněvadž to byla jediná metoda, k níž byla vůbec nějaká teorie. Ukázalo se, že to je hrozný generátor náhodných čísel; teorii ušlo, že průměr přes první polovinu je  $+1$  a přes druhou polovinu  $-1$ ! Vždy v celé historii si lidé vybírali metody a nedošla jim jejich omezení.

**CLB:** Co to bylo za metodu?

**Knuth:** No, ve většině knihoven podprogramů se jí říkalo RANDU. Touto dobou už by měla být dostatečně vymýcena; nicméně, jak někdo uvidí podprogram s názvem RANDU, pryč s ním!

**CLB:** Vedlo vás ke spojení WEBu s Céčkem to, že je dnes tolik programátorů používá, anebo se vám osobně C líbí a píšete v něm?

**Knuth:** Myslím si, že C má spoustu velmi důležitých vlastností. Například způsob, jímž C zachází se směrnicí, byl znamenitou inovací; vyřešil řadu problémů se strukturováním dat, s nimiž jsme se dříve potýkali, a vylepšil také vzhled výsledných programů. C ovšem není dokonalý jazyk, nakonec dokonalý není žádný z existujících jazyků, ale má podle mého spoustu kladů, a když se vám něco nelíbí, tak se tomu můžete vyhnout. Takže C jako jazyk se mi určitě líbí, zvláště proto, že je dobře sladěno s operačním systémem (pokud například používáte UNIX).

Po celý život si vždy vybírám ten jazyk, který je nejlépe sladěn s debuggerem a operačním systémem, které zrovna používám. Pokud bych měl pro nějaký jazyk X lepší debugger a X se dobře snášel s operačním systémem, používal bych tento jazyk.

Extrémní případ nastal, když jsem rok pracoval v laboratoři, kde operační systém navrhl Ned Irons. Byl to systém pro jednoho z raných Crayů a Irons také napsal kompilátor pro jazyk zvaný IMP. IMP měl řadu hrozných vlastností. Jednak to byl rozšiřitelný jazyk, a tak jej všichni neustále

rozšiřovali. Program, který v pondělí fungoval, vám v úterý fungovat přestal, čili po zřícení programu se vždycky muselo napřed zkontrolovat, zda je v pořádku strojový kód programu. Za druhé, IMP byl jazykem ohromně zhuštěným. Kupříkladu tam, kde v Pascalu píšete „if X>0 then ...“, v IMP řeknete „X+=>“. Jinými slovy, programy byly velmi krátké. Připadalo vám sice, že píšete elegantní programy, protože měly jen pár znaků, jenže druhý den už jste je nemohl vůbec přecíst! Cenou za úspornost tak bylo, že jste z té změti značek na stránce byl docela vedle...

**CLB:** V současné době zřejmě kladete důraz na „gramotné programování“, ale přitahoval vás někdy APL coby matematicky orientovaný jazyk?

**Knuth:** To je něco jiného. APL je pro lidi, kteří mají problémy k řešení a nestarají se tolik o efektivitu; chtějí pěkně a elegantně zformulovat řešení svého problému, i když toto řešení nemusí být nutně snadnou záležitostí pro počítač. Je to jazyk pro specifikaci problémů, nikoliv jazyk pro systémové programování... APL-WEB existuje.

Chtěl bych ale ještě něco říci k IMP. Třetí věcí proti němu bylo to, že v případě chyby se buď kompilátor dostal do nekonečné smyčky, anebo se zastavil na první chybě, oznámil „ERROR ERROR ERROR“ a skončil. Sám jste musel přijít na to, v čem je chyba. Žádný skvělý jazyk nebo kompilátor to tedy nebyl.

Nicméně... stejně jsem si tenhle jazyk zvolil, protože bezvadně zapadal do onoho operačního systému. Názvy polí se volily tak, že byly snadno vidět v debuggeru, a mohl jste zjistit, jakým způsobem se přiděluje paměť, věděl jste, co se děje, a konec konců jste mohl program zvládnout tak, aby běžel spolehlivě, poněvadž IMP byl sladěn s operačním systémem. Mohl jste psát v lepším jazyku, ale odpovědi byste jen tak nezískal a celá práce by vám trvala o pár týdnů déle. Já jsem používal IMP.

**CLB:** Používal se IMP ve Stanfordu?

**Knuth:** Tohle bylo ve výzkumné laboratoři v Princetonu. Rok před mým příchodem do Stanfordu jsem tam dělal na výzkumném projektu klasifikované analýzy šifer.

**CLB:** Řekněte nám, prosím, o vaší druhé knize, která se týká stanfordské *GraphBase*.

**Knuth:** Kniha o *GraphBase* je určena pro dva druhy lidí. Jeden její účel je výzkumný; lidé, zabývající se studiem algoritmů pro kombinatorické problémy, potřebují standardní sadu zkušebních dat, na nichž se mohou navzájem poměřovat, a také jako výkonnostní normu. Během přípravy

svazku IV řady *The Art of Computer Programming* jsem se rozhodl, že zpřístupním všem příklady a data, které používám. Byla tu potřeba standardních výkonnostních norem, a taky by to všechno mělo být uspořádáno tak, aby se to mohlo použít tisícem různých způsobů. Takže... mám teď sbírku tisíců standardních datových sad; kdokoli v Polsku může mít data přesně stejná, a zrovna tak kdokoli v Kalifornii nebo v Číně. Je to velmi přenositelné a lze to stáhnout z Internetu.

Druhým důvodem pro odbočku ke *GraphBase* je to, že jde o příklad programování v *CWEBu* — je to vlastně 32 příkladů programování v *CWEBu*. Jsou to krátké programy ilustrující ten styl programování, jež já dávám přednost. Příklady jsou cosi jako drobné eseje, kratičká vyprávění formou počítačových programů, které se snad hezky čtou.

**CLB:** Jaké je vaše současné hardwareové a softwareové prostředí?

**Knuth:** Pro své programování používám *CWEB*, velice využívám editor *Emacs*, a pro kreslení technických ilustrací používám výtečný jazyk vysoké úrovně, který se nazývá *META O T*. Je to nový jazyk, autorem je John Hobby a myslím, že by měl být už brzy zveřejněn. Je založen na *METAFONTu*. 75% kódu pochází ode mne z *METAFONTu*, ale je to upraveno tak, že z toho vychází *PostScript*. Moc se mi líbí.

Také používám program *Mathematica*. Lidé od *Maple* se mě pokoušejí přesvědčit, abych přešel k *Maple*, jinému výtečnému systému. Momentálně se mi zamlouvá *Mathematica*, poněvadž nemusíte psát značky násobení; stačí říct „2X“ místo „2\*X“. Taktéž má *Mathematica* výjimečně dobrý manuál.

**CLB:** Takže se vám líbí *Wolframův* styl psaní?

**Knuth:** Zejména rejstřík... V té knize se jen tak neztratíte. U prvního vydání jsem se díval do rejstříku, když jsem měl řešit nový problém, a skoro vždycky mě odkázal na správnou stránku. Snad ve třech nebo čtyřech případech tam hledané slovo nebylo, což jsem si vždy v knize označil tužkou, až se k tomu dostanu příště. V druhém vydání to už bylo všechno opraveno, a to jsem to ani nikomu nehlásil.

**CLB:** Rád bych se zeptal na vaše letmé dojmy o několika výzkumných směrech a také zda jste v nich něco studoval nebo udělal. První z nich jsou genetické algoritmy. Co říkáte té obecné myšlence, že algoritmus nemusí určovat člověk, ale může se nechat na stroji... .

**Knuth:** Plánuji s tím řadu experimentů, jakmile se dostanu ke svazku IV. Máme tu genetické šlechtění, pak simulované žíhání (*simulated annealing*), a lidé vymysleli ještě jiné strategie. V knize o stanfordské *GraphBase* mám metodu, které říkám „stratifikovaná lačnost“. Všechny tyto

postupy soupeřů o stejný druh problémů a já bych rád zkusil hodně příkladů; některé mohou být lepší tady než jinde, takže bych pro to rád získal nějaký cit. Určité problémy jsou jako stvořené pro neuronové sítě. . . genetické algoritmy budou nejspíš vycházet dobře v úlohách kolem rozpoznávání řeči a prý také v odhadech burzovních pohybů nebo v něčem takovém. Jaksi čím je problém blíže přírodě, tím více lze počítat s tím, že genetický algoritmus bude fungovat, zatímco čím je bližší teorii čísel nebo něčemu umělému, tím spíše se dá čekat pomoc od nějakého jiného přístupu. Je dost těžké porozumět vlivu velikosti úlohy na tyto metody; na malých úlohách mohou běžet božsky a pak najednou zcela zkrachují, když se úloha jen trošičku zvětší. . . anebo to může být obráceně.

**CLB:** Vypadá to tak, jako byste měl před sebou pár let disciplinovaného testování na vašich sadách dat.

**Knuth:** Stanfordská *GraphBase* mi dává neomezený zdroj úloh, na nichž mohu dělat já i jiní. Pročítám si, co jiní lidé prohlásili o svých metodách, ale také si je všechny zkuším. Můj původní přínos v *The Art of Computer Programming* spočívá v tom, že vezmu metody od dvou různých autorů a analyzuji metodu A z pozice autora B a metodu B z pozice autora A. Každý to popsal jen ze své strany, tak zkuším vyplnit mezery. . .

**CLB:** A co objektově orientované programování? Je to jen módní heslo anebo se vám opravdu zamlouvá?

**Knuth:** O programování jsem vždycky přemýšlel v tomto duchu, ale nepoužíval jsem jazyky, které pomáhají dodržovat disciplínu. K disciplíně jsem se vždy přinutil sám i v jiných jazycích. Programovací jazyky vás teď mohou chytit, když uděláte chybu, a ulehčují vám omezovat viditelnost informací mezi jednotlivými částmi programu navzájem. Ve svých programech napsaných ve starších jazycích jsem nepoužíval to, co jsem používat neměl; musel jsem se sám ukázněvat, abych se těmito pravidly řídil. Bylo to možné, a tak jsem to dělal. Nešlo o to, že bych nějaké programy nemohl napsat. . . ale ty nové prostředky určitě pomáhají.

Problém, který s nimi dneska mám, je v tom, že. . . C++ je moc komplikované. Momentálně je pro mne nemožné psát přenositelný kód, o němž bych věděl, že poběží na mnoha různých systémech, jedině snad pokud bych se vyhnul všem exotickým prvkům. Kdykoli se tvůrcům jazyka C++ naskytly dva konkurenční postupy pro řešení nějakého problému, řekli si „OK, dáme je tam oba“. Tím pádem se jazyk stal pro



moje gusto příliš barokním. Každý uživatel C++ si ale stejně vybírá jen nějakou oblíbenou podmnožinu, takže se nic neděje. CWEB plně podporuje C++ zrovna tak jako C.

**CLB:** Co si myslíte o takových oborech, jako je teorie chaosu a fraktály? Nedeterminovanost v nich obsažená se zdá být poněkud v rozporu s oblastmi, na něž jste se v minulosti soustředil.

**Knuth:** Zpočátku jsem se trochu zabýval fraktály a podobnými věcmi a zdá se mi, že je to vynikající nová abstrakce. Lidi mohou budovat modely, které by je dříve ani nenapadly a které přitom skutečně vystihují řadu věcí v přírodě, majících tu vlastnost, že vypadají stejně i při změně měřítka. Jak známo, když zvětšujete pobřežní čáru, vypadá stále jako pobřežní čára, a spousta jiných věcí má tento charakter. Příroda disponuje rekurzivními algoritmy, které používá pro vytváření mraků a ementálu a tak. Takže teď máme matematické postupy, s jejichž pomocí můžeme porozumět procesům, které sahají za hranice diferenciálních rovnic, na něž byli lidé zvyklí v předchozích staletích. Dostali jsme do rukou úplně nový nástroj, ale moje intuice mi toho o těchto metodách mnoho neříká. Jsem si vědom omezenosti své intuice; dovedu některé problémy dobře vyřešit, ale znám jiné lidi, kteří z fleku vyřeší něco, co mně sebere spoustu času. . . . Není to můj šálek čaje.

**CLB:** Do jaké míry jste kdy sledoval rozvoj umělé inteligence? Váš třetí program v životě byl „kámen–nůžky–papír“, který se učil z vlastních chyb, a Stanford je odjakživa jednou z předních institucí přes výzkum umělé inteligence. . . .

**Knuth:** No, umělá inteligence hodně souvisí se svazkem IV; badatelé v tomto oboru užívají kombinatorické postupy, které já studuji, takže vím o spoustě relevantní literatury. Mým úkolem je porovnání literatury o umělé inteligenci, která vyšla z kruhů elektrotechnického inženýrství, s tou, která pochází z jiných disciplín; každá komunita přistupuje k problémům trochu odlišně. Zkouším tyhle věci číst, očistit je od žargonu a sjednotit myšlenky. Z umělé inteligence pocházely odjakživa ty nejtěžší aplikace a nejatraktivnější problémy v *computer science* — umělá inteligence byla nejpłodnějším zdrojem metod v *computer science*. Vedla k řadě důležitých zlepšení, jako jsou datové struktury a práce se seznamy. . . . umělá inteligence toto ohromně stimulovala. Mnohá z nejlepších paradigmat pro ladění a vývoj programů, všechny systémy počítačové algebry, rané studie počítačové grafiky a počítačového vidění, atd., to vše mělo velmi silné kořeny v umělé inteligenci.

**CLB:** Nepatříte tedy mezi ty, kdož bagatelizují vše, co bylo v této oblasti uděláno. . .

**Knuth:** Ne, kdepak. Stalo se totiž to, že spousta lidí věřila, že umělá inteligence se stane všelékem. To máte, jako když nějaká firma udělá jen 15 % zisku, zatímco jí analytici předpovídali 18 %, a její kurs hned spadne. Bylo to jen nesplněné očekávání, které vzniklo z nabubřelých představ, že jedno paradigma vše vyřeší. Je to tak asi se všemi věcmi, které jsou teď na výsluní; lidi si časem uvědomí, že nejsou vyčerpávající odpovědi. Mnohé problémy jsou tak těžké, že se nám nikdy nepovede najít opravdu velkolepé řešení. Lidé jsou zklamání, když nenajdou elixír mládí. . .

**CLB:** Kdybyste měl těsně před promoci na univerzitě nebo byl doktorem a neměl žádnou „zátěž“, jaký druh výzkumu byste chtěl dělat? A vybral byste si vůbec znovu výzkum?

**Knuth:** Myslím, že nejbáječnější počítačový výzkum se teď dělá zčásti v robotice a zčásti v biochemických aplikacích. Robotika, kupříkladu, to je něco fantastického. Dělat stroje, které se skutečně pohybují a navzájem komunikují. Stanford má teď velkou laboratoř robotiky a naším plánem je umístit do nové budovy stovku robotů, kteří budou kráčet po chodbách. Chceme tím stimulovat studenty. Do nové budovy se nastěhujeme tak za dva za tři roky. Jenom to, že tam ty roboty uvidíte, vás přivede k hezkým projektům. Z takových projektů též vznikají mnohé dobré matematické a teoretické otázky. A grafické nástroje vysoké úrovně, tady je taky spousta skvělých věcí. Jo, to by se mi moc líbilo dělat. . . máme jen jeden život, to se ví, ale stejně. . .

**CLB:** Proč jste se zmínil o biochemii?

**Knuth:** Jsou tam milióny a milióny nevyřešených problémů. Biologie je tak digitální a přitom tak neskutečně složitá. Průšvih s biologií je ten, že práce biologa je nezáživná. Vaše experimenty vás stojí tři roky práce a pak během jedné noci vypnou proud a všechno vám to chcípne! Musíte začít znova. V počítači si můžeme vytvořit vlastní světy. Biologové zaslouží velký obdiv za to, že se tím dokáží prokousat.

Těžko mohu zodpovědně prohlásit, že po padesáti dalších letech explozivního růstu *computer science* tu ještě zbude hodně fascinujících nevyřešených problémů, že se nebude do značné míry pouze pracovat na drobných vylepšeních dobře prozkoumaných věcí. Může být, že všechny elementární věci a všechny opravdu dobré věci už budou v té době objeveny. Možná, že ne, ale rozhodně nemohu předpokládat růst bez konce. U *computer science* si nemohu být tak jistý, jako jsem si jist u biologie.

Ta má před sebou 500 let práce na vzrušujících problémech, v takovém je to stadiu.

**CLB:** Používání Internetu se teď bouřlivě rozvíjí, každý se připojuje. . .

**Knuth:** Jednoho dne taky zkusíme zjistit, kdo to vlastně všechno platí!

**CLB:** Vy sám jej teď používáte? Víím, že tomu tak bylo v minulosti.

**Knuth:** Po dobu patnácti let jsem používal elektronickou poštu na ARPANETu a Internetu. V lednu 1990 jsem s tím nakonec přestal, protože mě přehrabování toho smetí připravovalo o příliš mnoho času. Nemám emailovou adresu. Lidé, kteří mi zkoušejí posílat nevyžádané elektronické zprávy, dostávají slušnou odpověď „Profesor Knuth skončil se čtením elektronické pošty; můžete mu napsat na takovou a takovou adresu.“

Email se nedá vypnout! Pošlete někomu zprávu, ten ji pošle s poděkováním zpět a vy mu zas odepíšete „Děkuji za vaše poděkování. . .“

Email je báječný pro určitý druh lidí, je absolutně nepostradatelný pro jejich povolání a umožňuje jim dělat jejich práci lépe. Řekl bych, že pro lidi, jejichž úkolem je být na vrcholu dění, je email skvělý. Avšak mou rolí je sledovat dění zespodu. Sleduji různé myšlenky, pozorně o nich přemýšlím a snažím se je přepsat načisto. . . Probírám se zvolna věcmi již dříve udělanými a zkusím tento materiál zorganizovat. Ale nemám ponětí o tom, co se děje tento měsíc.

Takže teď nečtu elektronickou poštu, ale občas ji stejně používám. Řekněme když cestuji do Izraele a musím něco zařídit na poslední chvíli. Když navštívím na pár dnů jinou univerzitu nebo výzkumné středisko, musím odtud poslat email. Naučil jsem se používat emailové prostředky v Emacsu, ale nechci se v tom stát přeborníkem.

**CLB:** Máte spoustu zájmů nesouvisejících s počítači a matematikou — hudbu, náboženství, literární tvorbu. Je pro vás hudba tvůrčím ventilem, prostředkem pro rekreaci, nebo duchovním ventilem?

**Knuth:** Momentálně jde o rekreaci. Rád si zvu domů přátele, s nimiž hraje čtyřruční klavírní skladby. Kdybych to mohl dělat každý týden, tak bych to dělal. Doufám, že budu žít dost dlouho na to, abych po ukončení svého životního díla *The Art of Computer Programming* mohl komponovat hudbu. To je jen sen. . . ovšemže by to třeba byla mizerná hudba.

**CLB:** Už jste ale nějaké skladby napsal, že?

**Knuth:** Ano, ale byla to většinou aranžmá motivů jiných lidí. Tedy, když jsem byl na vyšší škole, napsal jsem krátkou hudební komedii nazvanou „Říše Nešťastníků“. Vzpomínáte si, jakým hitem byli Nešťastníci

(*Nebbishes*) koncem padesátých let? „Říše Nešťastníků“ byla jen asi desetiminutovou parodií, ale mělo to všechno původní hudbu a libreto.

**CLB:** Máte ty noty ještě někde na půdě?

**Knuth:** Jo. . . vlastně ne, myslím, že jsem je ztratil. Mám jich jen část. Doufám, že na ně znovu narazím. Procházím teď své složky a dělám si počítačový rejstřík všeho, co doma mám.

**CLB:** Nezdá se, že byste měl domácnost bez papírů!

**Knuth:** Kdepak!

**CLB:** Zkoušel jste hudebně počítačovou technologii MIDI, anebo jste se od ní záměrně držel stranou?

**Knuth:** Baví mě to. Minulé vánoce jsem koupil pro svého syna syntezátor a hrál jsem na něj sám dlouhé hodiny. Moc se mi to líbilo. Kdysi před lety, bylo to v domě Marvina Minskyho, jsem hrál na syntezátor Kurzweil, což byla imitace koncertního křídla. Ne tak dávno odjel jeden můj přítel na tři roky do Anglie a nechtělo se mu brát s sebou své křídlo, a tak si koupil šestihlasou Yamahu. Když jsem ho navštívil, mohl jsem po tři nádherné dny procházet všechny kusy, které jsem se na piano naučil, a hrál jsem je jakoby na vibrafon, nebo na cemballo, nebo jinými hlasy. Jeho „piano“ má zvuk cemballa, ale klávesnice je citlivá na sílu stisku, takže můžete hrát nahlas a potichu, což na opravdovém cemballu nejde. Tyhle syntezátory jsou skvělé.

**CLB:** Kdy jste odešel ze Stanfordu do výslužby?

**Knuth:** Letos. Byl jsem na dvouroční dovolené, než jsem mohl oficiálně odejít do výslužby. Neoficiálně jsem odešel v roce 1990, ve stejný den, kdy jsem se vzdal emailu. Své plány jsem ohlásil ještě o tři roky dříve. Ujasnil jsem si, že mým hlavním životním cílem je dokončení *The Art of Computer Programming*; promyslel jsem to do budoucnosti a vyšlo mi, že by to chtělo dvacet let práce na plný úvazek. Kdybych pokračoval ve všem ostatním, co jsem tehdy dělal, stalo by se z toho čtyřicet nebo padesát let práce. Nemohl jsem se nikam pohnout, měl jsem čím dál větší zpoždění. Tak jsem řekl „dost“. Přirozeně mi bylo proti srsti vzdát se mnoha z těch věcí, které dělám s velkým potěšením. Ale některých věcí jsem se milerád zbavil, například psaní návrhů. Těch jsem se vzdal s nadšením!

**CLB:** Vy jste musel psát nějaké návrhy?? Domníval jsem se, že jste od toho byl nějak izolován.

**Knuth:** Máte obdivuhodný smysl pro humor! Teď už to nemusím dělat; ale jako profesor jsem musel najít sponzory, když jsem chtěl zajistit

pro své studenty slušné vybavení, nebo přijímat hosty v rámci aktivních výzkumných programů, publikovat zprávy, atd. Žebření o peníze stojí spoustu úsilí. System Development Foundation mi nabídla milión dolarů, abych mohl dodělat  $\TeX$  a vrátit se k *The Art of Computer Programming*.

**CLB:** Vzal jste ty peníze?

**Knuth:** Jistě, ale dokončení  $\TeX$ u stejně ještě trvalo mnoho, mnoho let. Řekl jsem si, že jediná naděje na dokončení *The Art of Computer Programming* spočívá v přechodu na psaní na plný úvazek, na život poustevníka a v tom, že začnu lidem říkat „ne“. V prvních letech to šlo dost těžko. Teď už si připadám opravdu výkonný a psaní jde dobře. Příjemný ustálený stav.

Tak jednou za měsíc mám přednášku ve Stanfordu, když se tam dostanu. Nazývá se „Počítačové hrátky“. Chci v tom pokračovat dvacet let a každý měsíc pohovořit o něčem, co mne zrovna zaujalo, o pěkných myšlenkách, které jsem posbíral. . . Vytahuji problémy, s nimiž si nevím rady, aby je někdo udělal za mne. Když teď nedokážu pohnout s problémem za dvě hodiny, nezbývá mi než toho nechat a požádat někoho, aby se tomu věnoval; jinak budu zase zaostávat. Jak píšu knihu, musím přecházet od námětu k námětu a jednomu každému věnuji pozornost tak tři týdny.

**CLB:** Jste ponejvíce znám svými publikacemi a výzkumem; líbilo se vám učit a být ve styku se studenty?

**Knuth:** Měli jsme nejlepší studenty na světě. Se studenty se stále setkávám při svém přednáškovém cyklu, až na to, že už neznám jejich jména. To je potíž.

**CLB:** Nemáte žádné studentské praktikanty?

**Knuth:** Představte si, že při „Počítačových hrátkách“ nadhodím otevřený problém, a řekněme, že mezi posluchači se najde nějaký student, který ten problém vyřeší, udělá z toho disertaci a tu během příštích dvou (nebo snad dvou a půl) týdnů dokončí a ukáže mi ji. V takovém případě by mě to ještě zajímalo, ještě bych si to přečetl a s radostí bych se mu podepsal pod obhajobu. . . , ale to je jediný možný způsob. Dvacet osm je celkový počet doktorandů, kteří u mne promovali, a více jich asi mít nebudu. . . jedině, když něco bleskově vznikne v rámci mých „Počítačových hrátek“.

**CLB:** Doktorské práce v reálném čase! Jakých změn jste si všiml během těch let u studentů přicházejících na obor *computer science*?

**Knuth:** Došlo k velké změně, kterou si neumím vysvětlit. V sedmdesátých letech se většina našich studentů velice zajímala o hudbu. Ptávali jsme se jich hned, jak přišli: „Na jaký nástroj hrajete?“ Měli jsme spousty komorních seskupení a podobně. Dneska skoro nikoho z našich studentů hudba nezajímá. Nevím, jestli je to proto, že se na *computer science* zapisuje jiný typ lidí, anebo zda to platí pro všechny dnešní studenty, nebo co. Když se dnes zeptáte studentů *computer science*, jakého mají koníčka, nejspíš vám většina z nich řekne „cyklistiku“. Nedávno jsem měl jednoho, který hrál na foukací harmoniku, ale jinak v té skupině nebyli skoro žádní muzikanti.

**CLB:** Nějaké změny v úrovni studentů?

**Knuth:** V úrovni ne... Ale o matematice toho ví méně, než bývalo dříve zvykem. Musíme se tak více věnovat doškolování, dokonce i na takové škole, jakou je Stanford.

**CLB:** A co změny v oboru samotném... při takovém pokroku a počtu zúčastněných lidí, liší se dnešní *computer science* hodně od té dřívější?

**Knuth:** No, jsou tu všechna ta média a vizuální věci, to je velký rozdíl proti dřívějšímu. Také je tu konkurence; je to teď o hodně obtížnější, než to bylo za mých časů. Když jsem začínal, dalo se přijít s něčím novým daleko snáze než teď, kdy tu máte tisíce a tisíce chytrých lidí, kteří všichni dělají skvělé věci. Onehdy mohlo být takových deset výborných doktorských disertací za rok; dneska prostě není šance držet s tím vším krok.

Ať jste v kterémkoli oboru *computer science*, ukazuje se všeobecně, že se těžko drží krok. Jednotlivé obory se stále zužují, protože nikdo už není schopen pokrýt celé území. Každý si musí vybrat dvě malé části *computer science* a nastudovat je; jestliže jeden zná části A a B, další B a C a jiný zas C a D, zůstává obor rozumně dobře propojen, i přesto, že se stále rozšiřuje.

**CLB:** Nahlížíte na sebe jako na jednoho z posledních „renesančních lidí“ v oboru *computer science*?

**Knuth:** Nemám tak široký záběr, jak byste si mohl myslet — nepracuji na více věcech zároveň. Řekl bych, že mám schopnost rychle studovat; umím se v momentu stát expertem přes něco. Po dobu třiceti let sbírám materiál, takže mohu číst literaturu o jakémkoli tématu v „dávkovém režimu“ — nemusím přeskakovat mezi mnoha různými tématy. Dokážu absorbovat daný předmět lokálně a být v tom chvilku dobrý... ale nechtějte pak po mně, abych předvedl to, čím jsem se zabýval před pár měsíci! A taky mám spoustu lidí, kteří mi pomáhají opravovat mé chyby.

**CLB:** Poslední otázka, pro vás ta nejprotivnější. . . Jaký je váš současný plán na dokončení všech sedmi svazků *The Art of Computer Programming*?

**Knuth:** Chystám se vydávat dvakrát ročně fascikly o zhruba 128 stranách. Nejdříve shromáždíme čtyři z nich a pak teprve vypustíme první dva; budeme si nechávat něco v záloze! První fascikly čkejte v roce 1995 nebo 1996; budou to testovací beta verze opravdových knih. Počítám, že bych mohl dokončit svazek IV (části A, B a C) v roce 2003, svazek V v roce 2008, pak přijít s novými vydáními svazků I, II a III, potom dělat na VI a VII. Bude i jedna verze jakožto „čtenářský výtah“ ze svazků I až V.

**CLB:** Jaká by byla vaše kariéra a život, kdybyste nebyl ohlášil ten sedmisvazkový soubor?

**Knuth:** Počkat, nejdříve jsem to neohlašoval. Myslel jsem si, že píšu jedinou knihu. Ale i kdybych to nebyl udělal, počítám, že bych stejně hodně psal. Nějak se zdá, že po celou tu dobu jsem vždy nacházel zálibení v pokusech vysvětlovat věci. Na střední škole jsem vydával studentský plátek; na vyšší škole jsem redigoval časopis. Vždycky jsem si hrál se slovy.

---

(překlad: *Ladislav Lhotka*)

*Následující příspěvek byl původně napsán v angličtině pro EuroTEX 94. Vždycky jsem si myslel, že je pro autora snadné napsat pak totéž i česky, takže do posledního okamžiku jsem doufal, že mě to nebude stát víc energie, než kolik je potřeba na přetukání. To, že článek vychází česky teprve teď, kdy hlavní důvody pro jeho napsání částečně pominuly, je bohužel jen moje ostuda.*

---

---

## Jak si poradit s velkými METAFONTovými obrázky

KAREL HORÁK

Před nějakým časem referoval o podobném problému na stránkách tohoto občasníku kolega Zdeněk Wagner [2]. Řešení, které je v jeho článku navrženo, předpokládalo rozklad obrázku na několik menších obdélníků