Hochschule
Augsburg University of
Applied Sciences

Zusammenfassung Embedded Linux SoSe 2015

Embedded Linux
Hochschule Augsburg
Fakultät für Informatik (Prof. Dr. Hubert Högl)
Studiengang Technische Informatik, 6. Semester

Sommersemester 2015

Datum: 2015-06-12 07:48

Matthias Pfitzmayer, <matthias.pfitzmayer@hs-augsburg.de>, #935691
Daniel Schneider, <daniel.schneider@hs-augsburg.de>, #
Matthias Struwe, <matthias.struwe@hs-augsburg.de>, #

Zusammenfassung

Inhalt

2.8 Welche Distribution für Embedded Linux werden erwähnt?

[?]

2.9 Ist der Apache Webserver eine gute Wahl? Welche anderen Webserver gibt es noch?

[?]

2.10 Welche Werkzeuge gibt es zur Programmierung grafischer Oberflächen?

[?]

2.11 Fazit: Was ist der Unterschied zwischen Embedded Linux und "gewöhnlichem" Linux?

[?]

3 Tips for planning an embedded Linux project

[?]

3.1 Welche Anforderungen sprechen für Linux? (5 Stück)

3.2 Wie "bezahlt" man die Vorteile von Linux?

3.3 Aus welchen Einzelteilen besteht Embedded Linux? (6 Stück)

3.4 Welche Fähigkeiten braucht man, um einen Rechner mit Embedded Linux auszustatten und zu programmieren?

3.5 Welche Kosten sind zu erwarten?

3.6 Welche Risiken gibt es?

3.7 Was umfasst der Lebenszyklus des Produktes?

4 Speichertechnologien und Filesystems

[?]

4.1 Wie funktioniert Flash Speicher?

4.2 Welche grundsätzlichen Bausteintypen gibt es?

4.3 Was verstehen Sie unter dem Kürzel "MTD"?

4.4 Was ist CRAMFS?

4.5 Was ist SQUASHFS?

4.6 Was ist JFFS2?

4.7 Was ist YAFFS?

5 Anatomy of Flash Filesystems

[?]

5.1 Was ist das Virtual File System VFS?

5.2 Was sind “bad blocks” und “wear leveling”?

5.3 Was kann man über die Mount-Zeit von verschiedenen Flash Filesystemen sagen?

6 Embedded Linux Primer (Bootloader)

[?]

Fragen siehe index.html#bootloader.

7 Fast Startup Linux

[?]

7.1 Welche Möglichkeiten gibt es, die Bootzeit zu reduzieren?

8 Linux Kernel Module

[?]

8.1 Kann man in einem Kernelmodul auf die C Bibliothek zugreifen?

8.2 Beschreiben Sie, wie man ein einfaches Kernelmodul nativ als auch cross-kompiliert.

8.3 Wie lädt man ein Kernelmodul in den Kernel?

8.4 Was sind Gerätenummern?

8.5 Wie legt man Gerätedateien an?

8.6 Was sind zeichenorientierte Treiber (character driver)?

9 Freie / offene Software

Freie_Software

[?]

Kriterium	GPL	LGPL	AGPL	BSD	MPL	CC
kann für jeden Zweck verwendet werden	X	X	X			
kann den eigenen Bedürfnissen angepasst werden	X	X	X	X		
kann mit Freunden und Nachbarn geteilt werden	X	X	X	X		
eigene Änderungen dürfen verbreitet werden	X	X	X	X		
auf Freier Software basierende Programme müssen freie Software bleiben	X					
Aufzählung aller Copyright besitzer der vorhergehenden Sources			X	X		
Änderungen müssen dem Urheber der Software angezeigt und übermittelt werden			X			

9.1 Erläutern Sie das Prinzip des Copyleft.

Das Copyleft-Prinzip besagt, dass Software, die auf unter der GNU General Public License (GPL) veröffentlichten Software basiert nur mit den gleichen Freiheiten veröffentlicht werden darf. Dies stellt sicher, das freie Software für alle zugänglich und frei bleibt.

9.2 Welche vier Freiheiten gewährt die GPL?

Niemand sollte von der Software, die er verwendet eingeschränkt werden. Es gibt vier Freiheiten, die jeder Benutzer haben sollte.

- die Freiheit die Software für jeden Zweck zu verwenden
- die Freiheit die Software an die eigenen Bedürfnisse anzupassen
- die Freiheit die Software mit Freunden und Nachbarn zu teilen
- die Freiheit die eigenen Änderungen zu teilen

Erfüllt ein Programm diese Kriterien kann es als freie Software bezeichnet werden.

9.3 Wie unterscheidet sich die LGPL von der GPL?

Die Lesser GNU Public License (LGPL) unterscheidet sich von der GNU Public License (GPL) in der Copyleft Regelung während bei der GPL alle von der Software abgeleiteten Programme ebenfalls unter der selben Lizenz veröffentlicht werden müssen kann aus unter der LGPL veröffentlichten Quellen auch proprietäre Software geschrieben werden.

9.4 Wie unterscheidet sich die BSD Lizenz von der GPL?

Die BSD_Licence verlangt, dass nach einer Änderung der Urheber entsprechende des Musters "Copyright (c) <YEAR>, <OWNER>" auch in der geänderten Version bekannt gegeben wird.

9.5 Was sind Duale Lizenzen? Nennen Sie ein Beispiel.

Mehrfach_Lizenzen erlauben es

Beispiele: Qt, MySQL, Asterisk, Sendmail und Perl

9.5.1 Vorteile

Mit diesem Lizenzierungsprinzip können Unternehmen eigene Anwendungen entwickeln, ohne diese selbst wieder der Open-Source-Lizenz unterwerfen zu müssen. So ist es einerseits möglich, das Open-Source-Modell zur Entwicklung und Distribution von Software zu fördern, und andererseits ein professionelles Geschäftsmodell zu etablieren, bei dem Softwarelizenzen ohne Einschränkungen verkauft werden können. Für eine Mehrfachlizenzierung spricht des Weiteren die Möglichkeit, die Software mit proprietären Erweiterungen zu versehen.

Käufer einer mehrfach lizenzierten Software sind darüber hinaus nicht an die teils sehr restriktiven Vorgaben zur Weitergabe von Software unter einer freien Lizenz gebunden. Die Entwicklung freier Software wird gefördert, ohne auf Spenden angewiesen zu sein.

9.5.2 Nachteile

Gegen eine Mehrfachlizenzierung spricht, dass das lizenzierende Unternehmen die Arbeit der Open-Source-Entwickler ausnutzen könnte. Darüber hinaus könnte das entwickelnde Unternehmen es sich vorbehalten, irgendwann die Weiterentwicklung der offenen Version zu stoppen.

Umstritten und selten erwähnt ist jedoch die Tatsache, dass bei einigen Open-Source-Lizenzen die Kontrolle über die Entwicklung der Software leicht an die Community oder an finanziell überlegene Unternehmen gehen kann. Ein Entwicklungsstopp einer unter GPL lizenzierten Software besagt noch lange nicht, dass die Software nicht mehr weiterentwickelt wird. Die GPL räumt jedem Nutzer das Recht ein, die Software beliebig zu modifizieren, also auch weiterzuentwickeln.

Heftig umstritten ist, ob der Initiator eines Open-Source-Projektes, welcher seine Software unter einer dualen Lizenz auf den Markt bringt, die Weiterentwicklungen der Community in sein Produkt einfließen lassen kann, um diese unter einer anderen Lizenz als der GPL zu vertreiben.

Kleine Unternehmen laufen generell Gefahr, ihre Open-Source-Softwareprojekte an große Unternehmen zu „verlieren“.

9.6 Darf man Programme im Linux Userspace unter einer proprietären Lizenz vertreiben?

9.7 Darf man mit dem freien GNU C Compiler kommerzielle geschlossene Programme schreiben?

9.8 Was halten Sie von geschlossenen Kernelmodulen?

10 Linux auf eingebetteten Systemen

Suchen Sie Geräte, die unter Linux laufen. Sie haben sicher schon selber welche entdeckt. Sie können sich aber auch von folgenden Seiten inspirieren lassen:

linuxgizmos elinux linuxfordevices

10.1 Was erwartet Sie hinsichtlich der Lizenzen, wenn Sie Linux wählen?

[?]

10.2 Warum Linux? (6 Gründe)

10.3 Was bedeutet “GPL”?

10.4 Was ist “Open Source”?

10.5 Was verstehen Sie unter “Linux Standard Base”?

[?]

Beantworten Sie folgende Fragen:

10.6 Was zeichnet Embedded Systems aus?

10.7 Vergleichen Sie BIOS und Bootloader.

10.8 Aus welchen wesentlichen Funktionsblöcken bestehen Embedded Systems?

10.9 Aus welcher Hardware besteht die Entwicklungsumgebung?

10.10 Wie startet Embedded Linux? (von U-Boot bis init Prozess)

10.11 Welche Arten von Flash Speicher gibt es?

10.12 In welche Regionen ist der Flash Speicher bei Linux in der Regel aufgeteilt?

10.13 Wie sieht grob die memory map des gesamten Rechners aus?

Beschreiben Sie den Ausführungskontext mit Applikation, C Bibliothek, Kernel und Gerätetreibern. Was läuft im Kernel-Kontext, was im User-Kontext?

10.14 Was macht die MMU und wo ist sie eingebaut?

10.15 Wie nennt man die nicht-native Kompilierung auch?

"Cross-Kompilierung" - Das kompilieren von nativer Software auf einem Hostrechner für ein spezifisches Target mit Hilfe von Cross-Kompilierungs Toolchain und im Terminal gesetzten Variablen wie z.B. "$ARCH", "$CC", etc.

10.16 Was ist native Kompilierung? Ab welcher Rechenleistung macht sie Sinn?

10.17 Was ist eine Embedded Linux Distribution?

10.18 Aus wie vielen Paketen besteht eine Distribution ungefähr?

10.19 Welche anderen CPUs unterstützt Linux neben dem x86?

10.20 Was ist ein monolithischer Kern?

10.21 Warum lässt sich Linux einfach auf fast beliebige Prozessoren portieren?

10.22 Wie unterscheiden sich die Befehlssätze der Prozessorfamilien?

10.23 Virtual Memory Management

10.24 Wie heisst die Linux Variante, die ohne diese Hardware-Komponente auskommt?

10.25 Was ist ein "System on Chip" (SoC)?

10.26 Wie sieht die Schichtung der Software eines Embedded Linux Systems in etwa aus?

10.27 Welches Dateisystem wird bei Flash Speicher häufig verwendet?

10.28 Welche Sprachen werden zur Programmierung verwendet?

10.29 Welche Bootloader werden bei Embedded Linux verwendet? Ist grub auch eine Option?

11 Der Entwicklungsrechner

Installieren Sie Linux auf Ihrem Hostrechner, entweder in einer eigenen Partition oder in einer virtuelle Maschine wie z.B. VirtualBox.

Sehen Sie sich auf Ihrem GNU/Linux Desktop PC oder Laptop den Kernel und das Root Filesystem an. Wie viel Platz beanspruchen die einzelnen Teile?

11.1 Für welche Aufgaben wird der Hostrechner (Entwicklungsrechner) verwendet?

Im Skript steht eine Liste von Werkzeugen für den Entwicklungsrechner. Was macht jedes einzelne dieser Werkzeuge?.

11.2 Lernen Sie die Bedienung Ihres Linux Rechners ausschliesslich über die Kommandozeile.

Installieren Sie einen Teminal Multiplexer, z.B. screen oder tmux. Auch auf dem Zielrechner ist so ein Werkzeug sehr praktisch.

Welche Terminalprogramme gibt es unter Linux, um auf die Konsole über die serielle Schnittstelle zu gehen? Wie lauten die üblichen Einstellungen?

Wie kann man Daten im Terminalprogramm mit dem Protokoll X/Y/Z-Modem übertragen? Warum sollten nur kleinere Dateien bis zu ein paar 100 KByte übertragen werden?

11.3 Wie arbeitet man mit folgenden Programmen: ssh, scp, ftp (ncftp)?

11.3.1 ssh

11.3.2 scp

11.3.3 ftp

11.3.4 ncftp

Wie ändert man die Netzwerkeinstellungen für die Schnittstellen eth0 und eth1 mit den Werkzeugen ifconfig, route, ethtool und iptables.

Wie richtet man auf dem Host eine zweite Ethernet-Schnittstelle ein, um ein Netzwerkkabel mit dem Target zu verbinden? Wie lauten die Einstellungen auf dem Target und auf dem Host?

11.4 Installieren Sie den Quelltext des Linux Kernels auf dem Hostrechner und kompilieren Sie ihn.

11.5 Installieren Sie auf Ihrem Entwicklungsrechner eine Toolchain für die ARMv5 Architektur (ELDK oder Debian).

11.6 Wie richtet man auf dem Host einen NFS Server ein?

11.7 Wie richtet man auf dem Host einen TFTP Server ein?

12 Literatur und sonstige Quellen

References

[AELS] Architekturen eingebetteter Linux Systeme, Rene Rebe, 2008

http://elk.informatik.fh-augsburg.de/cdrom-elinux/Lesen/rebe-linux-ueberall.pdf

[ATES] Johan Thelin, Introduction: A Typical Embedded System, 12/2009

http://www.linuxjournal.com/article/10565

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/intro/thelin/

[AFFS] M. Tim Jones, Anatomy of Flash Filesystems, 2008

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/speicher/l-flash-filesystems-pdf.pdf

[BSD2] Berkley Software Distribution

[ELPB] Hallinan, Kap. 7 (Bootloader)

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/boot/hallinan_embedded_linux_primer_chap_7_bootloaders.pdf

[FSTL] Fast Startup Linux

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/boot/637_Fast_Startup_Linux.pdf

[GNU] GNU Software Foundation

[HELI] Embedded Linux, Hallinan, Kapitel 1 (“Introduction”)

http://proquest.tech.safaribooksonline.de/book/operating-systems-and-server-administration/embedded-linux/9780137061129

[LKMS] Boguslaw Sylla, Patrick Schorn, Linux Kernel Module.

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/kernel/linuxkernel.pdf

[STFS] Plattner/Schnepp Speichertechnologien und Filesystems

http://elk.informatik.fh-augsburg.de/pub/rtlabor/elinux/speicher/speichertechnologie.pdf

[TPEL] Tipps for planning an embedded Linux project, Cliff Brake, 2006

http://elk.informatik.fh-augsburg.de/cdrom-elinux/planning-tips.txt

[TBP] The Big Picture, Hallinan

http://proquest.tech.safaribooksonline.de/book/operating-systems-and-server-administration/embedded-linux/9780137061129

[WIKI] Wikipedia the free Encyclopedia