

## A SIMPLE FOOBAR SAMPLE L<sup>A</sup>T<sub>E</sub>X FILE

STUPID STUFF I WISH SOMEONE HAD TOLD ME FOUR YEARS AGO

*(Read the .tex file along with this or it won't make much sense)*

The first thing to realize about L<sup>A</sup>T<sub>E</sub>X is that it is not “WYSIWYG”. In other words, it isn’t a word processor; what you type into your .tex file is not what you’ll see in your .dvi file. For example, L<sup>A</sup>T<sub>E</sub>X will completely ignore extra spaces within a line of your .tex file. Pressing return in the middle of a line will not register in your .dvi file. However, a double carriage-return is read as a paragraph break.

Like this. But any carriage-returns after the first two will be completely ignored; in other words, you

can’t  
add  
more  
space  
between

lines, no matter how many times you press return in your .tex file.

In order to add vertical space you have to use “vspace”; for example, you could add an inch of space by typing `\vspace{1in}`, like this:

To get three lines of space you would type `\vspace{3pc}` (“pc” stands for “pica”, a font-relative size), like this:

Notice that L<sup>A</sup>T<sub>E</sub>X commands are always preceeded by a backslash. Some commands, like `\vspace`, take arguments (here, a length) in curly brackets.

The second important thing to notice about L<sup>A</sup>T<sub>E</sub>X is that you type in various “environments”...so far we’ve just been typing regular text (except for a few inescapable usages of `\verb` and the centered, smallcaps, large title). There are basically two ways that you can enter and/or exit an environment;

this is the first way...

this is the second way.

Actually there is one more way, used above; for example, THIS WAY. The way that you get in and out of environment varies depending on which kind of

environment you want; for example, you use `\underline` “outside”, but `\it` “inside”; notice `\this` versus `this`.

The real power of L<sup>A</sup>T<sub>E</sub>X (for us) is in the math environment. You push and pop out of the math environment by typing `$`. For example,  $2x^3 - 1 = 5$  is typed between dollar signs as `$2x^3 - 1 = 5$`. Perhaps a more interesting example is  $\lim_{N \rightarrow \infty} \sum_{k=1}^N f(t_k) \Delta t$ .

You can get a fancier, display-style math environment by enclosing your equation with double dollar signs. This will center your equation, and display sub- and super-scripts in a more readable fashion:

$$\lim_{N \rightarrow \infty} \sum_{k=1}^N f(t_k) \Delta t.$$

If you don’t want your equation to be centered, but you want the nice indicies and all that, you can use `\displaystyle` and get your formula “in-line”; using

our example this is  $\lim_{N \rightarrow \infty} \sum_{k=1}^N f(t_k) \Delta t$ . Of course this can screw up your line spacing a little bit.

There are many more things to know about L<sup>A</sup>T<sub>E</sub>X and we can’t possibly talk about them all here. You can use L<sup>A</sup>T<sub>E</sub>X to get tables, commutative diagrams, figures, aligned equations, cross-references, labels, matrices, and all manner of strange things into your documents. You can control margins, spacing, alignment, *et cetera* to higher degrees of accuracy than the human eye can percieve. You can waste entire days typesetting documents to be “just so”. In short, L<sup>A</sup>T<sub>E</sub>X rules.

The best way to learn L<sup>A</sup>T<sub>E</sub>X is by example. Get yourself a bunch of .tex files, see what kind of output they produce, and figure out how to modify them to do what you want. There are many template and sample files on the department L<sup>A</sup>T<sub>E</sub>X page and in real life in the big binder that should be in the computer lab somewhere. Good luck!

## 1 Introdução

Tecnologias de banco de dados são o componente principal de muitos sistemas de computação. Elas permitem que a informação seja retida e compartilhada eletronicamente e a quantidade de informação armazenada nesses sistemas continua a crescer à uma taxa exponencial.

Porém, danos e uso indevido de dados afetam não apenas um único usuário ou aplicação, mas pode ter consequências desastrosas para toda a organização. A proliferação rápida de aplicações baseadas na web e sistemas de informação aumentou ainda mais o risco de exposição de bancos de dados e, assim, a proteção de dados é hoje mais crucial do que nunca.

Brechas de segurança são tipicamente categorizadas como observação não autorizada de dados, modificação incorreta de dados, e indisponibilidade de dados. Observação não autorizada de dados resulta em revelação de informação

para usuários sem dinheiro de ganhar acesso a tal informação. Todas as organizações, variando de organizações comerciais para organizações sociais, numa variedade de domínios podem sofrer grandes perdas tanto do ponto de vista financeiro como humano como consequências da observação não autorizada de dados.

Modificação incorreta de dados, tanto intencional como não intencional, resulta em um estado incorreto no banco de dados. Qualquer uso de dados incorretos pode resultar em grandes perdas para a organização. Quando a informação está indisponível, informação crucial para o funcionamento adequado da organização não está prontamente disponível quando preciso.

Assim, uma solução completa para a segurança de dados deve atender os seguintes três requisitos:

1) Sigilo ou confidencialidade refere-se a proteção dos dados contra revelação não autorizada, 2) integridade refere-se a prevenção da modificação imprópria de dados não autorizada, e 3) disponibilidade refere-se a prevenção e recuperação do hardware e software de erros e de negações de acesso de dados maliciosas tornando o banco de dados indisponível.

Esses três requisitos surgem em praticamente todos os ambientes de aplicação.

## **2 Um pouco de história**

## **3 Controle de acesso**