# LaTeX: un sistem de tehnoredactare pentru matematică

Mihai Budiu — mailto:mihaib+@cs.cmu.edu
http://www.cs.cmu.edu/~mihaib/

septembrie 2000

**Subiect:** sistemul de tehnoredactare computerizată LaTeX.

**Cunoştinţe necesare:** cunoştinţe elementare despre tehnoredactare

**Cuvinte cheie:** tehnoredactare, macro, TeX, LaTeX, Knuth, Lamport

## Cuprins

# 1 Introducere

Acest articol este dedicat unei scule software, care pentru mine este de neînlocuit în activitatea zilnică. Voi descrie în cele ce urmează un limbaj, numit LaTeX, şi un set de programe folosite pentru tehnoredactarea computerizată. Dintru început simt nevoia însă să avertizez cititorul asupra *dezavantajelor* acestui sistem, comparat cu alte sisteme similare:

**E un limbaj:** LaTeX este un limbaj, şi nu un program. LaTeX seamănă cu limbajul HTML sau cu RTF (Rich Text Format). Pentru a formata un text folosim comenzi inserate între elementele textului pentru a descrie în ce fel trebuie să "arate" acest text.

**Nu e vizual:** înainte de a folosi LaTeX trebuie să ne familiarizăm cu o mulţime de comenzi noi. Pentru HTML există editoare speciale, care ne permit să compunem paginile de web în mod vizual, fără să ştim ce fel de comenzi trebuie inserate pentru a obţine înfăţişarea dorită. În cazul LaTeX însă aceste scule sunt mult mai primitive, şi în general pentru a obţine rezultate satisfăcătoare şi pentru a beneficia de întreaga lui putere, limbajul trebuie cunoscut.

**E complicat:** din motive istorice, pe care le vom trece pe scurt în revistă, LaTeX este un limbaj destul de obscur şi greu de folosit. Chiar dacă unele elemente fundamentale sunt relativ uşor de învăţat, efortul necesar pentru stăpînirea majorităţii comenzilor este substanţial.

**Specializat:** pentru o mulţime de aplicaţii LaTeX este departe de a fi limbajul ideal. De exemplu, deşi există translatoare automate din LaTeX în HTML, LaTeX nu este limbajul cel mai potrivit pentru a descrie pagini de web.

**Scule neprietenoase:** În fine, deşi LaTeX este implementat pe o mulţime de sisteme, incluzînd Windows, modul preferat de folosire este "în linie de comandă". Textul de formatat, conţinînd comenzile LaTeX de aranjare în pagină, trebuie compus folosind un editor separat. După aceea este invocată o comandă pentru a procesa textul, după care un alt program trebuie folosit pentru a vizualiza rezultatul; adesea alte programe trebuie folosite pentru a genera indexuri, bibliografii, sau pentru a tipări produsul final. (Notă: în ultima vreme au început să apară scule mai prietenoase, care integrează aceste funcţionalităţi sub o singură interfaţă vizuală).

**Nu e "curat":** anumite construcţii din limbaj, care funcţionează bine izolat, atunci cînd sunt folosite împreună dau rezultate surprinzătoare, sau erori. Acestea sunt probleme care rezultă din design-ul iniţial al limbajului şi sunt extrem de neplăcute, chiar şi pentru utilizatori avansaţi.

**Mesaje criptice:** atunci cînd un document conţine erori în comenzile de formatare, mesajele de eroare sunt adesea extrem de greu de interpretat, şi sursa erorii este foarte greu de găsit. Acesta este un obstacol substanţial pentru novici.

**Acest articol e foarte sumar:** în secţiunea de bibliografii recomand cîteva cărţi despre LaTeX. Descrierea completă a limbajului şi a o sumedenie de pachete suplimentare este cuprinsă în nu mai puţin de trei cărţi! Ca atare textul de faţă este o introducere extrem de sumară în capabilităţile LaTeX.

Perfect. Dacă aţi ajuns pînă aici cu cititul înseamnă că sunteţi gata să înfruntaţi toate aceste obstacole. Iată şi unele dintre avantajele LaTeX:

**Formule:** LaTeX rămîne cea mai eficientă metodă pentru a tehnoredacta texte matematice. În primul rînd, rezultatele grafice sunt extrem de plăcute, şi nici un alt program existent nu se ridică la înălţimea calităţii sale. Editorul de ecuaţii din Microsoft Word este absolut jalnic în comparaţie cu LaTeX. Vedeţi şi figura 5 pentru nişte exemple succinte.

**Standard:** LaTeX a fost conceput cu grijă pentru a se comporta la fel pe orice sistem. LaTeX şi TeX sunt lingua-franca pentru schimbul de documente în multe domenii tehnice. De exemplu, majoritatea, dacă nu toate articolele, trimise spre publicare în conferinţe şi jurnale ale societăţii americane de matematică, de fizică, sau de astronomie *trebuie* să fie scrise folosind LaTeX. În plus, LaTeX este folosit foarte mult în comunitatea informaticienilor; toate articolele mele, inclusiv cel de faţă, sunt scrise în LaTeX[1].

**Nu e vizual:** faptul că LaTeX nu e un sistem vizual de editare (WYSIWYG: What You See is What You Get) este considerat de unii ca un avantaj: în felul acesta autorul se concentrează asupra formei *logice* a documentului, şi nu asupra înfăţişării. Mesajul textului este atunci mai bine transmis.

**E extensibil:** vom vedea imediat că LaTeX este un limbaj în care se pot defini noi construcţii. Din cauza asta o sumedenie de adiţii au fost făcute de-a lungul timpului, care permit comenzi de formatare deosebit de sofisticate şi puternice. Manualul LaTeX de bază descrie limbajul în anexa C în 70 de pagini, dar cartea despre module suplimentare are 500 de pagini!

**E configurabil:** Din cauză că este un limbaj bazat pe macro-uri (vom discuta despre asta un pic mai încolo), multe din comenzile de bază se pot redefini sau configura. Din punct de vedere al ingineriei programării acesta nu e un lucru prea înţelept, şi unele din erorile criptice de aici provin. Pe de altă parte, capacitatea de a reconfigura foarte multe elemente face schimbarea stilului unei prezentări extrem de simplă. Fişiere de “stil” conţin configuraţii complexe predefinite (asemănător cu Cascaded Style Sheets din HTML); schimbînd un cuvînt într-un fişier LaTeX putem schimba radical felul în care este înfăţişat textul.

[1]PC Report foloseşte însă ca program de tehnoredactare QuarkXpress, care nu suportă LaTeX; ca atare, în textul pe care îl aveţi în faţa ochilor în revistă, numai unele dintre figuri au fost generate cu LaTeX şi apoi transformate în Postscript. Articolele prezente în pagina mea de web sunt însă tehnoredactate în întregime cu LaTeX.

# 2 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.

## 2.1 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 1: 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.

## 2.2 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 $\pi$: la ora actuală s-a ajuns la versiunea 3.14159.

### 2.2.1 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 $\chi$ 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 $\chi$ 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.

### 2.2.2 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.

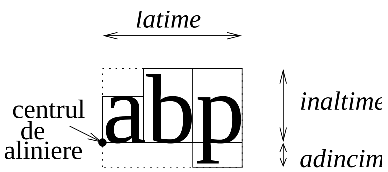


Figura 2: Trei litere şi cutiile din care fac parte, puse una lîngă alta. TeX operează tot timpul doar cu cutii, şi nu ştie nimic despre cum arată literele de fapt. Cu linie punctată este indicată cutia obţinută din alipirea celor 3 litere.

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).

### 2.2.3 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!

### 2.2.4 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).

## 2.3 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).

## 2.4 Leslie Lamport

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

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

[2]Splinele cubice sunt curbe compuse din bucăţi care se "unesc" în mod "continuu", şi pentru care fiecare fiecare bucată este descrisă de un polinom de gradul 3.

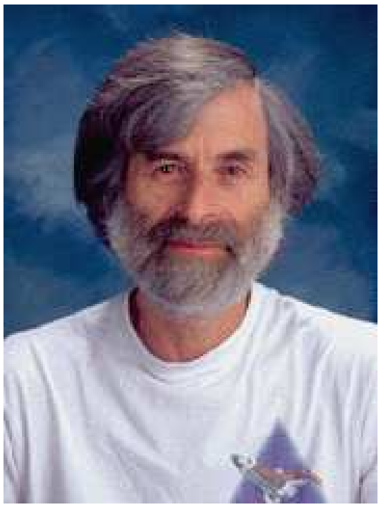

Figura 3: Leslie Lamport, creatorul LaTeX.

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.

### 2.5 LaTeX

În 1983 Lamport învăţase TeX de relativ puţină vreme, şi îşi construise un set de macro-uri foarte utile, care mutau operaţiile pe un nivel ceva mai înalt, în care descriai nu *cum* vrei să arate un document, ci care este *structura* lui logică.

Sistemul acesta de macro-uri a căpătat numele Lamport TeX, sau LaTeX. Editura Addison-Wesley dorea să tipărească un manual despre un sistem de tehnoredactare pentru matematicieni şi ingineri, aşa că un redactor l-a contactat pe Lamport. Ca urmare, Lamport a scris “cartea de LaTeX” (vedeţi şi secţiunea despre alte surse de informaţie), care a devenit un best-seller (s-au vîndut cîteva sute de mii de exemplare şi a fost retipărită de cel puţin şapte ori). Cartea a propulsat setul de macro-uri al lui Lamport în celebritate; ca şi TeX, acestea sunt disponibile în cod sursă pe Internet.

Ceea ce LaTeX încearcă să facă este să ascundă întreaga maşinărie încîlcită a TeX-ului sub o faţadă netedă: redefinirea comenzilor este în mod implicit interzisă (dar poate fi făcută, dacă utilizatorul chiar doreşte), un set de comenzi esenţiale este oferit, care organizează documentul într-o formă logică, şi sunt introduse *fişierele de stil*, care grupează declaraţii care influenţează arhitectura întregului document; schimbarea stilului se poate face separat de cea a conţinutului. LaTeX introduce de asemenea suport pentru referinţe încrucişate (genul “vezi figura X”, unde X este numerotat automat de sistem), etichete, cuprins, index, bibliografie, unele cu ajutorul unor programe suplimentare care procesează texte.

### 2.6 LaTeX2e şi LaTeX3

LaTeX face o treabă destul de bună; este un limbaj destul de rezonabil, în care se pot produce relativ repede rezultate atractive; matematica este tehnoredactată la întreaga putere a TeX-ului. Dar implementarea LaTeX era greu de menţinut şi dezvoltat, aşa că începînd din 1993 o serie de voluntari şi programatori din întreaga lume au pornit un nou proiect numit LaTeX3 (versiunea de LaTeX la acea dată era 2.09). LaTeX3 este un proiect în evoluţie, dar a fost deja lansată o versiune preliminară, numită LaTeX2e.

LaTeX a fost în întregime re-implementat într-o manieră asemănătoare cu programarea orientată pe obiecte, într-o serie de module separate. Unele module se pot înlocui unele pe altele, şi felul în care utilizatorii pot contribui cu noi module este standardizat. Asta a dat naştere la o cantitate remarcabilă de facilităţi noi aduse limbajului şi sistemul de folosire a font-urilor a fost extins şi perfecţionat, permiţînd LaTeX să folosească font-uri dezvoltate de terţi (şi nu numai Computer Modern).

LaTeX2e este compatibil cu 2.09, dar nu şi invers. Instrucţiunile pe care le prezentăm în continuarea acestui articol funcţionează în ambele versiuni. Limbajul de bază 2.09 este complet inclus în 2e.

## 3 O introducere rapidă în LaTeX

În cele ce urmează voi prezenta pe scurt unele din cele mai importante construcţii ale limbajului. Mi-e teamă că stăpînirea limbajului este imposibilă fără posesiunea cel puţin a cărţii de bază. Pe Internet există cîteva cărţi de introducere, şi cel puţin una a fost publicată şi în română. De asemenea, există o pleiadă de fişiere de “help” care descriu sumar comenzile. Dar nici una nu poate înlocui în acoperire manualul lui Lamport.

### 3.1 Un exemplu de document foarte simplu

Figura 4 arată un simplu document complet LaTeXşi rezultatul tehnoredactării sale. Orice document începe cu o declaraţie de stil (`\documentstyle`) şi este cuprins între `\begin{document}` şi `\end{document}`. În celelalte figuri voi omite aceste trei linii.

Înainte de a discuta unele dintre comenzile importante, va trebui să vă spun cum am obţinut fiecare din imaginile din acest text.

# Un exemplu LaTeX

Mihai Budiu

September 22, 2000

## Contents



## 1 Prima sectiune

Prima sectiune este foarte scurta, si imediat este impartita in subsectiuni astfel

### 1.1 O subsectiune

Observati cum subsectiunea este numerotata automat ca si...

### 1.2 Urmatoarea

...care este vida.

```
% acesta este un comentariu, care se termina odata cu linia
\documentstyle{article} % cream un document de tip 'articol'

\begin{document}  % aici incepe documentul

\title{Un exemplu \LaTeX}
\author{Mihai Budiu}

\maketitle % creaza titlul
\tableofcontents % va include cuprinsul

\section{Prima sectiune}

Prima sectiune este foarte scurta, si imediat este impartita in
subsectiuni astfel

\subsection{O subsectiune}

Observati cum subsectiunea este numerotata automat ca si\ldots

\subsection{Urmatoarea}

\ldots care este vida.

\end{document}
```

Figura 4: Un exemplu de document LaTeX împreună cu cuprinsul şi cîteva secţiuni.

## 3.2 Folosirea LaTeX

Pentru a scrie textul din partea de jos din figurii 4 am folosit un editor de texte obişnuit, în cazul meu emacs, care are un mod major care ajută editarea de documente TeX şi LaTeX (vedeţi şi articolul meu despre emacs din PC Report din mai 1997).

Am salvat fişierul sub numele `exemplu.tex`. Sufixul `.tex` este folosit atît pentru fişiere TeX cît şi pentru fişiere LaTeX.

Pentru a tehnoredacta fişierul am folosit apoi comanda: `latex exemplu.tex`. Această comandă porneşte interpretorul TeX, care la rîndul lui încarcă un set de macro-uri fundamentale, numite `plain.tex`, şi care apoi încarcă setul de macro-uri numit `latex.tex`. Apoi interpretorul citeşte fişierul `exemplu.tex` şi îl execută. Comenzile (în general indicate de cuvinte care încep cu semnul `\`, backslash), sunt macro-uri, care sunt expandate. Literele şi semnele speciale sunt tehoredactate, aşa cum am explicat mai sus în secţiunea despre cutiuţe.

(De fapt LaTeX trebuie executat de două, pentru că prima oară strînge informaţii despre unde se află fiecare secţiune pe care a la a doua rulare le foloseşte pentru a construi cuprinsul.)

Această execuţie produce mai multe fişiere, din care cel mai important este `exemplu.dvi`. Sufixul `.dvi` înseamnă DeVice Independent, adică "independent de dispozitivul de afişare". Fişierele dvi conţin o descriere a documentului tehnoredactat, în sensul că spun unde pe pagină trebuie plasată fiecare cutiuţă şi ce simbol se află în interiorul ei. Un fişier dvi nu conţine însă nici un fel de informaţii despre *cum* arată caracterele, sau eventualele imagini incluse.

Programul `dvips` transformă un fişierul `exemplu.dvi` într-un fişier Postscript `exemplu.ps` folosind comanda `dvips -o exemplu.ps exemplu.dvi`. Acest fişier extrage informaţii despre font-uri din locuri dinainte stabilite, adaugă imaginile incluse în text şi generează un program în Postscript. Acesta este un program de sine-stătător, care conţine toată informaţia despre cum arată rezultatul final; acest "program" poate fi fie vizualizat pe ecranul calculatorului folosind programul `ghostview`, fie tipărit la orice imprimantă care înţelege limbajul Postscript.

Toate aceste programe, `tex`, `latex`, `dvips`, `ghostview` sunt disponibile gratuit de pe Internet, laolaltă cu o sumedenie de fişiere de configurare, macro-uri şi font-uri, pentru o mulţime de sisteme de operare incluzînd Windows, Unix şi MacOS.

## 3.3 Cuvinte cheie şi caractere speciale

Programele LaTeX pot fi scrise folosind doar setul de caractere ASCII. Pentru a genera semne care nu sunt pe tastatură se folosesc macro-uri speciale care folosesc font-uri speciale. Acesta este un avantaj mare asupra programelor WYSIWYG: privind fişierul putem spune exact ce se va întîmpla; toate comenzile şi caracterele sunt descrise printr-un text clar (de exemplu în Microsoft Word trecerea de la text normal la text gras nu este vizibilă decît prin efectele sale).

Iată cum interpretează LaTeX diferitele construcţii:

**Toate literele şi cifrele** sunt caractere obişnuite, care apar în rezultatul final.

**Semne speciale:** Pentru a scrie macro-uri şi comenzi speciale, zece dintre caracterele ASCII au semnificaţii speciale; acestea sunt:

| Caracter | Nume | Semnificaţie |
|---|---|---|
| `#` | Diez | Număr de argumente într-un macro definit de utilizator |
| `$` | Dolar | Delimitează formule matematice |
| `%` | Procent | Începe un comentariu |
| `&` | Ampersand | Separă coloanele în tabele |
| `~` | Tilda | Spaţiu unde nu se poate despărţi rîndul |
| `_` | Subliniat | Pentru a scrie indici |
| `^` | Căciulă | Pentru a scrie exponenţi |
| `\` | Backslash | Urmează un nume de macro |
| `{}` | Acolade | Grupează mai multe caractere la un loc |

**Alte semne:** De asemenea, caracterele $| < >$ pot fi folosite numai în formule matematice (folosite în text dau naştere unor efecte ciudate).

**Un macro:** este indicat de un cuvînt format numai din litere în faţă căruia apare semnul `\`. Literele mari şi mici sunt distincte, deci `\latex` e diferit de `\LaTeX`.

**Semnul procent:** `%` este folosit pentru a introduce comentarii în text: tot ce urmează pînă la sfîrşitul liniei este ignorat de către interpretorul LaTeX.

**Acoladele** sunt extrem de importante în LaTeX; ele au o dublă funcţiune:

- Grupează la un loc mai multe simboluri, pentru a fi tratate ca o singură entitate. De exemplu, dacă scriem `$a^bc$` vom obţine ca rezultat $a^b$ urmat de $c$ ($a^bc$); dacă vrem să ridicăm $a$ la exponentul $bc$ trebuie să scriem `$a^{bc}$`, care generează $a^{bc}$.
- Creaza un nou *mediu* (environment); (aproape) toate definiţiile de macro-uri şi schimbările de font care sunt făcute între acolade dispar în momentul în care acolada închisă este întîlnită.

**Spaţiile** multiple sunt nesemnificative (includem aici caracterele blank, tab (`\t` în C) şi linie-nouă (`\n` sau `\r`); nu puteţi schimba deci alinierea textului tastînd mai multe spaţii. Excepţii la această regulă sunt însă următoarele:

- Sfîrşitul de linie termină un comentariu.
- Un spaţiu termină un nume de macro, dar nu e considerat ca spaţiu care separă două cuvinte. De aceea, dacă vreţi să puneţi un spaţiu după un macro, trebuie să folosiţi semnul "`\ `", (backslash urmat de spaţiu). Trebuie deci să scrieţi:

  ```
  \LaTeX\ este un program.
  ```

- Două sau mai multe semne consecutive de "linie nouă" indică începutul unui nou paragraf.

## 3.4 Formule matematice

După cum spuneam, TEX şi LATEX sunt excepţionale în tehnoredactarea formulelor matematice. O sumedenie de macro-uri sunt predefinite pentru toate soiurile de operatori matematici, care de care mai ciudaţi.

Formulele sunt delimitate de semne dolar `$`. Există două feluri de formule: care apar în text, care sunt delimitate de un singur semn `$`, şi formule care apar de sine-stătător, care sunt separate cu cîte două semne: `$$`.

$$A = \left| \begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \end{array} \right] \cdot B \tag{1}$$

```
\begin{equation}
A =
\left|
\begin{array}{cc}
1 & 0 \\
0 & 1 \\
\end{array}
\right]
\cdot B
\end{equation}
```

$$y(x) = \left\{ \begin{array}{lr} 0 & \mbox{ daca } x < 0 \\ \frac{x^2}{2} & \mbox{ altfel} \\ \end{array} \right. \tag{2}$$

$$z = \underbrace{x + \ldots + x}_{n} \tag{3}$$

```
\begin{eqnarray}
y(x) &=& \left\{
\begin{array}{lr}
0 & \mbox{ daca } x < 0 \\
\frac{x^2}{2} & \mbox{ altfel} \\
\end{array}
\right. \\
z &=& \underbrace{x + \ldots + x}_{n}
\end{eqnarray}
```

$$\sum_{i=1}^{n/2} x_i = \int_0^{\infty} {f(y)^{(\frac{1}{y})}}^2 dy = \sqrt{\frac{\alpha+\beta}{1 + \frac{\Omega}{\Gamma}}} \Longrightarrow {\mathcal F}(x) \approx 0.$$

```
$$\sum_{i=1}^{n/2} x_i =
\int_0^{\infty} {f(y)^{(\frac{1}{y})}}^2 dy =
\sqrt{\frac{\alpha+\beta}{1 + \frac{\Omega}{\Gamma}}} \Longrightarrow
{\mathcal F}(x) \approx 0.$$
```

Figura 5: Exemple de formule complexe în mod matematic şi programele LATEX care le generează.

Figura 5 arată unele simboluri folosite în formule matematice.

### 3.5 Enumerări şi liste

În LaTeX există două tipuri de macro-uri: macro-uri simple, care pot avea argumente, sau macro-uri de tip "bloc" (environment). Macro-urile de tip bloc apar întotdeauna în perechi begin-end; între `begin` şi `end` este activ macro-ul.

Trei macro-uri de tip bloc pot fi folosite pentru a obţine mai multe feluri de enumerări; acestea sunt `itemize`, `enumerate` şi `description`. `itemize` va pune biluţe (sau alte semne) în faţa fiecărui element, `enumerate` va pune cifre iar `description` va pune ceea ce indică utilizatorul. Vedeţi şi figura 6 pentru un exemplu.

### 3.6 Tabele şi matrici

Figura 7 arată unele dintre lucrurile care se pot face folosind tabele.

Mediul "tabular" este urmat de o listă de poziţii ale coloanelor:

| Literă | Engleză | Semnificaţie |
|---|---|---|
| l | left | aliniat la stînga |
| c | center | centrat |
| r | right | aliniat la dreapta |
| \| | bar | bară verticală |
| p{marime} | paragraph | coloană de lăţime fixă |

Coloanele sunt separate cu semnul ampersand `&`, şi liniile sunt terminate cu semnul `\\`. Texte pe mai multe coloane se pot face folosind macro-ul `\multicolumn`, care are trei argumente. Linii orizontale se pot face folosind `\hline`; linii orizontale parţiale cu `\cline`.

### 3.7 Font-uri

Rămînînd în interiorul aceleiaşi familii de font-uri se pot schimba font-urile cu diferite comenzi. Figura 8 arată cîteva exemple de comenzi şi fonturile obţinute.

### 3.8 Accente

TeX şi LaTeX oferă o serie bogată de accente, care se pot folosi pentru a scrie în majoritatea limbilor cu alfabete latine, ilustrate în figura 9.

### 3.9 Index-uri, referinţe încrucişate, bibliografii

LaTeX generează automat numere pentru pagini, capitole, secţiuni, paragrafe, ecuaţii, figuri, tabele, etc. Folosind macro-ul `\label` se generează automat nume pentru etichete. De exemplu, `\label{sec}` plasată în text va da variabilei `sec` numărul secţiunii curente. Plasînd `\label{fig}` în interiorul textului asociat unei figuri (vedeţi mai jos un exemplu), va da variabilei `fig` numărul figurii curente.

Folosind macro-ul `\ref` putem referi aceste variabile astfel: `vezi capitolul \ref{sec}`.

- Dar cred ca e mai interesant
- sa vedem cum se fac liste
    1. Sau chiar liste in liste
    2. unele numerotate
        (a) corect
        (b) chiar si atunci cind sunt imbricate
- altele cu bilute
- si altele cu descrieri:

  **itemize** face punctulete
  **enumerate** face numere
  **description** foloseste cuvintele indicate intre paranteze patrate

```
\begin{itemize}
\item Dar cred ca e mai interesant
\item sa vedem cum se fac liste

  \begin{enumerate}
  \item Sau chiar liste in liste
  \item unele numerotate

    \begin{enumerate}
    \item corect
    \item chiar si atunci cind sunt imbricate
    \end{enumerate}

  \end{enumerate}

\item altele cu bilute

\item si altele cu descrieri:

  \begin{description}
  \item[itemize] face punctulete
  \item[enumerate] face numere
  \item[description] foloseste cuvintele indicate intre paranteze patrate
  \end{description}

\end{itemize}
```

Figura 6: Exemple de liste şi enumerări în LaTeX şi codul care le-a generat.

### 3.10 Figuri

Figurile sunt tot dreptunghiuri în LaTeX, dar sunt speciale pentru că plasamentul lor în text nu este fixat de la început, ci este aranjat în funcţie de restul textului. Figurile se numesc de aceea "corpuri flotante", pentru că "plutesc" prin textul înconjurător pentru a ajunge mereu unde a indicat utilizatorul. Există două tipuri de figuri, care funcţionează asemănător, dar sunt numerotate

| Doua coloane | | si una |
|---|---|---|
| Stinga | Centru | Dreapta |
| prima coloana | a doua | a treia |
| fiecare | aliniata | diferit |
| inca | o linie | finala |

```
\begin{center}
\begin{tabular}{|l|cr|} \hline
\multicolumn{2}{|c|}{Doua coloane} & si una \\ \hline\hline
Stinga & Centru & Dreapta \\ \hline
prima coloana & a doua & a treia \\
fiecare & aliniata & diferit \\ \cline{2-3}
inca & o linie & finala \\ \hline
\end{tabular}
\end{center}
```

Figura 7: O tabelă şi codul LATEX care a generat-o.

independent: figuri şi tabele. Iată un exemplu de fragment cod LATEX care generează un tabel:

```
\begin{table}[ht]
\begin{tabular}{lll}
....

\end{tabular}
\caption{\label{culori}Culorile fundamentale.}
\end{table}
```

Acest tabel conţine chiar o tabelă (al cărei corp l-am omis), dar care asociază un text (capţiune) cu tabelul şi o etichetă, numită `culori`, care poate fi referită în text cu `\ref{culori}`.

Algoritmul de paginare va încerca să plaseze tabela după indicaţiile dintre parantezele pătrate (`[ht]` în exemplul de mai sus), după posibilităţi, în ordine în locul în care apare în text (h = here), la partea de sus a unei pagini (t = top) sau la partea de jos a unei pagini (b = bottom).

### 3.11 Macro-uri şi definiţii

După cum am văzut pînă acum în exemple, există mai multe feluri de macro-uri. Dar toate sunt construite din acelaşi obiect fundamental: un macro cu zero sau mai multe argumente. Utilizatorii pot defini şi ei macro-urile lor, şi sunt chiar încurajaţi să facă asta. Comanda `\newcommand` chiar asta face.

Pentru a defini un macro fără argumente folosim ceva de genul:

```
\newcommand{\pcr}{PC Report}
```

Iată şi un exemplu de macro cu două argumente, folosit pentru semnul "combinări de x luate cîte y":

Macro-uri care schimbă font-ul argumentului:

| Font | Abreviere | Engleză | Comanda exemplu | Rezultat |
|---|---|---|---|---|
| drept | up | upright | `\textup{Font drept}` | Font drept |
| cursiv | it | italic | `\textit{Font italic}` | *Font italic* |
| înclinat | sl | slanted | `\textsl{Font \^ \i nclinat}` | *Font înclinat* |
| majuscule | sc | small caps | `\textsc{Font cu majuscule}` | FONT CU MAJUSCULE |
| gras | bf | boldface | `\textbf{Font gras}` | **Font gras** |
| fără serif | sf | sans serif | `\textsf{F\u ar\u a serife}` | Fără serife |
| egal | tt | typewriter | `\texttt{Font egal}` | `Font egal` |

Macro-uri care schimbă fontul de la un punct încolo:

| Comanda | Engleză | Exemplu | Rezultat |
|---|---|---|---|
| `\rm` | roman | `{\rm Text drept (normal)}` | Text drept (normal) |
| `\bf` | bold | `{\bf Text gras}` | **Text gras** |
| `\sc` | small caps | `{\sc Text majuscule}` | TEXT MAJUSCULE |
| `\tt` | typewriter | `{\tt Text egal}` | `Text egal` |
| `\em` | emphasized | `{\em Text eviden\c tiat}` | *Text evidenţiat* |
| `\it` | italic | `{\it Text italic}` | *Text italic* |
| `\sl` | slanted | `{\sl Text \^ \i nclinat}` | *Text înclinat* |

Macro-uri care schimbă dimensiunea unui font:

| Comandă | Exemplu | Rezultat |
|---|---|---|
| `\tiny` | `{\tiny Minuscul}` | Minuscul |
| `\scriptsize` | `{\scriptsize Pitic}` | Pitic |
| `\footnotesize` | `{\footnotesize Foarte mic}` | Foarte mic |
| `\small` | `{\small Mic}` | Mic |
| `\normalsize` | `{\normalsize Normal}` | Normal |
| `\large` | `{\large M\u aricel}` | Măricel |
| `\Large` | `{\Large Mai mare}` | Mai mare |
| `\LARGE` | `{\LARGE Mare}` | Mare |
| `\huge` | `{\huge Uria\c s}` | Uriaş |
| `\Huge` | `{\Huge Gigantic}` | Gigantic |

Figura 8: Exemple de comenzi de schimbare a font-ului în LaTeX; unele comenzi se pot combina, de exemplu `\sl\tt`.

```
\newcommand{\comb}[2]{C^{#1}_{#2}}
```

După cum puteţi ghici, semnul `#1` înseamnă "primul argument". Pentru a folosi acest macro putem scrie ceva de genul:

```
\comb{5}{x}
```

LaTeX permite de asemenea definirea unor macro-uri bloc; asta se face cu ajutorul comenzii `\newenvironment`. Figura 10 arată exemple de macrodefiniţii.

Cînd expansiunea unui macro generează un alt macro, acesta este expandat la rîndul lui; acest proces se încheie cînd se generează caractere sau macro-uri fundamentale (de exemplu nume de simboluri speciale). Unele dintre macro-urile predefinite pot avea argumente opţionale, care sunt transmise între paranteze pătrate.

| | | | |
|---|---|---|---|
| ò | ó | ô | ö |
| õ | ō | ȯ | ŏ |
| ǒ | ő | o͡o | o̧ |
| ọ | o̲ | | |

```
\begin{tabular}{llll}
\` o & \' o & \^ o & \" o \\
\~ o & \= o & \. o & \u o \\
\v o & \H o & \t{oo} & \c o \\
\d o & \b o \\
\end{tabular}
```

Figura 9: Accente LATEX şi codul care a generat tabelul; pentru accente în mod matematic trebuie folosite alte comenzi.

Cît este $C^5_x$? Nu ştiu, dar îmi permit să citez din PC Report, dintr-un clasic: Cum spunea **Platon**: *Astazi nu ploua* $\odot$ Cîtă dreptate avea!

Dar nu numai atît: Cum spunea **Heidegger**: *Înseamnă că nu e soare!* $\odot$ Deci raspunsul e $C^{2y}_{z+5}$.

```
\newcommand{\pcr}{PC Report}
\newcommand{\comb}[2]{$C^{#1}_{#2}$}
\newenvironment{citat}[1]{Cum spunea {\bf #1}: \em}{$\odot$}

C\^ \i t este \comb{5}{x}?  Nu \c stiu, dar \^ \i mi permit s\u a
citez din \pcr, dintr-un clasic:
\begin{citat}{Platon}
Astazi nu ploua
\end{citat}
C\^ \i t\u a dreptate avea!

Dar nu numai at\^ \i t:
\begin{citat}{Heidegger}
\^ Inseamn\u a c\u a nu e soare!
\end{citat}
Deci raspunsul e \comb{2y}{z+5}.
```

Figura 10: Definirea şi folosirea unor noi macro-uri; rezultatul obţinut şi codul care îl generează.

### 3.12 Stiluri şi documente

Prima linie din document este `\documentstyle`; acest macro are un argument care indică tipul de document care urmează. Schimbînd argumentul puteţi schimba radical tipul de document. Tipurile standard de document sunt `article`, `report`, `book`, sau `letter`. Acest macro este înlocuit în versiunea LATEX2e de macro-ul `\documentclass`.

`\documentstyle` poate avea tot felul de argumente opţionale, care pot influenţa dramatic stilul. Font-ul standard e de 10 puncte, dar putem mări caracterele cu

```
\documentstyle[12pt]{article}
```

Dacă vrem să scriem pe două coloane, ajunge să facem:

```
\documentstyle[twocolumn]{article}
```

şi aşa mai departe, avem la dispoziţie o mulţime de opţiuni.

## 4 Pachete suplimentare

În varianta LaTeX2e a fost introdusă o nouă metodă (care nu funcţionează în LaTeX2.09) de a încărca pachete de macro-uri, folosind comanda `\usepackage`. Există efectiv mii de pachete de macro-uri, de la modificări şi extensii ale comenzilor standard pînă la pachete ultra-specializate pentru diferite domenii (chimie, fizică, matematică, algoritmi, etc.), marea majoritate fiind disponibile pe Internet, din arhivele CTAN (vedeţi mai jos secţiunea despre alte surse de informaţie).

## 5 Alte scule legate de LaTeX

O grămadă de alte scule pot opera cu fişiere LaTeX; iată aici unele dintre cele mai importante:

**latex2html** converteşte din LaTeX în HTML; face o treabă destul de bună, mai ales dacă documentele sunt scrise în LaTeX standard, fără cod TeX.

**makeindex** este un program care generează automat un index dintr-un document. Cuvintele care trebuie să apară în index sunt marcate de editor cu comanda `\index` în text, dar restul procesării este automat.

**bibtex** este un utilitar care extrage, crează şi formatează bibliografii în mod automat din documente LaTeX şi fişiere speciale care conţin descrieri bibliografice, cu extensia `.bib`.

**xdvi** este un program care permite vizualizarea fişierelor dvi direct pe un sistem de ferestre X Windows (standard sub Unix)

**ghostview** este un utilitar care poate afişa sau converti fişiere Postscript.

**amstex** este un pachet de macro-uri scris de American Mathematical Society, pe care autorii de manuscrise matematice trebuie să-l folosească cînd vor să publice în revistele editate de societate.

**pdftex** este un program care generează din TeX direct fişiere în limbajul `pdf`, Portable Document Format.

**AUC-TeX** este un mod major pentru emacs care permite editarea şi procesarea fişierelor TeX şi LaTeX.

**dvips** este un utilitar care converteşte din format `dvi` în format `Postscript`.

**LyX** este un editor vizual, în curs de dezvoltare, care generează cod LaTeX.

**MathType** pentru Microsoft Word este un editor WYSIWYG de ecuaţii care generează TeX.

**Convertoare** între LaTeX şi alte procesoare de text sunt listate la
`http://www.kfa-juelich.de/isr/1/texconv/texcnv.html`.

**texinfo** un limbaj în care proiectul GNU (vedeţi şi articolul meu din PC Report din iunie 1998) construieşte documentaţia programelor; din `texinfo` se pot genera automat fişiere TeX pentru documentaţie tipărită şi fişiere `info` pentru manuale interactive hipertext.

## 6 Încheiere

LaTeX este un program foarte complicat, dar cu care se pot obţine cu uşurinţă rezultate spectaculoase, mai ales în tehnoredactarea de formule matematice. Rămîne să decideţi dacă vă este sau nu util. Învăţarea este cîteodată anevoioasă, şi limbajul este plin de idiosincrazii. Dar, fie ca ne place, fie că nu, LaTeX va rămîne pentru viitorul apropiat singura alternativă decentă pentru tehnoredactarea computerizată de mare calitate a textelor matematice.

## 7 Alte surse de informaţie

Cele trei cărţi de LaTeX:

1. "*LaTeX: A document Preparation System, User's guide and Reference manual*", de Leslie Lamport; Addison-Wesley, ediţia a doua, 1994, ISBN 0-201-52983-1.
2. "*The LaTeX Companion*", de Michel Goossens, Frank Mittelbach, Alexander Samarin; Addison-Wesley, 1994, ISBN 0-201-54199-8.
3. "*The LaTeX Graphics Companion; Illustrating Documents with TeX and PostScript*", de Michel Goossens, Sebastian Rahtz, Frank Mittelbach; Addison-Wesley, 1994, ISBN 0-201-85469-4.

Cartea de TeX: "*The TeX book* de Donald E. Knuth; Addison-Wesley, 1986, ISBN 0-201-13447-0 / 0-201-13448-9.

Tot software-ul şi documentaţia free despre TeX se află la arhiva numită Comprehensive Tex Archive Network şi oglinzile sale: `http://www.ctan.org/`.

Site-ul de web al proiectului LaTeX: `http://www.latex-project.org/`

Legături despre METAFONT: `http://www-cgrl.cs.mcgill.ca/~luc/metafont.html`

Un articol despre formatarea logică şi vizuală scris de Leslie Lamport "Document Production: Visual or Logical?",
`http://www.research.digital.com/SRC/personal/lamport/pubs/document-production.ps`.

Un articol despre impactul LaTeXde Leslie Lamport "How (La)TeX changed the face of Mathematics",
`http://www.research.digital.com/SRC/personal/lamport/pubs/lamport-latex-interview.ps`.