

Linux in Embedded Anwendungen

Möglichkeiten des Betriebssystems erkennen und richtig nutzen



Bild 1: Wer Linux in Embedded Systemen verwenden will, hat die Wahl zwischen einer Linux-Standard-Distribution, einer Embedded Distribution und einer kundenspezifischen Distribution.

Die Linuxwelt in der Automatisierung birgt für den Einsteiger viele Fehlerquellen, Irrwege und Gefahren für den Erfolg seines Embedded Produktes. Folgender Artikel zeigt auf, wie man die Vorteile von Linux in der Automatisierung nutzt, schnell und sicher ans Ziel kommt.

Wer jemals unter einem erzwungenen Versionswechsel seines Betriebssystems litt, weil der Hersteller gerade eine Abkündigung ausgesprochen hat, der wird die Herstellerunabhängigkeit von Linux zu schätzen wissen. Linux kennt keine Abkündigungen, auch Lizenzgebühren gibt es nicht, und auf dem Markt sind viele Spezialisten mit breitem Know-how verfügbar. Zudem kann der Entwickler Einblick in den Source-Code des Betriebssystems nehmen – sei es für Anpassungen oder bei der Fehlersuche. Fast jedes Problem ist unter Linux schon einmal gelöst worden; damit präsentiert sich Linux flexibel und vielseitig. Entwicklungsplatzlizenzen gibt es bei Linux ebenfalls nicht – damit ist der Einstieg preiswert. Der offene Quellcode schützt die Investitionen über den gesamten Produktlebenszyklus. Gleichzeitig wird langfristig in Wissen um den Betriebssystemkern investiert: Die notwendigen Systemtools sind lizenzkostenfrei, und eine wachsende Zahl von Beratern, Systemintegratoren und Entwicklern verhindert Monopolpreise.

Anforderungen von Embedded

Ein Embedded System kann beliebig einfach oder auch sehr komplex sein. So unterschiedlich sie sein mögen, all diese Systeme müssen weltweit und rund um die Uhr zuverlässig arbeiten. Auch nach fünf oder zehn Jahren muss der Anwender sie noch warten, reparieren und gegebenenfalls nachrüsten können. Jederzeit müssen Verbesserungen und Erweiterungen möglich sein. Das setzt zum einen voraus, dass die Hardware nachgeliefert werden kann, zum anderen muss die Software reproduzierbar sein. Die exakte Reproduzierbarkeit des Software-Images ist oft Voraussetzung für eine Zertifizierung des Gesamtsystems, sei es in der Medizin- oder Fahrzeugtechnik. Die Ressourcen eines Embedded Systems unterscheiden sich grundlegend von Serverarchitekturen oder Bürorechnern. Statt einer großen Festplatte hat das Embedded System oft nur eine Flash Disk. Weil lüfterlose Rechner weniger anfällig sind, läuft der Prozessor mit geringerer Taktrate. Damit heizt er

das System zwar weniger auf, aber er rechnet auch entsprechend langsamer. Die Ressourcen eines Embedded Systems sind vielfältig optimiert, und viele Elemente wie Thermodrucker, Feldbuskomponenten, Sensoren, Aktoren oder Mensch-Maschine Schnittstellen gibt es nur im Embedded System. Die Entwickler von Embedded Systemen interessieren vor allem Bootzeiten, Echtzeit, Verhalten bei Stromausfall und Datenspeicherung, Updatemöglichkeiten über USB oder Ethernet sowie Fernwartbarkeit, vorhandene Schnittstellen und unterstützte Protokolle.

Linux auf den ersten Blick

Während bisher dem Entwickler der Einblick in die Tiefen des Betriebssystems verwehrt blieb, muss er jetzt genau hier zupacken. Wie stellt sich dem begeisterten Einsteiger aber die Linux-Welt dar? Da gibt es für wenig Geld die Standard-Distributionen wie Redhat, Suse oder Debian. Andererseits findet man aber auch Distributionen mit dem schmückenden

Attribut 'Embedded' von Wind River, Monta Vista oder das Elinos von Sysgo. Der Anwender findet verschiedene Kernel-Versionen – aktuell sind derzeit 2.4 und 2.6 – und verschiedene Möglichkeiten, Linux zur Echtzeitfähigkeit zu verhelfen – im Gespräch sind z.B. RTAI, Xenomai oder eine weitere echtzeitfähige Kernelerweiterung.

Software-Komponenten

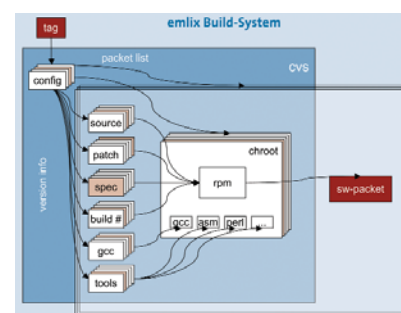
Kauft sich der Einsteiger für rund 100€ eine Standarddistribution, erwirbt er gleichzeitig mehr als 1.000 Anwendungen als Software-Päckchen. Zuzüglich aller Treiber sind das mindestens 4.000 bis 10.000 Software-Komponenten, aus denen er sein Betriebssystem zusammenstellen kann. Allein die Entscheidung, welche Teile davon wichtig, nötig oder überflüssig sind, ist eine Herausforderung. Erschwerend kommt hinzu, dass sich einzelne Päckchen gegenseitig bedingen. Die Anpassung ist mühsam und erfordert Detailkenntnis. Nach intensivem manuellen Zusammenstellen hat der Anwender ein erstes Image von mindestens 200MByte bis hin zu mehreren GByte zusammengestellt. Ein typisches Linux im Embedded Bereich braucht aber nur rund 15-40 Software-Pakete der 4.000 bis 10.000 Verfügbaren. 90% der Funktionen einer Standard-Distribution werden im Embedded Bereich nicht gebraucht. Was nützt die Textverarbeitung, wenn die Touchscreen Unterstützung fehlt? Wofür soll der multi-userfähige Printserver gut sein, wenn der Thermodrucker nicht angesteuert werden kann? Feldbusanbindung ist mit einer Standard-Distribution nicht möglich, doch MySQL Datenbank könnte man nutzen. Hinzu kommt, dass die mitgelieferten Tools die individuelle Zusammenstellung nicht automatisch dokumentieren. In der ersten Euphorie übersieht der Einsteiger jedoch, dass hier die fehlende Dokumentation der Reproduzierbarkeit im Wege steht. Gern wird auch die Abhängigkeit zwischen Entwicklungsrechner und Distributionsversion übersehen. Ein

Rechner mit einer neueren Distribution ist unter Umständen gar nicht mehr in der Lage, das identische Image zu erzeugen, weil andere Tools in der neueren Version zur Verfügung stehen. Das bedeutet den Verlust der Entwicklungsumgebung durch Updates. Standarddistributionen sind für Server- und Desktop-Anwendungen optimiert. Der Entwickler des Embedded Systems findet hier ein Linux vor, das intensiv cached, aber kein Flash Filemanagement hat. Für eine Festplatte ist das kein Problem, eine Flash Disk erreicht dadurch vorzeitig ihre max. Anzahl Schreibzyklen und nimmt Schaden. Von diesem Verhalten sollen bereits weltweit Anlagen stillgelegt worden sein. Wenn der Controller dann 100€ Anschaffungskosten gegen mehrere 100.000€ Folgekosten rechnet, bewilligt er das nächste Software-Budget wahrscheinlich sofort.

Embedded Distribution

Damit wird die Daseinsberechtigung der deutlich teureren Embedded Distributionen klar. Sie unterscheiden sich mit Anschaffungskosten im Bereich von mehreren tausend Euro deutlich von denen der Standarddistribution. Dafür kann man aber im Embedded Bereich sofort mit einem vorkonfigurierten und voroptimierten Linux starten. Kleine, schlanke GUIs wie nanoX oder QT/Embedded sparen Platz, für Flash optimierte Gerätetreiber/Filesysteme wie jffs2 vermeiden die oben geschilderte Gefahr des Flash Worn-Outs, und Tools mit kleinem Memory-Footprint wie busybox erleichtern das Embedded Leben. Dennoch muss der Anwender Optimierungsfragen im Detail klären, weiterhin ist er abhängig von der Produktphilosophie des Herstellers der Embedded Distribution und seiner Hardware, die unter Umständen nicht unterstützt wird. Das schränkt die Flexibilität deutlich ein. Aber als stressfreien Start und ersten Einblick sind die Embedded Distributionen durchaus zu empfehlen.

Reproduzierbarkeit durch Build-System



Das Emlix Build-System besteht aus einem Software-Repository, in dem alle verwendeten Kernel- und Applikations-Sourcen im Quelltext gepflegt werden. Repositories werden u.a. zum Versionsmanagement verwendet. Es sind eine Verzeichnisstruktur oder Datenbank, die Datenobjekte und deren Methoden zur Datentransformation enthalten. Deshalb enthält das Build-System eine definierte chroot-Umgebung, in der die Quellen zu RPM-Paketen kompiliert und zusammengestellt werden. RPM sind Archive, die ein Softwarepaket enthalten. Für das Software-Repository kommt das Open-Source Produkt Concurrent Version System (CVS) zum Einsatz. Beim CVS werden Quellcode-dateien oder andere Textdateien aus dem Repository auf den Rechner eines Programmierers geladen. Nach der Bearbeitung werden die geänderten Dateien wieder in das Repository gepackt, wobei die Veränderung protokolliert wird. Das Build-System ist die zentrale Versionsverwaltung für alle Ressourcen, auch die Integration von Kundensoftware sowie eigener Patches für Spezialanwendungen ist möglich. Insgesamt wird damit eine Unabhängigkeit der Target-Distribution vom Host bei der RPM-Erstellung erreicht.

Kundenspezifische Distribution

Lautet das Ziel, schnell und zuverlässig ein langzeitverfügbares und reproduzierbares System auf den Markt zu bringen und dabei gezielt Know-how aufzubauen, ohne schmerzhaftes Lehrgeld zu zahlen, dann bietet sich die Möglichkeit der kundenspezifischen Distribution. Hier liegen die Anschaffungskosten zwischen 5.000 und 20.000€. Ein erfahrener Linux-Scout hilft, aus den vorhandenen Möglichkeiten die optimal passenden auszuwählen. Und am Ende hat der Anwender – neben einem beträchtlichen Know-how Transfer durch begleitende Schulung, Coaching und Support – auch sein individuelles Paket, das mit seiner Hardware die optimale Verbindung darstellt. Systerra Computer GmbH bietet für ihre langfristig verfügbare Hardware diese Linuxbegleitung gemeinsam mit Emlix an. Merkmal der kundenspezifisch zusammengestellten Pakete ist die Reproduzierbarkeit auch noch nach Jahren und unter unterschiedlichen Entwicklungsumgebungen. Damit werden die Kern-

forderungen nach Herstellerunabhängigkeit, Reproduzierbarkeit und Flexibilität erfüllt. Allerdings setzt diese Herangehensweise ein Budget für Software im Projekt voraus, das jedoch in einer professionellen Projektplanung sowieso eingeplant sein sollte. Schneller, preiswerter und sicherer als jedes eigenständige Experiment ist die Begleitung durch einen erfahrenden Linux-Dienstleister. ■



Autorin: Dipl.-Ing. Frauke Frerichs-Waldmann arbeitet im Marketing bei der Systerra Computer GmbH und betreut u.a. die Linuxaktivitäten.

www.systerra.de