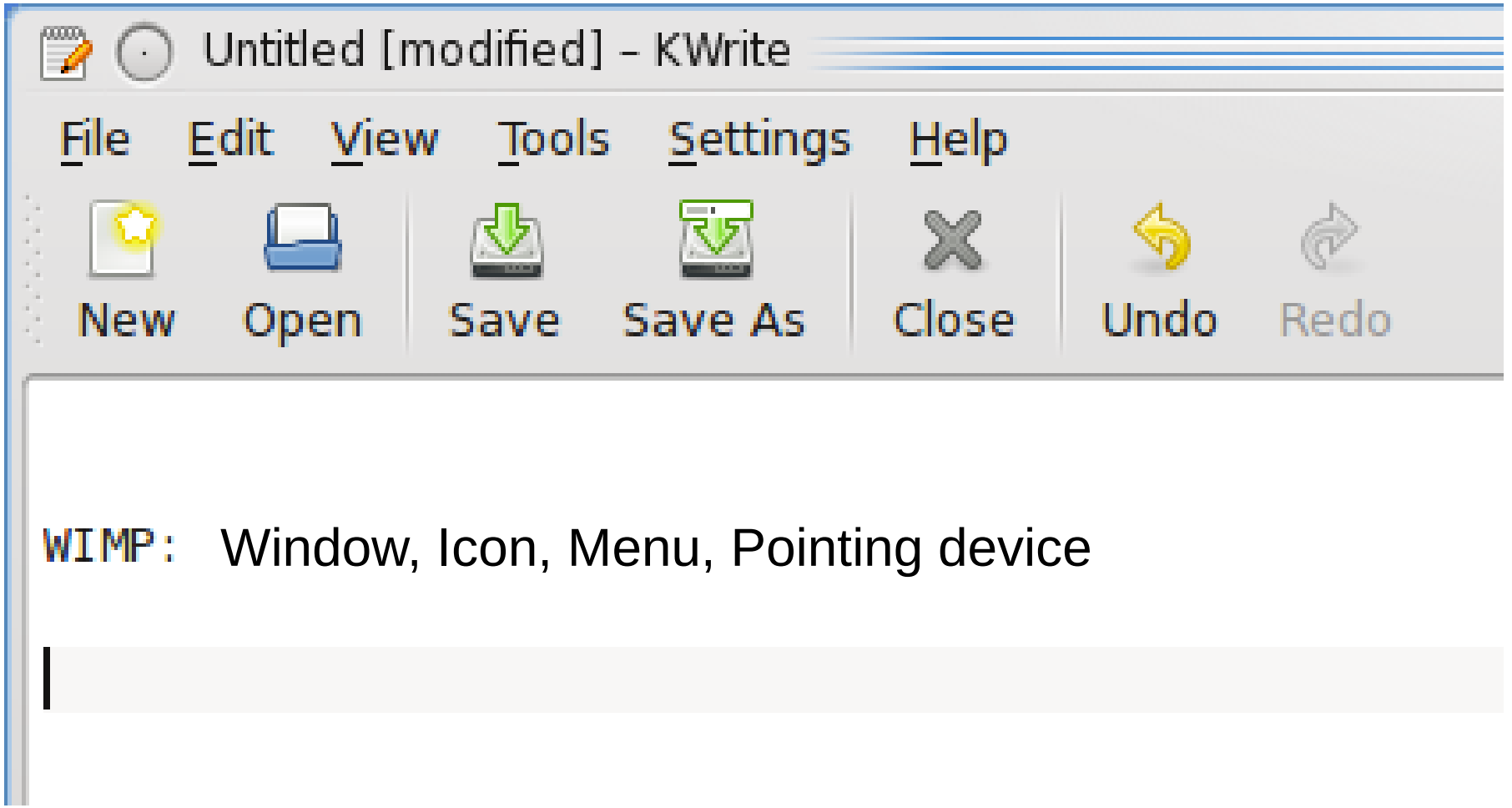
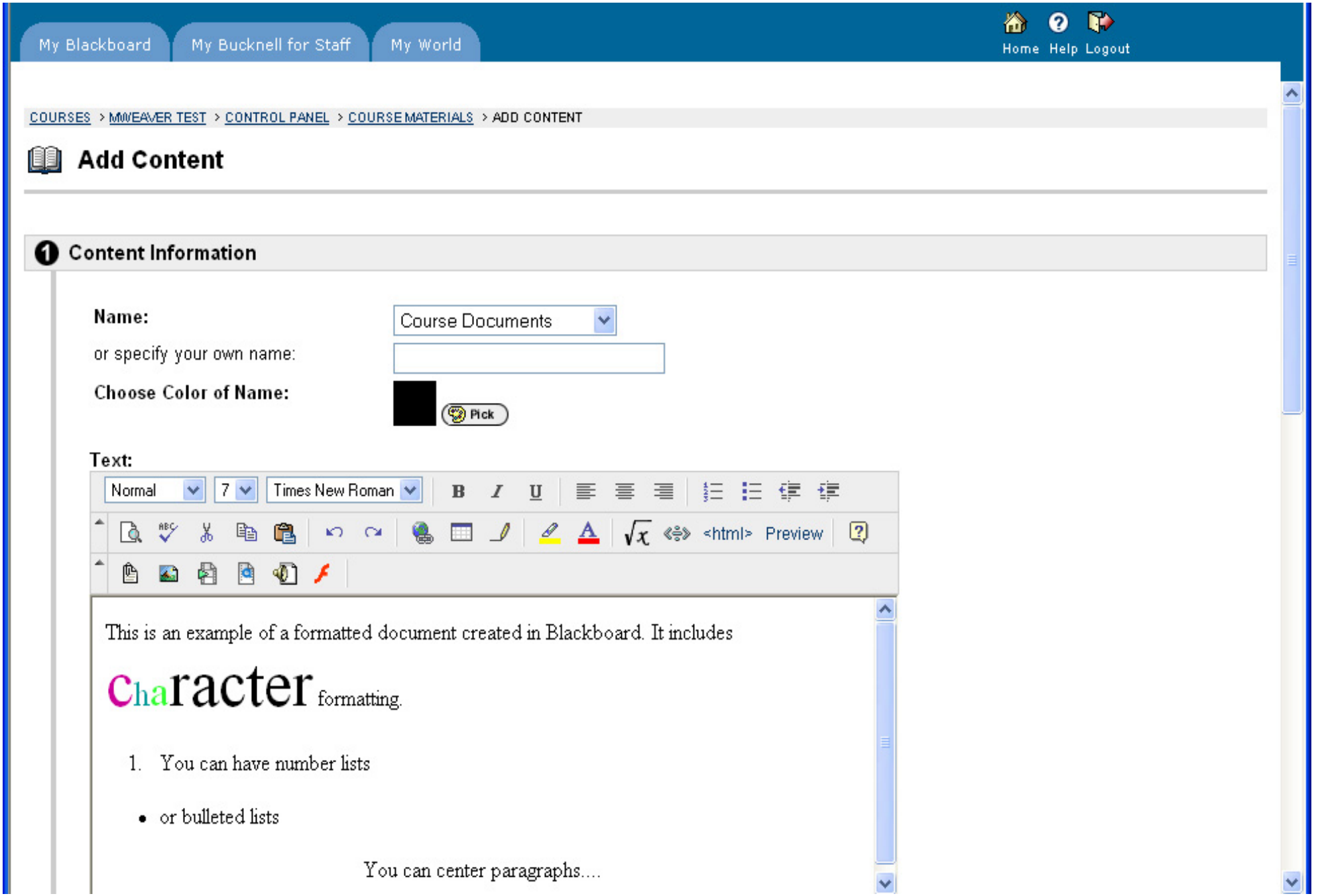
Text Editors … and “vi” in particular

Hans Vangheluwe

Text Editors ...

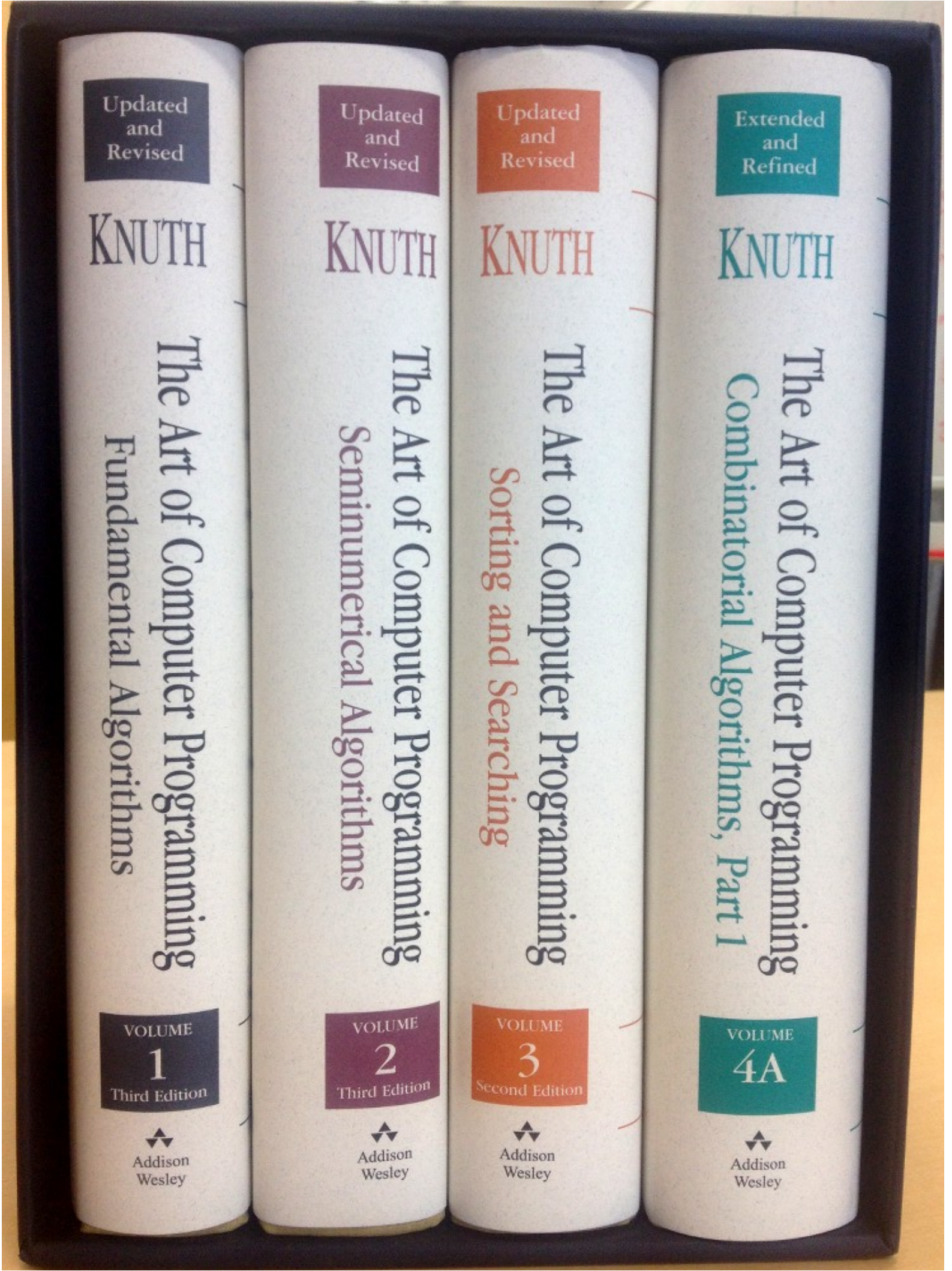


WYSIWYG (or not quite ...)



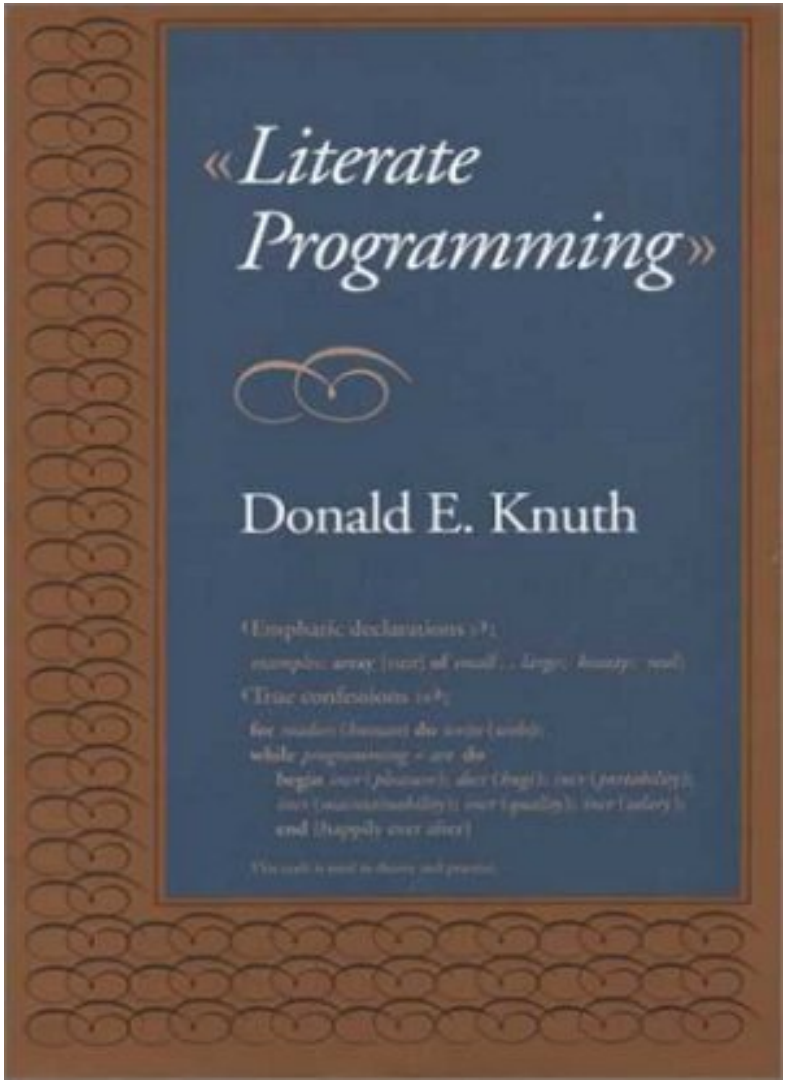
A.M. TURING AWARD

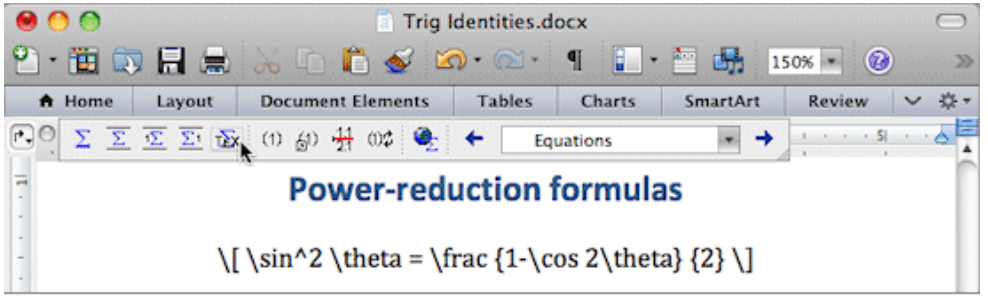
1974 TAOCP


Updated and Revised
KNUTH
The Art of Computer Programming
Fundamental Algorithms
VOLUME 1 Third Edition
Addison Wesley
Updated and Revised
KNUTH
The Art of Computer Programming
Seminumerical Algorithms
VOLUME 2 Third Edition
Addison Wesley
Updated and Revised
KNUTH
The Art of Computer Programming
Sorting and Searching
VOLUME 3 Second Edition
Addison Wesley
Extended and Refined
KNUTH
The Art of Computer Programming
Combinatorial Algorithms, Part 1
VOLUME 4A
Addison Wesley

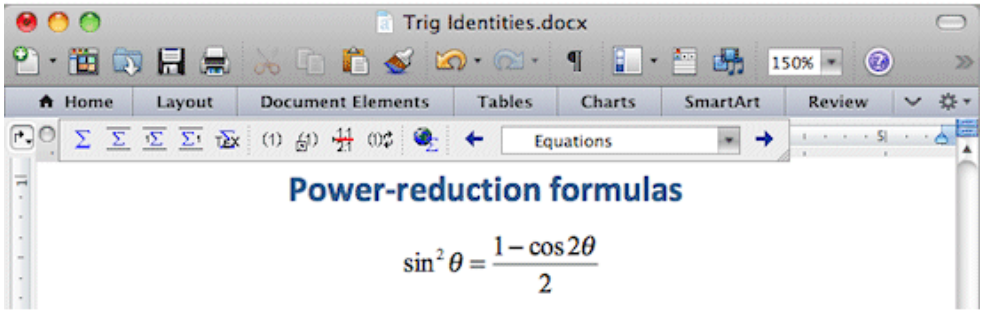
Explicitly state the **thoughts** behind the program:

- **Explain** the link between requirements (what) and design (how)
- **Explain** design decisions (algorithms and data structures)
- **Explain** code-level decisions (for example, for efficiency)

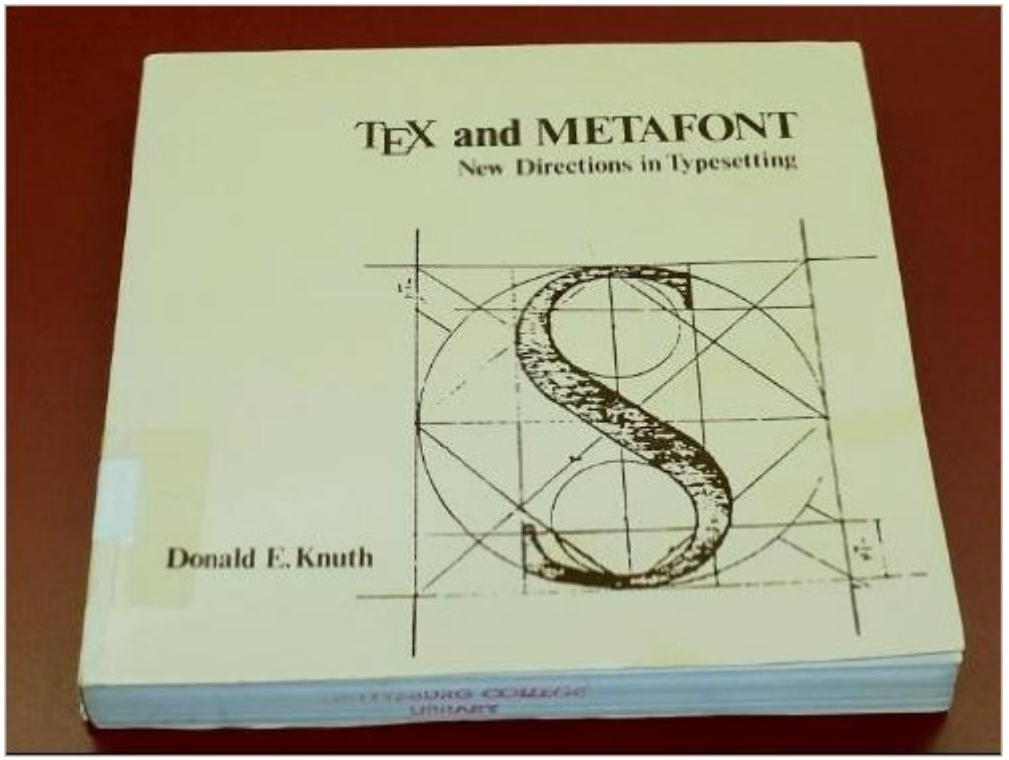



Trig Identities.docx
150%
Home
Layout
Document Elements
Tables
Charts
SmartArt
Review
Equations
Power-reduction formulas
\[\sin^2 \theta = \frac {1-\cos 2\theta} {2} \]

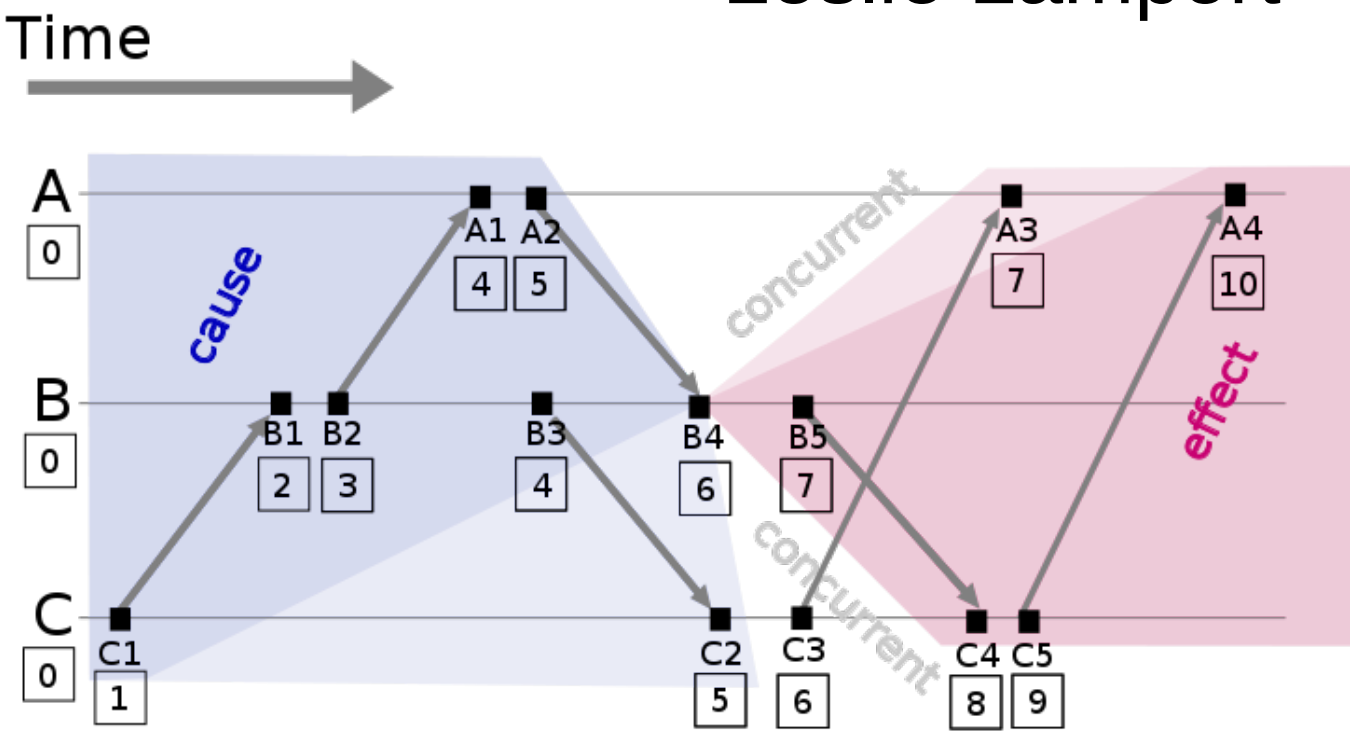
becomes...


Trig Identities.docx
150%
Home
Layout
Document Elements
Tables
Charts
SmartArt
Review
Equations
Power-reduction formulas
$\sin^2\theta = \frac{1-\cos 2\theta}{2}$

TEX


TEX and METAFONT
New Directions in Typesetting
Donald E. Knuth

Leslie Lamport



LaTeX

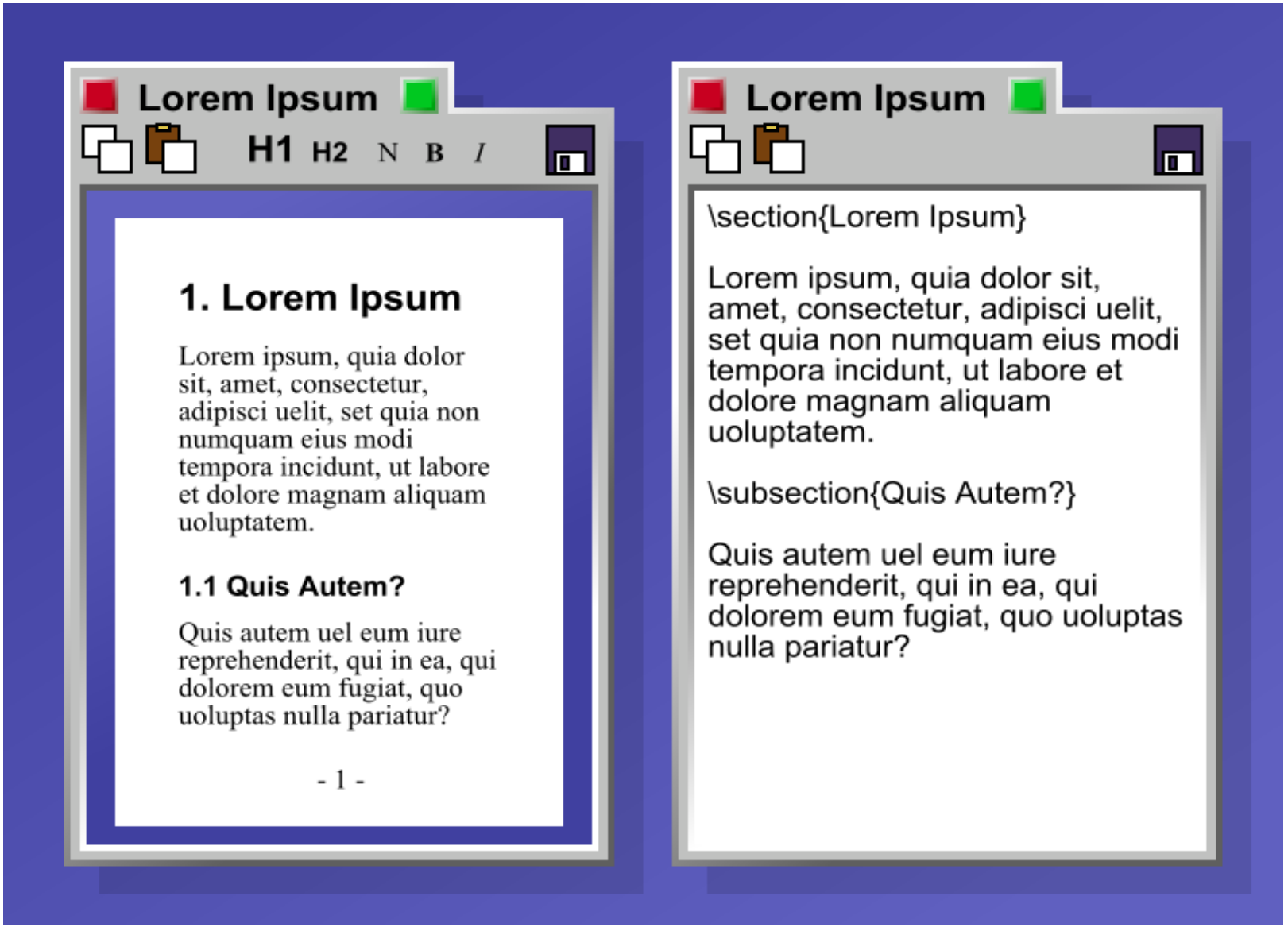
A.M. TURING AWARD

2013

Non-WYSIWYG ...

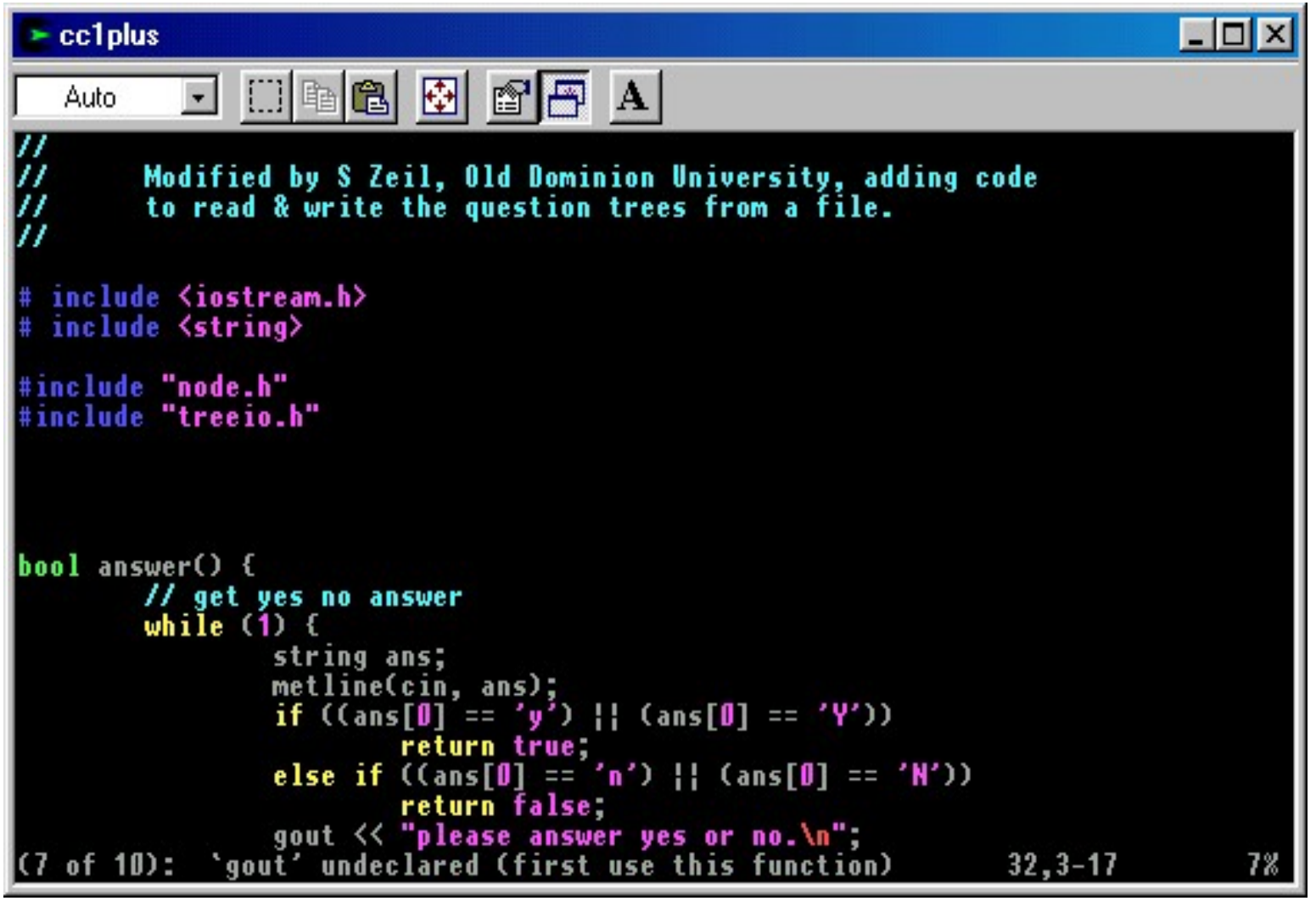
but optimal “typesetter”

LaTeX

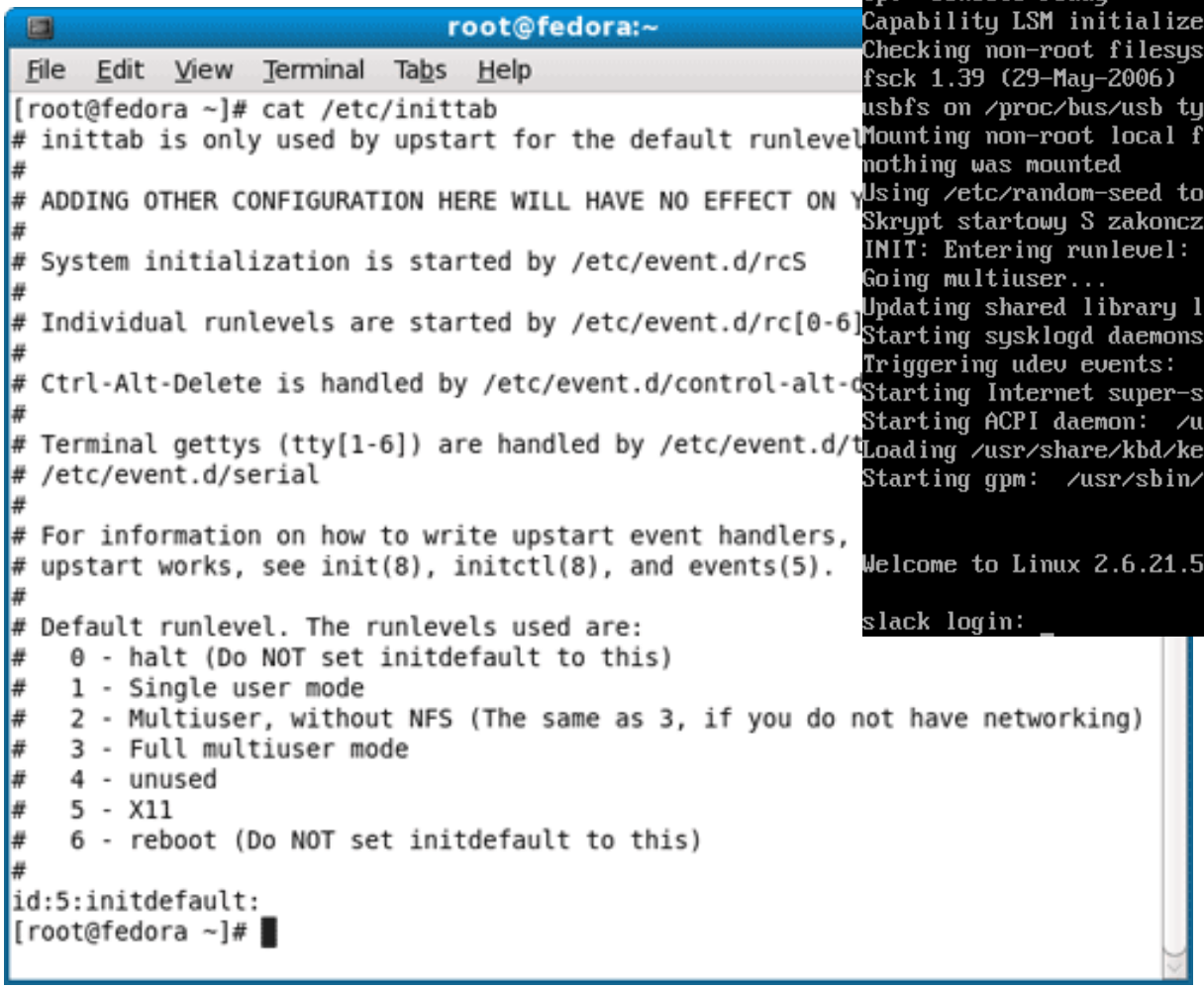


Who needs a mouse anyway ... “space cadet” keyboard

Who needs a mouse anyway ... the **vi** editor (and variants ... vim)



Who needs a mouse anyway ... the **vi** editor



use in minimal environments, ubiquitous

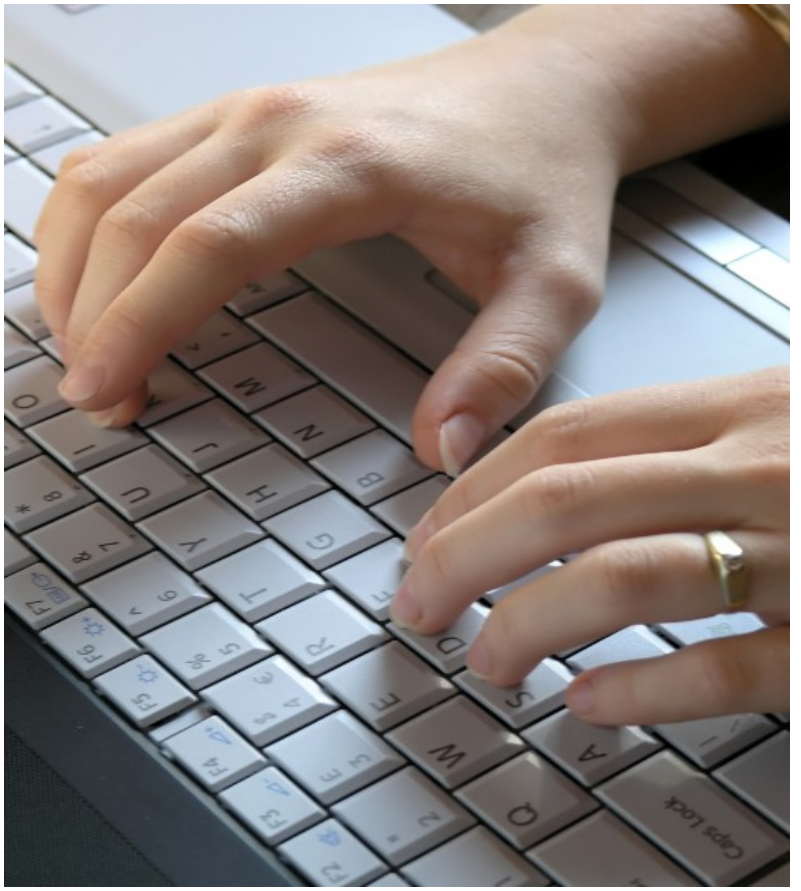
Who needs a mouse anyway ...
the **vi** editor (based on **ex** line editor)

Two **modes**:

- **C**ommand
- **I**nsert

Keyboard only!

... **fast** ...



vi help sheet

some usefull tips and command for the vi editor — *version of 29th Oct. 2006*

You have to use ESC (escape key) to get into the vi command mode. You will need to press RETURN Key for executing a command starting with the ponctuation character « : » or « / » or « ? ». Use CTRL+c for cancel a command. Use « . » for repeat your last command and « :!cmd » for execute a shell command (where cmd is the command to execute).

to get into insert mode

i	insert text before the cursor
a	append text after the cursor
I	insert text at the beginning of the current line
A	append text at the end of the current line
o	insert text in a new line below the cursor
O	insert text in a new line below the cursor

navigation commands

h or ←	move cursor left
l or →	move cursor right
k or ↑	move cursor up
j or ↓	move cursor down
G	goto the end of file
nG or **:n**	goto the line number « n »
0	move to the beginning of line
$	move to the end of line
ctrl+f	move one screen forward
ctrl+b	move one screen backward

file commands

:q	quit current open file
:q!	force to quit (without saving open file)
:w	save file
:w file	save file as « name »
:wq!	overwrite file then quit
:x,yw file	write from line « x » to line « y » into « file »
:w >> file	append buffer to « file »
:e file	edit another file
:e! file	edit another file without saving the current open file
:r file	insert file content at the current cursor position
:n	edit next file in vi arguments file list
ctrl+G	get file status

You can launch vi with some arguments. One usefull is « -c » for execute vi commands in a file directly from the command line.

ex.:
vi -c « %s/false/true/g\|wq » file.txt

search commands

/string	search forward for « string »
?string	search backward for « string »
n	repeat last search
:,$s/str1/str2/gc	search and replace « str1 » by « str2 » from the current line (a line number can be specified before the comma) to the end of file. Ask for confirmation before replace.
:%s/str1/str2/g	replace all « str1 » by « str2 » in all the file without confirmation.
:%s/str1/str2/	replace « str1 » by « str2 » for first occurrence of each line of file.

edit commands

r	replace a character at the cursor position
u	undo last change
mA	set mark « A » (can be any letter, case sensitive) at the beginning of current line
y'A	yank from current line to the mark « A »
d'A	delete from current line to the mark « A »
P (caps P)	put the buffer before the cursor
p (small p)	put the buffer after the cursor
x	delete character at cursor position
dw	delete first word after cursor position
d$ or **D**	delete from cursor position to the end of line
dd	delete curent line
J (caps J)	join curent line with the following line

Look at :

:map for mapping a key in command mode to a group of commands (ex. **:map de :,$d^M** will delete all file when using **de** command)
:set for define or show your editor current options
:ab for define a text abreviation in insert mode (ex. **:ab VIM Vi Improved** will auto complete **VIM** in insert mode)