Fachhochschule Nordwestschweiz
Hochschule für Technik

Modul Betriebssysteme (bsys-iC)

Feedback aus der Hausaufgabe

- Läuft Linux MINT in ihrer VirtualBox?
- Gab es grundlegende neue Erkenntnisse?
- Was hat gefehlt? Gab es Probleme?
- Wieviel Zeit haben Sie aufgewendet?

Lektion 10: System-Start/Stopp, Boot Manager, Hardware Abstraction Layer

Inhalt

- Prozeduren und Optionen zum Starten und Stoppen eines Linux-Systems
- Run Levels, Single User / Multi User Modus, Inhalt und Bedeutung der rc-Dateien im /etc Directory
- Funktionalität und Benutzung des Boot Managers
- BIOS/UEFI
- Der Hardware Abstraction Layer

Motivation

Starten und Stoppen eines Systems

- Startsequenz:
  - Rechner-Hardware mit „Mini-OS“ (z.B. BIOS), ggf. mit Profilen
  - Auswahl durch den Benutzer oder per Default
  - Bootbarer Betriebssystem-Kern (Master Boot Record) und Prozess 0 / 1
  - Single User Modus / abgesicherter Modus
  - Multi User Modus (mehrstufig, z.B. mit oder ohne Netzwerk-Dienste usw.)
- Stoppsequenz (Multi User zu Single User oder Vollstopp):
  - Benutzer informieren
  - Aktive Dienste abmelden und beenden
  - Laufende Prozesse terminieren, offene Dateien schliessen etc.
  - Benutzer ausschliessen
  - Disk / Caches synchronisieren, Peripherie deaktivieren
  - Shutdown-Befehl an die Rechner-Hardware

Prozeduren und Optionen zum Starten eines Linux-Systems

- Bei manchen Unix-/Linux-Systemen wird zunächst ein Mini-Linux in den Hauptspeicher geladen, das dann den Benutzer zur Eingabe der nächsten Aktionen und Optionen auffordert, z.B. bei Knoppix.

- Ubuntu Linux bootet ebenfalls in Stufen, aber für den Benutzer nicht unterbrechbar direkt in den Multi User Modus / das GUI.

Prozeduren und Optionen zum Stoppen eines Linux-Systems

- Bei Unix-/Linux-Systemen mit Benutzer-Zugang zu den Boot-Optionen kann meist auch der Systemstopp in mehreren Teilschritten vorgenommen werden.
- Ubuntu Linux führt einen Stopp- oder Reboot-Befehl nach einer Rückfrage zum Benutzer direkt aus.
- Ein Unix-/Linux-System kann auch direkt durch das manuelle Terminieren des Prozesses 0 bzw. 1 (init bzw. upstart) gestoppt werden („kill -1 1“), dies kann jedoch zu Datenverlust etc. führen.

Run Levels, Single User / Multi User Modus I

- Viele Betriebssysteme durchlaufen beim Start (Booten) mehrere abgestufte Systemzustände, bzw. starten in einen bestimmten Zustand, den Runlevel.
- Jedem Runlevel sind bestimmte System-Dienste zugeordnet, die beim Booten als Prozesse in definierter Reihenfolge innerhalb des Betriebssystems gestartet werden. Auf diese Weise werden Betriebsmittel des Computers stufenweise in Betrieb genommen.
- Bei Beendigung des Betriebssystems (Shutdown) werden die Runlevel in umgekehrter Reihenfolge durchlaufen, die laufenden Prozesse werden stufenweise beendet, bis der Computer ausgeschaltet werden kann. Daneben kann auch direkter Wechsel von einem Runlevel in einen anderen erfolgen. [Wikipedia]

Run Levels, Single User / Multi User Modus II

- *Runlevel* kennt man vor allem aus den unterschiedlichen Unix- und Linux-Systemen. Doch auch in Windows entsprechen die Startoptionen *Abgesicherter Modus*, *Abgesicherter Modus mit Netzwerk* oder *Windows normal Starten* im Windows-Bootmenü jeweils genau einem *Runlevel*.

- Idee der unterschiedlichen Runlevel ist es, abgestufte Sicherheitsstufen bereitzustellen, in denen nur bestimmte Systemprozesse aktiv sind. Dies ist wichtig, falls beispielsweise ein System von Computerviren befallen ist und ohne Netzwerk-Anbindung laufen soll. [Wikipedia]

Definition von Linux Run Levels

- Alle Systemdienste, die innerhalb eines bestimmten Runlevels gestartet werden sollen, werden in der Datei /etc/inittab festgelegt (der genaue Ort kann je nach Linux-“Dialekt“ variieren, z.B. /etc/rc.local bzw. /etc/init.d/rc.local in Ubuntu).
- Diese Datei wird von dem Systemprozess „init“ interpretiert.

„Typische“ Linux Run Levels

Runlevel	Beschreibung
0	Shutdown. Alle Netzverbindungen werden geschlossen, Dateipuffer werden geschrieben, Mounts auf Partitionen werden entfernt (d. h. die im Verzeichnisbaum eingebundenen Datenträger werden ausgehangen).
S	Single-User-Runlevel; niedrigster Systemzustand für Wartungsarbeiten, in dem ausschließlich Systemressourcen wie Festplatten oder Dateisysteme aktiv sind.
1	Einzelnutzerbetrieb ohne Netzwerk mit ausschließlich lokalen Ressourcen. In vielen Implementationen identisch mit 'S'.
2	Lokaler Mehrnutzerbetrieb ohne Netzwerk mit ausschließlich lokalen Ressourcen. Unter einigen Linuxdistributionen (z. B. Debian) wird in Runlevel 2 auch das Netzwerk konfiguriert.
3	Netzwerkbetrieb, über das Netzwerk erreichbare Ressourcen sind nutzbar, eine grafische Oberfläche steht nicht zur Verfügung. Firewalls sollten aktiviert werden.
4	Ist normalerweise nicht definiert. Kann aber für diverse Dienste genutzt werden.
5	Wie 3, zusätzlich wird die grafische Oberfläche bereitgestellt.
6	Reboot. Alle Netzverbindungen werden geschlossen, Dateipuffer werden geschrieben, Mounts auf die Partitionen werden entfernt.

[Wikipedia]

Wechseln des Run Level

- telinit <<runlevel>>
- shutdown / halt / poweroff / reboot
- Default Run Level: /etc/inittab, /etc/rc.local oder /etc/init/rc-sysinit.conf (Ubuntu)

telinit

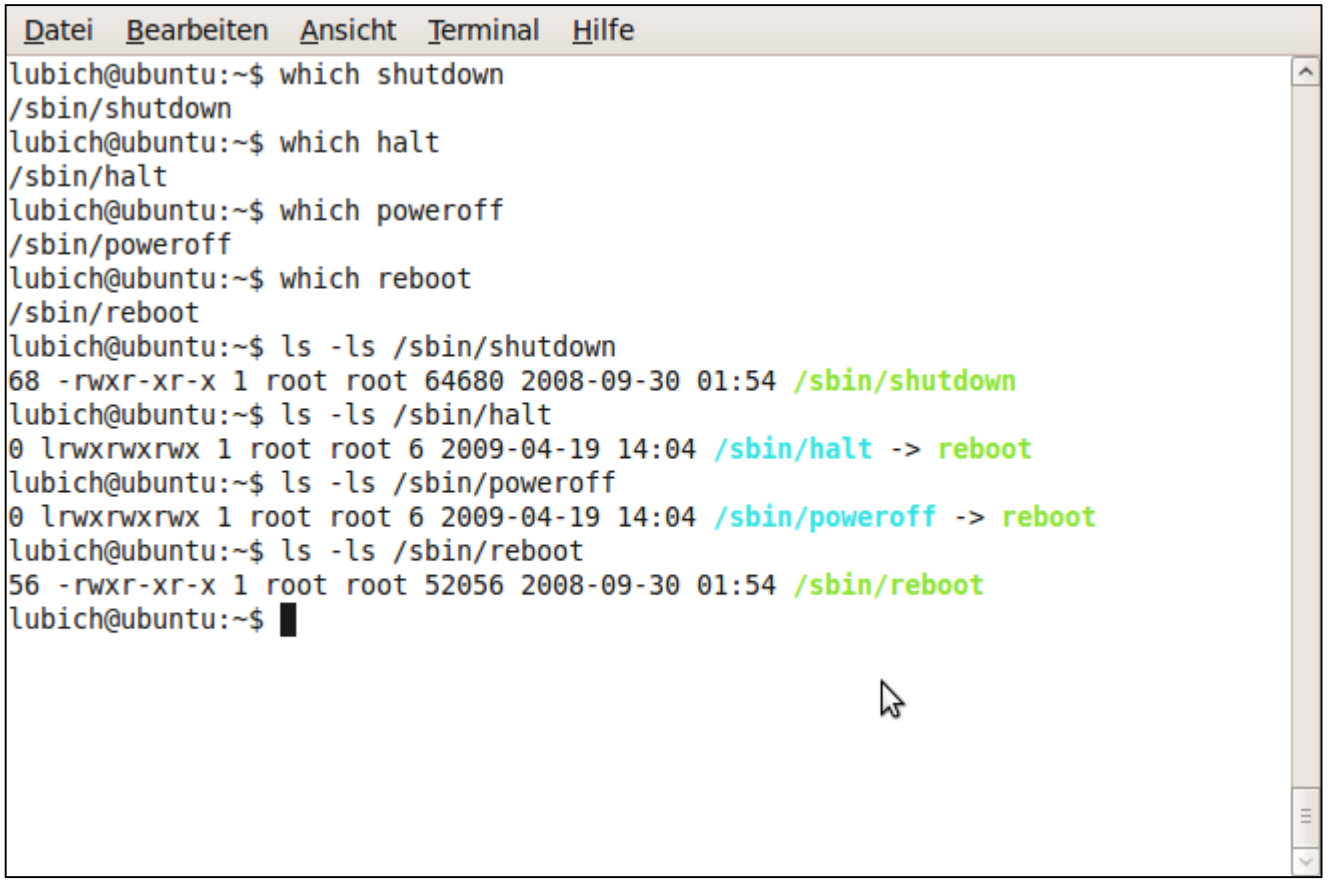
```
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
lubich@ubuntu:/etc$ telinit --help
Usage: telinit [OPTION]... RUNLEVEL
Change runlevel.

Options:
  -q, --quiet                reduce output to errors only
  -v, --verbose              increase output to include informational messages
      --help                 display this help and exit
      --version              output version information and exit

RUNLEVEL should be one of 0123456S.

Report bugs to <upstart-devel@lists.ubuntu.com>
lubich@ubuntu:/etc$
```

Shutdown & Friends




Ein offenes Betriebssystem kann schon mal mutieren. Bei Windows 2000 hingegen gibt es alle Services und Dienste aus einer Hand. Das spart Zeit und somit wirklich Geld. Mehr Infos unter www.microsoft.com/germany/windows2000
Microsoft
ein offenes betriebssyst em hat nicht nur vorteile

Upstart (Ubuntu)

- Ordner /etc/init
  enthält die migrierten Job Definition Files

- File /etc/init/rc-sysinit.conf
  löst SysV-init Model aus, d.h. Wechsel in die /etc/init.d Skripts aus (-> telinit)

- Nur in Ubuntu, Durchsetzung zweifelhaft

Inhalt und Bedeutung der /etc/rc-Dateien I

- Jeder definierte Run Level hat im /etc Directory ein Subdirectory rc*x*.d, wobei *x* die Run Level Nummer angibt.
- In jedem Run Level Directory sind Scripte pro Service abgespeichert (bzw. Links auf die Scripts in /etc/init.d) – jedes Script bietet mindestens die Aktionen „start“ und „stop“ an.
- Das Kommando „runlevel“ zeigt den letzten sowie den aktuellen Run Level an.

Inhalt und Bedeutung der /etc/rc-Dateien II

```
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
lubich@ubuntu:/etc/rc1.d$ ls -ls
insgesamt 4
0 lrwxrwxrwx 1 root root  13 2009-04-19 14:04 K01gdm -> ../init.d/gdm
0 lrwxrwxrwx 1 root root  17 2009-04-19 14:04 K02usplash -> ../init.d/usplash
0 lrwxrwxrwx 1 root root  17 2009-04-19 14:04 K11anacron -> ../init.d/anacron
0 lrwxrwxrwx 1 root root  13 2009-04-19 14:04 K11atd -> ../init.d/atd
0 lrwxrwxrwx 1 root root  14 2009-04-19 14:04 K11cron -> ../init.d/cron
0 lrwxrwxrwx 1 root root  20 2009-04-29 07:25 K15pulseaudio -> ../init.d/pulseau
dio
0 lrwxrwxrwx 1 root root  13 2009-04-19 14:04 K16hal -> ../init.d/hal
0 lrwxrwxrwx 1 root root  22 2009-04-19 14:04 K20acpi-support -> ../init.d/acpi-
support
0 lrwxrwxrwx 1 root root  16 2009-04-19 14:04 K20apport -> ../init.d/apport
0 lrwxrwxrwx 1 root root  22 2009-04-19 14:04 K20hotkey-setup -> ../init.d/hotke
y-setup
0 lrwxrwxrwx 1 root root  19 2009-04-19 14:04 K20powernowd -> ../init.d/powernow
d
0 lrwxrwxrwx 1 root root  15 2009-04-29 07:23 K20rsync -> ../init.d/rsync
0 lrwxrwxrwx 1 root root  15 2009-04-29 07:24 K20saned -> ../init.d/saned
0 lrwxrwxrwx 1 root root  17 2009-05-07 16:20 K20winbind -> ../init.d/winbind
0 lrwxrwxrwx 1 root root  15 2009-04-19 14:04 K21acpid -> ../init.d/acpid
0 lrwxrwxrwx 1 root root  20 2009-04-29 07:25 K31atieventsd -> ../init.d/atieven
tsd
0 lrwxrwxrwx 1 root root  13 2009-04-19 14:04 K39ufw -> ../init.d/ufw
0 lrwxrwxrwx 1 root root  31 2009-04-29 07:20 K70system-tools-backends -> ../ini
t.d/system-tools-backends
0 lrwxrwxrwx 1 root root  19 2009-04-19 14:04 K74bluetooth -> ../init.d/bluetoot
h
0 lrwxrwxrwx 1 root root  14 2009-04-19 14:04 K80cups -> ../init.d/cups
0 lrwxrwxrwx 1 root root  22 2009-04-29 07:25 K86avahi-daemon -> ../init.d/avahi
-daemon
0 lrwxrwxrwx 1 root root  14 2009-04-19 14:04 K88dbus -> ../init.d/dbus
0 lrwxrwxrwx 1 root root  15 2009-04-19 14:04 K89klogd -> ../init.d/klogd
0 lrwxrwxrwx 1 root root  18 2009-04-19 14:04 K90sysklogd -> ../init.d/sysklogd
0 lrwxrwxrwx 1 root root  21 2009-04-19 14:04 K99laptop-mode -> ../init.d/laptop
-mode
0 lrwxrwxrwx 1 root root  19 2009-04-19 14:04 K99policykit -> ../init.d/policyki
t
4 -rw-r--r-- 1 root root 369 2009-03-31 11:01 README
0 lrwxrwxrwx 1 root root  19 2009-04-19 14:04 S30killprocs -> ../init.d/killproc
s
0 lrwxrwxrwx 1 root root  21 2009-04-29 07:23 S70bootlogs.sh -> ../init.d/bootlo
gs.sh
0 lrwxrwxrwx 1 root root  19 2009-04-29 07:23 S70dns-clean -> ../init.d/dns-clea
n
0 lrwxrwxrwx 1 root root  18 2009-04-29 07:23 S70pppd-dns -> ../init.d/pppd-dns
0 lrwxrwxrwx 1 root root  16 2009-04-19 14:04 S90single -> ../init.d/single
lubich@ubuntu:/etc/rc1.d$
```

Ein rc Script

```
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
lubich@ubuntu:/etc/init.d$ more reboot
#! /bin/sh
### BEGIN INIT INFO
# Provides:          reboot
# Required-Start:
# Required-Stop:
# Default-Start:
# Default-Stop:      6
# Short-Description: Execute the reboot command.
# Description:
### END INIT INFO

PATH=/sbin:/usr/sbin:/bin:/usr/bin

. /lib/lsb/init-functions

do_stop () {
        # Message should end with a newline since kFreeBSD may
        # print more stuff (see #323749)
        log_action_msg "Will now restart"
        reboot -d -f -i
}

case "$1" in
  start)
        # No-op
        ;;
  restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
  stop)
        do_stop
        ;;
  *)
        echo "Usage: $0 start|stop" >&2
        exit 3
        ;;
esac
lubich@ubuntu:/etc/init.d$
```

Services bewirtschaften

- Service-Verzeichnis: /etc/init.d
- Service zufügen:
  - update-rc.d <<Service>> default
- Service löschen:
  - update-rc.d <<Service>> remove, sofern der Service in /etc/init.d schon gelöscht wurde.

→ update-rc.d = Perl-Script + man-Page

Pause

Übung
(ca. 30 min.)

- Aufgabe(n) gemäss separatem Aufgabenblatt
- Lösungsansatz: Einzelarbeit oder Gruppen von max. 3 Personen
- Hilfsmittel: beliebig
- Besprechung möglicher Lösungen in der Klasse (es gibt meist nicht die eine «Musterlösung»)

Übungsbesprechung (ca. 15 min.)

- Stellen Sie Ihre jeweilige Lösung der Klasse vor.
- Zeigen Sie auf, warum ihre Lösung korrekt, vollständig und effizient ist.
- Diskutieren Sie ggf. Design-Entscheide, Alternativen oder abweichende Lösungsansätze.
- Gibt es Unklarheiten? Stellen Sie Fragen.

Pause

Meldungen beim Systemstart und -stopp

```
Datei  Bearbeiten  Ansicht  Terminal  Hilfe
lubich@ubuntu:/var/log$ ls
apparmor          debug.3.gz          kern.log.3.gz     syslog.5.gz
apt               dist-upgrade        lastlog           syslog.6.gz
auth.log          dkms_autoinstaller  lpr.log           udev
auth.log.0        dmesg               mail.err          unattended-upgrades
auth.log.1.gz     dmesg.0             mail.info         user.log
auth.log.2.gz     dmesg.1.gz          mail.log          user.log.0
auth.log.3.gz     dmesg.2.gz          mail.warn         user.log.1.gz
boot              dmesg.3.gz          messages          user.log.2.gz
bootstrap.log     dmesg.4.gz          messages.0        user.log.3.gz
btmp              dpkg.log            messages.1.gz     wpa_supplicant.log
btmp.1            dpkg.log.1          messages.2.gz     wpa_supplicant.log.1.gz
ConsoleKit        dpkg.log.2.gz       messages.3.gz     wpa_supplicant.log.2.gz
cups              dpkg.log.3.gz       news              wpa_supplicant.log.3.gz
daemon.log        faillog             pm-powersave.log  wpa_supplicant.log.4.gz
daemon.log.0      fontconfig.log      pycentral.log     wpa_supplicant.log.5.gz
daemon.log.1.gz   fsck                samba             wtmp
daemon.log.2.gz   gdm                 syslog            wtmp.1
daemon.log.3.gz   installer           syslog.0          wvdialconf.log
debug             kern.log            syslog.1.gz       Xorg.0.log
debug.0           kern.log.0          syslog.2.gz       Xorg.0.log.old
debug.1.gz        kern.log.1.gz       syslog.3.gz
debug.2.gz        kern.log.2.gz       syslog.4.gz
lubich@ubuntu:/var/log$
```

Meldungen beim Systemstart

```
Jul 13 14:09:24 ubuntu syslogd 1.5.0#5ubuntu3: restart.
Jul 13 14:09:24 ubuntu kernel: Inspecting /boot/System.map-2.6.28-13-generic
Jul 13 14:09:24 ubuntu kernel: Cannot find map file.
Jul 13 14:09:24 ubuntu kernel: Loaded 67962 symbols from 180 modules.
Jul 13 14:09:24 ubuntu kernel: [    0.000000] BIOS EBDA/lowmem at:
0009fc00/0009fc00
Jul 13 14:09:24 ubuntu kernel: [    0.000000] Initializing cgroup subsys cpuset
Jul 13 14:09:24 ubuntu kernel: [    0.000000] Initializing cgroup subsys cpu
Jul 13 14:09:24 ubuntu kernel: [    0.000000] Linux version 2.6.28-13-generic
(buildd@yellow) (gcc version 4.3.3 (Ubuntu 4.3.3-5ubuntu4) ) #45-Ubuntu SMP Tue
Jun 30 22:12:12 UTC 2009 (Ubuntu 2.6.28-13.45-generic)
Jul 13 14:09:24 ubuntu kernel: [    0.000000] Command line:
root=UUID=46A6C65FA6C64F5D loop=/ubuntu/disks/root.disk ro ROOTFLAGS=syncio
quiet splash
Jul 13 14:09:24 ubuntu kernel: [    0.000000] KERNEL supported cpus:
Jul 13 14:09:24 ubuntu kernel: [    0.000000]   Intel GenuineIntel
Jul 13 14:09:24 ubuntu kernel: [    0.000000]   AMD AuthenticAMD
Jul 13 14:09:24 ubuntu kernel: [    0.000000]   Centaur CentaurHauls
Jul 13 14:09:24 ubuntu kernel: [    0.000000] BIOS-provided physical RAM map:
Jul 13 14:09:24 ubuntu kernel: [    0.000000]  BIOS-e820: 0000000000000000 -
000000000009fc00 (usable)
Jul 13 14:09:24 ubuntu kernel: [    0.000000]  BIOS-e820: 000000000009fc00 -
00000000000a0000 (reserved)
Jul 13 14:09:24 ubuntu kernel: [    0.000000]  BIOS-e820: 00000000000ef000 -
0000000000100000 (reserved)
Jul 13 14:09:24 ubuntu kernel: [    0.000000]  BIOS-e820: 0000000000100000 -
00000000bae4c000 (usable)
```

Meldungen beim Systemstopp

```
Jul 13 10:07:38 ubuntu console-kit-daemon[2978]: GLib-GObject-WARNING:
IA__g_object_get_valist: value location for `gchararray' passed as NULL
Jul 13 10:07:38 ubuntu console-kit-daemon[2978]: GLib-GObject-WARNING:
IA__g_object_get_valist: value location for `gchararray' passed as NULL
Jul 13 10:07:38 ubuntu init: tty4 main process (2640) killed by TERM signal
Jul 13 10:07:38 ubuntu init: tty5 main process (2641) killed by TERM signal
Jul 13 10:07:38 ubuntu init: tty3 main process (2649) killed by TERM signal
Jul 13 10:07:38 ubuntu init: tty6 main process (2650) killed by TERM signal
Jul 13 10:07:38 ubuntu init: tty1 main process (3518) killed by TERM signal
Jul 13 10:07:38 ubuntu init: tty2 main process (2648) killed by TERM signal
Jul 13 10:07:38 ubuntu console-kit-daemon[2978]: GLib-GObject-WARNING:
IA__g_object_get_valist: value location for `gchararray' passed as NULL
Jul 13 10:07:38 ubuntu console-kit-daemon[2978]: GLib-GObject-WARNING:
IA__g_object_get_valist: value location for `gchararray' passed as NULL
Jul 13 10:07:43 ubuntu bluetoothd[3145]: bridge pan0 removed
Jul 13 10:07:43 ubuntu bluetoothd[3145]: Stopping SDP server
Jul 13 10:07:43 ubuntu bluetoothd[3145]: Exit
Jul 13 10:07:43 ubuntu exiting on signal 15
```

systemd

- systemd ist ein Hintergrundprogramm (Daemon) für Linux-Systeme, das als init-Prozess als erster Prozess (Prozess-ID 1) zum Starten, Überwachen und Beenden weiterer Prozesse dient.
- Systemd ist abwärtskompatibel zu SysVinit-Skripten. Allerdings werden bewusst Features benutzt, die nur unter Linux zur Verfügung stehen, nicht aber auf anderen unixoiden Betriebssystemen. Es kann daher nur auf Systemen mit Linux-Kernel laufen.

https://de.wikipedia.org/wiki/Systemd

systemd

- systemd soll den gegenseitigen Abhängigkeiten von Prozessen besser gerecht werden, durch mehr Parallelisierung zu einer besseren Auslastung beim Systemstart führen und somit weniger Verzögerungen verursachen als das ältere SysVinit oder das inzwischen auch von Ubuntu aufgegebene Upstart.
- Grundlegendes Konzept dafür ist es, weitgehend alle Prozesse gleichzeitig zu starten. Um nicht, wie bei anderen zwar grundsätzlich auf Parallelisierung setzenden Systemen, anhand der in einem Modell erfassten wechselseitigen Abhängigkeiten der Prozesse teilweise noch mit Serialisierung zu arbeiten, werden die D-Bus-Verbindungen und Sockets zur Interprozesskommunikation schon vor dem Start des zugehörigen Dienstes bereitgestellt und vom Kernel eventuell auflaufende Nachrichten bis zur Bereitschaft des Dienstes gepuffert.

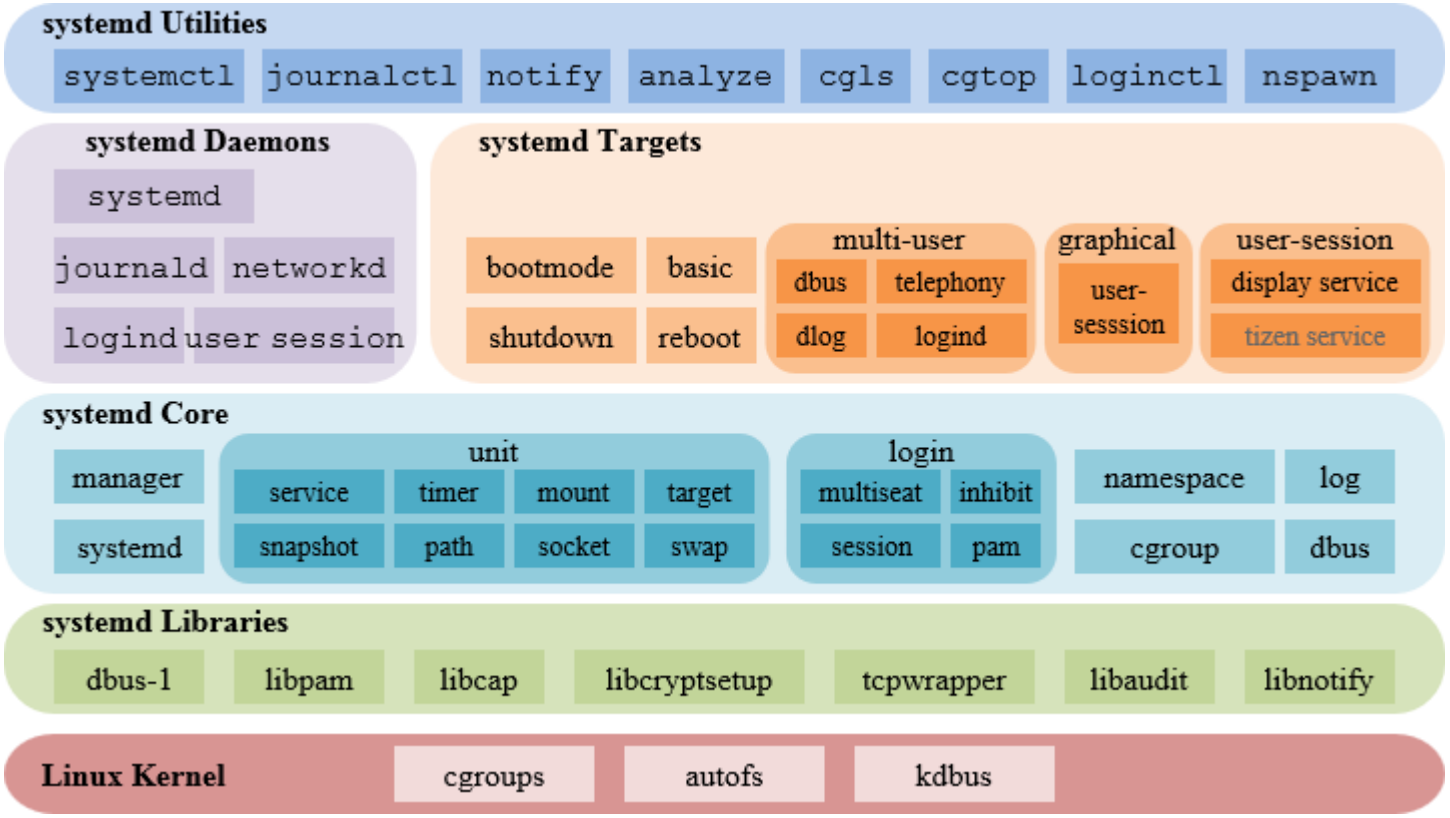
https://de.wikipedia.org/wiki/Systemd

systemd

- Daneben kann es nur gelegentlich benötigte Dienste ereignisbasiert erst bei Bedarf starten und so beim Systemstart weniger Dienste starten. Damit nimmt es Aufgaben wahr, die bei klassischen Unix-Systemen von inetd übernommen werden.
- Weiterhin sollen alle Shell-Boot-Skripte durch deklarative Konfigurationsdateien ersetzt werden, in denen definiert wird, wie die jeweiligen Dienste gestartet werden. Diese Dateien sind in der Regel deutlich einfacher zu schreiben als init-Skripte und vermeiden den erheblichen Overhead von Shell-Skripten.

https://de.wikipedia.org/wiki/Systemd

Aber:



https://de.wikipedia.org/wiki/Systemd

Aber:

- Systemd polarisiert die Community und wird von den Gegnern teils äußerst stark kritisiert. Die Kritik vermischt sich dabei mit Kritik an den Entwicklern, insbesondere Poettering. Die Diskussion, ob man in Debian weiter SysV-Init verwenden oder auf systemd oder aber ein anderes Init-System umsteigen sollte, führte zu monatelangen Streitereien und schließlich zu einer Abstimmung („General Resolution“),zahlreichen Rücktritten und einem Fork unter dem Namen Devuan.

https://de.wikipedia.org/wiki/Systemd

Aber:

- Der Hauptkritikpunkt an systemd liegt in seinem Anspruch, deutlich mehr verschiedene Aufgaben als das alte SysV-Init erledigen zu wollen, was es recht kompliziert und fehleranfällig mache und überdies die Unix-Philosophie verletze, dass ein Programm ein Problem gut lösen sollte, anstatt viele Aufgaben schlecht zu lösen.
- Außerdem wird vielfach bemängelt, dass systemd Log-Dateien im Binärformat und nicht als einfache Textdateien speichert. Ein weiterer Kritikpunkt besteht in der Entscheidung, systemd explizit nur für Linux zu entwickeln

https://de.wikipedia.org/wiki/Systemd

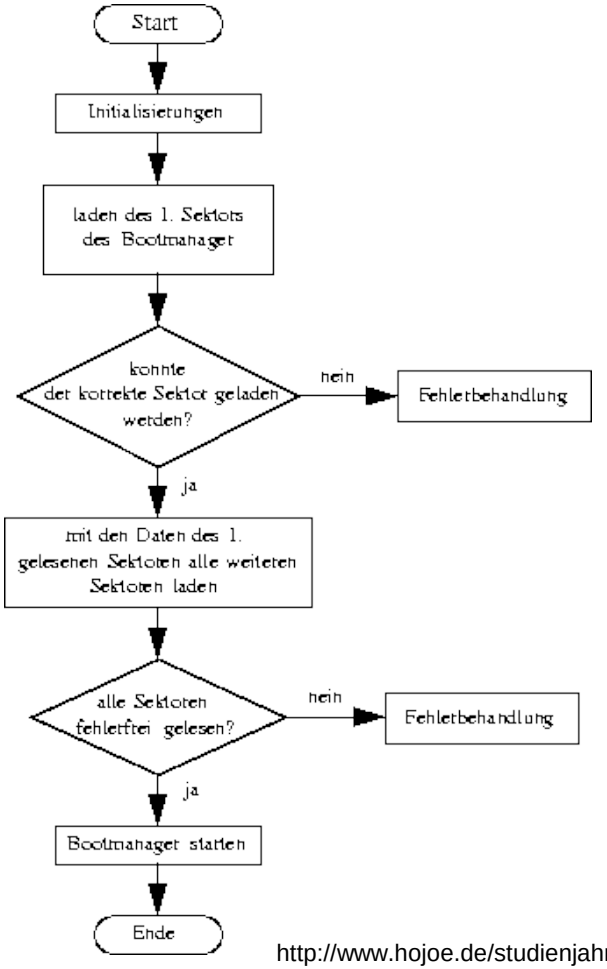
Boot Loader & Manager

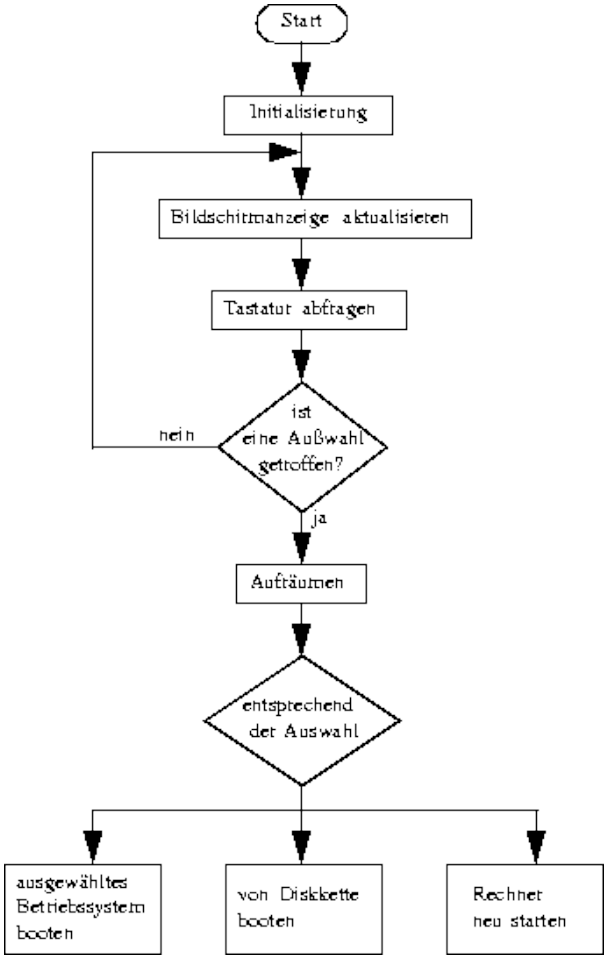
- Jedes Betriebssystem, welches alternative Betriebssystem-Installationen erlaubt, hat seinen eigenen Boot-Manager, dazu kommen diverse generische Boot Manager.
- Master Boot Record: bei BIOS-basierten Computern der erste Datenblock eines in Partitionen aufgeteilten Speichermediums, wie beispielsweise einer Festplatte. Der MBR enthält eine Partitionstabelle, die die Aufteilung des Datenträgers beschreibt, und optional einen Boot-Loader, ein Programm, das vom BIOS aufgerufen wird und ein Betriebssystem auf einer der Partitionen startet – danach startet der dort definierte Boot-Manager.
- Speichermedien, die nicht in Partitionen unterteilt sind, z. B. USB-Sticks oder CD-ROMs, enthalten keinen MBR. Hier wird der erste Datenblock als *Bootsektor* oder auch *Boot Record* bezeichnet.

→ Es gibt mehr als eine „richtige“ Lösung und Reihenfolge, um das Gleiche zu tun.

Fachhochschule Nordwestschweiz
Hochschule für Technik

Ablauf des Boot Managements

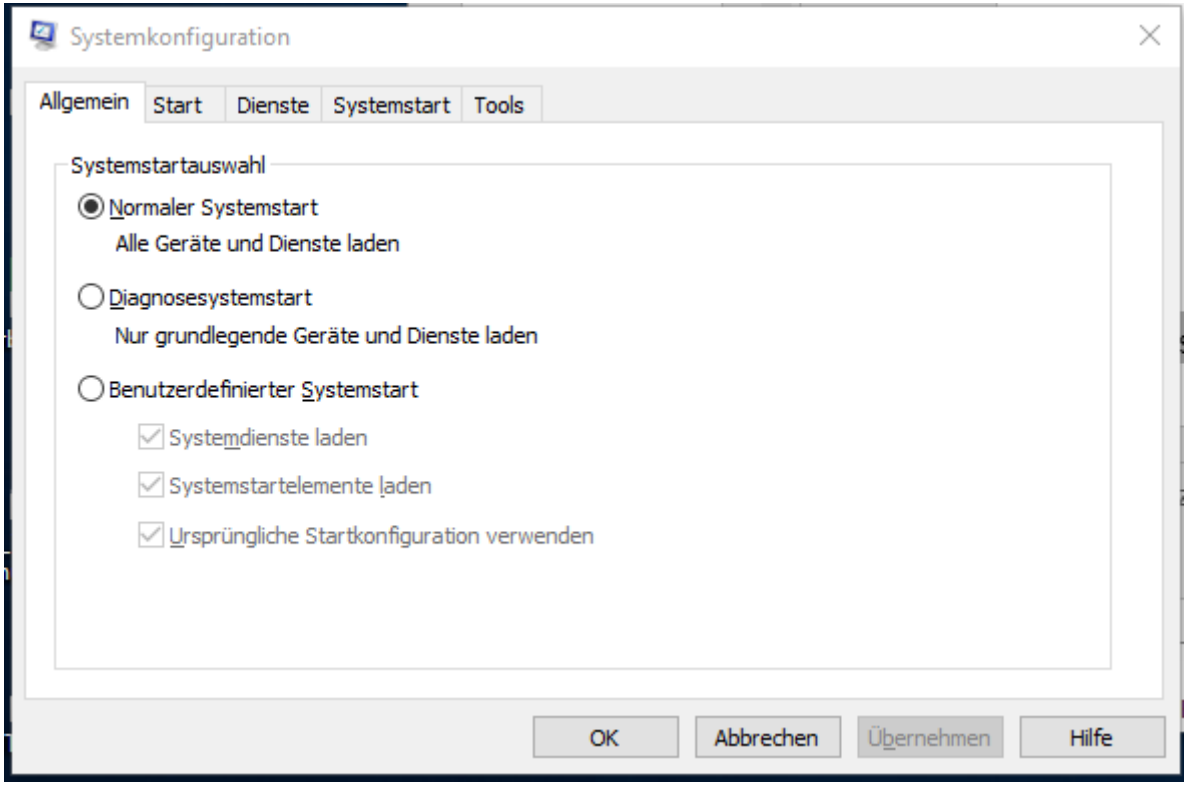




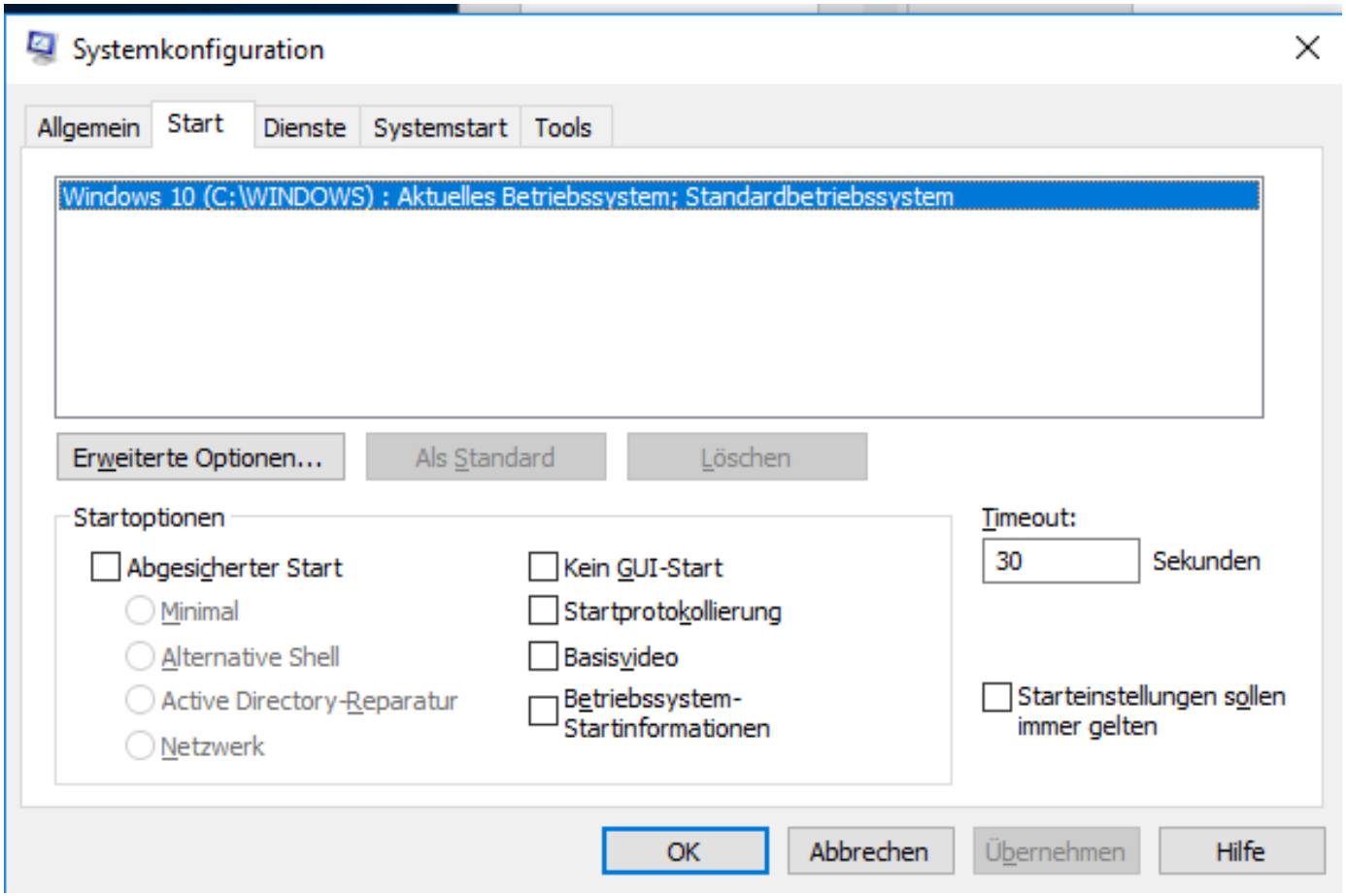
http://www.hojoe.de/studienjahresarbeit/node23.html

Windows 10 Boot-Einstellungen

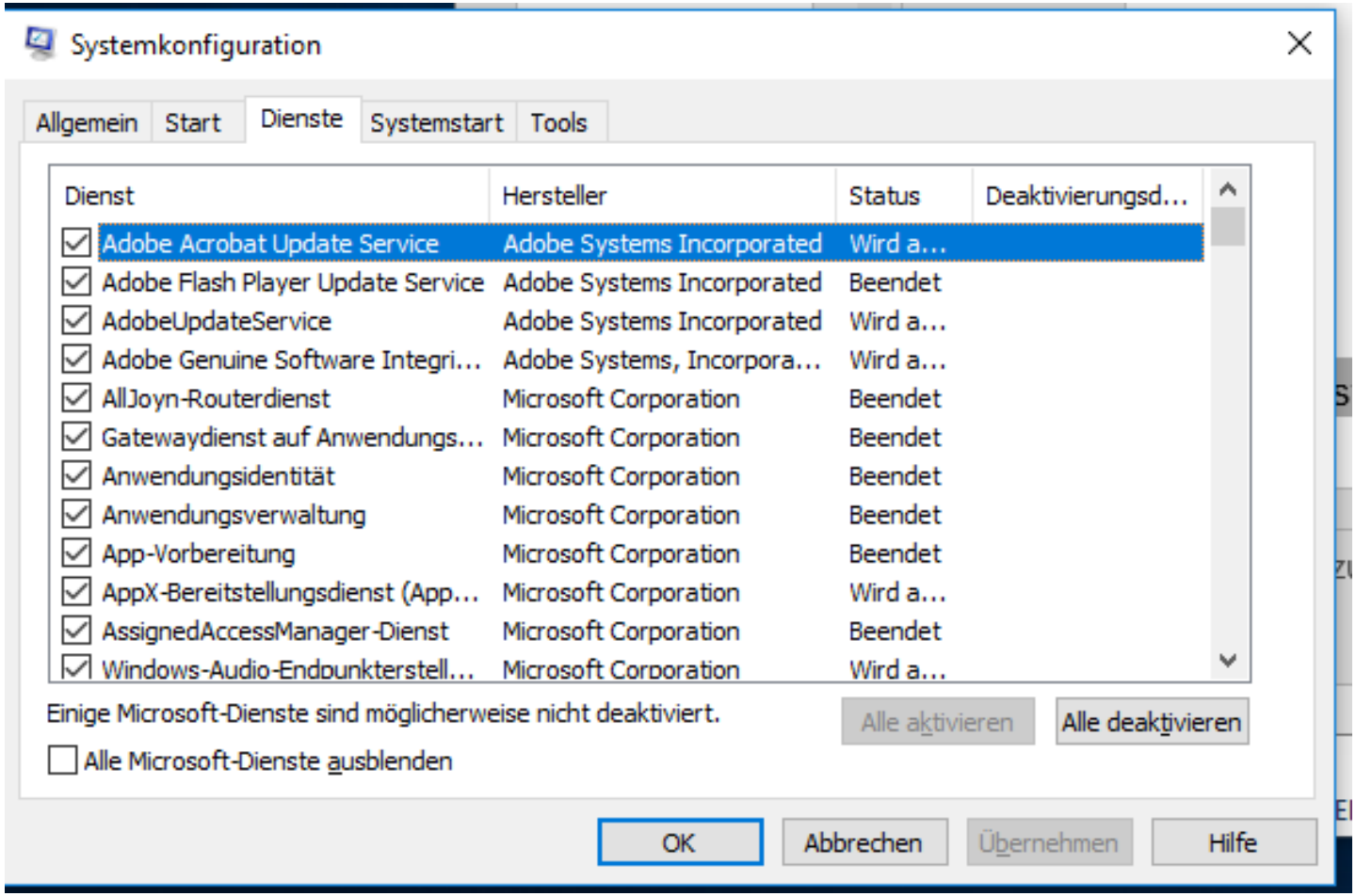
Windows-Suche: Systemkonfiguration



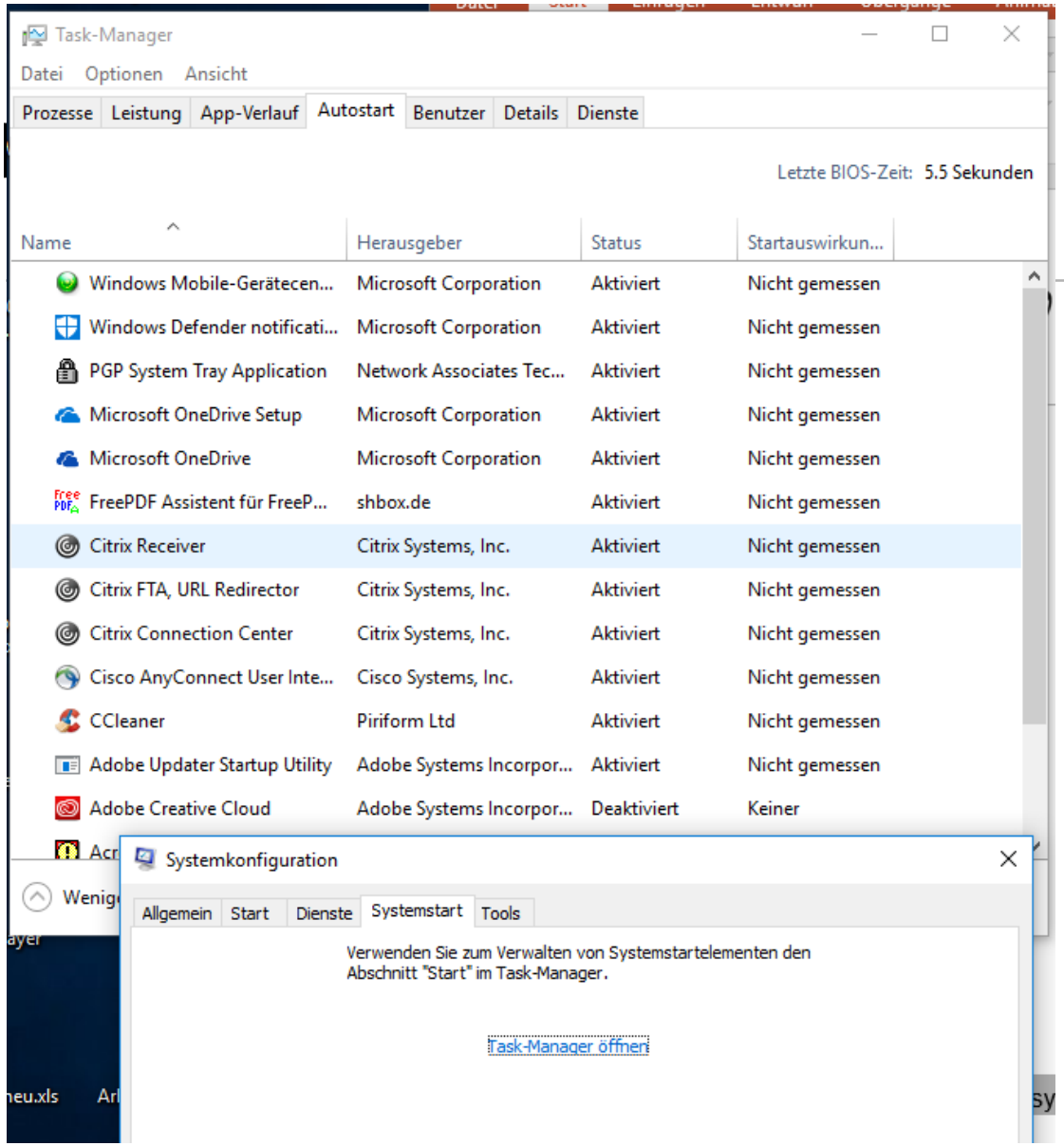
Windows 10 Boot-Einstellungen



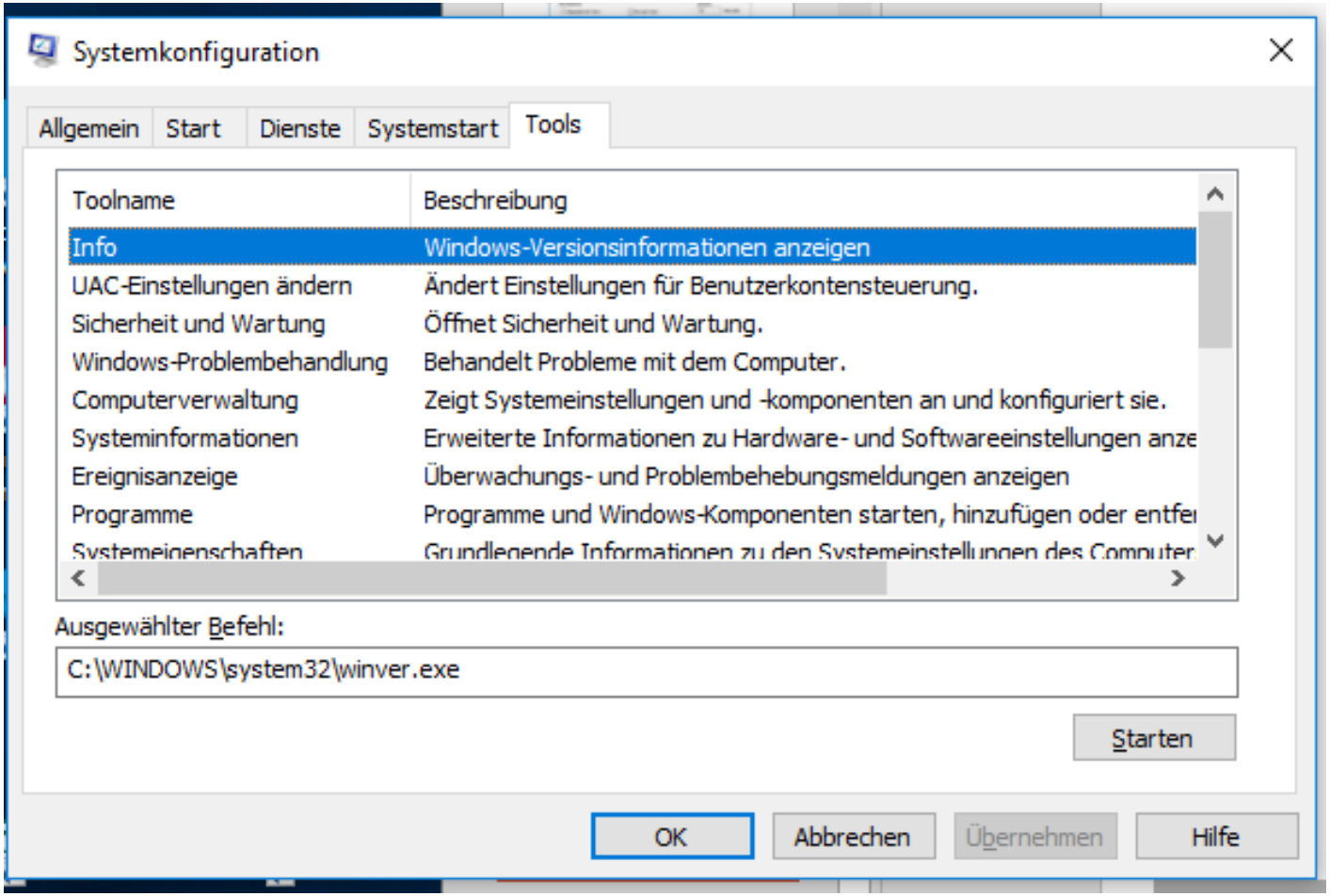
Windows 10 Boot-Einstellungen



Windows 10 Boot-Einstellungen



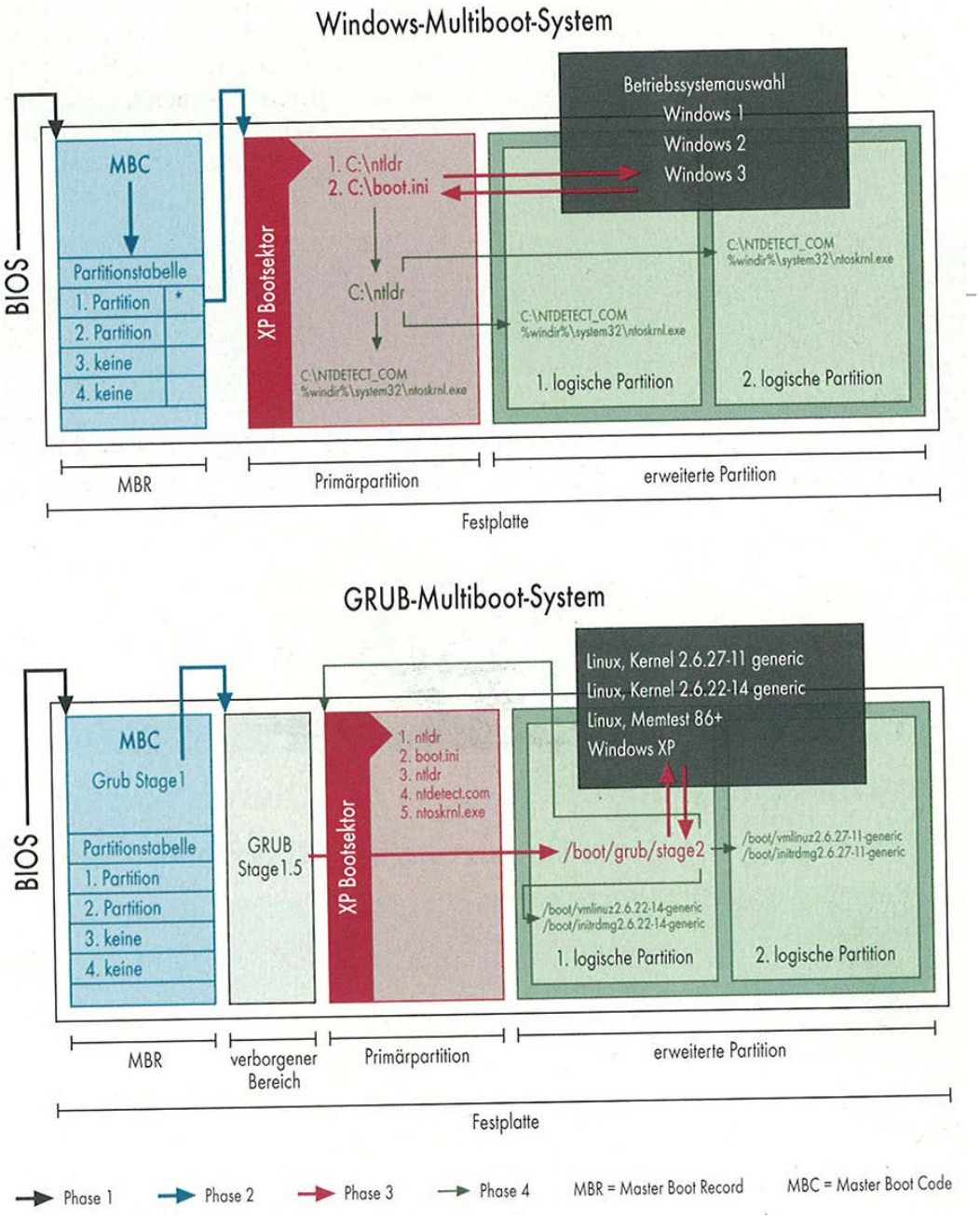
Windows 10 Boot-Einstellungen



Linux Boot Manager

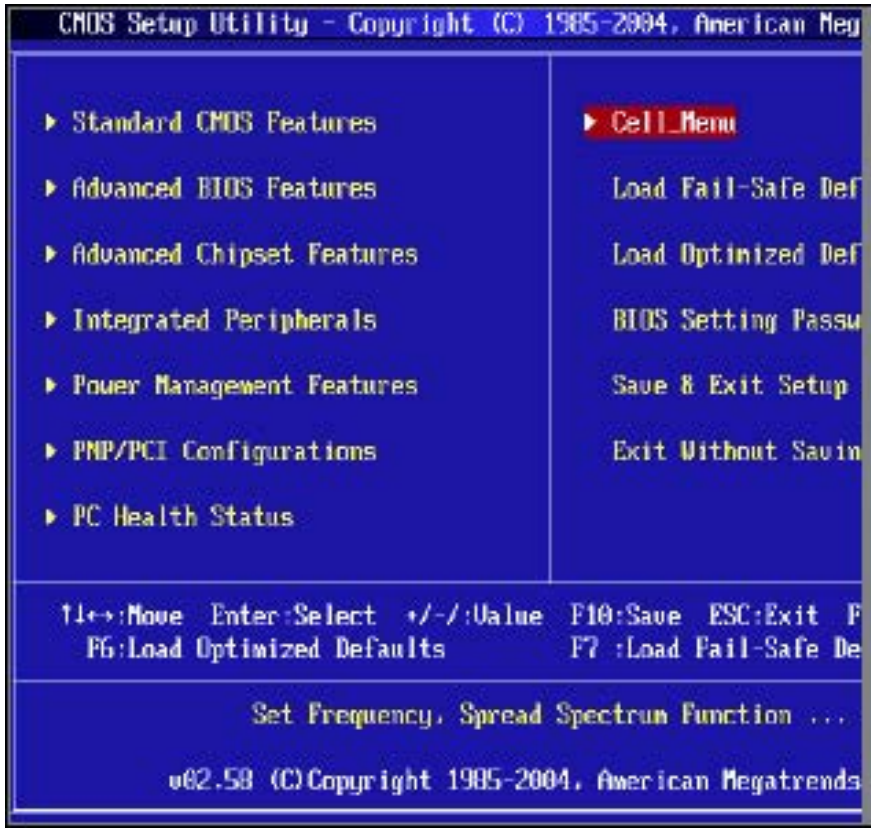
- Ubuntu installiert einen eigenen Boot-Manager (grub).
- Konfiguration: boot/grub/menu.lst
- Bei der Installation unter Windows mittels wubi-installer befinden sich die nötigen Dateien im Verzeichnis ubuntu\winboot (z.B. D:\ubuntu\winboot\menu.lst)

Grub versus Windows Multi-Boot



Quelle: iX 12/2010, p. 46

BIOS vs. UEFI


CMOS Setup Utility - Copyright (C) 1985-2004, American Meg
Standard CMOS Features
Advanced BIOS Features
Advanced Chipset Features
Integrated Peripherals
Power Management Features
PNP/PCI Configurations
PC Health Status
Cell_Menu
Load Fail-Safe Def
Load Optimized Def
BIOS Setting Passu
Save & Exit Setup
Exit Without Savin
↑↓←→:Move Enter:Select +/-/:Value F10:Save ESC:Exit F
F6:Load Optimized Defaults F7 :Load Fail-Safe De
Set Frequency, Spread Spectrum Function ...
v02.58 (C)Copyright 1985-2004, American Megatrends


EFI BIOS Utility - Advanced Mode
Main
Ai Tweaker
Advanced
Monitor
Boot
CPU Temperature
MB Temperature
CPU Fan Speed
Chassis Fan 1 Speed
Power Fan 1 Speed
Chassis Fan 2 Speed
Power Fan 2 Speed
CPU Q-Fan Control
CPU Fan Speed Low Limit
CPU Fan Profile
Chassis Q-Fan Control
Chassis Fan Speed Low Limit
Chassis Fan Profile
Enabled
600 RPM
Standard
Enabled
600 RPM
Standard

BIOS vs. UEFI


ASUS EFI BIOS Utility - EZ Mode
Exit/Advanced Mode
06:06
Tuesday[11/16/2010]
P8P67 DELUXE
English
BIOS Version : 0304
Build Date : 10/21/2010
CPU Type : Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
Speed : 3100 MHz
Total Memory : 2048 MB (DDR3 1333MHz)
Temperature
CPU +116.6°F/+47.0°C
MB +96.8°F/+36.0°C
Voltage
CPU 1.200V
5V 5.120V
3.3V 3.408V
12V 12.288V
Fan Speed
CPU_FAN 1834RPM
PWR_FAN1 N/A
CHA_FAN1 N/A
CHA_FAN2 N/A
System Performance
Quiet
Performance
Energy Saving
Normal
Boot Priority
UEFI
UEFI
Use the mouse to drag or keyboard to navigate to decide the boot priority.
Boot Menu(F8)
Default(F5)

Der Hardware Abstraction Layer

- HAL (kurz für *Hardware Abstraction Layer*) ist eine freie Software, die es Anwendungen ermöglicht, Informationen über verfügbare Hardware abzurufen und mit ihr zu kommunizieren. Mit HAL können Anwendungen auf das Anschließen und Entfernen von Hardware reagieren (Plug and Play).
- HAL arbeitet als Daemon und benutzt D-Bus, um Informationen an Anwendungssoftware weiterzugeben. HAL verfügt über eine eigene Datenbank, die detaillierte Beschreibungen von Hardwarekomponenten enthält. So kann Anwendungssoftware beispielsweise in die Lage versetzt werden, eine Digitalkamera als solche anzusprechen, auch wenn sie sich am Universal Serial Bus nur als Datenspeicher zu erkennen gibt (Wikipedia)

Fazit

- Boot Management ist:
  - Komplex
  - Historisch und unkoordiniert gewachsen
  - Rückwärtskompatibel
  - Unter ständigem Druck, neue Hardware korrekt zu integrieren
  - Mit (zu) vielen Optionen ausgestattet

→ Die Default Werte sind in der Regel nicht schlecht.

Zusammenfassung der Lektion 10 und Hausaufgabe

- Die Prozeduren und Optionen zum Starten und Stoppen ihres Linux-Systems.
- Das Konzept von Run Levels in Unix / Linux.
- Die Funktionalität und Benutzung des Boot Managers.
- BIOS/UEFI
- Der Hardware Abstraction Layer

- Hausaufgabe:
  - Repetieren Sie den Stoff dieser Lektion.
  - Studieren Sie das Dokument «10-grub.pdf»
  - Studieren Sie die Web-Seite «https://de.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface»