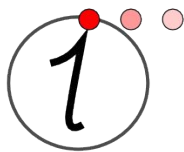


# UMLMON

Ein Monitor für User Mode Linux  
Technische Produktbeschreibung

FÜR VERSION 1.0



Informatikbüro  
Dipl.-Inform. Gerd Stolpmann

Viktoriastr. 45 • 64293 Darmstadt  
<http://www.gerd-stolpmann.de>



# Inhaltsverzeichnis

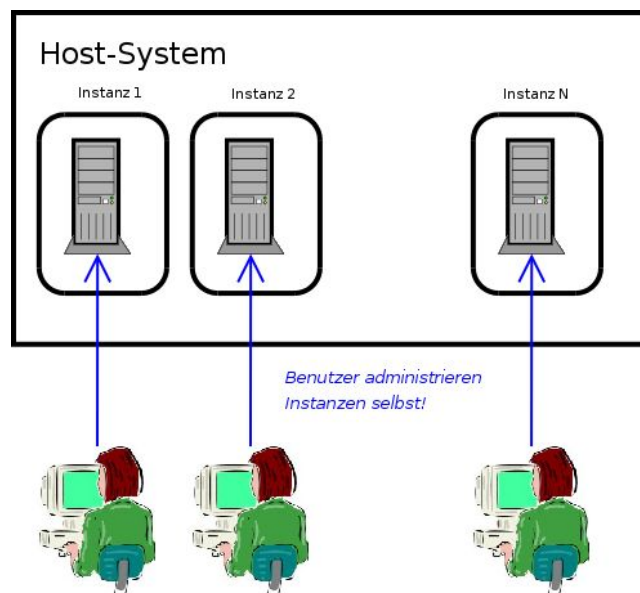
1 Einführung.....	4
2 Probleme beim Einsatz von User Mode Linux.....	5
2.1 Problem: UML ist kein Dämon.....	6
2.2 Problem: UML erlaubt keine Konsolen-Konzentration.....	6
2.3 Problem: UML enthält keine Logging-Funktion.....	6
2.4 Problem: UML hat kein Konzept für Privilegien-Separation.....	7
2.5 Problem: UML kommt ohne Betriebskonzept.....	7
2.6 Problem: UML ist nicht fernsteuerbar.....	7
3 UMLMON steuert User Mode Linux.....	7
3.1 Prozessmodell.....	8
3.2 Konfiguration von UMLMON.....	8
3.3 Administration durch die Benutzer.....	9
3.4 Zentrale Administration.....	10
3.5 Sicherheitsaspekte.....	10
4 Administration mit UMLMON-Web.....	11
5 Notwendige technische Skills.....	11
6 Weitere Informationen.....	12
Daten UMLMON.....	13
Daten UMLMON-Web.....	14

# 1 Einführung

UMLMON ist ein Monitor für den professionellen Einsatz von User Mode Linux in Rechenzentren. User Mode Linux (UML) ist eine Virtualisierungstechnik, bei der mehrere Linux-Instanzen auf einem Linux-Host-System ablaufen können. Der Einsatz von UML ist nicht-trivial, da UML von Hause aus zwar eine Reihe von Konfigurationsmöglichkeiten, aber kaum Management-Funktionen mitbringt. UMLMON ergänzt diese Funktionen.

Die Grundidee von UML ist, dass die virtuelle Linux-Instanz als einfacher, unprivilegierter Unix-Prozess auf dem Host-System abläuft. Jede Linux-Instanz kann individuell konfiguriert werden, ohne dass Administrationsrechte auf dem Host-System benötigt werden. Die einzelnen virtuellen Instanzen sind von einander isoliert. UML erlaubt damit kostengünstiges Hosting, bei dem die Benutzer volle Administrator-Rechte auf der ihnen zugänglichen virtuellen Instanz haben, aber keine solchen Rechte auf dem Host-System benötigen.

Für die Ergänzung der fehlenden Management-Funktionen durch UMLMON wurde von einer bestimmten Aufteilung von Aufgaben zwischen dem Host-Administrator und dem Gast-Benutzer ausgegangen. UMLMON geht diesem Szenario zu Folge davon aus, dass der Host-Administrator nur die Grundzüge der virtuellen UML-Instanzen konfiguriert. Die virtuellen Instanzen sind Linux-Benutzerkonten zugeordnet, die ihre Instanzen weitgehend selber administrieren können, z.B. Starten und Stoppen der UML-Instanzen, Änderung der Plattenkonfiguration etc:



Die Selbstadministration der virtuellen Instanzen geschieht entweder durch das Kommandozeilentool `umlwatch` oder durch das Zusatzprodukt UMLMON-Web. Um `umlwatch` benutzen zu können, ist ein Login auf dem Host-System oder einem

System im selben LAN erforderlich, der mit ssh realisiert werden sollte. Dieses Login kann allerdings stark beschränkt werden (z.B. auf den Aufruf von umlwatch selbst).

Für UMLMON-Web ist kein Login erforderlich. Aus Sicherheitsgründen wird allerdings der Einsatz von SSL (d.h. HTTPS) empfohlen.

UMLMON bietet folgende Features:

- Start und Stopp des User Mode Linux (UML)-Prozesses: UMLMON startet die UML-Instanz und beaufsichtigt den Prozess. UMLMON weiß zu jedem Zeitpunkt, ob die UML-Instanz abläuft oder inaktiv ist. Start und Stopp können in den Boot-Vorgang des Host-Systems eingebunden werden.
- Ablauf von UML in einem chroot-Jail: UMLMON baut einen chroot-Jail auf, in dem der jeweilige UML-Prozess eingesperrt ist. Dies erhöht die Sicherheit beim Einsatz von UML beträchtlich. Durch die „Bauweise“ von UMLMON wird dabei die Konfigurierbarkeit von UML in keiner Weise beeinträchtigt.
- Interaktiver Zugriff auf UML-Konsolen: UMLMON erlaubt es den Benutzern, auf die Linux-Konsolen der virtuellen Instanz direkt zuzugreifen. Die Konsolen können zeitweise auf ein Terminal geschaltet werden, so dass die Benutzer sich in das Gast-System einloggen können, ohne dass Netzwerkfunktionen der virtuellen Instanz benötigt werden. Diese Funktion ist vergleichbar mit einem Terminal-Konzentrator.
- Logging der UML-Konsolen: UMLMON loggt optional die Ausgabe der virtuellen Instanz. Log-Rotation wird unterstützt.
- Benutzer-Management: Die virtuellen Instanzen sind nicht-privilegierten Unix-Benutzern zugeordnet. UMLMON erlaubt es, dass die Benutzer ihre virtuellen Instanzen weitgehend selbst konfigurieren können, d.h. es sind hierzu keine Root-Rechte erforderlich. Durch Privilegien-Separation wird erreicht, dass für das Management erforderliche Administrationsrechte fallweise eingeräumt werden (z.B. Einrichtung eines chroot-Jails).
- Disk-Management: UMLMON erlaubt auf einfache Weise die Administration der virtuellen Festplatten der UML-Instanzen.
- Fernsteuerung: UMLMON kann über eine RPC-Schnittstelle angesteuert werden. Dadurch ist eine Fernsteuerung von einem anderen Rechner möglich. Dies erlaubt es z.B., den Benutzern ein Web-Interface zur Verfügung zu stellen, das zentral für eine ganze Farm von UML-Hosts den Web-Zugang ermöglicht. Ein exemplarisches Web-Interface steht mit UMLMON-Web bereit.

Falls darüber hinaus weitere Funktionen benötigt werden, können auf die Bedürfnisse des Rechenzentrums zugeschnittene Versionen hergestellt werden.

## 2 Probleme beim Einsatz von User Mode Linux

In diesem Kapitel wird der Frage nachgegangen, welche Features das frei

verfügbare User Mode Linux bietet und welche Features für den Rechenzentrumsbetrieb eigentlich benötigt werden. Ergebnis dieser Analyse ist der Funktionsumfang von UMLMON, das ja UML um die fehlenden Funktionen ergänzt.

Auf den ersten Blick scheint der Einsatz von UML einfach zu sein. Auf der Kommandozeile kann man UML mit einem Befehl starten, und einen Augenblick später poppen auf dem Bildschirm die Konsolen-Terminals auf. UML ist mit einer Reihe von Features ausgestattet, die den „Erstkontakt“ zu einem überraschend angenehmen Erlebnis machen. Für den Einsatz in Rechenzentren ist die Ausstattung jedoch ungenügend, und man läuft rasch in eine Reihe von Problemen.

## **2.1 Problem: UML ist kein Dämon**

In der Unix-Welt wird ein im Hintergrund ablaufender Dienst als Dämon bezeichnet. UML ist per se kein Dämon, d.h. die virtuelle Instanz läuft normalerweise im Vordergrund ab, und ein eingeloggter Benutzer wird vorausgesetzt. Diese Einstellung ist im Rechenzentrum natürlich nicht zu gebrauchen; die virtuellen Instanzen sollen natürlich auch ablaufen, wenn gerade niemand eingeloggt ist.

UMLMON führt dagegen die virtuelle Instanz im Hintergrund aus. UMLMON überwacht die virtuelle Instanz, d.h. sorgt für Start und Stopp, und weiß, ob die Instanz gerade läuft.

## **2.2 Problem: UML erlaubt keine Konsolen-Konzentration**

Wie jedes Linux haben auch die virtuellen Instanzen Konsolen, d.h. textorientierte Administrationszugänge. Die Konsolen sind die primäre Möglichkeit, eine Linux-Instanz zu konfigurieren, solange kein Netzwerk-Login möglich ist (z.B. weil das Netzwerk noch nicht konfiguriert oder überhaupt nicht vorgesehen ist).

In der Standard-Einstellung versucht UML, die virtuellen Konsolen dadurch für den Benutzer zugänglich zu machen, dass Fenster geöffnet werden, die dann mit den Konsolen verbunden sind. Alternativ kann man die virtuellen Konsolen auch als Netzwerk-Dienst einrichten. Beide Einstellungen sind im Rechenzentrum nicht zu gebrauchen: Ein Fenstersystem kann nicht vorausgesetzt werden, und die Netzwerk-Option ist unsicher (da unverschlüsselt).

UMLMON ist dagegen ein Konsolen-Konzentrator. Die virtuellen Konsolen werden mit sog. Pseudo-TTYs verbunden, die auf Anforderung des Benutzers mit vorhandenen (echten oder Pseudo-)Terminals weiterverbunden werden können. Der Benutzer kann sich auf diese Weise zeitweise mit einer virtuellen Konsole verbinden (und diese Verbindung auch jederzeit wieder lösen).

## **2.3 Problem: UML enthält keine Logging-Funktion**

Die Konsolen haben neben der Administrationsfunktion auch die Aufgabe, wichtige Log-Meldungen darzustellen.

Die Ausgaben, die die virtuelle Instanz auf die Konsolen schreibt, landen in der Standard-Einstellung von UML entweder im Nirvana, oder sind nur in der interaktiven Sitzung zugänglich. Ein einfaches Logging ist zwar möglich, aber keine Log-Rotation.

UMLMON ergänzt die Logging-Funktion: Alle Ausgaben, die durch den Konsolen-Konzentrator laufen, können auch in Dateien geloggt werden. Die Dateien können selbstverständlich rotiert werden.

## **2.4 Problem: UML hat kein Konzept für Privilegien-Separation**

Obwohl ein UML-Prozess als unprivilegierter Benutzer ablaufen kann, sind für einige Setup-Funktionen dennoch Root-Rechte erforderlich. Ein Beispiel ist die Einrichtung des tmpfs-Dateisystems, das den virtuellen Hauptspeicher simuliert.

UMLMON übernimmt die Setup-Funktionen, die als Root ausgeführt werden müssen, und erlaubt so auch unprivilegierten Benutzern die Benutzung dieser Funktionen. Damit können auch diese Benutzer UML vollständig in Anspruch nehmen.

## **2.5 Problem: UML kommt ohne Betriebskonzept**

In UML gibt es keine Vorstellungen, in welche Verzeichnisse welche Dateien installiert werden, und es ist leicht möglich, die Performance sehr bremsende Konfigurationen einzustellen. Letzteres ist sehr ernst zu nehmen: Durch Falschkonfiguration kann z.B. der virtuelle Hauptspeicher der Gast-Systeme durch eine echte Plattendatei repräsentiert werden. Die Performance sinkt dadurch auf einen Bruchteil des Möglichen.

UMLMON bringt ein Konzept mit, in welchen Dateien und Verzeichnissen Platten, Konfigurationen und Sockets installiert werden. UMLMON erschwert zudem die Falschkonfiguration erheblich, da ein verständlicheres Konfigurationskonzept verwendet wird.

## **2.6 Problem: UML ist nicht fernsteuerbar**

Wenn die Administration der Instanzen über eine Web-Oberfläche erfolgen soll, wird eine Fernsteuerungs-Funktion benötigt, da die Web-Oberfläche in der Regel auf einem anderen Rechner installiert ist. UML bringt diese Funktion nicht mit.

UMLMON implementiert durch die RPC-Schnittstelle die Fernsteuerung.

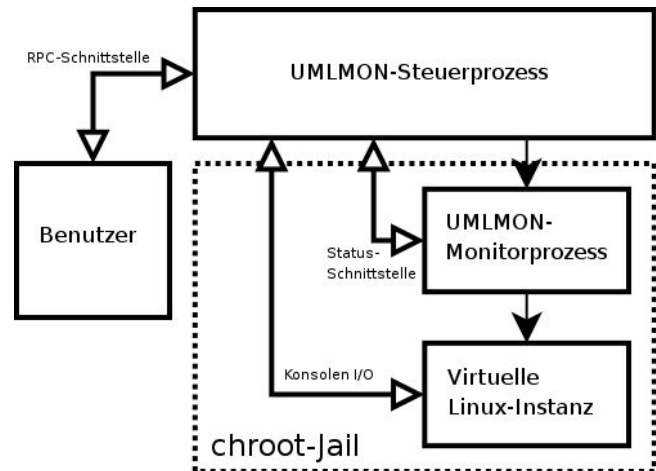
# **3 UMLMON steuert User Mode Linux**

In diesem Abschnitt soll eine technische Beschreibung von UMLMON gegeben werden.

### 3.1 Prozessmodell

Nebenstehendes Diagramm zeigt das Prozessmodell von UMLMON. Normalerweise gibt es drei Prozesse:

- Der Steuerprozess richtet den chroot-Jail ein und sorgt für den Setup der UML-Konfiguration. Um dies durchführen zu können, läuft der Steuerprozess als Root-Benutzer. Der Steuerprozess kann über die RPC-Schnittstelle angesprochen werden.
- Der Monitorprozess überwacht die virtuelle Linux-Instanz, d.h. prüft ob diese noch läuft, und gibt den Status an den Steuerprozess zurück. Der Monitorprozess sorgt außerdem dafür, dass die virtuelle Instanz unter einem nicht-privilegierten Benutzer läuft, d.h. ist für die Privilegien-Separation zuständig.
- Die virtuelle Linux-Instanz ist der eigentliche UML-Prozess. Dieser Prozess kann lediglich auf die Dateien zugreifen, die im chroot-Jail zur Verfügung stehen – das Host-System an sich ist unsichtbar. Die Kommunikation mit der Außenwelt erfolgt über Dateideskriptoren, insbesondere der Konsolen-I/O.



Benutzer können nur mit dem UMLMON-Steuerprozess kommunizieren, der seine Funktionen als RPC-Dienst nach außen darstellt.

### 3.2 Konfiguration von UMLMON

Die Konfiguration von UMLMON erfolgt über eine zentrale Konfigurationsdatei. Es können zusätzliche Konfigurationsdateien angelegt werden, die die Anteile des Setups enthalten, die von den jeweiligen Benutzern modifiziert werden können.

Ein Beispiel für eine Konfigurationsdatei:

```

[vm]
jaildir = /data/uml/vm/jail
logfile = /data/uml/vm/log/monitor.log
vmuser = vm
kernel = /shared/linux
mem = 64
kernelarg = eth0=tuntap,tap2
kernelarg = devfs=nomount
ubd0 = /disks/root.cow
ubd1 = /disks/swap.dsk
ubd2 = /disks/var.dsk
ubd3 = /disks/data.dsk
con0 = pty:/data/uml/vm/log/con0
con1 = pty:/data/uml/vm/log/con1
  
```



```

con2 = pty:/data/uml/vm/log/con2
con3 = pty:/data/uml/vm/log/con3
con4 = pty:/data/uml/vm/log/con4
con5 = pty:/data/uml/vm/log/con5
con6 = pty:/data/uml/vm/log/con6

```

Die Datei hat das bekannte „ini“-Format und ist in Sektionen aufgeteilt, die jeweils eine virtuelle Instanz beschreiben. Hier ist die Sektion für die Instanz „vm“ abgebildet. Die Parameter haben folgende Bedeutung:

- Mit `jaildir` wird festgelegt, welches Verzeichnis für den chroot-Jail benutzt wird. Jede virtuelle Instanz benötigt einen eigenen chroot-Jail. Der Jail wird automatisch aufgesetzt.
- In `logfile` wird die Log-Datei für den Monitor selbst festgelegt.
- Mit `vmuser` wird der Unix-Benutzer festgelegt, als der die virtuelle Instanz ablaufen soll.
- In `kernel` wird der Linux-Kernel angegeben, der gestartet werden soll. (Der Dateiname ist relativ zu `jaildir`. Für ein `/shared`-Verzeichnis gibt es spezielle Unterstützung.)
- In `mem` wird die Größe des Hauptspeichers in Megabyte angegeben. UMLMON erzeugt automatisch ein `tmpfs`-Dateisystem mit der passenden Größe, das in den Jail hinein gemountet wird.
- In `kernelarg` können zusätzliche Argumente an den Linux-Kernel übergeben werden. Die meisten Argumente werden von UMLMON erzeugt und brauchen hier nicht angegeben werden.
- Die `ubd`-Parameter legen die Plattenkonfiguration fest.
- Die `con`-Parameter legen die Konsolenkonfiguration fest, d.h. diese Konsolen sind über den Konsolenkonzentrator zugänglich.

### 3.3 Administration durch die Benutzer

UMLMON enthält das Kommandozeilen-Tool `umlwatch`, mit dem eingeloggte Benutzer auf ihre virtuellen Instanzen zugreifen können. Beispiele für die Benutzung von `umlwatch`:

```
umlwatch -start vm
```

Dieses Kommando startet die virtuelle Instanz namens „vm“. Der eigentliche Startvorgang findet im Hintergrund statt.

```
umlwatch -ctrlaltdel vm
```

An die virtuelle Instanz wird ein CTRL-ALT-DEL-Request geschickt (als ob jemand diese Tasten auf der – nicht vorhandenen – Tastatur gedrückt hätte). Dies ist die normale Vorgehensweise, um die virtuelle Instanz wieder herunterzufahren.

Um sich mit einer Konsole zu verbinden, wird folgendes Kommando benutzt:

```
umlwatch -connect con1 vm
```

Hierdurch wird die Konsole con1 auf das Terminal geschaltet, in dem der Benutzer gerade eingeloggt ist. Der Benutzer kann sich jetzt in die virtuelle Maschine einloggen und dort Kommandos ausführen. Die Verbindung mit con1 wird durch die Tastenkombination Strg-Altgr-9 (d.h. CTRL-]) wieder getrennt.

Mit weiteren umlwatch-Kommandos lassen sich Platten und Konfigurationsdateien administrieren. Die Log-Dateien können ebenfalls abgerufen werden.

umlwatch kann auch als interaktive Shell benutzt werden.

### **3.4 Zentrale Administration**

Wie bereits in der Einführung angedeutet, können die Benutzer nur diejenigen Teile der Administration durchführen, die für die Integrität des Host-Systems unbedeutend sind. Die übrigen Teile der Administration müssen durch die zentrale Administration des Host-Systems erledigt werden. Hierzu gehören:

- Zuweisung von Verzeichnissen: In welchen Verzeichnissen dürfen die Benutzer welche Dateien ablegen?
- Festlegung von Limits
- Regelung des Zugangs zu den Administrationsfunktionen der virtuellen Instanzen
- Konfiguration des Netzwerks

Es gibt die Möglichkeit, Site-Skripte zu hinterlegen, mit deren Hilfe ein Teil dieser Funktionen wieder zurück an die Benutzer delegiert werden kann (z.B. automatische Netzwerkkonfiguration). Im Lieferumfang ist ein Beispiel-Skript enthalten, das es erlaubt, virtuelle Instanzen zu duplizieren.

Für die zentrale Administration steht desweiteren das Kommando umladmin zur Verfügung. Im Kern regelt umladmin den Start und Stopp der virtuellen Maschinen zum Boot- bzw. Shutdown-Zeitpunkt. Es hat aber auch weitere Funktionen, mit denen vor allem die Konfiguration durch Skripte editiert werden können.

### **3.5 Sicherheitsaspekte**

Da die virtuelle Instanz in einem chroot-Jail abläuft, ist es nach heutigem Wissensstand kaum möglich, dass Eindringlinge, die es geschafft haben, in die virtuelle Instanz einzubrechen, auf das Host-System durchbrechen können. Dies schützt effektiv gegen Konfigurationsfehler in den virtuellen Instanzen. (Um durchbrechen zu können, müsste ein Eindringling mehrere Sicherheitslücken zugleich vorfinden und ausnutzen können - ein sehr unwahrscheinliches Szenario.)

Die RPC-Schnittstelle wird normalerweise nur über einen Unix-Domain-Socket verwendet. Dadurch ist eine sichere Authentisierung der Benutzer gegeben. Wenn die RPC-Schnittstelle für TCP freigeschaltet wird, ist ein Passwort-Schutz vorgesehen, der per HMAC abgesichert ist.

UMLMON selbst ist in der Programmiersprache Objective Caml geschrieben. Dadurch wird der Monitor besonders sicher, denn viele Sicherheitslücken sind in dieser Sprache nicht möglich, u.a. Buffer Overflows.

## 4 Administration mit UMLMON-Web

Mit UMLMON-Web wird ein einfaches und recht vollständiges System zur Web-Administration von UMLMON mitgeliefert. UMLMON-Web wird, anders als UMLMON selbst, im Quelltext geliefert, und ist als Programmierbeispiel gemeint.

UMLMON-Web kommuniziert über die RPC-Schnittstelle mit dem UMLMON-Steuerprozess. In der Regel wird die RPC-Schnittstelle für das TCP-Protokoll freigeschaltet, so dass die Web-Schnittstelle auch auf einem anderen Rechner installiert sein kann.

Die RPC-Schnittstelle ist vollständig dokumentiert. Sie hält sich an den ONC-RPC-Standard, der bei Linux auch für die System-RPC-Dienste (z.B. NFS) verwendet wird. Das Web-Interface kann daher in einer beliebigen Programmiersprache verfasst werden, die die Entwicklung von RPC erlaubt (z.B. C).

UMLMON-Web implementiert folgende Funktionen:

- Es ist möglich, sich mit beliebigen UMLMON-Instanzen zu verbinden und diese fernzusteuern, sofern das richtige Passwort hinterlegt ist.
- Die virtuellen Maschinen können gestartet und angehalten werden.
- Es ist Lesezugriff auf die Log-Dateien möglich.
- Die Plattenkonfiguration kann administriert werden.
- Die Web-Konsole ermöglicht das Umschalten auf eine virtuelle Konsole. Bedingt durch das Medium sind allerdings starke Bedienungsabstriche zu machen. Die Konsolenausgabe erfolgt zwar relativ zeitnah, aber bei Tastatureingaben ist die Web-Konsole umständlich und langsam. Dennoch bietet die Web-Konsole einen einzigartigen Zugang, da der Benutzer neben seinem Web-Browser keine Zusatzsoftware benötigt, um sich in das Gast-System einzuloggen.
- Der Hosting-Anbieter kann Zusatzfunktionen anbieten, z.B. die Duplikation von virtuellen Instanzen.

## 5 Notwendige technische Skills

Wer plant, UMLMON zusammen mit UML einzusetzen, sollte wissen, dass trotz der Vereinfachung durch UMLMON immer noch bestimmte Administrationsfertigkeiten unumgänglich sind. UMLMON ist keine click-and-run-Software.

Generell sind Linux-Administrationskenntnisse unumgänglich. Im Einzelnen werden folgende Fertigkeiten benötigt (mit Hilfe einer ausführlichen Anleitung):

- Aufruf eines Installationskripts und Festlegung von Installationsverzeichnissen
- Editierung einer textbasierten Konfigurationsdatei
- Einrichtung von Linux-Benutzerkennungen
- Übersetzung eines Linux-Kernels nach Vorgabe aus der Bedienungsanleitung (Einspielen von Patches, Einhalten bestimmter Konfigurationen)
- Zusammenstellen eines Boot-Images nach Vorgaben aus der Bedienungsanleitung (Boot-Images sind Dateiabzüge existierender Linux-Systeme, die als Gast-Systeme verwendet werden)
- Übliche Linux-Administrationskenntnisse

## 6 Weitere Informationen

Unter <http://www.gerd-stolpmann.de/umlmon> erhalten Sie:

- Darstellung der Produktmerkmale und der Einsatzmöglichkeiten von UMLMON
- Die Bedienungsanleitung von UMLMON
- Eine funktionsbeschränkte Demo-Version von UMLMON
- Die Vollversion von UMLMON-Web inkl. Quelltext

Sie finden dort auch die **Bestellunterlagen**. UMLMON ist ausschließlich direkt bei Gerd Stolpmann erhältlich.

Für weitere Informationen wenden Sie sich bitten an folgende Mail-Adresse:

- [anfrage@gerd-stolpmann.de](mailto:anfrage@gerd-stolpmann.de)

Gerd Stolpmann unterstützt Sie gerne auch persönlich bei der Einrichtung Ihrer UML-Architektur.

# Daten UMLMON

<b>Systemvoraussetzungen:</b>	
	<ul style="list-style-type: none"> <li>● Linux-System für x86, Kernel 2.4 oder 2.6</li> <li>● 256 MB Hauptspeicher oder besser</li> <li>● glibc Version 2.3 oder besser</li> </ul>
<b>Skill-Voraussetzungen:</b>	
	Kenntnisse in Linux-Systemadministration sind unabdingbar
<b>Lieferumfang:</b>	
	<ul style="list-style-type: none"> <li>● UMLMON wird nur elektronisch geliefert (Download), der Anwender muss einen Lizenzschlüssel einspielen.</li> <li>● Enthalten sind der umlmon-Dämon, der umldir-Dämon sowie die Administrationswerkzeuge umladmin und umlwatch. Diese Programme werden nur als Binary geliefert.</li> <li>● Enthalten ist außerdem die IDL-Definition der RPC-Steuerung im Quelltext.</li> <li>● Die Lieferung erfolgt als tar-Datei mit Installationskript. Eine Anleitung kann separat heruntergeladen werden.</li> <li>● Es erfolgt <b>keine</b> Lieferung von Linux-Kerneln mit UML- oder SKAS-Patch. (D.h. die UMLMON-Lieferung enthält nicht UML! Anwender müssen sich gepatchte Kernel selber übersetzen oder aus geeigneter Quelle besorgen.)</li> <li>● Es erfolgt <b>keine</b> Lieferung von bootfähigen Images.</li> </ul>
<b>Lizenzbedingungen</b>	
	UMLMON wird generell mit einer Lizenz geliefert, die den unternehmensweiten Einsatz erlaubt.
<b>Funktionen:</b>	
	<ul style="list-style-type: none"> <li>● Start und Stopp von virtuellen Instanzen, sowohl manuell als auch per Boot-Skript</li> <li>● Logging des Monitors sowie der virtuellen Konsolen, inkl. Log-Rotation</li> <li>● Konsolen-Konzentrator</li> <li>● Privilegien-Separation: Die virtuelle Instanz kann als unprivilegierter Benutzer ablaufen. Der Monitor läuft allerdings immer als root-Benutzer.</li> <li>● Ablauf der virtuellen Instanz in einem chroot-Jail</li> <li>● Die Administration der virtuellen Instanz durch einen unprivilegierten Benutzer ist weitgehend möglich.</li> <li>● Die Steuerung des Monitors per RPC ist möglich (Netzwerk-Steuerung). Das RPC-Protokoll ist dokumentiert.</li> </ul>
<b>Limitierungen:</b>	
	<ul style="list-style-type: none"> <li>● Die Größe des virtuellen Hauptspeichers ist auf ca. 1GB pro Instanz beschränkt.</li> </ul>
<b>Ausführung:</b>	
	<ul style="list-style-type: none"> <li>● UMLMON ist in der Programmiersprache Objective Caml entwickelt worden.</li> <li>● Die parallele Abarbeitung von Anforderungen erfolgt durch klassisches Unix-Multiplexing.</li> </ul>

# Daten UMLMON-Web

<b>Systemvoraussetzungen:</b>	
	<ul style="list-style-type: none"> <li>● Linux-System für x86, Kernel 2.4 oder 2.6</li> <li>● 256 MB Hauptspeicher oder besser</li> <li>● glibc Version 2.3 oder besser</li> </ul>
<b>Skill-Voraussetzungen:</b>	
	Kenntnisse in Linux-Systemadministration und Web-Administration sind unabdingbar
<b>Lieferumfang:</b>	
	<ul style="list-style-type: none"> <li>● UMLMON-Web wird nur elektronisch geliefert (Download)</li> <li>● Enthalten sind übersetzte Fassungen von UMLMON-Web für cgi, sowie der Quelltext. Zur eigenständigen Übersetzung sind weitere Tools und Libraries erforderlich, die nicht Bestandteil der Lieferung sind.</li> <li>● Die Lieferung erfolgt als tar-Datei mit Installationskript. Eine Anleitung ist enthalten.</li> </ul>
<b>Lizenzbedingungen</b>	
	UMLMON-Web wird generell unter den Bedingungen der General Public License geliefert.
<b>Funktionen:</b>	
	<ul style="list-style-type: none"> <li>● Administration von UMLMON-Instanzen, die auf beliebigen im LAN erreichbaren Rechnern ablaufen</li> <li>● Einbindung in einen Web-Server per CGI.</li> <li>● Das Web-Layout kann durch Editieren von XML-Dateien an Kundenbedürfnisse angepasst werden.</li> </ul>
<b>Ausführung:</b>	
	<ul style="list-style-type: none"> <li>● UMLMON-Web ist in der Programmiersprache Objective Caml entwickelt worden.</li> </ul>