Jnode and the way ahead

Jnode: Java New Operating System Design Effort

Contents

- Where we are now
- The ultimate goal
- How to go about it
- What we need NOW
- Technology wishlist
- Also helpful
- Watch out for traps

Where we are now

- Bootable in VMware and on selected Hardware
- 64-Bit and Multi-Processor support (currentIs broken!)
- Prototypical UI
- Drivers for networking, human interface and multiple file systems

The ultimate goal

- Java developers' choice of heart
- The OS choice for Java server software
- Be an OS with real-world use cases
- Keep native code out
- Offer a playground for stunning new Java technologies

Interlude

What with Jnode? Why don't ya contribute to Linux or something?

Jnode's written in Java.

Cool!

Yeah!

...erm...

What?

Why Jnode is more than cool

- Use of Java eco-system provides for
  - Mature APIs for many purposes
  - Mature components, which quicken development
  - Being able to use standards and run applications that have been around for years

How to go about it

1. Put strengths into main project, weaknesses into sandbox
2. Stabilize and make it usable
3. Employ standards where applicable
4. Get major companies to use and support Jnode
5. Put more force into sandbox

What we need NOW

- Allow System.exit
- Allow System.setSecurityManager
- Get Tomcat/JBoss to work
- Get Isolates to work
- Enable remote debugging

Technology wishlist

- JINI
- JMS
- JMX
- JNDI
- db4o / Hibernate
- AspectJ
- JAAS
- JNLP
- JAIN
- JMI
- JBoss drools
- JBPM
- Lucene
- JavaMail
- OSGi

Also helpful

- Maven-Like run-time dependency management
- Eat-your-own-dogfood: push jnasm
- Terracotta clustering support
- JavaDesktop integration
- Use UML to describe inner workings
- Use MDA to ease development

Watch out for traps

- ✗ Looking like a toy OS
- ✗ Starving sandbox due to lack of integration
- ✗ Preferring bad “standards” over good API