

# Praktikum angewandte Systemsoftwaretechnik

## Aufgabe 1

Benjamin Oechslein, Daniel Lohmann, Jens Schedel, Michael Gernoth,  
Moritz Strübe, Reinhard Tartler, Timo Hönig

Lehrstuhl Informatik 4

Oktober, 2011

# SSH-Authentifizierung mit Schlüsseln (Host)

- SSH-Authentifizierung mit einem Schlüsselpaar **ohne** Passwort
  - **Privaten** und **öffentlichen** Schlüssel mit `ssh-keygen` erzeugen

```
$ ssh-keygen -f <gruppen_name> -N ""  
Generating public/private rsa key pair.  
Your identification has been saved in <gruppen_name>.  
Your public key has been saved in <gruppen_name>.pub.  
(...)
```

- Erzeugte Schlüssel (`gruppen_name = gruppe0`)

```
$ ls -l  
-rw----- 1 thoenig users 1675  8. Mai 11:29 gruppe0  
-rw-r--r-- 1 thoenig users  394  8. Mai 11:29 gruppe0.pub
```

- Port-Weiterleitung für SSH-Verbindungen (QEMU)

```
$ qemu <...> -redir tcp:'id -u':22
```

- Hinweis: In der Betriebsumgebung des Host-Rechners ausführen.

# SSH-Authentifizierung mit Schlüsseln (VM)

- Installation des SSH-Servers in der virtuellen Maschine

```
$ apt-get install ssh openssh-server
```

- Zugriffe auf die virtuelle Maschine unter Zuhilfenahme des generierten **öffentlichen** Schlüssels
- Hinterlegen des **öffentlichen** Schlüssels

```
$ su - <vm_user>  
$ mkdir .ssh  
$ scp <user>@<host_ip>:~/<gruppen_name>.pub \  
/home/<vm_user>/.ssh/authorized_keys
```

- Alternative: `ssh-copy-id(1)`
- Hinweis: In der Betriebsumgebung der virtuellen Maschine ausführen

# Verbindungsaufbau vom Host zur virtuellen Maschine

- Kontrollverbindung zur virtuellen Maschine aufbauen

```
$ ssh -p 'id -u' \
    -i <gruppen_name> \
    <vm_user>@localhost
```

- Datenverbindung

```
$ scp -P 'id -u' \
    <datei1> (...) \
    <vm_user>@localhost: /<vm_path>
```

- Alternative: SSHFS (benötigt root-Rechte!)

```
$ sshfs -p 'id -u' \
    -o IdentityFile=<absolute_path>/<gruppe_name> \
    <vm_user>@localhost: /<vm_path> \
    <mount_point>
```

- Hinweis bei Verwendung von sshfs(1): Absoluter Pfad zum Schlüssel zwingend notwendig

# Unterschiede zwischen Source-Code

- Grundlegendes Werkzeug um Unterschiede zweier Programm-Code-Revisionen zu erhalten: `diff(1)`
  - zeilenorientiert
  - bevorzugter Modus: „unified diff“
- Beispiel: „Normaler“ Diff

```
$ diff -r a b
diff -r a/linux/drivers/staging/crystalhd/crystalhd_hw.c
      b/linux/drivers/staging/crystalhd/crystalhd_hw.c
87c87
<      * (63MHz * 40us = 0x9D8)
---
>      * (63MHz * 20us = 0x4EC)
89c89
<      crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x9D8);
---
>      crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x4EC);
```

# Unterschiede zwischen Source-Code

- Beispiel: „Vereinheitlichter“ Diff (unified diff)

```
$ diff -ur a b
--- a/linux/drivers/staging/crystalhd/crystalhd_hw.c 16:55
+++ b/linux/drivers/staging/crystalhd/crystalhd_hw.c 16:57
@@ -84,9 +84,9 @@
 /*
  * Bus Arbiter Timeout: GISB_ARBITER_TIMER
- * (63MHz * 40us = 0x9D8)
+ * (63MHz * 20us = 0x4EC)
 */
- crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x9D8);
+ crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x4EC);

/*
 * Decoder clocks: MISC_PERST_DECODER_CTRL
Only in b/linux/drivers/staging/crystalhd: crystalhd-fix.msg
```

# Unterschiede zwischen Source-Code

- Beispiel: „Vereinheitlichter Diff (unified diff), zusätzlich: -N und -p

```
$ diff -urNp a b
--- a/linux/drivers/staging/crystalhd/crystalhd_hw.c 16:55
+++ b/linux/drivers/staging/crystalhd/crystalhd_hw.c 16:57
@@ -84,9 +84,9 @@ static bool crystalhd_bring_out_of_rst(s
 /*
  * Bus Arbiter Timeout: GISB_ARBITER_TIMER
- * (63MHz * 40us = 0x9D8)
+ * (63MHz * 20us = 0x4EC)
 */
- crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x9D8);
+ crystalhd_reg_wr(adp, GISB_ARBITER_TIMER, 0x4EC);

 /*
  * Decoder clocks: MISC_PERST_DECODER_CTRL
diff -urNp a/linux/drivers/staging/crystalhd/crystalhd-fix.msg
      b/linux/drivers/staging/crystalhd/crystalhd-fix.msg
--- a/linux/drivers/staging/crystalhd/crystalhd-fix.msg 01:00
+++ b/linux/drivers/staging/crystalhd/crystalhd-fix.msg 17:10
@@ -0,0 +1 @@
+Initial patch description.
```

# Diffs gezielt durchsuchen

- Funktioniert auch mit gepackten Diffs
- Kann führende Unterverzeichnisse überspringen

```
$ lsdiff -z --strip=1 big-diff.gz | grep ^tools
tools/perf/.gitignore
tools/perf/CREDITS
tools/perf/Documentation/Makefile
tools/perf/design.txt
tools/perf/perf.c
tools/perf/perf.h
[...]
```



# Diffs gezielt durchsuchen

- Funktioniert auch mit gepackten Diffs
- Kann führende Unterverzeichnisse überspringen

```
$ lsdiff -z --strip=1 big-diff.gz | grep ^tools
tools/perf/.gitignore
tools/perf/CREDITS
tools/perf/Documentation/Makefile
tools/perf/design.txt
tools/perf/perf.c
tools/perf/perf.h
[...]
```

- Änderungen auf Unterverzeichnisse beschränken:

```
$ filterdiff -z --strip=1 -i '*/tools/perf/*' big-diff.gz | lsdiff
tools/perf/.gitignore
tools/perf/CREDITS
tools/perf/Documentation/Makefile
tools/perf/Documentation/asciidoc.conf
[...]
```

# Größe von Diffs ermitteln

- Ausgabe aus dem Kommando `diff` kann direkt weiterverwendet werden:

```
$ diff -Nurp linux-2.6.18/ linux-2.6.32/ | diffstat
.gitignore | 36
.mailmap | 107
CREDITS | 367
Documentation/00-INDEX | 240
[...]
virt/kvm/ioapic.c | 362
virt/kvm/ioapic.h | 76
virt/kvm/iodev.h | 70
virt/kvm/iommu.c | 233
virt/kvm/irq_comm.c | 422
virt/kvm/kvm_main.c | 2830 +
36994 files changed, 7819725 insertions(+), 3043036 deletions(-)
```

# Extra-Aufgabe: Mainline- vs. Distributions-Kernel

- Distributions-Kernel weichen häufig vom Mainline-Kernel ab:
  - Backports
  - Fixes
  - Branding
- **Aufgabe:** *Analyse eines Distributions-Kernels im Vergleich zum Mainline-Kernel.*
- Mainline: Linux **2.6.32**
- Distributionen: Debian, Gentoo, Red Hat, SUSE, Ubuntu
- Kriterien:
  - Anzahl der Patches
  - Bezugsquelle, evtl. zusätzliche benötigte Werkzeuge
  - Umfang der Änderungen zum Mainline-Kernel
  - Bewertung der Dokumentation der Patches
  - Existenz von Details über Auswirkungen der einzelnen Patches
- Pro Gruppe maximal drei Folien (etwa zehn Minuten Referat)