Istoric

Evoluţia acestui limbaj de tehnoredactare este extrem de interesantă în ea însăşi, şi ne va cauza întîlnirea cu nişte giganţi ai informaticii. Pe lîngă aceste aspecte anecdotice, multe din idiosincraziile limbajului provin din deciziile făcute de designer-ul iniţial, Donald Knuth.

Donald E. Knuth

Donald E. Knuth este una dintre figurile cele mai impresionante din istoria informaticii. Deşi pagina sa de web este foarte comprehensivă, http://www-cs-faculty.stanford.edu/~knuth/, merită să ne oprim pe scurt asupra personajului. Este genul de individ care are nevoie de o pagină şi jumătate pentru a face lista tuturor premiilor, distincţiilor academice şi doctoratelor onorifice pe care le-a primit (acestea din urmă deocamdată 21 în număr). Excelenţa sa academică în informatică este dublată de o cultură enciclopedică, care se reflectă în stilul său literar deosebit de interesant. Chiar dacă, din punct de vedere tehnic, multe din cărţile sale sunt relativ greu de citit, ele sunt întotdeauna o combinaţie foarte plăcută de ştiinţă, literatură, umor şi înţelepciune.

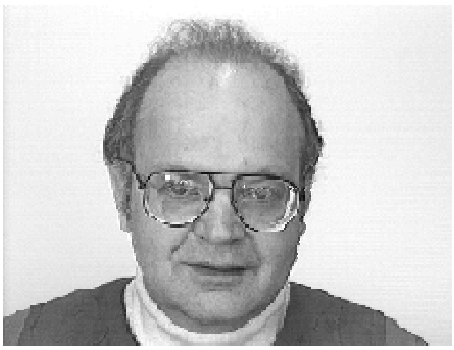

Figura: Donald Knuth, creatorul TEX.

Domnul Knuth este începînd din 1968 profesor la universitatea Stanford din California. Doctoratul său este însă în matematică, obţinut în 1963 la universitatea Caltech. Articolele sale în general sunt la intersecţia dintre matematică şi calculatoare, dar are foarte multe publicaţii de matematică pură; Don Knuth (cum este cunoscut printre prieteni şi admiratori) este responsabil de altfel pentru multe din metodele matematice de analiză a algoritmilor folosite astăzi în mod curent. Multe din aceste metode au fost

expuse în volumele publicate din ampla lucrare ``Arta programării calculatoarelor''. Această serie a fost începută în 1969, şi trebuia să conţină şapte volume masive. Primele trei au fost publicate în 1968, 1971 şi respectiv 1973, şi au fost traduse şi în româneşte, la Editura tehnică, între 1974 şi 1976; sunt cu siguranţă disponibile la o mulţime de biblioteci şi anticari.

Aceste cărţi se doreau a fi o incursiune în toate tehnologiile majore din informatică la vremea respectivă, dar conţin şi o mulţime de rezultate originale ale lui Don Knuth. Ca urmare, în 1974 i-a fost decernat cel mai prestigios premiu din informatică, ``Nobel''-ul din calculatoare, premiul Turing.

Frustrat de dificultatea de a produce documente tipărite de bună calitate, care să conţină formule matematice, înainte de a începe volumul 4 din serie, Knuth s-a apucat să studieze problemele tipografiei digitale. Astfel a început o excursie de un deceniu în acest domeniu, care a dus la crearea sistemelor TEX şi METAFONT, pentru tehnoredactare computerizată şi pentru crearea automată de seturi de caractere (font-uri). Pentru a putea împărtăşi întregii lumi creaţiile sale, Knuth a făcut codul-sursă al lui TEX disponibil în întregime, gratuit dintru început. Pentru a putea apoi explica codul sursă, Knuth a creat un sistem numit WEB (a nu se confunda cu World-Wide-Web), care permite compunerea simultană a unui program şi a documentaţiei sale. Sistemul WEB este explicat de Knuth într-un articol celebru numit ``Literate Programming'', despre programarea ca o formă de literatură! Rezultatele acestui proiect în tipografia digitală au fost tipărite în 1986 sub forma a cinci volume numite ``Computers and Typesetting'' (Calculatoarele şi tehnoredactarea). Două dintre cărţi sunt manualele TEX şi respectiv METAFONT, alte două cărţi conţin codul sursă al programelor TEX şi METAFONT, descrise în sistemul CWEB (care este o implementare a sistemului WEB pentru limbajul C), iar ultimul volum conţine descrierile parametrilor şi programelor METAFONT utilizate de Knuth pentru a genera setul de caractere Computer Modern.

Creaţia lui Knuth nu se termină aici; este autorul a peste douăzeci de cărţi şi a cîteva sute de articole, de la matematică, analiza algoritmilor, tipografie digitală, pînă la analiza unor texte biblice!

Domnul Knuth are o pagină de web foarte comprehensivă, care reflectă propensiunile sale literare, dar a renunţat de mai bine de zece ani să mai folosească poşta electronică, din cauză că primea prea multe scrisori de la nenumăraţii fani. În urmă cu cîţiva ani a ieşit la pensie, dar asta nu înseamnă că s-a oprit din activitatea didactică în întregime; ţine în continuare prelegeri

la universitatea Stanford şi în alte locuri unde este invitat, şi afirmă că vrea să-şi dedice majoritatea timpului rămas completării celor şapte volume din ``Arta programării", a căror serie a fost întreruptă temporar în urmă cu 26 de ani. Afirmă de asemenea că nu mai primeşte studenţi în calitate de îndrumător la doctorat, dar că va semna teza de doctorat a oricui rezolvă în timp scurt (2-3 săptămîni) una din problemele nerezolvate pe care le propune din cînd în cînd în prelegerile sale (aviz amatorilor de un doctorat pe fugă).

Aceasta este schiţa de portret a unuia dintre cele mai pitoreşti personaje din informatică. Chiar dacă ne-am abătut de la subiectul acestui articol, nu am rezistat să nu vorbesc puţin despre el.

TEX

Vom vedea mai departe că LATEX nu e nimic altceva decît un pachet de definiţii (macro-uri) în limbajul TEX creat de Knuth. Ca atare vom arunca întîi o privire rapidă asupra acestui limbaj.

Versiunile succesive de TEX sunt numerotate cu zecimalele numărului π : la ora actuală s-a ajuns la versiunea 3.14159.

Numele

Primul lucru care trebuie clarificat este pronunţia: TEX se citeşte de fapt ``tech", şi nu ``tex". Iată explicaţia lui Don Knuth pentru etimologie, extrasă din primul capitol al cărţii de TEX:

English words like `technology' stem from a Greek root beginning with the letters $\tau\epsilon\chi\ldots$; and this same Greek word means *art* as well as technology. Hence the name TEX, which is an uppercase form of $\tau\epsilon\chi$.

Insiders pronounce the χ of TEX as a Greek chi, not as an `x', so that TEX rhymes with the word blecchhh. It's the `ch' sound in Scottish words like *loch* or German words like *ach*; it's a Spanish `j' and a Russian `kh'. When you say it correctly to your computer, the terminal may become slightly moist.

în traducere:

Cuvinte ca ``tehnologie'' sunt formate dintr-o rădăcină grecească care începe cu literele $\tau\epsilon\chi$ (tau, epsilon, chi); acest cuvînt grecesc înseamnă el însuşi ``artă'' dar şi ``tehnologie''. De aceea numele TEX, care este scrierea cu litere majuscule a cuvîntului $\tau\epsilon\chi$.

Cunoscătorii pronunţă litera χ din TEX ca pe grecescul ``chi'', şi nu ca pe un ``x'', astfel încît TEX rimează cu cuvîntul blecchhh. Este ca sunetul ``ch'' din cuvinte scoţiene ca *loch* sau cuvinte germane ca *ach*; este ``j''-ul spaniol şi ``kh''-ul rusesc. Cînd este pronunţat corect în faţa calculatorului, va face ca terminalul să se umezească.

Tehnoredactat corect, litera E din TEX trebuie să apară mai jos decît celelalte (vedeţi şi figura 4), tocmai pentru a indica faptul că e vorba de un program de tehnoredactare. Cînd se foloseşte un terminal ASCII cuvîntul se scrie cu T şi X mare şi E mic: TeX.

Cutiuţe

Conceptele fundamentale cu care TEX operează sunt *cutiuţele* şi *lipiciul* (boxes and glue). Motorul care tehnoredactează nu ştie mare lucru despre fiecare caracter; tot ceea ce îi trebuie este să ştie dimensiunile unei cutii care cuprinde caracterul, ca în figura 2. TEX nu ştie nimic despre care puncte din cutie vor fi pictate cu cerneală şi care nu; aceasta este treaba altor scule.

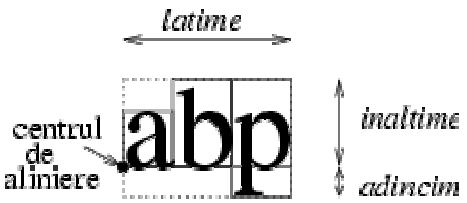

Figura: Trei litere şi cutiile din care fac parte, puse una lîngă alta. TEX operează tot timpul doar cu cutii, şi nu ştie nimic despre cum arată literele de fapt. Cu linie punctată este indicată cutia obţinută din alipirea celor 3 litere.

Textul este asamblat din astfel de cutiuţe pusă una lîngă alta. TEX aranjează cutiuţe una după alta pe orizontală formînd rînduri. Cînd un rînd se termină, TEXface din el o cutiuţă mai mare şi pune cutiuţele-rînduri una peste alta. Fiecare pagină este tot o cutie, asamblată din mai multe rînduri puse unul peste altul.

Utilizatorul poate crea cutii cu dimensiuni negative; în felul acesta se pot obţine efecte speciale, micşorînd distanţa dintre caractere sau scriind caractere suprapuse. Comenzi speciale pot muta cutiile (de exemplu dacă vrem să facem un exponent putem folosi semnul ^, care va ridica cutia care urmează şi va micşora font-ul).

În TEX mai există şi un alt tip special de cutie, care se cheamă ``clei". Ca şi cutiuţele, clei-ul are o anumită dimensiune, dar în plus are capacitatea de a se întinde şi comprima ca un elastic. Între cuvintele de pe o linie TEX pune automat lipici; asta permite spaţiilor dintre cuvinte să se întindă sau să se comprime pentru a ajusta frumos marginea din dreapta a liniei. Cu cît se întinde mai mult lipiciul, cu atît mai ``dureros" este rezultatul, pentru că aşezarea în pagină se depărtează de la cea ideală.

TEX foloseşte conceptul de lipici pentru a calcula despărţirea în rînduri ``optimă" a unui text. TEX foloseşte pentru asta o formulă matematică, care în funcţie de cît de întins este lipiciul din fiecare rînd calculează ``frumuseţea" unui paragraf. Cînd programul desparte în rînduri un paragraf, foloseşte un algoritm deştept inventat de Knuth (folosind programare dinamică) pentru a calcula tăieturile care oferă cel mai plăcut efect estetic, dintre toate cele posibile.

Folosind tipuri speciale de lipici, care se întinde la infinit fără ``dureri", se pot obţine efecte speciale: de exemplu punînd lipici la marginea din dreapta a fiecărui rînd se pot obţine pagini la care rîndurile nu sunt egale, ci sunt aliniate la stînga. Punînd lipici la ambele părţi se obţin rînduri centrate.

Asta e tot! Folosind cutiuţe şi lipici împreună cu algoritmi inteligenţi de aranjare în pagină, TEX obţine tipografie de calitate extraordinară.

Trebuie menţionate aici formulele matematice: modul în care TEX formatează matematica este încă neîntrecut, deşi sistemul este vechi de peste douăzeci de ani. Cînd are de tehnoredactat o formulă, TEX foloseşte un algoritm sofisticat care re-calculează dimensiunile şi plasamentele tuturor cutiuţelor în funcţie de tipurile de simboluri care apar: fiecare simbol este catalogat ca fiind într-una din 13 categorii: operator binar, operator relaţional, accent, un semn de punctuaţie, un tip de paranteză închisă sau deschisă, semn de fracţie, radical, etc. Fiecare din aceste categorii de simbol se comportă într-un anume fel în raport cu vecinii săi; unele simboluri cresc cît este necesar pentru a se adapta la cutiile cu care sunt vecine (de exemplu linia de fracţie sau parantezele).

Macro-uri

TEX este un limbaj foarte original. Principalul concept din acest limbaj este *macro*-instrucţiunea, numită familiar şi *macro*. Cei care programează în C sunt familiarizaţi cu macro-urile, care sunt acolo definite cu comanda `#define`. Un macro este practic o instrucţiune care arată cum un anumit text este substituit cu un altul.

Limbajul TEX este în întregime bazat pe macro-uri. Fiecare variabilă este de fapt un macro care este substituit în text cu valoarea sa. Limbajul are un set restrîns de operaţii primitive (cea mai primitivă operaţie fiind cea de a ``desena'' un caracter), şi toate celelalte operaţii sunt definite din acestea folosind macro-uri. Unele macro-uri se pot expanda la rîndul lor într-un text care conţine macro-uri, care trebuie expandate la rîndul lor, etc. În felul acesta, folosind macro-uri recursive, limbajul foloseşte un concept rudimentar de buclă.

Limbajele bazate pe macro-uri sunt extrem de puternice; de fapt sunt la fel de puternice ca orice alt limbaj; pentru demonstraţie Andrew Marc Greene de la universitatea MIT a scris în 1992, în 470 de linii de TEX (inclusiv comentarii) un interpretor complet de BASIC! (Un alt limbaj bazat pe macro-uri foarte folosit este m4.) Dar anumite complicaţii cu macro-urile fac de fapt folosirea lor extrem de dificilă. Cele mai multe probleme apar din *ordinea de evaluare*: cînd avem un macro aplicat altui macro, care dintre ele trebuie expandat întîi? În funcţie de ordine putem avea rezultate diferite.

În TEX problemele sunt exacerbate din cauză că *nu există cuvinte rezervate*, ca în toate limbajele cu care suntem obişnuiţi. Fiecare cuvînt sau chiar fiecare caracter poate fi transformat într-un macro, care se expandează în altceva! Expansiunea unui macro poate genera o definiţie care schimba semnificaţia macroului însuşi!

TEX: un limbaj imposibil

Cu tot respectul pentru dl. Knuth, care a jucat un rol foarte important în 1968 în construirea primului compilator de Algol, şi care a contribuit la crearea unor importante concepte în teoria compilatoarelor, limbajul TEX este o varză completă. Utilizatorul trebuie să consume un efort de imaginaţie substanţial pentru a face programele să funcţioneze, fiind foarte atent la ordinea de evaluare şi la efectele laterale ale fiecărui macro.

Pentru că unele macro-uri le redefinesc pe altele, avem o neplăcută problemă de dependenţă: efectul unui ``program" TEX depinde de mediul în care este invocat. Pentru că efectele prin variabile globale şi definiţii de macro-uri nu sunt întotdeauna controlate clar, programele TEX sunt foarte greu de scris şi de depanat. Mai mult, unele concepte care funcţionează foarte frumos în izolare, au mari probleme cînd sunt folosite laolaltă (de exemplu, în LATEX e foarte complicat să faci note de subsol în tabele).

Utilizatori sofisticaţi pot folosi TEX pentru a tehnoredacta texte relativ simple, dar doar o mînă de oameni poate exploata TEX cu uşurinţă. Eu cred că design-ul mizerabil al limbajului este unul dintre motivele fundamentale pentru care TEX nu este folosit pe scară mai largă. De fapt TEX ar fi trebuit de mult să fie scos din circulaţie de produse mai uşor de folosit; singura explicaţie pentru longevitatea sa este faptul că piaţa pentru un program de tehnoredactare de matematică este extrem de îngustă, iar dificultatea implementării unui program de tehnoredactare atît de sofisticat (cu toată publicarea algoritmilor şi a codului sursă) este enormă (Knuth este un programator colosal, cu o atenţie nemărginită pentru detaliu).

METAFONT

Simultan cu TEX, Knuth a construit un alt program foarte interesant, numit METAFONT, care poate genera familii întregi de font-uri dintr-o singură specificaţie. Astfel, programatorul descrie forma generală a fiecărui caracter; fişiere de configuraţie pot apoi genera toate familiile de font-uri dintr-o singură descriere: caracterele *oblice*, **grase**, `egale`, CAPITALIZATE, *italice* şi variatele dimensiuni şi mărimi, cu şi *fără serife* sunt toate generate apoi automat din aceeaşi descriere.

METAFONT este tot un limbaj de programare care permite desenarea a felurite primitive geometrice, declaraţii de variabile, definirea de macro-uri şi bucle, folosirea a diferite creioane cu vîrfuri eliptice pentru a desena curbe spline cubice[2], sisteme de ecuaţii lineare pentru definirea de variabile (rezolvate automat de interpretor).

Leslie Lamport

Înainte de a afla mai multe detalii despre LATEX, ne întîlnim din nou cu o figură colosală a informaticii, Leslie Lamport.

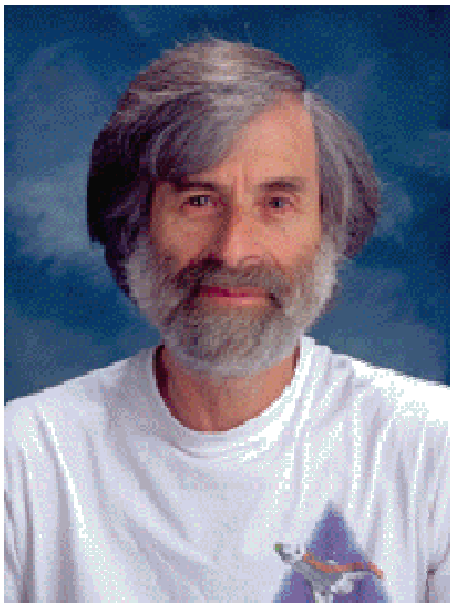

Figura 3: Leslie Lamport, creatorul LaTeX.

Deşi mai puţin faimos decît Donald Knuth, Leslie Lamport este unul dintre cercetătorii cei mai renumiţi din domeniul calculatoarelor. Domeniul său de specialitate este teoria sistemelor distribuite şi verificarea formală. După ce a obţinut un doctorat în matematică în 1972 de la universitatea Brandeis, a lucrat pentru o vreme la compania Stanford Research Institute (iniţial formată prin colaborare cu universitatea Stanford; SRI este unul din cele mai mari centre de cercetare din întreaga lume, care lucrează pe bază contracte). În timp ce lucra aici Lamport a creat sistemul LaTeX, care este pretextul acestui articol. După un scurt timp s-a mutat la centrul de cercetări Systems Research Center (SRC) al companiei Digital (care a fost între timp cumpărată de Compaq). Lamport lucrează în continuare la Compaq SRC, unde puteţi găsi şi pagina sa de web: research.compaq.com/SRC/personal/lamport/home.html.

Înainte de a discuta despre cea mai cunoscută creaţie a lui Lamport, LaTeX, se cuvine să menţionăm unele dintre contribuţiile lui fundamentale în teoria sistemelor concurente şi distribuite:

- Lamport a clarificat în 1978 noţiunea de timp şi sincronizare într-un sistem distribuit, folosind noţiunea de acum clasică de ceasuri Lamport.
- O problemă celebră în sisteme distribuite pentru care Lamport a oferit o soluţie în 1982 este cea a ``generalilor bizantini", care discută

despre posibilitatea ca un sistem distribuit să tolereze defecţiuni arbitrare în unele din componentele sale.

- În 1990 Lamport a introdus noţiunea de logică temporală a acţiunilor, Temporal Logic of Actions, TLA, despre care se spune că este în continuare cel mai bun formalism pentru a descrie şi raţiona despre sisteme distribuite. TLA şi extensii ale sale sunt folosite pentru a specifica, raţiona şi depana sisteme distribuite reale, cum ar fi de exemplu protocoale de coerenţă pentru multiprocesoare simetrice (vedeţi şi articolul meu din PC Report din noiembrie 1998 despre acest subiect), sau protocoale de securitate.