

Why I hate OpenOffice

Or, but is it UNIX?

Greg “groggy” Lehey
LEMIS (SA) Pty Ltd
grog@lemis.com
Sydney, 5 September 2003

UNIX: What a success!

UNIX has been a great success:

- Close integration of few concepts.
- A powerful environment to build solutions.
- Few, simple file formats.
- Like a “super-language” for talking to the computer.
- No “desktop”.
- Microsoft has a “desktop”.
- We must have one too!

The secret for UNIX's success

- Good framework.
- Programmers can build on the work of others.
- Write programs that do one thing and do it well.
- Write programs that work together.
- Write programs that handle text streams, because that is the universal interface.
- Use few, simple file formats for better interchange of data.

Other UNIX features

- UNIX is a multi-user system.
- UNIX has a standardized file system layout.
- Each user has a *home directory*.
- UNIX is a networking system.

But is it user friendly?

- None of these concepts relate to user friendliness.
- UNIX was written by programmers for programmers.
- Required learning a foreign language.
- It scared off non-programmers.
- A new generation of “user friendly” software arose.

How to communicate with humans

- Communication requires language.
- Four levels of language:
 - Gestures, grimaces, grunting.
 - Speech: requires additional hardware (larynx) and experience in using it.
 - Originally simple (verb, noun): “Find food”.
 - Became more refined: “Find good-tasting food”.
 - Abstract concepts: “If you see an elephant, hide”.

How to communicate with computers

- Communication requires language.
- Four levels of language:
- Gestures, grimaces, grunting (“point and click”).
- Language: requires additional hardware (keyboard) and experience in using it.
- Originally simple (verb and noun): `list files`.
- Became more refined: `list files in detail`.
- Abstract concepts:

```
if (ispresent (elephant))  
    hide ();
```

Talking to computers

- Initially non-interactive access (punched cards, paper tape).
- Round the birth of UNIX, teletypes became available for interactive access. UNIX made great use of it.
- The output was on paper, so interaction tended to be linear.
- Soon character-based CRT-based terminals (*glass ttys*) became available.
- Character based serial devices with top speed of 2 kB/s.

Talking to computers (2)

- Some “full-screen” software used this interface to create menus.
- User navigated the menu with the cursor keys.
- In the 1980s, bit-mapped graphics became available.
- These systems usually had a mouse.
- Initially available systems were proprietary.
- In the late 1980s the X Window System became generally available.

The early days

- X arrived too late to influence the “desktop” market.
- Microsoft’s first operating system was XENIX, but it left no mark on the desktop.
- Microsoft’s early software was like a primitive UNIX.
- UNIX had a powerful command language.

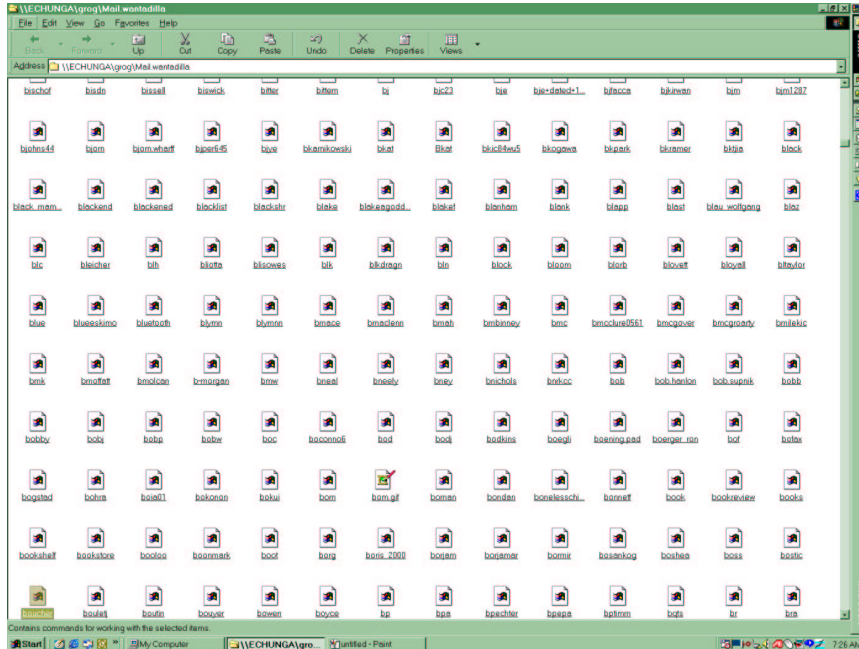
The new user syndrome

- Unsophisticated users had trouble with typing.
- Unsophisticated users had trouble with the command language.
- Unsophisticated users had trouble.
- Let the computer take charge!

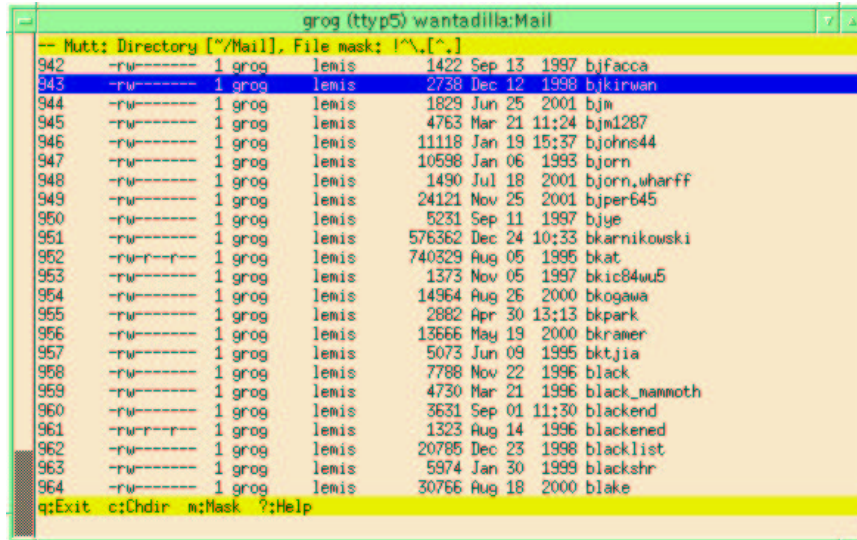
The advent of the menu

- Text-based menus had been around for a long time.
- Graphics-based menus came to be the standard interface to the system.
- Also used to represent objects such as directories (“*folders*”).
- They even simulated a keyboard.

Microsoft Directory Display



Mutt Mail Directory Display



The screenshot shows a terminal window titled "grog (ttp5) wantadilla:Mail". The window displays a list of email messages in a directory view. The messages are sorted by date and time, with the most recent at the top. Each line represents an email entry with the following fields: message number, permissions, flags, sender, recipient, date, and subject. The message number 943 is highlighted in blue. At the bottom of the window, there is a help prompt: "q:Exit c:Chdir m:Mask ? :Help".

```
-- Mutt: Directory [~/Mail], File mask: l^\.[^,]
942  -rw----- 1 grog lewis      1422 Sep 13 1997 bjfacca
943  -rw----- 1 grog lewis      2738 Dec 12 1998 bjkirwan
944  -rw----- 1 grog lewis      1829 Jun 25 2001 bjw
945  -rw----- 1 grog lewis      4763 Mar 21 11:24 bjmd287
946  -rw----- 1 grog lewis     11118 Jan 19 15:37 bjohns44
947  -rw----- 1 grog lewis     10598 Jan 06 1993 bjorn
948  -rw----- 1 grog lewis      1490 Jul 18 2001 bjorn.wharff
949  -rw----- 1 grog lewis     24121 Nov 25 2001 bjper645
950  -rw----- 1 grog lewis      5231 Sep 11 1997 bjye
951  -rw----- 1 grog lewis    576362 Dec 24 10:33 bkarnikowski
952  -rw-r--r-- 1 grog lewis     740329 Aug 05 1995 bkat
953  -rw----- 1 grog lewis      1373 Nov 05 1997 bkic84uu5
954  -rw----- 1 grog lewis     14964 Aug 26 2000 bkogawa
955  -rw----- 1 grog lewis      2882 Apr 30 13:13 bkpark
956  -rw----- 1 grog lewis     13666 May 19 2000 bkramer
957  -rw----- 1 grog lewis      5073 Jun 09 1995 bktjia
958  -rw----- 1 grog lewis      7788 Nov 22 1996 black
959  -rw----- 1 grog lewis      4730 Mar 21 1996 black_mammoth
960  -rw----- 1 grog lewis      3631 Sep 01 11:30 blackend
961  -rw-r--r-- 1 grog lewis      1323 Aug 14 1996 blackened
962  -rw----- 1 grog lewis     20785 Dec 23 1998 blacklist
963  -rw----- 1 grog lewis      5974 Jan 30 1999 blackshr
964  -rw----- 1 grog lewis     30766 Aug 18 2000 blake
q:Exit c:Chdir m:Mask ? :Help
```

Needles and haystacks

- Not a good use of menus.
- Microsoft also offers text listing.
- How to select? Much mouse pushing.
- Scrolling very difficult in large folder.
- File name completion better.
- Microsoft does file name completion too.

Menu trees

- Running programs is another matter.
- Organize programs by category.
- Make category top level of menu.

Executables on my laptop

<i>Directory</i>	Number of executables
<i>/bin</i>	40
<i>/usr/bin</i>	406
<i>/sbin</i>	104
<i>/usr/sbin</i>	229
<i>/usr/local/bin</i>	630
<i>/usr/local/sbin</i>	35
<i>/usr/X11R6/bin</i>	353
<i>Total</i>	1797

Menu trees, again

- How to organize a tree of 1,800 executables?
- How to add new programs?
- How to do it without a menu?
- An unsolved problem.

Using a menu tree

Assuming you have a menu tree and you know how it is organized, start programs like this:

1. Find the mouse.
2. Select the “start” icon, often at bottom left of the screen.
3. Select a submenu from the menu.
4. Possibly repeat, selecting a further submenu from the submenu.
5. Select the final program.
6. If the program requires arguments, supply them by whatever means the program provides.

Using the command line

Assuming you know the name of the program, start it like this:

- Type in the name of the program.
- If the program requires arguments, type them after the program name.
- Press **Enter**.

Unknown program names

- Menu can help find the program.
- In command line, need to use things like *man* to find what you're looking for.
- Menus win here.
- “Learn in a day, pay for the rest of your life”.

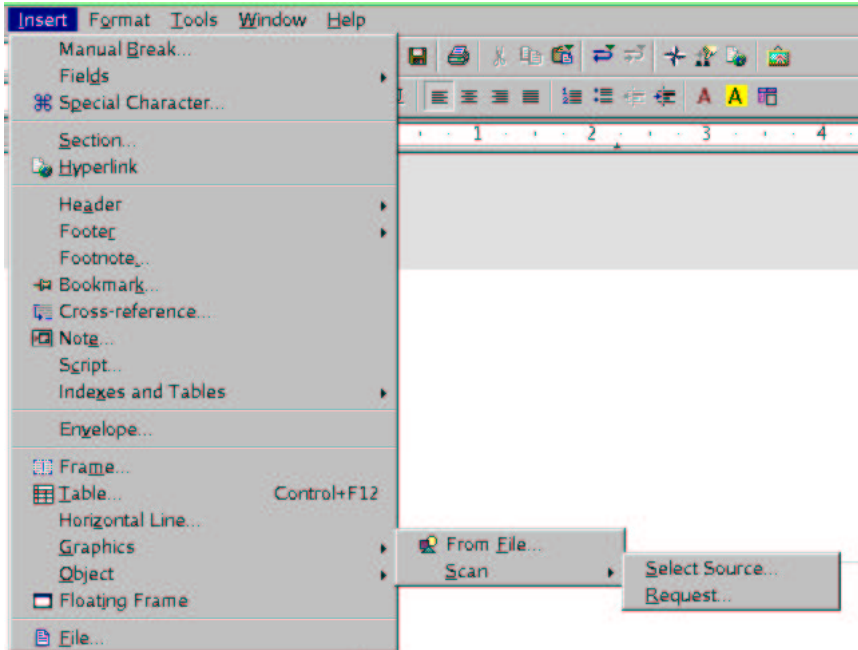
And the options?

From the previous slide, *Using a Menu Tree*:

- If the program requires arguments, supply them by whatever means the program provides.

One way is to have another menu tree.

OpenOffice menus



Making it easier

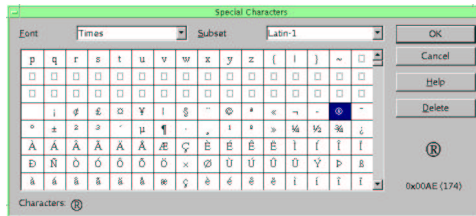
- This is the worst case in *OpenOffice*.
- Other programs can have much more convoluted trees.
- “Short cuts” are available.
- Instead of the tree, enter **Alt-IGS**.
- On command line, enter **scan**.
- Which character sequence is easier to remember?

Learning curve

- Example: insert special character ®
- Not on keyboard.
- How to find it?

®: first time with OpenOffice

- Select Insert with mouse. A submenu pops up.
- Select Special Character with mouse. A very long submenu containing all known characters in all fonts, pops up, armed with a monster scroll bar.
- Search for the character and select it. The character appears on the right of the window:



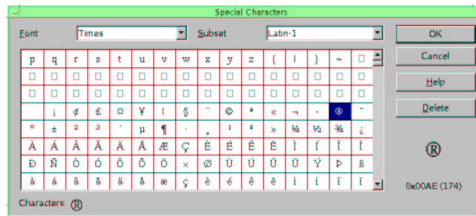
- Select OK and the character is transferred to the document.

Ⓜ: first time with T_EX

- Get out the T_EXbook. Read.
- Give up. Get out “T_EX for the impatient”. Read.
- Give up. Look at the contents of the T_EX macros.
- Watch eyes go funny.
- Give up. Call guru. Discover it’s a sequence called `\textregistered`.
- Type `\textregistered` into document.

®: next time with OpenOffice

- Select Insert with mouse. A submenu pops up.
- Select Special Character with mouse. A very long submenu containing all known characters in all fonts, pops up, armed with a monster scroll bar:
- Search for the character and select it. The character appears on the right of the window:



- Select OK and the character is transferred to the document.

®: next time with T_EX

- Type `\textregistered` into document.

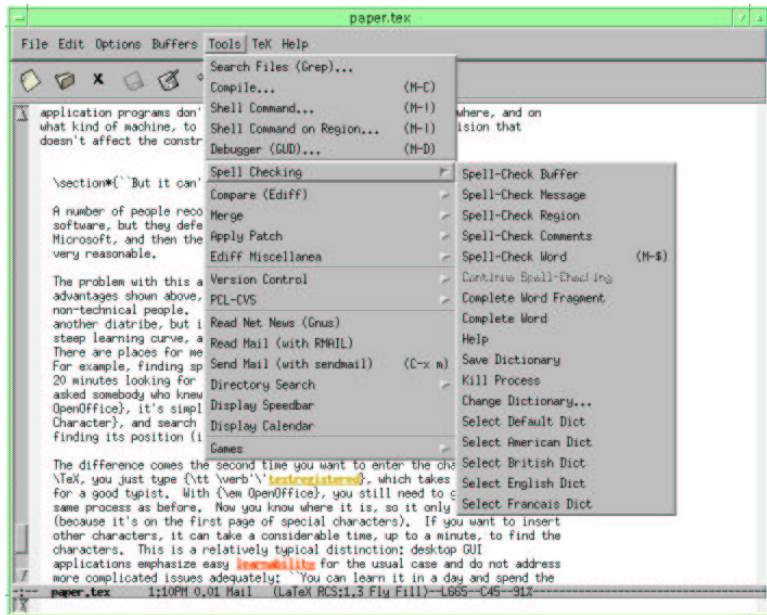
What's wrong here?

- UNIX way isn't perfect either.
- Plenty of reasons to use GUIs and desktops.
- The real question is how to use them.
- A good example: *Emacs*.

Emacs: the anti-GUI

- *Emacs* has a bad reputation.
- Difficult to learn.
- Only for gurus.
- Times have changed.

Emacs today



Emacs differences

- Many functions available.
- Special functions (modes) for editing particular types of file.
- Menus available for inexperienced users.
- System not built around menus.
- Experienced users don't use menus.

Editing and Email

- GUI users don't understand the power of an editor.
- Frequently mutilate mail messages and other texts.
- Look at example from *mozilla*.

mozilla-generated message

```
grog (tty5) wantadilla:Mail
--Mutt: /var/tmp/foomail [Msgs:1, Post:441 4.3K]---(threads/date)---(all)---
1 06-07-2003 Mailer Foo To Mailer Foo ( 47) Re: Mozilla 1.4/Opera 6.12 &
1/2 Mailer Foo Re: Mozilla 1.4/Opera 6.12 & Mime Types -- (55%)
Date: Tue, 08 Jul 2003 20:07:21 -0400
From: Mailer Foo <gso@trini0.org>
To: Mailer Foo <foob@sample.net>
Cc: ports@freshand.org
Subject: Re: Mozilla 1.4/Opera 6.12 & Mime Types

I didn't check in the preferences as yet, and I don't have Flash
installed.
I think that I did have flash installed once when I put this php
code
together, so that may explain it.
I have since set the mime types in Mozilla and Opera (haven't tested
Opera as yet but it should work),
but its fine now.
I was a bit worried over how things got broke, but in the end,
the code
or browsers didn't break.
Thanks for the tip...

Mailer Foo wrote:

>On Tue, 08 Jul 2003 18:37:56 -0400, Funny Bloke <bloke@sample.org>
>wrote:
>
>>In particular, application/octet-stream and
>application/x-shockwave-
>>flash,
>>I have a php script that uploads files, checking for
>allowed
>>filetypes by its mime type.
>>It was put into action some months ago, and today, I notice
>its not
>>allowing flash files.
!;Exit -;PrevPg <Space>NextPg v;View Attach. d;Del r;Reply j;Next ?;Help
```

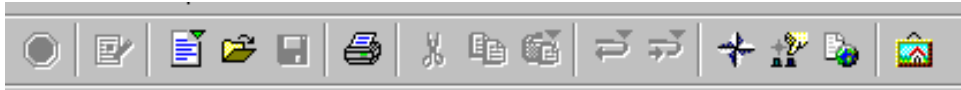
Icons

- GUI applications use many icons.
- Supposed to be easily recognizable.
- Many programs invent their own.
- Who knows what they mean?

Understanding icons: an example



More strange symbols



Problems with icons

- Need to be learnt before they can be understood.
- Opposite of the “don’t need to learn” menu.
- Don’t scale according to the size of the display.
- Using them requires taking hands off the keyboard.
- Difficult to position on, especially on a high resolution display.

Who's in charge?

- Traditionally, programs manipulate data.
- The data is the thing, not the program.
- You can change an image manipulating program.
- You seldom want to change the image.
- “Can you send that image in Illustrator version 8 format?”.
- “Can you send that document in Word format”?

File formats

- UNIX file formats are normally plain text.
- Helps interoperability.

Source example

Source text for *groff*:

Elsewhere, the UNIX philosophy has been defined as:

```
.Ls B
```

```
.LI
```

```
Write programs that do one thing and do it well.
```

```
.LI
```

```
Write programs that work together
```

```
.LI
```

```
Write programs that handle text streams, because that is the  
universal interface.
```

```
.Le
```

Converting to T_EX

Much can be done with an editor:

```
\begin{itemize}
\item
  Write programs that do one thing and do it well.
\item
  Write programs that work together.
\item
  Write programs that handle text streams, because that is the universal
  interface.
\end{itemize}
```

Converting to OpenOffice

Simple:

- Highlight the text in an *Emacs* window and copy it into *OpenOffice* with mouse button 2.
- Remove the markup manually: *OpenOffice* does not have advanced editing facilities.
- Not even simple functions like “delete to end of line”.
- Search for the tool which adds bullet points.
- Find it via the icon `Format-Numbering/Bullets`.

Converting to OpenOffice (2)

- For some reason, the first line came out underlined. Select it, then find the “underline” icon to turn off underlining.
- Press `ctrl-S` to save the document.
- A menu appears showing a quarter of the home directory (not the current working directory), ordered with directories first.
- At the bottom is a window for the file name.
- To save in the current directory, enter the entire pathname.

Converting to OpenOffice (3)

- The file is saved with a different name: instead of being called *sampletext*, it is called *sampletext.sxw*.
- This file is 5045 bytes long. It is in some binary format.

OpenOffice document format

- Look at the saved file with *file*:

```
$ file sampletext.sxw
sampletext.sxw: Zip archive data, at least v2.0 to extract
```

- *zip -l* tells us:

```
$ zip -l sampletext.sxw
Archive:  sampletext.sxw
Length      Date       Time       Name
-----
5582  07-08-03  04:10    content.xml
5147  07-08-03  04:10    styles.xml
1119  07-08-03  04:10    meta.xml
6183  07-08-03  04:10    settings.xml
 752  07-08-03  04:10    META-INF/manifest.xml
-----
18783                                5 files
```

OpenOffice document format (2)

content.xml contains only two lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE office:document-content PUBLIC "-//OpenOffice.org//DTD OfficeDocum
ent 1.0//EN" "office.dtd"><office:document-content xmlns:office="http://open
office.org/2000/office" xmlns:style="http://openoffice.org/2000/style" xmlns
:text="http://openoffice.org/2000/text" xmlns:table="http://openoffice.org/2
000/table" xmlns:draw="http://openoffice.org/2000/drawing" xmlns:fo="http://
www.w3.org/1999/XSL/Format" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns
:number="http://openoffice.org/2000/datastyle" xmlns:svg="http://www.w3.org/
2000/svg" xmlns:chart="http://openoffice.org/2000/chart" xmlns:dr3d="http://
openoffice.org/2000/dr3d" xmlns:math="http://www.w3.org/1998/Math/MathML" xm
lns:form="http://openoffice.org/2000/form" xmlns:script="http://openoffice.o
rg/2000/script" office:class="text" office:version="1.0"><office:script/><of
fice:font-decls><style:font-decl style:name="Arial Unicode MS" fo:font-famil
y="&apos;Arial Unicode MS&apos;" style:font-pitch="variable"/><style:font-de
cl style:name="HG Mincho Light J" fo:font-family="&apos;HG Mincho Light J&ap
os;" style:font-pitch="variable"/><style:font-decl style:name="Thorndale" fo
:font-family="Thorndale" style:font-family-generic="roman" style:font-pitch=
"variable"/></office:font-decls><office:automatic-styles/><office:body><text
:sequence-decls><text:sequence-decl text:display-outline-level="0" text:name
="Illustration"/><text:sequence-decl text:display-outline-level="0" text:nam
e="Table"/><text:sequence-decl text:display-outline-level="0" text:name="Tex
t"/><text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
</text:sequence-decls><text:p text:style-name="Standard">% Hey, Emacs! <text
:s/>Edit this file in *- tex-fill *- mode!</text:p><text:p text:style-name
="Standard">%</text:p><text:p text:style-name="Standard">% $Id: paper.tex,v
```


Deciphering the format

With the help of *Emacs*, it's possible to make this fractionally more intelligible:

```
<text:p>text:style-name="P1">Elsewhere, the UNIX philosophy
has been defined as:</text:p><text:p
text:style-name="P2"/><text:unordered-list
text:style-name="L1"><text:list-item><text:p
text:style-name="P3"><text:s/>Write programs that do one
thing and do it
well.</text:p></text:list-item><text:list-item><text:p
text:style-name="P3"><text:s/>Write programs that work
together.</text:p></text:list-item><text:list-item><text:p
text:style-name="P3"><text:s/>Write programs that handle
text streams, because that is the universal
interface.</text:p></text:list-item></text:unordered-list>
<text:p text:style-name="Standard"/><text:p
text:style-name="Standard">Add some text here.</text:p>
```

To quote Ken Thompson: “You are not supposed to understand this”.

Why bother?

This code is clearly not intended to be examined in this way. It's interesting to look at these steps for other reasons, though:

- *OpenOffice* does understand some of the X conventions, though there appear to be bugs in the implementation. For example, it can't paste from a different display.
- *OpenOffice* is not an editor. If you want an editor, you need to go elsewhere.
- *OpenOffice* does not have a freely exchangeable data format. Thus, after you've gone elsewhere for your editing, you have a data conversion issue.

Why bother? (2)

- *OpenOffice* does not understand UNIX directories adequately. When saving a new document, it goes to the home directory, not the current directory.
- *OpenOffice* recognizes its files by the file type, but it adds an “extension” to the file name when saving.
- This is not the UNIX Way, and it’s confusing for people who are used to free format file names.
- You can’t use UNIX tools like *diff* on OpenOffice documents, so *OpenOffice* includes a tool for version control. Of course, it only works on *OpenOffice* documents.

Multiple activities

- UNIX paradigm is “as many things as possible at a time”.
- Microsoft paradigm was “one thing at a time”.
- Multi-user support frequently ignores UNIX and relies on “profiles”.
- Can only start one web browser.
- Can only start one *OpenOffice*.
- Don’t understand X.
- It’s possible to open a window hundreds of kilometres away and not know about it.

But it can't do any harm!

- One argumentation is: “It provides people with an alternative to Microsoft”.
- It's not really an alternative, it's a clone.
- Stifles real UNIX-based alternatives.

What needs to be done

- Reading the future is difficult.
- The GUI needs to become better attuned to the needs of an experienced typist.
- Desktop GUI software needs to understand more of the UNIX environment.
- Desktop GUI software needs to interact better with the UNIX environment.
- Don't hold your breath.