



weil er eine Unix-Shell mit den dazugehörigen Werkzeugen anbietet. Die Symbiose mit dem stabilen Unixkern (ja, ich weiß... is kein reines Unix....) und der schönen Grafik oben drauf ist für Endbenutzer demnach auch eine ernstzunehmende Alternative. Wenn, verliert Windows Endnutzer nicht an Linux, sondern eher an den Mac. Nur meine persönliche Meinung. Ist zwar ein anderes geschlossenes „Eco“system, aber das soll jeder für sich entscheiden.

Nebenbei fand die Open Sourcebewegung unter Unix seinen Anfang. Open Source ist ein super Begleiter, wenn man programmieren will. Die Unix-Shell und das Dateisystem sind wie erwähnt unter dem Mac genauso vorhanden.

Für das Programmieren sind Open-Source-Programme deshalb ideal, weil man den Quellcode einsehen darf und nicht vordergründig die Kosten. Man kann genau nachvollziehen, wie sich beispielsweise Funktionen realisieren und eventuell an seine Bedürfnisse abändern lassen. In der Windows-Welt ist so etwas in der Richtung nicht da. Wenn es so etwas nicht gäbe, würde jeder Programmierer von rein proprietärer Dokumentation abhängig sein. Sämtliche Bücher, Blogs und Tutorials könnten gar nicht in dem Ausmaß geschrieben werden.

Wenn Du hingegen in die Windows Programmierung einsteigen willst, führt natürlich kein Weg an Windows vorbei. Wenn Du unter Apple... ditto. Wenn Du unter Linux....ditto.

Das Ganze fängt mit dem Binär- und Source-Kram an. Hört sich kompliziert an, aber eine Zylinderkopfdichtung zu tauschen, ist schwerer. Die Installation von Software zwischen Linux und Windows unterscheidet sich erheblich. In Linux holst Du Dir Deine Installationsdateien von einem „Repository“. Betrachte es einfach als „Lager“ im Internet, wo Du permanenten Zugriff hast unter Linux. Dort sind die verschiedenen Installationsdateien in einem Paket zusammengefasst.

„Firefox.deb“ wäre so ein typisches Binärpaket. Das heißt, dass das deb-Paket schon kompiliert wurde. Kompilieren heißt, vom menschenlesbaren Quelltext eines Programmes wurde dieses mittels eines Compilers in ein binäres Format umgewandelt. So ein Compiler gehört

auch zur Standardausrüstung unter Linux (gcc).

Und damit das Ganze auch eine Struktur und Übersicht bekommt, verwaltet ein sogenannter Paketmanager (apt, yum, synaptic...) die binären Pakete. So etwas wie einen Paketmanager gibt es unter Windows nicht, was ich sehr schade finde. Der „Microsoft Store“ ist so eine Art Repository (Lager, Quelle), das sich aber um keine Abhängigkeiten oder verschiedene Bibliotheken kümmern kann. Dies sind aber inhärente Aufgaben eines Paketmanagers unter Linux. Bibliotheken sind nicht selbständig lauffähige Hilfsmodule, die andere Programme zu deren eigenen Ausführung anfordern müssen. DLL unter Windows wären solche Bibliotheken beispielsweise. Der größte Vorteil eines Paketmanagers liegt aber in der Verfügbarkeit der vorkompilierten Open Source Software. Eben in Binärform, und das alles von einem Platz aus, ohne da im Internet auf die Suche zu gehen. Nebenbei kümmert sich dieser auch noch um die Updates, Up/Downgrades, De/Installationen, Reparaturen von Paketen, benötigte Bibliotheken und das Auflösen von Abhängigkeiten.

Ein System, das typischerweise nur mit Binärformaten arbeitet, wäre Windows. (exe, msi...)

Unter Linux heißen Binärformate beispielsweise deb und rpm. Dort könnte ich aber auch die Software von den Quelldateien her einsehen, abändern und auch dementsprechend installieren. Oder eine bestimmte Version eines Programmes mit genau definierten Features installieren. OpenLdap, ein Open-Source-Produkt, bezeichnet den Verzeichnisdienst unter Linux. Dieser stellt aber keine Binärdistribution zur Verfügung, obwohl dies grundsätzlich unter Windows möglich wäre. Der Verzeichnisdienst unter Windows heißt „Active Directory“. Dieser läuft sehr wohl unter Windows, aber nicht unter Linux.

Die Mehrzahl der Programme ist unter Windows nicht im Quellcode verfügbar. Außerdem ist Windows per se nach der Installation, im Gegensatz zu Linux nicht programmierbar. Es enthält keine Werkzeuge, um Programme (im größeren Rahmen) selber effizient zu entwickeln. Die entsprechenden Entwicklungswerkzeuge bietet Microsoft separat gegen Münzein-

wurf an. (z. Bsp. Visual Studio als IDE...und ja, es gibt auch eine Communityversion).

Linux hat konträr dazu seine Entwicklerwerkzeuge schon an Board.

Wer also freie Software unter Windows mit allem nötigen Schnickschnack entwickeln will, muss sich erst mal kratzen. Ist so. Da alle Windows-Programme aber sowieso nur in Binärform vorliegen, wird der gemeine Nutzer nie zum Lernen veranlasst, eventuell mit einem Compiler zu arbeiten.

Kurz gefasst: Linux ist von Haus aus in alle Richtungen programmierbar, da die Programme in der Regel auch im Quellcode verfügbar sind. (Firewall, Server, Desktop, Raspberry Pi, ERP-Systeme, Cluster, Supercomputer, Router, Modems, Tablets, Mobilgeräte, Telefonanlagen, Security-Distros....). Open-Source-Software (zum Beispiel Linux) ist immer im Quellcode verfügbar. Sieh Dir mal auf die Schnelle ein Beispiel mit dem Dateimanager „Dolphin“ an.

Alle ausführbaren Dateien unter Linux liegen im Ordner „bin“ (binaries-Binärdateien). Das ist jetzt alles sehr vereinfacht, sollte aber des Dobermanns Kern treffen. Sieh Dir **Bild 8** an.

Gelber Pfeil. Dolphin ist rot ausgegeben und mit einem Sternchen am Ende versehen. heißt für mich, dass diese Datei ausführbar ist. Der darunter liegende Befehl „file dolphin“ gibt mir in Textform die Art der Datei aus. Roter Pfeil. „Executable“ steht da. Also ausführbar. Passt. Jetzt will ich mir in der letzten Zeile mit einem Ausgabeprogramm „less“ diese Datei ansehen. Es kommt eine Meldung, dass diese Datei möglicherweise eine Binärdatei ist. Wenn ich mir diese jetzt so ansehen will, werde ich nicht glücklich werden, da nur unleserliches Zeug dargestellt wird.

Darum kann man Binärdateien nicht mit einem gewöhnlichen Texteditor einfach so anschauen (ja, anschauen kannst es schon), sondern braucht auf das jeweilige Dateiformat abgestimmte Editoren. „xxd“ wäre so eine einfache Lösung auf der Kommandozeile. **Siehe Bild 9.**

Wie siehst Du unter Windows Binärdateien an? Du brauchst auch dort einen eigenen Editor, logo. „regedit“ (Registrierungseeditor) heißt der. Links in die Suchbox regedit eingeben, und schon öffnet sich ein spezieller Editor, der Dir die Binärdateien anzeigen kann. **Siehe Bild 10.**

Zusammengefasst heißt das, dass Binärdateien immer einen speziellen Editor zum „Anschauen“ brauchen. Egal, von welchem Betriebssystem wir hier reden.

Die Mehrzahl der Microsoftprogramme ist nur in Binärdateien verfügbar und Entwicklerwerkzeuge sind zudem kostenpflichtig. Somit hat Herr **Mikl** die Plattform (Linux) gewählt, um jetzt beispielsweise sein Warenwirtschaftsprogramm zu realisieren. Es ist wie bei den Programmiersprachen. Es gibt keine perfekte, genauso

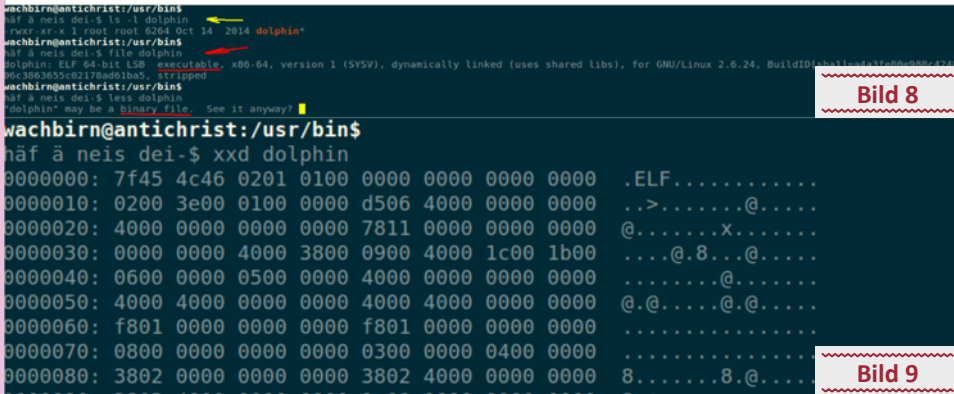


Bild 8

Bild 9