

Version 1.0

**Autor: Oliver Böhm (oliver.boehm@agentes.de)**

© **Oliver Böhm**

**Dieser Text wird unter der UVM Lizenz (Freie Inhalte) zur Verfügung gestellt.**

*Dieser Vortrag gibt einen Überblick von über 10 Jahren Java unter Linux. Während man in der guten alten Zeit noch mit Performance-Problemen und der Duldung der Blackdown-Portierung durch Sun zu kämpfen hatte, zeigte sich mit dem Aufkommen von Eclipse, welche Bedeutung Linux für IBM hatte (Eclipse 1.0 gab es nur für Windows und Linux).*

## **Java unter Linux - es war einmal...**

Es war einmal ein Entwickler, der auszog, um Java zu lernen. Das war vor über 10 Jahren. Damals war Java-Entwicklung noch gleichbedeutend mit Java-Entwicklung unter Windows. Dies hat sich zum Glück geändert, oder?

Heute ist Java-Entwicklung unter Linux nichts Exotisches mehr und auch nicht gleichbedeutend mit Emacs als die alleinige Java-Entwicklungsumgebung. Allerdings hat es lange gedauert, bis Linux als Entwicklungsplattform für Java attraktiv wurde. Während man in den Anfängen noch mit Performance-Problemen und der Duldung der Blackdown-Portierung durch Sun zu kämpfen hatte, gab es erst mit dem Aufkommen von Eclipse 1.0 eine freie IDE-Alternative, die eine komfortable Java-Entwicklung ermöglichte.

Heute hat sich Java unter Linux durchgesetzt, nicht nur im Serverbereich. Und auch die Zukunft wird nicht ohne Linux stattfinden, wie uns ein Blick in die Glaskugel verraten wird. Und wenn er nicht gestorben ist, entwickelt er noch immer Java unter Linux bis an sein System.exit()...

*Ich habe mir immer gewünscht, dass mein Computer so leicht zu bedienen ist wie mein Telefon; mein Wunsch ging in Erfüllung: mein Telefon kann ich jetzt auch nicht mehr bedienen.*

**Bjarne Stroustrup**

## **Einleitung**

Als ich Anfang des Jahres meinen Rechner aufräumte, stolperte ich über eine Datei mit der Endung .sxi auf, die wohl noch aus Zeiten von OpenOffice 1.x stammte. Es waren die Folien zum Vortrag „Java unter Linux“, den ich 2003 auf dem Linuxtag in Chemnitz hielt.<sup>1</sup> Alte Erinnerungen wurden wach, als ich in den Folien blätterte (ich war zu dieser Zeit gerade mit meinem Buch „Java SW-

---

<sup>1</sup> <http://chemnitzer.linux-tage.de/2003/vortraege/detail.html?index=28>

Engineering unter Linux“ fertig). Viele der Folien sind heute überholt, während manche der Folien erstaunlicherweise auch heute noch aktuell sind, über andere Folien muss ich heute, 7 Jahre danach, schmunzeln.

## Vorsätze 2010

Ein neues Jahr birgt immer das Risiko, dass man sich gute Vorsätze fasst, diese aber nicht immer in die Tat umsetzt. Hatte ich mir nicht vermerkt, mal wieder einen Vortrag auf einem Linux-Tag zu halten? Stimmt.

Linux-Tage haben meist einen eigenen Charakter. Es ist, als ob man zu einem großen Familien-Treffen geht. Und vor allem bekommt man hier Hilfe, wenn die Graphikkarte mal wieder zickt oder die UMTS-Karte konfiguriert werden will. Von daher kam ich diesem Vorsatz für 2010 gerne nach und reichte diesen Beitrag ein – auch deswegen, weil ich dieses Mal auch direkte Unterstützung durch die agentes AG für die Einreichung bekam.

## Folien-Recycling

Als dann mein Vorschlag Anfang April angenommen wurde, stand ich vor einem Problem: Soll ich die Folien aktualisieren oder einen neuen Foliensatz erstellen? Für letzteres sprach allein die Tatsache, dass 60 Folien einfach zuviel für 30 Minuten sind.

Ich entschied mich dann doch für einen anderen Ansatz: ich ließ die Folien (mit kleinen Änderungen, die ich aber entsprechend markiert habe) so, wie sie waren. Schließlich macht aus meiner Sicht gerade das den Reiz der Folien aus. Sogar das Format (OpenOffice 1.x) habe ich aus diesem Grund gelassen, auch wenn eigentlich das ODP-Format vorgeschrieben war (die Organisatoren mögen mit dies verzeihen). Änderungen und Aktualisierungen, die sich seitdem ergeben haben, habe ich in dieses Dokument, das du gerade vor Augen hast, ausgelagert.

**D.h. die Folien von damals und dieses Dokument von heute sind als eine Einheit zu sehen!**

## Folien-Übersicht

Genug der einleitenden Worte. Wenden wir uns den einzelnen Folien zu.

### Folie 3 (Java unter Linux)

Heute ist natürlich Java 6 aktuell und Java 7 steht kurz vor der Tür. Sun existiert nicht mehr als eigenständige Firma, sondern unter dem Dach von Oracle.

## Folie 6 (Warum Linux?)

Windows NT war 2003 vereinzelt noch anzutreffen. Inzwischen gehören Total-Abstürze auf aktuelleren Windows-Version eher der Vergangenheit an (wenn man mal von Windows Vista absieht). Wenn heute ein Windows-Rechner abstürzt, liegt es eher an Viren und Trojaner, die sich dort gerne einnisten.

## Folie 9 (Welche JVM?)

Inzwischen ist natürlich Java 6 aktuell (und Java 7 steht vor der Tür). Aber man hat nach wie vor die Auswahl zwischen:

- Sun JDK (bzw. Oracle JDK)<sup>1</sup>
- IBM JDK<sup>2</sup>
- OpenJDK<sup>3</sup>

## Folie 10/11 (CaffeineMark)<sup>4</sup>

Vergleichbare Zahlen für ein aktuelles Java 6 oder 7 zu bekommen, sind schwierig, da mein Test-Rechner von damals (ein Pentium I mit 133 MHz) nicht mehr existiert. Ausserdem ist Performance heute (fast) kein Thema mehr und große Performance-Sprünge durch die VM sind nicht zu erwarten, auch wenn Oracle angekündigt hat, ihre Erfahrungen aus JRockit in die nächsten VMs einzubringen.

Ein Testlauf auf aktuellen Systemen brachte folgendes Ergebnis zu Tage:

	IBM T40 (1,5 GHz, Ubuntu 9.10, Sun-JDK 1.6.0)	IBM T40 (1,5 GHz, Ubuntu 9.10, IBM-JDK 1.6.0)	MacBook Pro (2,26 GHz, OS-X 10.6.3)
CaffeineMark 3.0	17000	31000	56000
embedded CM	31000	77000	191000

Alle Tests fanden mit Java 6 statt. Die Zeiten des OpenJDKs sind mit den Zeiten des Sun-JDKs vergleichbar. Erstaunlich ist, dass das IBM-JDK immer noch rund doppelt so schnell ist.

---

1 <http://www.java.com/de/download/manual.jsp#lin>

2 <http://www.ibm.com/developerworks/java/jdk/linux/download.html>

3 <http://openjdk.java.net/install/>

4 <http://www.benchmarkhq.ru/cm30/>

## Folie 12 (Weitere VM's)

Inzwischen existiert mit OpenJDK eine OSS-Implementierung von Java. Trotzdem gibt es auch heute noch die anderen VMs

- Kaffe<sup>1</sup>: letztes Release: 2008
- Japhar:<sup>2</sup> scheint bei Java 1.1. stehen geblieben zu sein
- ORP<sup>3</sup>: letztes Release: 2002
- GNU Classpath<sup>4</sup>: letztes Release: 2009

Daneben gibt es inzwischen ein Reihe weiterer VMs, die hier nicht aufgeführt sind.

## Folie 22 (CVS-Alternative)

Subversion kann inzwischen als ausgereift bezeichnet werden (akt. Version: 1.6.9). Hinzugekommen sind aber noch git (<http://git-scm.com/>) oder Mercurial (<http://mercurial.selenic.com/>).

## Folie 24 (ArgoUML)

Inzwischen auch für andere Sprachen verfügbar. Aktuelle Version: 0.30

<http://argouml.tigris.org/>

## Folie 25/26 (Fujaba)

Inzwischen auch als Eclipse-Plugin verfügbar: <http://www.fujaba.de/>

## Folie 27 (kUML)

Projekt steckt immer noch fest („under heavy development“), aber seit 2000 kein Update mehr

## Folie 28 (Umbrello)

Umbrello gibt es immer noch und läuft sogar unter Windows und Mac.

Homepage: <http://uml.sourceforge.net/>

---

1 <http://www.kaffe.org>

2 <http://www.hungry.com/old-hungry/products/japhar/>

3 <http://orp.sourceforge.net/>

4 <http://www.gnu.org/software/classpath/>

## Folie 29/30 (Together)

Ist inzwischen in Borland Together aufgegangen.

## Folie 42/43 (Datenbanken)

SAP-DB heißt inzwischen MaxDB.

Ab Java 6 ist Derby als Java-DB enthalten.

aktuelle Versionen:

- MySQL 5.1
- Postgres 8.4

## Folie 45 (EJB Application Server)

Hier muss natürlich als freier Server noch Apaches Geronimo<sup>1</sup> genannt werden.

## Folie 48 (JBoss)

Das Logo stammt noch aus Zeiten von JBoss 2.x (aktuell: 5.x). Inzwischen gehört JBoss zu RedHat.

aktuelles Logo:



## Folie 49 (Test-Frameworks)

Inzwischen ist JUnit Standard und in vielen IDEs bereits integriert. Weitere Test-Frameworks wie TestNG oder FIT/FITnesse sind hinzugekommen.

expect gibt es nach wie vor und gehört zum Standard-Umfang vieler Linux- und Unix-Derivate (z.B. auch MacOS-X).

## Folie 53 (Installations-Hilfen)

Das Installations-Kit von IBM<sup>2</sup> wird wohl nicht mehr weiterentwickelt.

---

1 <http://geronimo.apache.org/>

2 <http://alphaworks.ibm.com/tech/installtoolkit>

Und neben RPM gibt es natürlich auch weitere Installer (wie z.B. den Debian-Installer).

## **Folie 55 (Ein Blick in die Zukunft)**

Mit Android hat Linux und Java nochmals richtig Schub bekommen. Während Linux vom Embedded Bereich nicht mehr wegzudenken ist (man denke nur an die ganzen DSL-Router, NAS-Geräte, ...) tut sich Java noch etwas schwerer in diesem Bereich.

## **Folie 59 (Bücher)**

Beide Bücher sind bereits Geschichte (d.h. nicht mehr erhältlich).

## **Schlussbemerkung**

*...und wenn sie nicht gestorben sind, dann leben sie noch heute.*

Zunehmend spielt die Frage des Betriebssystems keine Rolle mehr. Zumindest eine kleinere Rolle, als es noch vor 7-10 Jahren der Fall war. Trotzdem treffe ich bei vielen Kunden noch Windows als Entwicklungsplattform an. Daran habe ich mich gewöhnt. Ärgerlich dabei ist bloß, dass ich oft durch den Virenschanner ausgebremst werde.

Immer öfters habe ich aber die Wahl. Und hier habe ich bis vor kurzem immer zu Linux (erst SuSE, dann Ubuntu) gegriffen. Warum? Es bietet mir die Freiheit und Tools, die ich als Entwickler brauche. Fehlende Software lässt sich bequem über die Softwareverwaltung nachinstallieren. Und Linux bringt eine vernünftige Kommandozeile mit, über die ich viele Dinge schneller als mit der Maus erledigen kann.

Dass ich inzwischen zu Mac gewechselt bin, liegt an der Grafik (sorry, Linux) – ich muss mich hier nicht zwischen KDE oder Gnome entscheiden (will ich auch nicht), sondern hab' hier ganz einfach die beste Oberfläche, die es derzeit gibt. Aber ich bin dadurch auch abhängiger geworden (leider) – auf das Wohlwollen von Apple (bzgl. Java und bzgl. Security-Fixes). Aber notfalls gibt es ja immer noch Ubuntu...

*May the Source be with you!*