

---

## Klausur

Prüfungsfach: Open-Source Software  
Datum/Uhrzeit: 24. Juli 2008 / 8:30 Uhr  
Raum: J201  
Prüfer: Dr. Hubert Högl  
Dauer: 90 Minuten  
Hilfsmittel: keine

### Hinweise:

1. Dieses Klausurangabenblatt hat auch eine **Rückseite!** Bitte sofort überprüfen.
2. Schreiben Sie bitte nicht auf das Angabenblatt. Verwenden Sie für Ihre Antworten die separat ausgeteilten Bögen. Die Angaben dürfen Sie behalten.
3. Schreiben Sie nicht mit Bleistift.

---

Viel Glück!

---

### Aufgabe 1 (6 Punkte)

Welche technische Infrastruktur würden Sie einem grösseren Open-Source Projekt geben. Machen Sie eine Zusammenfassung von Kapitel 3 des Buches „Producing OSS“ von Karl Fogel.

### Aufgabe 2 (6 Punkte)

Ihnen obliegt die Verwaltung eines neuen offenen Projektes und sie sollen es für die Welt zugänglich machen. Sie wollen aber keine eigene technische Infrastruktur aufbauen.

- (a) Welche Möglichkeiten gibt es in diesem Fall? Nennen Sie **drei** dieser Möglichkeiten.
- (b) Gehen Sie näher auf das Angebot von einem dieser Dienste ein (siehe Artikel von Timo Hoenig).

### Aufgabe 3 (5 Punkte)

Autoconf/Automake (Artikel über Autotools von Peter Novotnik)

- (a) Wie kann man beim Konfigurationsprozess bestimmte Features ein- und ausschalten?
- (b) Was muss man tun damit ein Programm nicht in `/usr/...` installiert wird, sondern in `/home/meinname/local/`?
- (c) Was enthalten `configure.in`, `Makefile.am` und `config.h.in`?
- (d) Wie werden `configure.in`, `Makefile.am` und `Makefile.in` von den Autotools weiterverarbeitet? Zum Veranschaulichen verwenden Sie ein Diagramm mit Pfeilen.

(e) Wie wird das Paketarchiv erstellt?

#### **Aufgabe 4** (10 Punkte)

Verteilte Versionierung mit Git

- (a) Worin unterscheidet sich die Entwicklung mit Subversion (svn) und mit Git grundlegend?
- (b) Wozu gibt es sowohl in svn als auch in git die **branches**?
- (c) Skizzieren Sie wie ein grosses Projekt (z.B. Linux Kernel oder U-Boot Bootloader) mittels Git verwaltet wird. Wie ist das Ganze hierarchisch gegliedert?
- (d) Was bedeutet in diesem Zusammenhang **upstream**? Wieso ist es schlecht, wenn der Upstream-Maintainer wie so oft keine Änderungen bekommt?
- (e) Wie werden Änderungen an der Software genannt und wie werden diese mit Git erstellt?

#### **Aufgabe 5** (4 Punkte)

Beschreiben Sie kurz die Entwicklung des Urheberrechts so wie es im Artikel von Spielkamp (Die Zukunft der Ideen) zu lesen ist.

#### **Aufgabe 6** (4 Punkte)

Beschreiben Sie die **vier Freiheiten**, die einem von der GNU *General Public License* gewährt werden.

#### **Aufgabe 7** (4 Punkte)

GPL-lizenzierter Code darf durchaus Teile von BSD-lizenzierten Code enthalten und das Ergebnis darf wieder unter die GPL gestellt werden. Umgekehrt funktioniert das Ganze leider nicht. Wie könnten Sie es trotzdem erreichen, dass Ihr freier Code von beiden Lagern verwendet werden kann?

#### **Aufgabe 8** (4 Punkte)

Angelehnt an Lizenzen für freie und offene Software verbreiten sich Lizenzen für freie Texte, Bücher, Diplomarbeiten und künstlerische Werke immer weiter. Nennen Sie zwei dieser Lizenzen und beschreiben Sie grob, was diese jeweils leisten.

#### **Aufgabe 9** (6 Punkte)

Man kann mit Open-Source Software durchaus Geld verdienen. Nennen Sie **drei Open-Source Geschäftsmodelle** und beschreiben Sie diese kurz. Geben Sie jeweils eine Firma als Beispiel an.

#### **Aufgabe 10** (4 Punkte)

Die Open-Source Grundidee wird zunehmend in Bereichen angewendet, die nichts mit Software zu tun haben. Nennen Sie zwei dieser Bereiche und erläutern Sie diese kurz.