



Bild: kongkiat chairat/stock.adobe.com/nosoft

Die Zukunft der Automatisierung mit Linux

von Stefan Niermann

Nach Einstellen des Supports für ‚Windows CE‘ gewinnt Linux in der Welt der Automatisierung mehr Bedeutung. Der Vorteil des Systems: Offenheit und damit Unabhängigkeit von einem einzelnen Software-Anbieter.

Für viele Anwender ist Linux nach wie vor ein Mysterium, insbesondere im industriellen Umfeld. Während vielen das geschlossene System von Microsoft Windows mit seinen bekannten Installationsprozessen und Benutzeroberflächen vertraut ist, wirkt das offene und in vielen unterschiedlichen Distributionen verfügbare Linux oft abschreckend. Doch mit der Einstellung des Supports von ‚Windows CE‘ im Oktober 2023 gewinnt Linux zunehmend an Bedeutung in der Welt der Automatisierung. Dessen Offenheit und Flexibilität bietet entscheidende Vorteile. Das Betriebssystem kann für Embedded-Systeme angepasst werden und ist echtzeitfähig, was es zur idealen Wahl für SPSen macht, die als Software auf Hutschienen-Geräten laufen. Dank seiner großen Entwicklergemeinschaft ist Linux im Bereich der Webserver längst zum Goldstandard geworden. Aktuell wächst die Verbreitung im industriellen Sektor. Ein nicht zu unterschätzender Vorteil: Gerätehersteller sind bei Supportthemen nicht mehr von einem einzelnen großen Software-Anbieter abhängig, sondern können diesen selbständig gewährleisten.

Eines ist klar: Linux ist gekommen, um zu bleiben. Deshalb ist es für Maschinenbauer essenziell wichtig, Berührungspunkte abzubauen. Ein anschauliches Beispiel für einen einfachen Einstieg in die Linux-Welt ist die Nutzung der HMI- und Scada-Software ‚VisiWin‘ auf dem populären Raspberry Pi. Obwohl dieser meist als „Bastel-Plattform“ für Heimanwender oder im Bildungsbereich eingesetzt wird, hat der Raspberry Pi auch im

Automatisierungsumfeld viele Fans und findet in industriellen Varianten bei immer mehr Anwendungen seinen Platz.

VisiWin auf Rasp Pi: Einfacher Einstieg in Linux

Der Raspberry Pi ist vielen als günstige und flexible Plattform bekannt, für die es unzählige Anwendungen und sogar Peripherie-Bastelprojekte gibt. Durch die Lauffähigkeit der ‚VisiWin Cross-Platform Runtime‘ auf dem Einplatinen-Minicomputer möchte Insoft den Einstieg in offene Embedded-Systeme mit Linux erleichtern.

Auch wenn es mit ‚Windows 10 IoT Core‘ eine abgespeckte Microsoft-Betriebssystem-Variante für den Raspberry Pi gibt, laufen die meisten Geräte mit Linux-Betriebssystemen wie ‚Raspberry Pi OS‘ / ‚Debian Linux‘ oder ‚Ubuntu‘. Aufgrund der hohen Popularität, geringen Kosten und einfachen Handhabung ist der „RasPi“ für Insoft das perfekte Mittel, um die Hemmungen gegenüber Linux abzubauen.

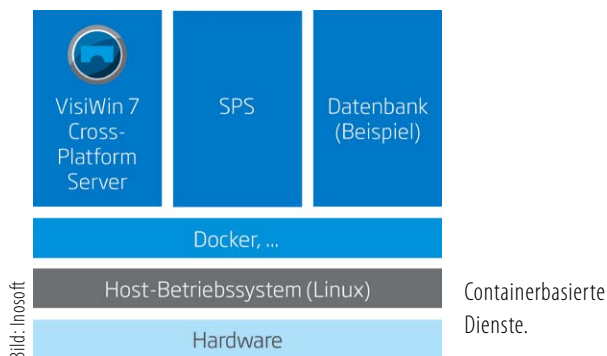
Herausforderung: Software-Installation unter Linux

Eine häufig gestellte Frage lautet: „Wie installiere ich Software unter Linux?“ Während Windows-Nutzer gewohnt sind, mit Installationsdateien wie *.exe oder *.msi zu arbeiten, gibt es bei Linux zahlreiche Möglichkeiten, Software zu installieren: von Paketmanagern wie APT bis hin zur selbstständigen Kompilierung des Codes. In der Automation wird diese Vielfalt jedoch oft mit Skepsis betrachtet. Es entsteht der Eindruck,

Linux sei eine „Lösung für Nerds“. Doch durch die Nutzung von Containern wird dieser Prozess erheblich vereinfacht.

Container: Effizienz und Flexibilität

Container-Technologien wie Docker ermöglichen es, Software vom zugrunde liegenden Betriebssystem zu separieren. Es wird nichts in das Host-Betriebssystem installiert, was die Stabilität anderer Anwendungen gefährden oder bei der Deinstallation ungewollte Fragmente hinterlassen könnte. Der Container enthält alle notwendigen Abhängigkeiten und Konfigurationen, um eine Anwendung isoliert auszuführen. Dadurch können mehrere Instanzen oder verschiedene Versionen derselben Software parallel betrieben werden, ohne sich gegenseitig zu beeinflussen. Der Linux-Kernel spielt dabei eine zentrale Rolle, indem er die Prozesse der einzelnen Container voneinander trennt und so für die notwendige Isolation sorgt. Container-Lösungen werden oft mit virtuellen Maschinen (VM) vergli-



chen. Auch wenn sich die Funktionsweisen ähneln, so gibt es doch grundlegende Unterschiede in der Art und Weise, wie sie Ressourcen nutzen und isoliert sind.

Eine VM läuft auf einem Hypervisor, der auf der Hardware des Host-Systems installiert ist – wie zum Beispiel VMware oder Microsoft Hyper-V. Jede VM enthält ein eigenes, vollständiges Gast-Betriebssystem wie Windows oder Linux und arbeitet isoliert vom Host und von anderen VMs. Da jede VM ein komplett eigenes Betriebssystem für die Anwendungen mitbringt, benötigt sie viel Speicherplatz und Rechenleistung, kann aber unterschiedliche Betriebssysteme parallel und isoliert voneinander laufen lassen. Das sorgt für ein hohes Maß an Sicherheit und Flexibilität, da eine VM auch vom Host-Rechner, also der Hardware, unabhängig läuft.

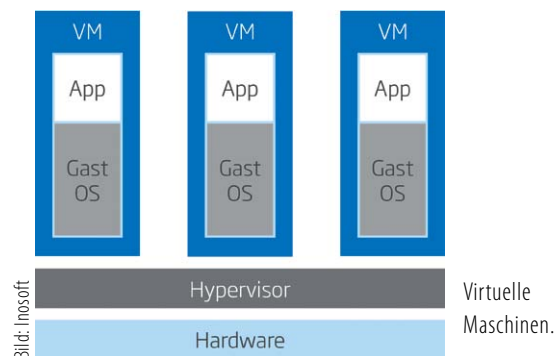
Container hingegen nutzen dasselbe Host-Betriebssystem, das in der Regel eine Linux-Distribution ist. Jeder Container enthält nur die Anwendung und deren Abhängigkeiten wie Bibliotheken oder Konfigurationen, nicht das gesamte Betriebssystem. Dies führt zu einer effizienteren Ressourcennutzung in Bezug auf Arbeitsspeicher und Speicherplatz und lässt die Dienste schneller starten. Container-Dienste wie Docker nutzen den Linux-Kernel und schaffen eine isolierte Umgebung für die Anwendungen, ohne ein vollständiges Betriebssystem pro Container bereitzustellen. Diese Effizienz macht Container für den Einsatz in der Automatisierung attraktiv und auch auf leistungsbegrenzter Hardware mit passiver Kühlung lauffähig.

Verwaltung und Orchestrierung von Containern

Für die Verwaltung und Orchestrierung von Containern gibt es zahlreiche Tools, die Anwendern die Arbeit mit Linux-Diensten vereinfachen sollen. Als Beispiele seien Portainer.io und das unter anderem von Google entwickelte Open-Source-System Kubernetes genannt.

Portainer.io richtet sich vor allem an Einsteiger, die mit Container-Technologien wie Docker arbeiten. Es verfügt über eine grafische Benutzeroberfläche (GUI), die auch ohne tiefgehende Kenntnisse der Kommandozeile eine Container-, Netzwerk- und Volumeverwaltung ermöglicht.

Kubernetes hingegen ist für erfahrene Anwender und komplexere sowie systemkritische Container-Anwendungen gedacht. Dank seiner automatisierten Skalierung und Lastverteilung kann Kubernetes das Maximum aus der zur Verfügung stehenden Hardware holen. Da Kubernetes auch abgestürzte Container selbstständig erkennen und neu starten kann, eignet



es sich vor allem für Anwendungen, die eine hohe Zuverlässigkeit und Ausfallsicherheit benötigen. Es kann sowohl in der Cloud als auch lokal auf eigener Hardware betrieben werden.

Keine Angst vor Linux

Mit der zunehmenden Verbreitung von Linux im industriellen Umfeld wird es immer wichtiger, die Hemmungen vor offenen Systemen zu überwinden. Der Einsatz von Containern, Tools zur Containerverwaltung und Ausfallsicherheit sowie der einfache und kostengünstige Zugang zu (Test-)Plattformen wie dem Raspberry Pi bieten Maschinenbauern und Automatisierern einen praxisnahen Einstieg in die Welt von Linux. Eine der größten Herausforderungen beim Einstieg – die Software-Installation und -Verwaltung – lässt sich dank umfassender Paketlösungen und der Verfügbarkeit der VisiWin Cross-Platform Runtime als Container über Docker Hub sowie die App-Stores Flecs und Nupano einfach bewältigen. Für neue Nutzer stellt Inosoft eine detaillierte Schritt-für-Schritt-Anleitung zur Verfügung, die auch erklärt, wie erste Projekte über die IDE auf das Zielsystem kommen. ag



Stefan Niermann

ist Head of Business Development und Prokurist bei Inosoft.