

Singularity, Cosmos... des systèmes d'exploitation écrits en C#

Microsoft a mis à profit la TechFest 2008 pour présenter les dernières évolutions de son système d'exploitation [Singularity](#). Ce dernier est programmé en C#, avec une base en C. Le Singularity Research Development Kit (RDK) 1.1 [est aujourd'hui disponible](#). Il est librement téléchargeable pour un usage non commercial. L'archive comprend le code source du système, des outils de compilation ainsi que d'autres ressources (jeux...).

Notez que ce projet dédié à l'étude et à la recherche n'a pas pour objectif de devenir un concurrent des systèmes d'exploitation de la firme. En quelque sorte, il est un peu pour Microsoft ce qu'un *concept car* est pour un constructeur automobile, chose que confirme Rick Rashid, senior vice-président de Microsoft Research : « *Singularity n'est pas le prochain Windows. Pensez-y comme à un concept car. Il s'agit d'un prototype de système d'exploitation conçu dès le départ pour tester un nouveau paradigme portant sur la façon dont le système d'exploitation et les applications interagissent les uns avec les autres. Nous le mettons à la disposition de la communauté dans l'espoir qu'il permettra aux chercheurs de tester rapidement de nouvelles idées* ».

L'utilisation d'un langage de haut niveau, comme le C#, est censée réduire le nombre de failles de sécurité qui s'insèrent dans le code (par exemple, C# n'est pas sujet aux dépassements de tampon). La machine virtuelle .NET, qui exécute du code intermédiaire CIL (*Common Intermediate Language*), est également une surcouche appréciable pour rendre portable un tel système d'exploitation.

Des concurrents libres

L'idée de développer un système d'exploitation en C# a été adoptée sur d'autres projets. [Cosmos](#), sous licence BSD et [SharpOS](#), sous licence GPL, sont les deux concurrents de Singularity. L'approche consiste ici à abandonner totalement le C, pour n'utiliser que C#. Les parties du noyau qui ne peuvent être qu'en code machine utilisent alors un convertisseur « C# vers assembleur ». À cet effet, un compilateur AOT (*Ahead-of-Time*) est présent.

Ces deux systèmes d'exploitation sont encore en cours de développement. SharpOS se montre cependant légèrement plus avancé. Une interface graphique (SharpWS) et une base de données relationnelle (SharpSQL) sont ainsi en cours d'écriture.

```
SharpOS [In esecuzione] - Intel® VirtualBox
Macchina Depositi Auto

[commands]      Displays a list of all the commands
[cpuid]         Displays info about the CPU and its capabilities
[egg]           The mandatory easter egg
[halt]          Halts the system
[keymap]        Lists keymaps and changes the active one
[memdump]       Displays memory usage
[panic]         This tests the panic mode
[reboot]        Restarts the machine
[stage]         Prints the current kernel stage
[version]       Displays version information about SharpOS
[show]          Shows information about SharpOS
[snake]         Plays a quick game of Snake#

#OS> show
Syntax:
  show w : shows the warranty.
  show c : shows the copyright.
  show d : shows the developers.
Use 'help commands' to get a list of commands.

#OS> stage
Current kernel stage: (B) diagnostics

#OS>
Timer ticks: 2400
```

L'alternative : le système d'exploitation écrit en Java !

C# et la machine virtuelle .NET n'ont pas été les seuls à inspirer des systèmes d'exploitation. C'est dans le même esprit que le projet libre [JNode](#) a vu le jour en mai 2003. Java est ici à l'œuvre. Aujourd'hui, JNode est 99% compatible avec l'API de Java 6 et fait fonctionner les applications graphiques AWT et Swing. L'intégration d'openJDK est également en cours d'achèvement (elle est terminée à plus de 80%). Enfin, JNode dispose de multiples pilotes et supporte les systèmes de fichiers les plus courants (FAT32, NTFS, EXT2, NFS, *etc.*).

