Übungen zu Foliensatz G

1. Starten Sie die virtuelle Linux-Maschine. Suchen Sie dann nach eingebauten bzw. angeschlossenen PCI- und USB-Geräten.
2. Identifizieren Sie die numerische ID der im virtuellen PC eingebauten Grafikkarte (in der Form `0123:4567`) und googeln Sie nach dieser ID; Sie sollten einen Treffer auf der Seite http://pci-ids.ucw.cz/read/PC finden.
3. Es gibt zwei Geräte mit der Herstellerbezeichnung „InnoTek Systemberatung GmbH“ (der ursprüngliche Anbieter von VirtualBox). Bei den Geräte-IDs hat InnoTek sich damals einen Spaß erlaubt, den man auch in aktuellen Versionen von VirtualBox noch erkennen kann – welchen?
4. Prüfen Sie auf zwei Arten, welche Kernel-Version Ihr Linux-System verwendet.
5. Der Ordner `/lib/modules` hat auf Ihrer virtuellen Maschine nur ein einziges Unterverzeichnis (`2.6.32-5-686`) für die installierte Kernel-Version; im Allgemeinen liegen dort mehrere Ordner. Das richtige Modulverzeichnis für den laufenden Kernel finden Sie immer bequem mit dem Befehl

   ```
   echo /lib/modules/$(uname -r)
   ```

   heraus – probieren Sie aus, dass das funktioniert. Die Konstruktion `$(...)` bewirkt übrigens, dass der in den Klammern stehende Befehl ausgeführt und der `$(...)`-Ausdruck durch die Ausgabe dieses Befehls ersetzt wird.
6. Die Ausgabe von

   ```
   cat /proc/cpuinfo
   ```

   ist sehr umfangreich. Finden Sie heraus, welche Bedeutungen die Felder `stepping`, `bogomips` und `address sizes` haben.
7. Vergleichen Sie die Ausgaben von `fdisk -l` und `cat /proc/partitions`. Was fällt Ihnen bei der Blockgröße der erweiterten Partition auf?
8. Das auf den Folien erwähnte Verzeichnis `/proc/scsi` gibt es auf Ihrer virtuellen Maschine nicht. Über eine Google-Suche nach `/proc/scsi` und der Kernel-Version (nur in der Kurzform 2.6.x) finden Sie heraus, woran das liegt. Wo Sie alternativ Informationen über das SCSI-Subsystem finden können, beantwortet die folgende Aufgabe.
9. Installieren Sie die Pakete `strace` und `lsscsi` nach. Das Tool `lsscsi` listet (analog zu `lspci`, `lsusb`) alle SCSI-Geräte auf, während `strace` die Ausführung eines Programmes beobachtet und alle System Calls anzeigt. Mit dem Befehl

   ```
   strace lsscsi 2>&1 | grep open | less
   ```

   können Sie sich die Liste der Dateien anzeigen lassen, die `lsscsi` öffnet. Falls Ihnen das nicht gelingt (z. B.: kein Internet) – die ersten Zeilen der Ausgabe sehen so oder ähnlich aus:

   ```
   open("/etc/ld.so.cache", O_RDONLY)      = 3
   open("/lib/i386-linux-gnu/libc.so.6", O_RDONLY) = 3
   open("/proc/mounts", O_RDONLY)          = 3
   open("/sys/bus/scsi/devices", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DI-
   RECTORY|O_CLOEXEC) = 3
   open("/sys/bus/scsi/devices/2:0:0:0/type", O_RDONLY) = 3
   open("/sys/bus/scsi/devices/2:0:0:0/vendor", O_RDONLY) = 3
   ```

   Wie findet man also etwas über das SCSI-Subsystem heraus, wenn es `/proc/scsi` nicht gibt?
10. Lassen Sie sich die Liste der geladenen Kernel-Module anzeigen und finden Sie für drei der Module (mit `modinfo`, nicht durch Googeln) heraus, welche Funktion sie haben.
11. Ermitteln Sie den vollen Pfad zu den Programmen `ls`, `fdisk` und `vi`. Wo liegen die zugehörigen Manpages? (Wieder ist der volle Pfad gesucht.)

12. Wenn Sie auf gleiche Weise wie in Aufgabe 11 nach `cd` suchen, werden Sie nichts finden. Wie klären Sie, woran das liegt?
13. Suchen Sie mit `find` alle Dateien auf Ihrem virtuellen Linux-System, die weniger als einen Tag alt sind. An einem Blick in die Manpage führt kein Weg vorbei: Die Möglichkeit einer solchen Suche wurde auf den Folien zwar erwähnt, aber kein passender `find`-Aufruf gezeigt.
14. Erstellen Sie in `/tmp` zwei Dateien mit Leerzeichen, z. B. mit dem Befehl

    ```
    touch "/tmp/Datei 1.txt" "/tmp/Datei 2.txt"
    ```

    (Die Anführungszeichen sind wichtig, damit `touch` erkennt, wo ein Dateiname anfängt und aufhört.) Suchen Sie nun mit find im Ordner `/tmp` nach Dateien, welche die Dateiendung `.txt` haben, und leiten Sie die Ausgabe über eine Pipe an `xargs` weiter, das die Dateien mit `ls -l` anzeigen soll – der Befehl

    ```
    find /tmp -name '*.txt' | xargs ls -l
    ```

    wird aber nicht funktionieren. Warum erhalten Sie eine Fehlermeldung und wie beheben Sie das Problem?
15. Lösen Sie die Teilaufgabe mit Suche/Ausgabe aus der vorangehenden Aufgabe ohne die Weiterleitung an `xargs`. Das Problem aus der alten Aufgabe gibt es dann nicht.
16. Sie haben bei `find` die Aufrufoptionen `-exec ... \;` und `-exec ... \+` kennengelernt. Warum ist es in den meisten Fällen besser, die Variante mit dem Pluszeichen am Ende zu verwenden?
17. Warum unterscheidet Linux zwischen ganz wichtigen und weniger wichtigen Programmen und Bibliotheken? (Die wichtigen liegen in `/bin`, `/sbin` und `/lib`, die weniger wichtigen in `/usr/bin`, `/usr/sbin` und `/usr/lib`.)
18. Wie können Sie mit `find` nach leeren Dateien suchen? Suchen Sie alle solchen Dateien auf Ihrem System; Sie sollten dabei u. a. die in Aufgabe 14 mit `touch` erzeugten Dateien finden.
19. Zur Erinnerung: Sie haben `touch` in dieser Aufgabe benutzt, um leere neue Dateien anzulegen – was ist die eigentliche Hauptfunktion von `touch`?
20. Bei der Besprechung von `locate` haben Sie gesehen, dass es i. d. R. nicht möglich ist, damit Dateien zu finden, auf die man selbst nicht zugreifen darf. Nennen Sie zwei Gründe, aus denen dieses Verhalten sinnvoll ist.
21. OpenSuse löst die Aufgabe, Dateien in der `locate`-Ausgabe zu verstecken (siehe Aufgabe 20), indem es den `updatedb`-Prozess mit den Rechten des Benutzers `nobody` laufen lässt. Debian hinggen verwendet das `mlocate`-Paket, das mit Root-Rechten nach Dateien sucht, um die Datenbank aufzubauen, und erst beim `locate`-Aufruf prüft, welche Dateien es anzeigen darf. Warum ist der Ansatz des Debian-Pakets besser?