

[Webmaster-IMHO](#)

Blog für Webmaster und solche die es werden wollen

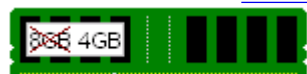
- [Home](#)
- [DoFollow-Blogliste](#)
-

[RSS](#)

[Home](#) > [Allgemein](#), [Windows](#) > Mehr als 4GB RAM mit 32-Bit Windows nutzen

Mehr als 4GB RAM mit 32-Bit Windows nutzen

26. Oktober 2010 [IMHO](#)



8GB unter 32Bit?

Jaa, Speicher ist preiswert geworden und mit 4 GB RAM ausgestattete PCs sind nichts besonderes mehr. Und nicht nur das, immer öfter finden sich in modernen PCs 6, 8, 12 oder 16 GB RAM. Die sich immer mehr auch auf dem heimischen Rechenknecht durchsetzenden 64-Bit Betriebssysteme machen solche Speicher-Mengen nutzbar. Doch was ist wenn man, aus welchem Grund auch immer, doch noch ein 32-Bit System fahren muss? Ist dann bei 4 GB Schluss? Ich meine, gibt es keinen Weg, keinen Trick oder Kniff doch etwas mehr als 4 GB RAM zu nutzen? Doch, es gibt einen Trick, einen ganz einfachen sogar. Und wie der funktioniert möchte ich, samt etwas Hintergrund-Wissen, in diesem Artikel am Beispiel von Windows XP 32 Bit beschreiben.

Warum 32 Bit wenn es 64 Bit gibt?

Wer mehr als 4 GB RAM in seinem PC nutzen möchte dessen erste Wahl sollte natürlich ein Betriebssystem sein, welches auch mehr als 4GB unterstützt und adressieren kann. Aber es gibt eben doch Fälle in denen man gezwungen ist das gute alte Windows XP 32-Bit oder dessen Nachfolger in der 32-Bit Version zu nutzen. Auch bei mir ist das so. Deshalb nutze ich auf meinem PC nicht nur Windows 7 64-Bit sondern parallel auch noch Windows XP 32-Bit. Nötig wird das immer dann wenn noch alte Programme in Verwendung sind welche partout nicht auf 64-Bit laufen. In meinem Fall wären das z.B. diverse ältere VST-Plugins für meinen Sequenzer (Cubase) welche entweder unter 64-Bit ganz streiken oder mit der 64-Bit VST-Bridge nicht zurecht kommen. Von den 8 GB RAM in meinem Computer liegt aber unter Windows XP die Hälfte brach. Und ich seh nun mal nicht gerne Speicher-Riegel die sich langweilen und am gähnen sind. 😊

Etwas technischer Hintergrund:



gewusst wie!

Wer nun sofort erfahren möchte wie man der 4GB-Grenze ein Schnippchen schlägt der sollte jetzt das Mauseisern bemühen und nach unten bis zum Abschnitt "**Windows XP 32-Bit – mehr als 4GB RAM**"

nutzen... Der Trick" scrollen. Alle anderen lade ich, bevor es zur Sache geht, auf einen kurzen Ausflug ein, der Licht ins dunkel bringt und "beleuchtet" warum es die 4GB Grenze überhaupt gibt, was der Unterschied zwischen einem 32-Bit und einem 64-Bit Prozessor ist, warum ein 32-Bit Prozessor eben doch mehr als 4GB nutzen kann und warum es für Otto-Normal-User dann doch nicht geht (gehen soll). Ich versuche das mit verständlichen Worten und stark vereinfacht zu erklären damit es jeder nachvollziehen kann... soll ja hier keine Abhandlung in Fachchinesisch werden. 😊

Der Unterschied zwischen 32-Bit und 64-Bit Prozessoren:

Überall in den Zeitschriften, Prospekten, Foren, Chats, Blogs ist davon die Rede... 32-Bit, 64-Bit... fast jeder gibt seinen Senf dazu aber die wenigsten wissen was genau sich dahinter verbirgt. Wer die 4GB-Grenze aber verstehen will muss einen Blick hinter die Kulissen werfen: Was genau ist also der Unterschied zwischen einem 32 und einem 64-Bit Prozessor?

Ein 32-Bit Prozessor besitzt einen 32-Bit Datenbus und einen 32-Bit Adressbus. Datenbus und Adressbus nennt man die beiden Verbindungen zwischen dem Prozessor und dem Arbeitsspeicher, also dem RAM. Und wie das eben so ist in der Elektronik, besteht so ein Bus im Grunde aus nichts anderem als Leitungen... also Drähten die die "Beinchen" (also Kontakte) des Prozessors mit den Kontakten des RAM verbinden. Der Datenbus und der Adressbus ist also quasi ein "Bündel" aus jeweils 32 "Drähten" zwischen Prozessor und RAM (32 Daten-Drähte und 32 Adress-Drähte). Die Kommunikation zwischen Prozessor und RAM funktioniert nun folgendermaßen: Zuerst "sagt" der Prozessor über den Adressbus (also die 32 Adressleitungen) dem RAM für welche der vielen Speicher-Stellen im RAM die nun folgende Aktion bestimmt ist, und danach liest oder schreibt er ein oder mehrere Zeichen über den Datenbus (die 32 Datenleitungen) in den RAM. Es ist also eigentlich ganz einfach: Der Adressbus sagt wo und der Datenbus sagt was im RAM gespeichert oder ausgelesen wird. Dabei ist der ganze Befehls-Satz eines 32-Bit Prozessors eben auf diese 32 Daten- und Adress-Leitungen ausgelegt. Ein Programmierer muss also mit 32-Bit Befehlen programmieren wenn er ein Programm für einen 32-Bit Prozessor schreibt.



64Bit mal anders...

Ein 64-Bit Prozessor unterscheidet sich nun, wer hätte das gedacht, in der Anzahl der Daten- und Adress-Leitungen von einem 32-Bit'er, sprich: Ein 64-Bit Prozessor hat einfach doppelt so viele "Drähte" die Prozessor und RAM elektrisch miteinander verbinden. Also 64 Daten-Leitungen (Datenbus) und 64 Adress-Leitungen (Adressbus). Und damit diese doppelt so vielen Leitungen auch von Programmen verwendet werden können besitzt eine 64-Bit CPU auch einen dazu passenden 64-Bit Befehls-Satz. Programmierer müssen hier nun mit 64-Bit Befehlen programmieren.

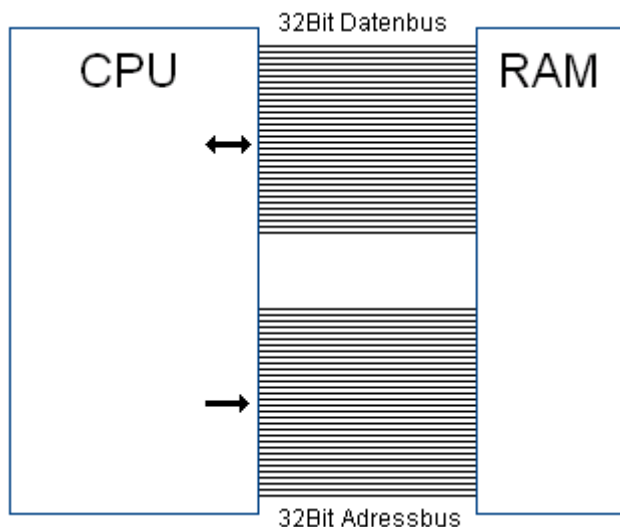
(Die Experten unter Euch mögen mir meine sehr starke Vereinfachung verzeihen 😊)

Die 4 GB Grenze:

Die 4GB-Grenze bei einem 32-Bit Prozessor ergibt sich zwangsläufig aus den 32 Adress-Leitungen über die die CPU mit dem RAM verbunden ist. In den Leitungsverbindungen eines PC passieren nämlich sehr einfache Dinge... entweder ist "Strom drauf" oder es ist "kein Strom drauf". In der Fachsprache sagt man zu "Strom da" auch "1" und zu "Strom nicht da" sagt man "0". Wenn man sich jetzt so eine einzelne Adressleitung vorstellt und überlegt wie viele Informationen man über so einen einzelnen Draht übertragen kann, so ist die Antwort einfach: Man kann 2 Informationen übertragen nämlich "Strom da" (1) oder "Strom nicht da" (0). Ein Prozessor mit einem 1-Bit Adressbus könnte also genau 2 Speicher-Stellen im RAM, also 2 Byte adressieren. Zu wenig um einen Computer zu bauen. Wir brauchen also mehr

Adress-Leitungen. Versuchen wir es mit einem 2-Bit Adressbus, also 2 Drähten zwischen Prozessor und RAM. Wie viele Speicher-Stellen könnten wir nun adressieren? 0,0 (kein Strom, kein Strom); 0,1 (kein Strom, Strom); 1,0 (Strom, kein Strom); 1,1 (Strom, Strom). Es sind also 4 Speicher-Stellen bzw. 4 Byte. Das ist schon doppelt so viel wie mit unserem 1-Bit Adressbus, aber immer noch zu wenig. Versuchen wir es mit einem 3-Bit Adressbus, also 3 Adress-Drähten zwischen Prozessor und RAM. Mit 3 Leitungen ergeben sich folgende Kombinations-Möglichkeiten: 000, 001, 010, 011, 100, 101, 110, 111. Wir könnten mit 3 Bit also 8 Speicher-Stellen im RAM adressieren, das sind wieder doppelt so viele wie mit dem 2-Bit Adressbus.

Wer dieses Spiel weiter führt merkt schnell daß jede weitere Adress-Leitung die Anzahl der adressierbaren Speicherstellen im RAM verdoppelt. Mit einem 4-Bit Adressbus wären es also schon 16 Byte RAM und mit einem 5-Bit Bus 32 Byte. Das Prinzip welches dahinter steckt dürfte damit klar sein, ich kürze das also jetzt mal ab: Mit 8 Bit kann man 256 Byte RAM adressieren, mit 16 Bit sind es 65.535 Byte, mit 24 Bit wären es schon 16.777.216 und mit 32 Leitungen im Adressbus kann man nun satte 4.294.967.296 Byte, oder kurz 4GB RAM adressieren.



32 Datenbus- und 32 Adressbus-Leitungen

An Hand dieser Betrachtung, die jeder mit einem einfachen Taschenrechner nachprüfen kann, ergibt sich eindeutig, unwiderruflich und ohne Zweifel: Mit einem 32-Bit Prozessor der einen 32-Bit Adressbus besitzt kann man maximal 4GB RAM adressieren und nicht ein Byte mehr! Und da haben wir sie... die 4GB-Grenze.

Grenzverletzer... Intels PAE-Trick:

Das Problem, das bei 32-Bit Prozessoren bei 4 GB RAM Schluss ist, war lange nicht störend denn, es ist noch gar nicht so lange her da waren 4GB RAM im PC undenkbar und unerschwinglich. Dennoch hat sich Intel schon vor langer Zeit diesem Problem angenommen und darüber nachgegrübelt wie man diese Grenze sprengen kann. Dabei ist es doch ganz einfach... wir haben ja soeben herausgefunden das es weiterer Adressleitungen bedarf um mehr Speicher zu adressieren. Daran führt kein Weg vorbei! Auch Intel kann nicht zaubern, und so blieb ihnen nichts anderes übrig als dem Pentium (zum ersten mal 1995 im Pentium PRO) weitere 4 Adressleitungen zu spendieren. Ab dem [Pentium PRO](#) (bei AMD seit dem Athlon) besitzen also die in PCs üblichen x86 32-Bit Intel-Prozessoren nicht 32 sondern 36 Adress-Leitungen. Das Ganze nennt sich [PAE, was die Abkürzung für "Physical Address Extension"](#) oder zu deutsch "physische Adress-Erweiterung" ist. Das gilt übrigens auch für die 64-Bit Prozessoren von Intel und AMD wenn sie im 32Bit-Modus laufen, sich also wie ein "normaler" 32-Bit Prozessor verhalten. Spitzfindig betrachtet haben wir es also seit dem Pentium PRO schon nicht mehr mit einem reinrassigen 32-Bit Prozessor zu tun denn der Befehls-Satz und der Datenbus ist 32 Bit, der Adressbus dagegen 36 Bit breit.

Und wie viel Speicher kann man jetzt mit 36 Bit adressieren?

Naja, wie oben schon beschrieben... jede Adressleitung verdoppelt den adressierbaren Speicherbereich. Mit 32 Bit sind es noch 4GB, mit 33 Bit wären es schon 8GB, 34 Bit = 16GB, 35 Bit = 32 GB und mit einem 36-Bit Adressbus können wir wahnsinnige 64 GB RAM adressieren. Und das seit dem Pentium PRO von 1995. Im Prinzip ist es also schon seit Mitte der 90er Jahre möglich mit einem normalen PC-Prozessor ganze 64 GB RAM zu verwalten. Mal ganz davon abgesehen das auch das Motherboard alle 36 Adressleitungen und entsprechende Speicher-Steckplätze unterstützen muss.

Also doch mehr als 4GB RAM mit 32-Bit Windows?

Ja, so ist es! Aber es gibt einen Haken. Seit Windows 2000 unterstützt das Microsoft Betriebssystem die PAE-Erweiterung der 32-Bit Intel- und AMD-CPU's. Seit Windows 2000 kann Windows also mit PAE und dadurch mit bis zu 64 GB RAM umgehen. Und nun der Haken: Microsoft hat die Unterstützung von mehr als 4GB RAM in den 32-Bit Windows-Systemen nur in den Server-Varianten freigeschaltet, in den Arbeitsstation-Varianten aber deaktiviert. Sprich: "Windows 2000" ab "Advanced Server", "Windows-Server 2003" ab "Enterprise" und "Windows-Server 2008" ab "Enterprise" können mehr als 4GB Speicher verwenden. Windows 2000, XP, Vista und Windows 7 können es nicht. Der Grund: Produkt-Politik seitens Microsoft und sonst nichts! Es gibt keinen Hardware- oder Software-technischen Hinderungsgrund der dazu führt daß die normalen 32-Bit Windows Desktop-Varianten nicht mehr als 4GB adressieren könnten! Es geht nur deshalb nicht weil Microsoft es so will.



Auch Windows XP 32-Bit kann PAE!

Und mehr als 4GB RAM unter 32-Bit Linux?

Unter Linux ist die Unterstützung der PAE-Erweiterung seit Kernel 2.3.23 vorhanden. Linux kann also mit mehr als 4GB RAM umgehen. Ohne künstliche Einschränkungen wie bei Windows. Linux-Anwender die noch einen 32-Bit Linux-Kernel nutzen müssen oder wollen und deren Board mehr als 4GB RAM unterstützt haben also an dieser Stelle ein Problem weniger. Ich sehe förmlich die Linux-User unter Euch die dies mit einem Lächeln im Gesicht lesen und denken: "Sag ich doch!" 😊

Windows XP 32-Bit – mehr als 4GB RAM nutzen... Der Trick:

Es soll zwar eine Möglichkeit geben die Unterstützung von mehr als 4GB RAM mittels PAE unter 32-Bit Windows freizuschalten, schließlich können es die Server-Varianten und auf dem Desktop ist es nur deaktiviert, aber das ist aufwendig und nicht ganz ungefährlich. Es müssen System-Dateien gepatcht oder ausgetauscht, in der Registry herum-geändert und am Kernel geschraubt werden. Sicher nur was für die

Freaks unter uns, und ob Windows Updates und Patches dann noch funktionieren darf stark bezweifelt werden.

Aber es geht auch viel einfacher...: Was macht das Windows-System wenn der Arbeitsspeicher nicht ausreicht? Richtig! Es nutzt virtuellen Speicher, also die Auslagerungsdatei. Auf diese Art kann Windows auch jetzt schon "quasi" mehr Arbeitsspeicher nutzen, wenn auch über den Umweg der Auslagerung. Die Auslagerung auf Festplatte ist allerdings ein zeitaufwändiges Unterfangen, weshalb der virtuelle Speicher eben wesentlich langsamer ist als der echte RAM. Wenn man es also schaffen könnte den Auslagerungs-Speicher so schnell zu machen wie den normalen RAM, dann hätte man den virtuellen Speicher quasi zu "echtem" RAM aufgewertet. Bewerkstelligen kann man dies recht einfach mit einer RAM-Disk welche den Speicher oberhalb 4GB nutzt. Auf diese RAM-Disk packen wir nun die Windows-Auslagerungs-Datei. Und schwupp-di-wupp wird auch der Speicher oberhalb der 4GB-Grenze genutzt.

So wird's gemacht:

Für diese kleine Anleitung gehe ich von einem aktuellen Rechner-System mit 8GB RAM und Windows-XP SP3 32-Bit aus. Zuerst laden wir uns die [VSuite-Ramdisk FreeEdition](#) herunter. Die FreeEdition hat zwar einige Einschränkungen, ist aber kostenlos und für unsere Zwecke komplett ausreichend. Nach der Installation geht es ans Einrichten der RAM-Disk. Da wir ja den Speicher der für Windows nicht erreichbar ist nutzen wollen, also alles was sich oberhalb der 4GB RAM befindet, aktivieren wir zuerst die Option "*Enable OS Invisible Physical Memory*".



RAM-Disk-Speicher oberhalb 4GB

Nun wird die eigentliche RAM-Disk angelegt. In der FreeEdition kann diese maximal 4GB groß sein, was für unsere 8GB im PC ja genau richtig ist. Wir geben also bei "*Disk Size*" 4096 ein und wählen einen Laufwerks-Buchstaben, bei mir wäre das Laufwerk F. Als File-System wählen wir natürlich NTFS und geben diesem auch gleich einen Namen... "RAMDISK" bietet sich hier an. Die *NTFS-Kompression* sollte abgeschaltet sein. Wichtig für unseren Zweck ist der Haken bei "*Use OS Invisible Memory*". Dadurch wird diese RAM-Disk in dem vom Windows nicht nutzbaren Speicher oberhalb 4GB eingerichtet. Die Option "*Enable Image File*" sollte deaktiviert bleiben, dadurch wird die RAM-Disk bei jedem Windows-Start neu und leer angelegt. Jetzt noch speichern, neu starten und schon haben wir eine RAM-Disk.



RAM-Disk für Auslagerungs-Datei erstellen

Nun müssen wir Windows noch sagen das die Auslagerungs-Datei von nun an auf der RAM-Disk liegen soll, bei mir also Laufwerk F. Dazu machen wir einen Rechts-Klick auf "Arbeitsplatz" -> "Eigenschaften" -> Register "Erweitert" -> Systemleistung: "Einstellungen" -> "Erweitert" -> Virtueller Arbeitsspeicher: "Ändern". Hier wählen wir nun die RAM-Disk aus und aktivieren "Benutzerdefinierte Größe". Als Anfangs- und Maximal-Größe empfehle ich einen Wert der knapp unter der Größe der RAM-Disk liegt, 4064 MB wäre ein gute Größe. Dann ein Klick auf "Festlegen". Nun wählen wir Laufwerk C: aus und deaktivieren die Auslagerungsdatei mit dem Punkt "keine Auslagerungsdatei". Wieder ein Klick auf "Festlegen", dann "OK" -> "OK" -> "OK" und fix den Rechner neu gestartet. Wenn alles geklappt hat nutzt Windows nun die Auslagerungs-Datei auf der RAM-Disk.



Auslagerungs-Datei auf der RAM-Disk

Auf diese Weise können wir fast die ganzen im PC befindlichen 8GB RAM unter Windows auch nutzen. Nicht umsonst nennt Windows ja die Auslagerungs-Datei "virtuellen Arbeitsspeicher", denn er erweitert tatsächlich den in Windows nutzbaren Speicher. Nur das dieser nun nicht mehr "Festplatten-langsam" sondern "RAM-schnell" ist. Sicher dürfte als erstes auffallen das wesentlich weniger Festplatten-Zugriffe stattfinden. Klar, es wird ja nun nichts mehr auf die Platte sondern eben in unsere RAM-Disk, und damit in der Speicher oberhalb der 4GB ausgelagert. Auch dürften Programme nun etwas schneller starten und überhaupt... man hat den Eindruck alles läuft irgendwie flüssiger. Jedenfalls war das mein erster Eindruck nachdem ich die Auslagerungs-Datei auf die RAM-Disk verlagert hatte. Leider hält dieser Effekt nicht lange an denn... man glaubt gar nicht wie schnell man sich daran gewöhnt und es dann nicht mehr wahr

nimmt.

Fazit:

Es ist schon ärgerlich ein Betriebssystem sein eigen zu nennen welches ein Feature (PAE-Unterstützung) aufweist das dann aber doch aus rein Produkt-Politischen Gründen "beschnitten" ist. Aber klar, wer kauft noch teure Server-Betriebssysteme wenn die Desktop-Version das Selbe kann. Aus kommerzieller Sicht verständlich, aus User-Sicht einfach nur Mist! Mit Hilfe einer RAM-Disk den Speicher oberhalb der 4GB zu nutzen ist sicher ein gangbarer Weg für alle, die sich von 32-Bit nicht trennen können oder wollen. Für alle anderen wäre der Umstieg auf ein 64-Bit Betriebssystem logischer Weise die bessere Wahl. Aber selbst unter 64-Bit kann eine RAM-Disk für die Auslagerungs-Datei oder auch für das Temp-Verzeichnis sinnvoll sein. Wer also gerne mal was probiert sollte das Thema "RAM-Disk" unbedingt auf seine ToDo-Liste setzen. Denn... wie immer geht probieren über studieren und eigene Erfahrungen sind immer besser als jede Anleitung. 😊

Google Anzeigen: [Windows XP](#), [4GB RAM](#), [Aber 3GB RAM](#), [RAM Disk](#)

Kategorien [Allgemein](#), [Windows](#) Tags: [4 GB Grenze](#), [4GB RAM](#), [RAM-Disk](#), [Windows XP 4 GB RAM](#)
[Kommentare \(4\)](#)

1.



[H.-Peter Pfeufer](#)

30. Oktober 2010, 05:33 | [#1](#)

[Antwort](#) | [Zitat](#)

Ich sehe förmlich die Linux-User unter Euch die dies mit einem Lächeln im Gesicht lesen und denken: "Sag ich doch!"

Verdammt, erwischt 😊

OK, zurück zum Ernst der Lage.

Ich bin selbst schon durch diverse Rechner im Bekanntenkreis auf diese ominöse 4GB Grenze gestoßen und habe bisher nie wirklich eine Erklärung parat gehabt. Aber auch keine Lösung. Und dann lese ich hier das Stichwort "RAM-Disk" und musste mir erst mal an dem Kopf fassen mit der Frage, wieso ich da nie drauf gekommen bin. Mit den Dingen hat man doch zu DOS-Zeiten auch gerne mal rumgespielt.

Gut, man muss zugeben, das war wieder mal viel zu naheliegend diese für so etwas zu nutzen.

Danke für diesen Artikel, ich hab nun einige PCs zu "tunen" 😊

Grüße

Peter

2.



IMHO

30. Oktober 2010, 21:59 | [#2](#)

[Antwort](#) | [Zitat](#)

Du hast Recht, an RAM-Disks erinnert sich heut kaum noch jemand. Dabei gibt es einige schöne Einsatzmöglichkeiten.

Was die 4GB-Grenze betrifft... Ich find es immer wieder sehr spannend aufzudecken was in einem Computer wirklich abläuft wenn der normale User "Klick" macht. 😊 Aber naja, ich komm ja auch noch aus einer Zeit als es den Z-80 gab, mit 8 Bit Daten und 8 Bit Adressbus. Übrigens gab es damals die "64kByte-Grenze" welche auch mit diversen Tricks (Bank-Umschaltung) umgangen wurde. Aber zu der Zeit hab ich auch noch Leiterplatten geätzt, Löcher für die Chips gebohrt, und die 40-Pin Z-80 CPU, PIO, SIO und RAM-Chips von Hand eingelötet... lang ist's her. Und mach

das jetzt mal mit ner 1000-Pin CPU von heute... 😊



3.

Alexander

30. Oktober 2010, 22:14 | [#3](#)

[Antwort](#) | [Zitat](#)

Vielen Dank für diese Erklärung. Bin seit einigen Wochen am suchen was ich aus meinem schon etwas in die Jahre gekommenen Rechner rausholen kann ohne mir einen neuen zu kaufen, denn das Problem ist bei mir nicht der Prozessor sondern schon immer der Arbeitsspeicher, da ich mit vielen Programmen gleichzeitig arbeiten muss. Durch deinen Tipp werde ich das morgen gleich beheben.



4.

Bernard

5. November 2010, 17:42 | [#4](#)

[Antwort](#) | [Zitat](#)

Danke für die Anleitung

hab in mein Macbook vor kurzem aufgerüstet und das mit bootcamp installierte XP läuft jetzt wesentlich flüssiger als vorher (kein Wunder. waren vorher nur 2GB Ram).

OS X hatte sowieso keine Probleme mit den 8 GB (im aktuellen Macbook wahrscheinlich erst darüber. laut hersteller verkraftet das system auch nur 4GB, haben aber keine eingebaute sperre)

Kommentare sind geschlossen

[Das Google Page Rank Update ist da... Parted Magic: Linux Partition-Manager](#)

Neueste Artikel

- [PHP 5.3.4 und PHP 5.2.15 erschienen](#)
- [HTTP-Protokoll: HTTP-Tutorial Teil-2](#)
- [HTTP-Statuscodes: HTTP-Tutorial Teil-1](#)
- [Linux Mint 10: Codename Julia](#)
- [Google Instant Preview](#)

Categories

- [Allgemein](#)
- [FTP-Server und Client](#)
- [Linux](#)
- [Offtopic](#)
- [PHP](#)
- [Programmierung](#)
- [SEO](#)
- [vServer und Rootserver](#)
- [Webmaster und Co.](#)
- [Webserver](#)
- [Webspace und Hosting](#)
- [Windows](#)

Blogroll

- [Dokumentation](#)
- [FAQ](#)

- [News-Blog](#)
- [Support Forum](#)
- [Themepool](#)
- [WordPress-Planet](#)

Schlagworte

[301 Redirect](#) [301 Umleitung](#) [301 Weiterleitung](#) [Abkürzungen](#) [Apache](#) [Atari](#) [Atari-Emulator](#) [Atari-Simulator](#) [Atari Janus](#) [Atari ST](#) [Betriebssystem](#)
[Blog](#) [Blog-Pause](#) [dofollow](#) [Domain Editor](#) [FileZilla](#) [FTP-Server](#) [GEM](#) [htaccess](#) [Janus 020](#) [Javascript](#) [kostenlos](#) [Linux](#) [Linux Mint 9](#) [Linux Mint Isadora](#)
[mod_rewrite](#) [Netzjargon](#) [nofollow](#) [Offtopic](#) [open-basedir](#) [PHP](#) [PHP-Sicherheit](#) [php.ini](#) [Piwik](#) [Piwik Analytics](#) [Piwik Installation](#) [Piwik](#)
[Plugins](#) [Piwik Wordpress](#) [Script](#) [SEO](#) [TOS](#) [Webmaster](#) [Webseiten-Analyse](#) [Webserver](#)
[Oben](#) [WordPress](#)
[Impressum](#)

Copyright © 2010-2011 Webmaster-IMHO «» Theme by [NeoEase](#). Modified by OllySoft for Webmaster-IMHO.