Wichtige Lizenzen, Shell Teil 2

Linux-Kurs der Unix-AG

Andreas Teuchert

26. April/2. Mai 2012

Kommerzialisierung des Softwaremarkts in den 80ern

- bis in die 70er/80er Jahre: Software wird (meist mit Quellcode) formlos verteilt
- Weitergeben und/oder Verändern von Software ist normal
- 80er Jahre: Firmen kommen auf die Idee, Software zu verkaufen
- Software wird nun ohne Quelltext ausgeliefert („Closed Source“)
- Weitergeben und Verändern nicht erlaubt

Nachteile von Closed-Source-Software

- Weiterbildung durch Studieren des Quelltextes nicht möglich
- Anpassen des Programms an eigene Bedürfnisse kaum möglich
- Verbesserungen können nur vom Hersteller vorgenommen werden

Gründung von GNU und der FSF

- vor allem Forscher stören sich an den Einschränkungen von Closed-Source-Software
- einer von ihnen ist Richard Stallman
- 1983: Gründung von GNU
- 1985: Gründung der Free Software Foundation (FSF)
- Ziele der FSF:
  - Unterstützung des GNU-Projekts
  - „Lobbyarbeit“ für freie Software

Definition freier Software nach FSF

Freie Software muss vier Freiheiten einräumen:

0. das Programm für jeden Zweck verwenden
1. die Funktionsweise des Programms untersuchen (z. B. Quellcode lesen) und es anpassen
2. das Programm weitergeben
3. das Programm verbessern und die Verbesserungen veröffentlichen

Freie Software vs. Open Source

- „Freie Software“ ist nicht eindeutig
- „frei“ wird häufig mit „kostenlos“ gleichgesetzt (Beispiel: „Freeware“)
- Frei wie Freibier vs. Frei wie Freie Rede
- viele Leute bevorzugen daher den Begriff „Open Source“
- „Open Source“ bedeutet strenggenommen nur, dass der Quellcode offenliegt
- daher auch kein treffendes Schlagwort

Gründung der OSI

- 1998: Gründung der Open Source Initiative (OSI)
- Gründer: Bruce Perens und Eric Raymond
- gleiche Ziele wie die FSF, aber weniger radikal
- Open Source statt Freie Software als zentraler Begriff

FOSS und FLOSS

- sowohl „Freie Software“ als auch „Open Source“ haben als Schlagwörter ihre Nachteile
- Kompromiss: FOSS bzw. FLOSS
- FOSS: **F**ree and **O**pen **S**ource **S**oftware
- FLOSS: **F**ree, **L**ibre and **O**pen **S**ource **S**oftware

Wofür Lizenzen?

- Software ist grundsätzlich durch das Urheberrecht geschützt
- oder durch entsprechende Gesetze in anderen Ländern (Copyright, etc.)
- Verwendung der Software setzt Erlaubnis des Rechteinhabers voraus (Lizenz)

Unterschiede zwischen proprietären und freien Lizenzen

Proprietäre Lizenzen

- schränken die Rechte des Nutzers i. d. R. ein (kein Weiterverkauf, Nutzung nur für bestimmte Zwecke)
- Nutzer muss vorher zustimmen (Vertrag, EULA)

Freie Lizenzen

- gestehen dem Nutzer zusätzliche Rechte zu (Veränderung, Weitergabe)
- kein Nachteil für den Nutzer
- daher keine Zustimmung nötig

General Public License (GPL)

- am weitesten verbreitete FOSS-Lizenz
- bei der Weitergabe von GPL-Software müssen dem Empfänger auch die ihm nach der GPL zustehenden Rechte eingeräumt werden (Copyleft)
- dafür muss der Quellcode mit weitergegeben werden (oder im Nachhinein angefordert werden können)
- GPL macht keine Vorschriften über den Preis
- GPL-Software kann also verkauft werden, der Käufer darf sie aber beliebig oft weiterverkaufen oder -schenken

BSD-Lizenz

- ursprünglich für BSD-Unix verwendet
- erlaubt die Weitergabe solange der Lizenz-Text erhalten bleibt
- bei der Weitergabe muss der Quelltext nicht mit weitergegeben werden (kein Copyleft)
- die ursprüngliche Version forderte eine Anerkennung des Original-Autors in Werbetexten

Creative Commons

- viele Lizenzen (wie GPL und BSD) sind für Software ausgelegt
- für Texte, Bilder, Musik, etc. nicht geeignet
- Creative-Commons-Lizenzen können auch für solche Werke verwendet werden
- Autor kann auswählen, ob das Werk für kommerzielle Zwecke benutzt werden darf und ob es in veränderter oder unveränderter Form weitergegeben werden darf
- das Werk kann auch gemeinfrei (Public Domain) gemacht werden (Verzicht auf alle Urheberrechte, soweit möglich)

Geld verdienen mit FOSS

- Geld verdienen durch den Verkauf von Einzellizenzen funktioniert bei FOSS nicht
  - Kunden dürfen Software einfach weitergeben
  - Konkurrenten dürfen das Produkt kopieren und selbst verkaufen
- Open-Source-Geschäftsmodelle:
  - Support
  - Schulungen
  - individuelle Weiterentwicklungen
  - kostenpflichtige (eventuell unfreie) Zusatzfunktionen

mv

- `mv` („move“) verschiebt Dateien und Verzeichnisse
- wichtige Optionen:
  - `-b`: legt Sicherungskopien an, wenn Dateien überschrieben werden
  - `-i`: fragt vor dem Überschreiben nach
- Argumente:
  - zwei Dateiennamen: Datei wird umbenannt
  - beliebig viele Dateien/Verzeichnisse und Verzeichnis als letztes Argument: Dateien werden in das Verzeichnis verschoben

cp

- `cp` („copy“) kopiert Dateien und Verzeichnisse
- wichtige Optionen:
  - `-b`: legt Sicherungskopien an, wenn Dateien überschrieben werden
  - `-i`: fragt vor dem Überschreiben nach
  - `-r` oder `-R`: kopiert Verzeichnisse rekursiv
  - `-u`: kopiert nur neuere Dateien
- Argumente:
  - zwei Dateinamen: Kopie der Datei wird angelegt
  - beliebig viele Dateien/Verzeichnisse und Verzeichnis als letztes Argument: Dateien werden in das Verzeichnis kopiert

ln

- `ln` („link“) legt Verknüpfungen an
- ohne Optionen wird ein zweiter Name für die gleiche Datei erzeugt (Hardlink); geht nicht mit Verzeichnissen (sonst Zyklen im Dateibaum und andere schlimme Dinge)
- `ls -l` zeigt in der zweiten Spalte den Link-Zähler an
  - zeigt bei Dateien die Anzahl der Links an
  - bei Verzeichnissen mindestens 2 wegen `.` (Link auf das Verzeichnis selbst) und für jedes Unterverzeichnis +1 wegen `..` (Link auf das Elternverzeichnis)
- `ls -i` zeigt die sog. Inode-Nummer an (laufende Nummer für Dateien); bei Links ist diese gleich
- wird ein Link gelöscht, kann man über die anderen noch auf die Datei zugreifen

ln -s

- Hardlinks sind verwirrend (verschiedene Dateinamen, aber gleicher Inhalt)
- `ln -s` legt einen Softlink an: Link zeigt auf das Ziel
- geht auch mit Verzeichnissen
- wird das Ziel gelöscht, zeigt der Link ins Nichts
- `ls -l` zeigt an, wo der Link hinzeigt

cat

- `cat` gibt den Inhalt von Dateien aus
- Wichtige Optionen:
  - `-A`: nicht darstellbare Zeichen (Tabs, Zeilenumbrüche, etc.) werden durch Ersatzzeichen dargestellt
- Argumente: eine oder mehrere Dateien

less

- `less` ist ein Pager: gibt soviel Inhalt einer Datei aus wie auf den Bildschirm passt und ermöglicht es zu scrollen
- Scrollen mit den Pfeiltasten (eine Zeile) oder Bild-Auf/-Ab (eine Seite)
- Pos1 und Ende springt zum Anfang bzw. zum Ende
- Suchen mit `/muster` (nach unten) oder `?muster` (nach oben)
- Weitersuchen mit `n` (gleiche Richtung) oder `N` (andere Richtung)
- Beenden mit `q`

head

- `head` gibt den Anfang einer Datei aus
- Wichtige Optionen:
  - `-n Zahl`: gibt soviele Zeilen aus wie angegeben; Standard sind 10 Zeilen

tail

- `tail` gibt das Ende einer Datei aus
- Wichtige Optionen:
  - `-n Zahl`: gibt soviele Zeilen aus wie angegeben; Standard sind 10 Zeilen
  - `-f`: wartet darauf, dass neue Zeilen in die Datei geschrieben werden und gibt diese dann aus

Merkhilfe für Optionen

- viele Befehle unterstützen einen gemeinsamen Satz an Optionen
- `-r` („recursive“): `rm`, `cp`
- `-i` („interactive“): `rm`, `mv`, `cp`, `ln`
- `-b` („backup“): `mv`, `cp`, `ln`
- `-v` („verbose“): `rm`, `mv`, `cp`, `ln`
- Faustregeln:
  - wenn man eine Option braucht, existiert sie wahrscheinlich
  - Optionen haben sinnvolle Namen
  - gleiche Namen haben (wenn es passt) auch die gleiche Bedeutung

Globs

- wenn man alle Dateien in einem Verzeichnis gleichzeitig bearbeiten will (z. B. alle Kopieren oder Löschen), ist es mühsam alle einzeln aufzulisten
- Lösung: Globs (auch Wildcards genannt)
- spezielle Zeichen, die auf mehrere Zeichen passen
- `?`: ein beliebiges Zeichen (außer `.`)
- `*`: beliebig viele Zeichen (außer `.`)
- `[...]`: die gewünschten Zeichen können angegeben werden (z. B. `[avx]` oder `[a-z]`)

Sonderzeichen in Dateinamen

- bestimmte Zeichen haben für die Shell eine Sonderbedeutung (wie Globs oder das Leerzeichen)
- können nicht einfach so als Dateinamen verwendet werden
- Lösung: Sonderzeichen „escapen“: \ vor das Sonderzeichen stellen
- oder: den Dateinamen in Anführungszeichen setzen
- `ls Komischer\ Date\*name` oder
  `ls "Komischer Date*name"`
- trotzdem nicht zu empfehlen!

Spickzettel

Alle Befehle

Befehl	Optionen
`mv`	`-b`, `-i`
`cp`	`-b`, `-i`, `-r`/`-R`, `-u`
`ln`	`-s`
`ls`	`-i`
`cat`	`-A`
`less`	(`/muster`, `?muster`, `n`, `N`, `q`)
`head`	`-n`
`tail`	`-n`, `-f`

Globs

`?`, `*`, `[...]`