



Integrated Development Environment

Eine integrierte Entwicklungsumgebung ist eine Sammlung von Anwendungsprogrammen, mit denen die Softwareentwicklung möglichst ohne Medienbrüche bearbeitet werden kann. In erster Linie sind IDEs hilfreiche Werkzeuge, die dem Softwareentwickler häufig wiederkehrende Aufgaben abnehmen, einen schnellen Zugriff auf wichtige Funktionen bieten, mit denen die Arbeits(zwischen)ergebnisse verwaltet und in spätere Bearbeitungsfunktionen direkt überführt werden können.

IDE

Moderne Software wird zunehmend komplexer, die Größe des Codes wächst und auch die Projektteams. Damit wachsen auch die Anforderungen an den Entwicklungsprozess. Eine flexible und komfortable Entwicklungsumgebung ist daher ein Muß. Unsere Entwicklungsumgebung bietet alles, was für die erfolgreiche Umsetzung Ihres Linux Projektes notwendig ist.

Eclipse als Basis

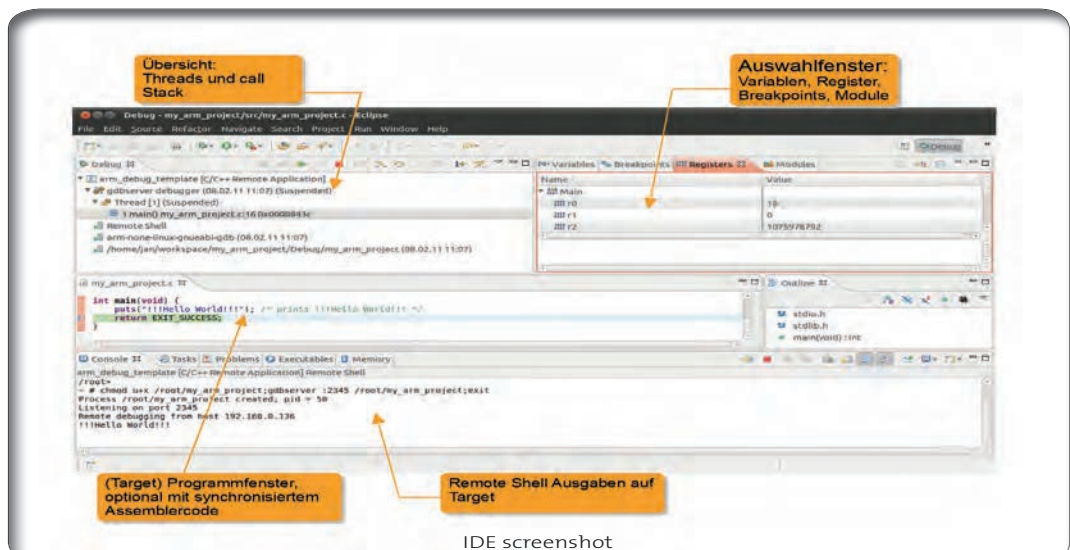
Eclipse, ursprünglich von IBM als Entwicklungsumgebung für Java konzipiert, ist seit langem der de-facto Standard für IDEs. Und auch wir nutzen Eclipse und seine Plug-Ins als Framework für unsere Lösung.

Editoren unterstützen den Arbeit des Entwicklers. Syntaxhighlighting, Funktionsbrowser, Textvervollständigung und ein Hintergrundparser, um etwaige Fehler frühzeitig zu erkennen sind eine Selbstverständlichkeit.

Editoren nehmen typischerweise einen großen Teil der Oberfläche ein, wobei mehrere Fenster gleichzeitig offen sein können. Um diese Fenster können weitere Informationen angeordnet werden (in Eclipse Views genannt). Typische Beispiele von Views für eine aktuelle IDE sind ein Projektbaum mit allen zugehörigen Dateien oder eine View, die alle Klassen, Funktionen und Variablen anzeigt oder ein Fenster, welches das Ergebnis eines Build-Laufs darstellt und eine direkte Navigation im Quelltext ermöglicht.

Views, Editoren, Toolbars und selbstkonfigurierte Menüanordnungen sind Bestandteil des Bedienkonzeptes von Eclipse und nennen sich „Perspektiven“.

Eclipse unterstützt durch seine Struktur der (Projekt-)Datenablage die Arbeit von Projektteams. Eine zentrale, starre Datenablage ist nicht notwendig, das Repository eines Versionskontrollsystems ersetzt sie. Der Workspace von Eclipse stellt immer ein aktuelles Bild des gesamten Quellcodes eines Projektes dar, wo-

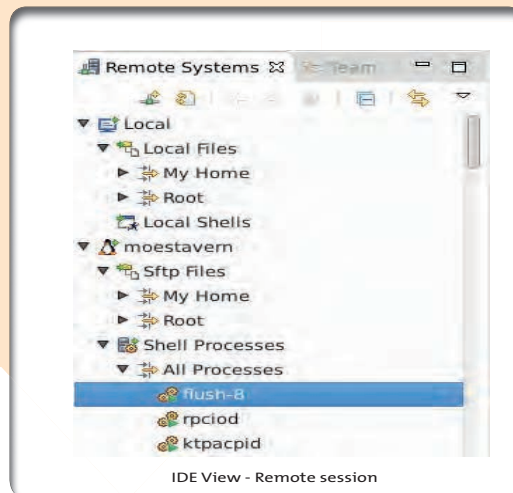




bei gleichzeitig die Änderungen daran überwacht werden. Damit ist gewährleistet, dass auch große Teams effizient an einem Projekt arbeiten können.

Toolchain Integration, Cross Compiling, Remote Debugging und Profiling sind nur einige der weiteren Funktionen, die Sie mit der Linutronix IDE bekommen.

Auch das Zielsystem kann vollständig über die integrierte Entwicklungsumgebung bedient werden. Starten / Stoppen von Pro-



zessen, Anzeige von Prozessen, senden von Signalen und Kopieren sowie Editieren von Daten auf dem Zielsystem sind nur einige der Aktionen, die direkt aus der IDE ausgeführt werden können.

Unterstützung vieler CPU Architekturen

Alle gängigen CPU Architekturen werden unterstützt: x86 und ARM, auch MIPS, PowerPC und viele mehr. Die jeweiligen Crosscompiler sind nahtlos in unsere Oberfläche integriert, das Zielsystem kann komfortabel vom Entwicklungssystem aus gedebuggt werden.

Auch die Simulation verschiedener Zielsysteme ist möglich! Hierbei unterstützen wir alle oben aufgeführten Architekturen.

Native Zielsystem Entwicklung

Im IT Sektor ist die Nutzung von Virtualisierungstechnologien seit Jahren üblich. Mit Linutronix können Sie Virtualisierung nutzen, um einfacher und effektiver für Ihr Zielsystem entwickeln zu können: Alternativ zum Einsatz eines Crosscompilers kann Ihr Zielsystem emuliert werden und Ihre Software wird nativ kompiliert. Insbesondere bei komplexen Abhängigkeiten, beispielsweise zu verschiedenen Bibliotheken, erspart dieser Ansatz sehr viel Zeit: Das Übersetzen der Abhängigkeiten (und die häufig damit

verbundenen Anpassungen) fällt weg und Sie nutzen vorkompilierte Pakete aus der Debian Distribution. Mit unserer VM Ihres Zielsystems können Sie dann direkt, ohne Cross-Compilierung, mit diesen Bibliotheken arbeiten.

Damit wird das Entwickeln für Ihr Zielsystem so einfach wie das Erstellen einer Applikation für Ihren PC.

Für dieses Konzept besteht eine nahtlose Integration in unser ELBE Buildsystem, mit dem Sie Ihre Linux Distribution erstellen können. Ergebnis eines jeden Buildprozesses ist nicht nur eine passende Distribution für das Zielsystem, sondern auch eine virtuelle Maschine, die zur nativen Entwicklung für genau diese Distribution genutzt werden kann.



Versionskontrolle

Mit unserer IDE sind Sie an keine bestimmte Versionskontrolle gebunden! Egal ob Sie CVS, Subversion, Mercurial oder GIT verwenden (um nur einige Beispiele zu nennen) - nahezu jedes Versionskontrollsystem kann in die Oberfläche integriert werden.

Für Windows und Linux

Egal in welcher Welt, Linux oder Windows, Sie sich auf Ihrem Entwicklungssystem zu Hause fühlen, unsere IDE ist für beide Welten erhältlich. Sie bekommen von uns ein vorkonfiguriertes und auf Ihre Anforderungen abgestimmtes Paket.

Unser Paket beinhaltet die integrierte Entwicklungsumgebung mit Crosscompiler, (Remote-) Debugger und allen notwendigen Werkzeugen, die Sie für die Applikationsentwicklung benötigen.

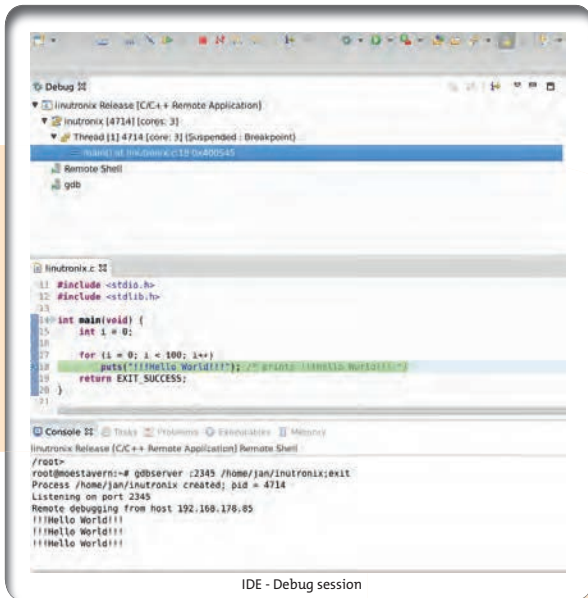
Die Oberfläche ist beliebig erweiterbar, unter anderem um eine Anbindung an unser ELBE Buildsystem zum Erstellen Ihrer eigenen Linux Distribution, um Tools zur Kernelentwicklung und um die Integration der Linux Tracing-Infrastruktur.

Und sollte Ihre IT spezielle Anforderungen haben, ist auch das kein Problem. Kundenspezifische Anpassungen sind problemlos möglich.

Debugging

Die Integration eines Debuggers ist für eine IDE eine Selbstverständlichkeit. Linutronix nutzt hierfür den GNU Debugger (GDB).

Unsere Entwicklungsumgebung stellt ein grafisches Frontend bereit, mit dem sich alle



IDE - Debug session

Funktionen dieses Debuggers nutzen lassen. Dabei wird das Remotedebugging des Zielsystems aus der Oberfläche heraus ermöglicht. Mehrere Ziel-Systeme lassen sich in der Oberfläche grafisch verwalten und für jedes System eine spezifische Debugsession nutzen.

Debugging von Echtzeit-Applikationen

Das Debuggen von echtzeitkritischen Applikationen ist eine herausfordernde Aufgabe, da die Debuggingschnittstellen unter Linux (PTRACE) nicht auf Echtzeitverhalten ausgelegt sind. Allein durch das Attachen des Debuggers wird das zeitliche Verhalten der Applikation beeinflusst. Dies gilt auch, wenn kein Breakpoint erreicht wird.

Abhilfe können hier die sogenannten **Up probes** schaffen. Mit diesen lassen sich Eventpunkte innerhalb der Applikation definieren. Linutronix hat diese Funktionalität um die sogenannten „**Global Breakpoints**“ erweitert, die auch zum Debuggen von Bibliotheken genutzt werden können.

Mit den **Global Breakpoints** können Sie Ihre Applikation stoppen lassen, sobald ein von Ihnen definiertes Up probe Event triggert. Da das zeitliche Verhalten nun so oder so beeinflusst wird, kann sich jetzt der Debugger attachen. Dies bietet den Vorteil, dass die Echtzeitapplikationen Ihr deterministisches Verhalten beibehalten können, bis Sie in einen Breakpoint laufen.

Diese von Linutronix entwickelte Technologie ist bei namhaften Herstellern schon länger im Einsatz.

Integration Tracing

Als Wegbereiter der Linux Tracinginfrastruktur bieten wir selbstverständlich auch eine nahtlose Integration aller Linux Tracingmethoden. Und dies unabhängig davon, ob Sie mit dem in Linux integrierten FTRACE oder mit dem Linux Tracing Toolkit (LTTng) arbeiten.

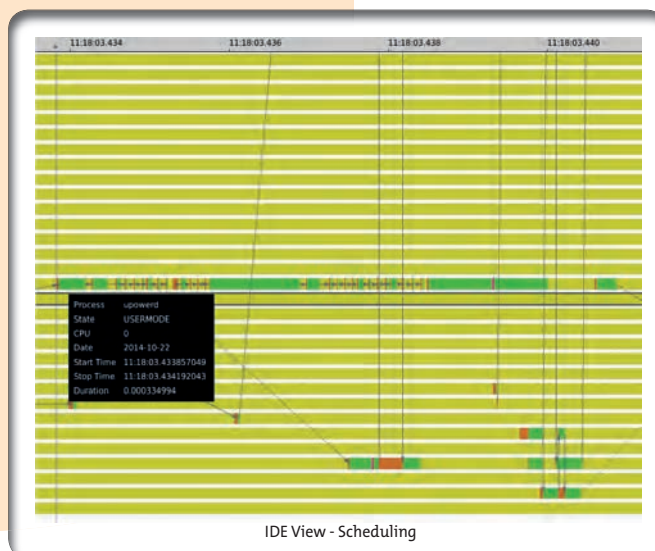
Alle Vorgänge im Betriebssystem und der Applikation können aufgezeichnet, grafisch ausgewertet und statistisch analysiert werden. Egal ob Performance oder Latenzzeitprobleme, unser Werkzeugkasten bietet alles, was Sie für eine effektive Fehlersuche benötigen.

Neben den üblichen Debugfunktionen können auch komplexe Vorgänge im Betriebssystem, wie zum Beispiel das Schedulingverhalten, untersucht werden. Hier spielt es keine Rolle, ob Sie live auf Ihrem Zielsystem tracen oder eine Aufzeichnung „offline“ auswerten. Für die Offline Aufzeichnung gibt es geeignete Werkzeuge von uns.

Automatisierte Auswertung

Vorgefertigte Heuristiken erleichtern die Analyse von Traceaufzeichnungen: Scheduling, CPU Auslastung und die statistische Auswertung einzelner Events sind nur einige Funktionen, die zur Verfügung stehen.

Die Visualisierung kann auch um eigene Parser erweitert werden, um automatisiert bestimmte Fehlerszenarien zu isolieren und grafisch hervorzuheben.



IDE View - Scheduling

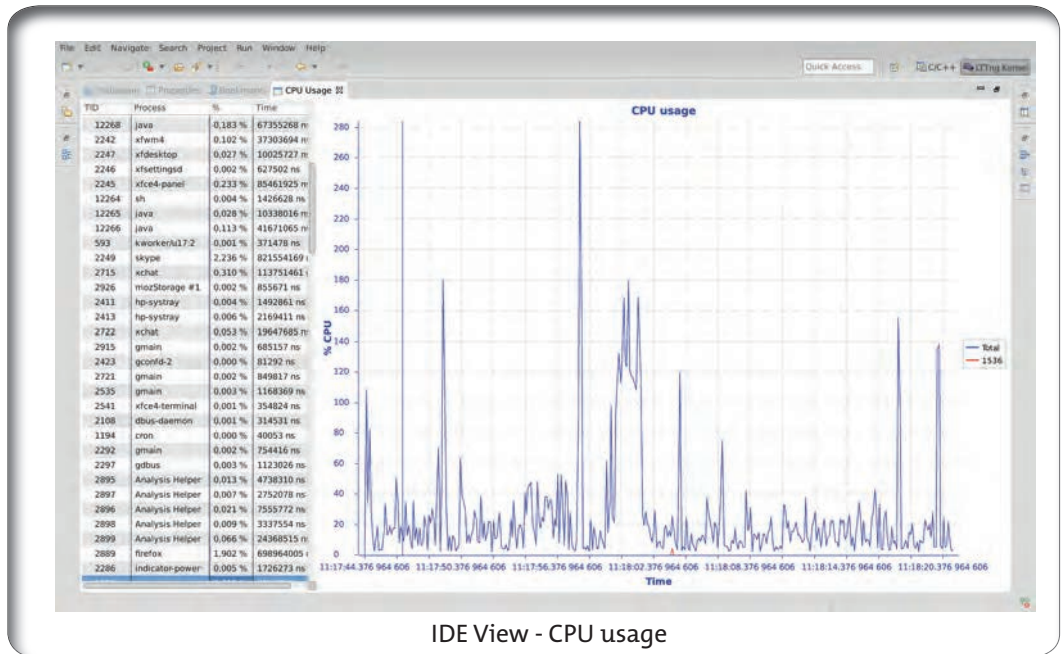


Entwickeln, Analysieren, Optimieren

Durch unser modulares Konzept lassen sich alle Funktionen in der Oberfläche beliebig kombinieren und ergänzen: Entwickeln, Über-

Linux bereitgestellten UPROBES (dynamische Events in der Applikation), als auch das von LTTng bereitgestellt Userspace Tracing kann mit unserer IDE ausgewertet werden.

Egal ob die eigene Applikation oder eine Grafikbibliothek, wie zum Beispiel Qt, alle Soft-



IDE View - CPU usage

setzen, Debuggen, Optimieren und Analysieren - alles mit einer Oberfläche! Unsere IDE unterstützt Sie von den ersten Gehversuchen mit Ihrer Applikation bis hin zur Performance-optimierung für Ihre CPU Architektur.

warekomponenten können instrumentiert und mit unserer Oberfläche grafisch analysiert werden.

Visualisierung Bootverhalten

Profitieren Sie von unserer langjährigen Erfahrung mit FastBOOT Systemen! Linutronix ist eines der führenden Unternehmen, wenn es um die Bootzeit-Optimierung von Linuxsystemen geht.

Schulungen und Workshops

Linutronix bietet nicht nur Werkzeuge zur Applikationsentwicklung und zum Debugging, sondern auch Schulungen und Kurse, in denen alle wichtigen Kenntnisse zu diesen Themen vermittelt werden: Komponenten einer Toolchain, Methoden zum Debugging, Anwendung der Entwicklungsumgebung und Integration der eigenen Applikation in eine Linux Distribution sind Bestandteil unseres festen Schulungsprogramms. Selbstverständlich bieten wir auch maßgeschneiderte Kurse, speziell abgestimmt auf Ihren Anwendungsfall und auf Ihre Bedürfnisse!

Sie erhalten das notwendige Tooling, um den kompletten Bootprozess inkl. U-Boot Bootloader aufzuzeichnen und zu visualisieren. Bei der Umsetzung der Erkenntnisse stehen wir Ihnen, wenn gewünscht, gerne zur Seite.

Analyse von Applikationen

Neben dem Bootloader und dem Verhalten des Betriebssystems läßt sich selbstverständlich auch die Applikation instrumentieren und mit den Werkzeugen zur Tracevisualisierung analysieren. Auch hier ist unsere Oberfläche äußerst flexibel: Sowohl, die von

Seit vielen Jahren schenken uns namhafte Firmen Ihr Vertrauen, wenn es um die Weiterbildung Ihrer Mitarbeiter geht. Profitieren auch Sie von unserem Know How.

Haben wir Ihr Interesse geweckt? Wollen Sie mehr wissen? Rufen Sie einfach an, oder senden Sie uns eine Email.

LINUTRONIX GMBH

Auf dem Berg 3 | D-88690 Uhltingen - Mühlhofen
Telefon +49 7556 4521 890 | Fax +49 7556 919 886
info@linutronix.de | www.linutronix.de

LINUTRONIX
L I N U X F O R I N D U S T R Y