Promoting BeOS by teaching it

Cedric Neve

1.0 Audience

BeOS is a great operating system. Most of the readers of this document surely know about it. I am willing to address this document to two type of readers:

- The ones that know about BeOS and that are willing to spread their admiration in different ways than casual promotion.
- The ones that don’t know about BeOS but that are open-minded enough to consider at least reading about another operating system than their own.

This second type of people may fear to step outside the Windows world (to jump through the Windows ;-)), or may have apprehensions to leave their beloved Apple system,... Anyone that has a definite answer to “Which operating system is the best ?” or that can’t give the name of another operating system than their own should read this document in order to get some perspective.

2.0 Best Operating System ? Based on which criteria ?

We could endlessly argue about which operating system has the most elegant user interface, which is the fastest, which has the best internal design, which crashes the least, but these are not the most important issue… The real question, when it comes to choosing an operating system, is probably this one: “Which system best suites my needs ?”

If I’m a Quake addict, I will choose the system that has the latest Quake available, that runs fast and doesn’t crash too often. Some of these requirements are conflicting, so I must forget about the last one ;-)…

If I’m a systems engineer supporting thousands of workstations and a couple of servers, I will probably choose a *nux approach since crashing should not happen on an every day basis, if I don’t want to get fired. But that may conflict with the view of workstation users and maybe a hybrid solution will emerge.

Anyway, it seems natural to think that the needs define the tools to use, including the operating system. In a perfect world where there is a perfect all-purpose operating system, this question of the needs does not exist. But we hardly live in such a world and thus most operating system have their place.

So, let’s face it. No operating system is the best in all aspects. Each has its flaws and its extraordinary components. In most situations, choosing the operating system is very important. Everybody working/playing with computers should be able to make their choice in an OS-aware way and not in an ignorant way that can be summerized by “My system was brought with Windows pre-installed. I didn’t even know about anything else and nobody offered me an alternative.”

3.0 How about BeOS

So, to return to the main subject of this paper, let’s talk about what are the main flaws and benefits of BeOS...

3.1 Disadvantages

- Not enough applications
- Not enough drivers
- Not enough users
- Proprietary system (not open-source and not free to use in its professional edition)

3.2 Advantages

- Fast
- Stable
- Great multi-processor and multi-threading support
- Great file system
- Great C++ API

3.3 Implications

Based on this quick list of (dis)advantages, something should be very clear now... The operating system base is solid, the system is great... but it is lacking some features (both drivers and applications) and it isn’t widely used. We can’t change the fact that it is proprietary but we can work on all other aspects. It would be great to find a situation were we can increase the three lacking areas (applications/drivers/users) by finding a situation where BeOS is the best suited operating system. In fact, one of these situations occurs when computer science students follow their operating system and C++ courses... They can write applications and drivers during their exercices, they could increase the number of users while taking advantage of the great features of BeOS.

4.0 Teaching BeOS

I happen to have the opportunity to get involved in some courses that will be given to computer science high school students in Belgium. These courses are: C++ and Operating Systems... What a coincidence !

In fact, many positions are vacant since computer science engineers are not paid enough for such positions. Another reason comes from the fact that some operating system courses still manage to explain how the MVS works. But the MVS is not the most interesting system for the students... Such IBM monsters are not very popular among students... Go figure why ?

Here is what I intend to do and what I am sure can be reproduced by many of the BeOS users since there are many C++ developers among us:

4.1 Operating System Course

I am willing to focus the Operating System course on BeOS and Linux. I will explain the different parts of the Operating System based on the BeOS architecture and I will illustrate some specific aspects by providing the Linux code (eg. scheduler). I will also make comparisons between operating systems and explain why the accent is placed on BeOS (for the concepts and the design) and Linux (for the implementation).

The exercices will include:

- familiarizing with the shell (in Linux or an *nux)
- driver writting

Of course, this last part will be made on BeOS, probably by using Linux prior implementations.

4.2 C++ Course

The C++ course will begin with all the usual C++ language explanations. The Object Oriented design will be explained as well as the design patterns. Finaly, some aspects of the BeOS API will be browsed.

The exercices will mainly be aimed at:

- Writting an application.

4.3 Liberty of choice

The students will have a rather large liberty of choice for the driver and the application to write. The expectations will not be expressed in terms of what functionalities should the application contain but rather what level of complexity it will have.

The exercices for both courses will contain:

- Seeking information about possible projects
- Propose the project for approval
- Write down the specifications

- Design the driver/application
- Implement it
- Submit the project for review (cotation)

Of course, I will encourage the student developers to release their useful applications/drivers to the community.

5.0 Conclusion

BeOS has interesting features (C++ API, excellent OS design,...) that make it an excellent system to play around with operating system level concepts and to implement applications. The friendly community and the free PE edition are valuable additions that should facilitate the learning processes and bring some more BeOS adoptions. A single class of users would double the number of BeOS users in Belgium which would already be an interesting achievement. The fact that this initiative will also lead to new drivers and applications just makes this more interesting...

So, for those who can take the time, don’t hesitate. Computer Science high schools are searching for professors and it just might be your chance to promote BeOS, make a valuable contribution for the community and get paid for it... all at once.

6.0 References

For comments, don’t hesitate to send an e-mail to Cedric.Neve@cediti.be.

I have heard of a brazilian operating system course which focuses on the BeOS, I don’t have the reference but you can contact me to get an update about it, as I am trying to find out who is teaching it and what his course includes.

If you wish to know how this experience is evolving, or if you wish to have some details, send me an e-mail and I’ll be most happy to answer to your questions.