

マイクロサーバをベースとしたアプライアンス製品

あなたに代わってネットワーク上の機器を自動監視・通知

監視BlockS Pro

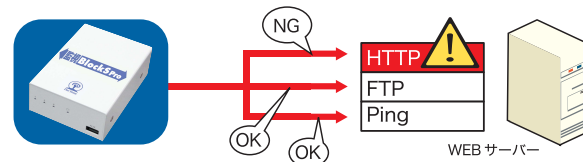
型番 KAPRO266/512/V6
WEB・直販価格 128,000円(税込) (本体価格 121,905円)
メーカー希望小売価格 オープン



監視BlockS Pro は、ネットワークにそのまま接続し設定するだけで、部門単位の小規模システムから監視対象100台超の大規模システムまで、簡単・低コストで導入・管理ができる監視アプライアンスです。

- 監視対象機器はメーカー・機種を問わず
- 駆動部品を排除した、コンパクトで堅牢な設計
- エージェントや専用ハードウェアの追加不要
- 運用・管理の大幅なコスト削減
- 日本語WEB設定ツールで柔軟な設定が可能
- 大規模システム向けのクオリティを容易に導入

監視例：ポートによるサービスの死活監視



- pingによる「機器の死活監視」
- 「SNMP OID値」による監視
- プロトコルレベルでの監視 (SMTP/POP3/IMAP)
- 「SNMP TRAP」受信による通知に対応
- 「認証」機能付きメールサーバーに対応
- 「MRTG」によるグラフ表示と閾値監視

国産L2-VPNソフトウェア「PacketiX VPN 2.0」を搭載
セキュアなVPNをワンストップで導入

PacketiX BOX

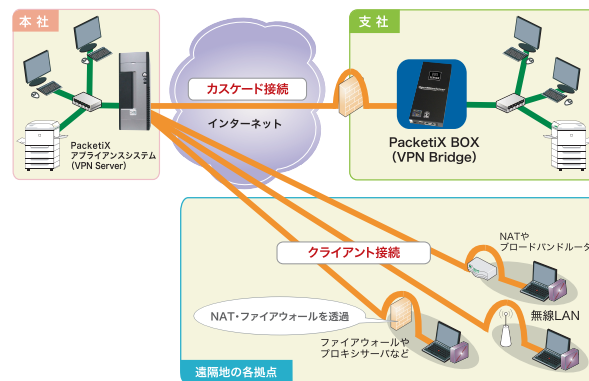
型番 P2-S/MS400
WEB・直販価格 153,200円(税込) (本体価格 145,905円)~
メーカー希望小売価格 オープン



PacketiX VPN 2.0は、SSL-VPNの持つネットワーク透過性とIPSec-VPNの持つ柔軟性を併せ持つ、新しいVPN技術です。インターネット (WEB) が閲覧可能であれば、どんな環境でもVPNを確立することが可能です。他のVPNソリューションで必要となる、ルーター・ファイアウォール等への特別な設定の追加や既存のネットワークの再設計、使用アプリケーションの再開を一切必要とせずに、容易かつ柔軟にVPNを導入できます。

- どんなアプリケーションでも利用可能
- あらゆる構成のネットワークを柔軟に構築可能
- 環境を選ばない接続性
- 高い安定性と通信効率

接続例



プロキシ型セキュリティアプライアンス

セキュアPROX

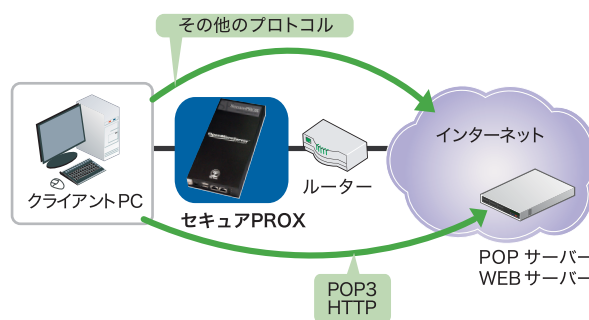
型番 SPROX400AL 他
WEB・直販価格 138,000円(税込) (本体価格 131,429円)~
メーカー希望小売価格 オープン



セキュアPROXは、WEBアクセス/E-mailの各種プロトコルに対応し、プロキシ型・ブリッジ型で利用できるセキュリティアプライアンスです。ウイルスメール対策・スパムメール対策・悪意のあるWEBサイトやドラッグ・アダルト系サイトなどの様々な有害コンテンツ対策に高い効果を発揮します。

- 万全のウイルス/スパイウェア対策
- 最小限のネットワーク設定で簡単導入
- PCの負荷を軽減し、トータルコストを削減
- 機能的で扱いやすい迷惑メール管理機能

ブリッジ型接続例



オープンソース思想に基づいた小型Linuxサーバ

マイクロサーバ シリーズ 2008.05

Plat'Home

TECHNOLOGY to serve you.

ラインナップ・マイクロサーバとは ... 01

ソフトウェアのアプライアンス化で、
導入提案を魅力的に ... 02

内部構造 ... 03

起動プロセスの詳細 ... 04

特集：実践OpenMicroServer
即運用術 - LAMP環境の実装 ... 05

製品仕様 ... 13

ソフトウェア仕様 適合・準拠規格
オプション ... 14



	TEL	FAX	メール	担当
法人のお客様窓口	03-3251-2600	03-3251-2602	sales@plathome.co.jp	営業部
オンライン販売	03-3251-6151	03-3251-6173	biz@plathome.co.jp (売掛)	オンライン営業チーム
故障・修理・技術的な質問	03-3251-6116	03-3251-0629	support@plathome.co.jp	カスタマーケア課
代表	03-3251-6111	03-3255-9506	-	管理部

本カタログに掲載の商品については、当社営業部およびオンラインショップにてご購入・ご相談いただけます。まずは上記まで、お気軽にお問合せください。

記載の価格はすべて2008年5月現在のものであり、変更される場合があります。(最新の価格についてはお問い合わせください) / 製品の仕様・外観は、予告なく変更される場合があります。

法人のお客様お問い合わせはこちらまで>>

TEL 03-3251-2600 FAX 03-3251-2602
sales@plathome.co.jp
http://www.plathome.co.jp/

ラインナップ

シリーズ累計出荷台数40,000台超
優れた耐環境性でアプライアンスに最適 オープンな設計の小型Linuxサーバー

優れた耐環境性能 利用範囲の広い仕様

OpenMicroServer™

型番 OMS-AL400/128
WEB・直販価格 56,400円(税込) (本体価格 53,715円)
メーカー希望小売価格 オープン

■ギガビットイーサを2ポート搭載
(10/100を1ポート、計3ポート搭載)

■PoE受電可能

■CPU: Alchemy au1550(mipsel) 400MHz, MEM: 128MB

RoHS
指令適合

3年保証
センドバック

動作保証周辺温度
0~50℃
(PoE受電時)
(ACアダプタ使用時40℃保証)



■USB 2.0を2ポート搭載

■センサ制御用 Digital I/O 各4ポート

シンプルな設計 手のひらサイズのロングセラー

OpenBlockS™ 266

型番 OBS266/128/16R
WEB・直販価格 45,800円(税込) (本体価格 43,620円)
メーカー希望小売価格 オープン

■2.5インチ SSD(Solid State Drive)搭載可能

■CPU: PPC 405GPr(powerpc) 266MHz, MEM: 128MB

RoHS
指令適合

3年保証
センドバック

動作保証周辺温度
0~40℃



マイクロサーバとは

オープンソース思想に基づいた設計でシステムをトータルにサポート

アプライアンス作成にマイクロサーバが最適なのは、他社製の組み込み系ボードや小型PCと一線を画す、アプライアンスプラットフォームのための開発思想を備えているからです。

■製品化に手間のかかる組み込み系ボード・小型PC

種類や基板サイズに幅があり、拡張性の広さにも優れる組み込み系ボード。しかし、それをアプライアンスとして製品化するには、他の部品を組み合わせたケースに格納した上での、熱処理や冷却構造の再検証が必要です。そのため組み込み系ボードから24時間365日連続運用可能な筐体を作り、製造体制の構築や電波・電源関連の規格適合、RoHSなどの環境規格適合を行うには長い時間と大きなリソースを必要とします。あるいは小型PCを用いる場合も、スペックは高いものの、ファンやHDDといった回転部品の存在による発熱、消費電力、消費スペースの問題が避けられません。

また、いずれの例でも、ハードウェア・OS・開発環境のそれぞれが別々のメーカー製のものとなるために、ソフトウェア開発者の負担が増大するという難点があります。

コンパクト、低コスト、一切の無駄を排除

現在は、DB用途など高速処理の必要なサーバーのほか、部門内の情報共有Wikiといった大きな負荷のかからないサーバーも多く利用されています。こうしたサーバーには、導入コストも消費電力も大きい通常のサーバーハードウェアは適していません。

マイクロサーバなら、必要十分なスペックを備えつつ、OSには独自デ

■アプライアンスのために生まれたマイクロサーバ

ソフトウェア開発者の視点で生まれたマイクロサーバシリーズは、ハードウェア・OS・開発環境というシステム全体をトータルで提供し、サポートします。ゼロスピンドルで低発熱、低消費電力、高耐久性を実現しているため、サーバーラームはもちろん、オフィス内・OAフロア・MDF(配電盤)内などにも柔軟に設置可能。国内外の主要な規格にも適合し、ごく高いパフォーマンスを必要とする高負荷DBなど以外の全てのアプライアンス作成に、最適なソリューションを提供します。

	マイクロサーバ	組み込み系ボード	小型PC
OS対応	Linux	Windows、Linux他 別途必要	Windows、Linux他 別途必要
開発環境	無償(製品に同梱)	別途必要	別途必要
堅牢性	0スピンドル	ボード単体販売	FAN、HDD
設置場所	OAフロアやMDF等	ボード単体販売	オフィス等
消費電力	低い	ボードによる	高い

ィストリビューション(SSD/Linux)を採用、無駄なライセンス料も発生しません。導入コストも運用コストも最小限に抑え、筐体も消費電力も非常にコンパクト。様々な無駄を排除し、グリーンITを目指す企業にも最適なサーバーハードウェアです。

数多い実績が示す、高い堅牢性

監視カメラや自動販売機などに合わせてサーバーが設置される例も、近年ではよく見られます。こうした用途では、周辺環境の整った専用サーバーラームでの用途とは異なり、ビルの配線ボックスや床下といった環境でも問題なく動作する、堅牢なサーバーが必要とされます。一切の回転部品を排除、開口部も持たず高い堅牢性を誇るマイクロサーバシリーズは、様々な環境でも問題なく設置・運用が可能です。



ビルのMDFへの設置



オフィスの床下へ

超低消費電力の省エネ仕様

近年は省エネに配慮された汎用サーバーが多く市場に出回っていますが、それら1台の消費電力が80W*程度(当社調査)とすると、超低消費電力で稼働のマイクロサーバシリーズなら、同じ消費電力で10台が稼働できます。また、持ち運びできる小型軽量サイズを実現しています。
* CPU: Intel E6300 1.86GHz, MEM: 4GB, HDD: 120GB x 1 の場合

PoE給電対応、電源のない場所でも設置可能*

PoE(Power over Ethernet)給電が可能なネットワークスイッチを利用すれば、ACアダプタなしでも、ネットワークケーブル経由での電力確保が可能。サーバーを設置できる環境が大きく広がります。

*OpenMicroServerのみ



■PoE対応ネットワークスイッチ 参考商品
メーカー: MNO 型番: MN27059

屋外の筐の中など、高温環境(動作環境50℃まで)に設置可能

ソフトウェアのアプライアンス化で、導入提案を魅力的に

ソフトウェアでのサービス提供は、導入障壁があります。

- 顧客のハードウェアやOS・他アプリケーションとの関連など導入環境に起因する問題
- インストールをはじめとした初期設定にかかわるサポートの問題
- 不具合発生時における切り分け作業や対応の問題



マイクロサーバでアプライアンス化すると...

- 顧客の導入環境に依存せずに、様々な提案が可能
- 初期設定済で出荷されるため、顧客は複雑なサポートを必要とせず短時間かつスムーズに導入が可能
- 高耐久でメンテフリー、ハードウェア不具合対応などを行う技術スタッフの増員や新たに設備追加は必要なし

ソフトウェアを堅牢な小型アプライアンス化することで、より魅力的な導入提案で、より多くの顧客へアプローチ可能!

■使用例

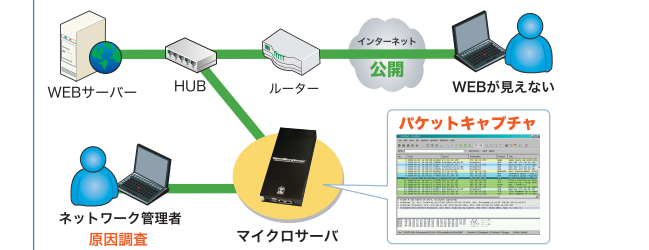
コンテンツマネジメントシステム(CMS)

専門知識不要のWEB作成アプリケーションツールとして



ネットワークモニタリング

パケットキャプチャによる通信状況や障害時の原因解析に



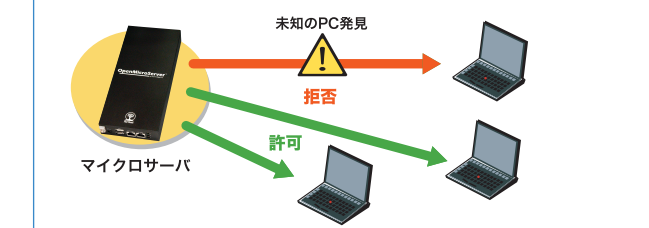
バージョン管理システム

簡易版SCM(SourceCodeManagement)として



ネットワークアクセスコントロール(NAC)

未知のPCのネットワーク進入防御・阻止システムとして



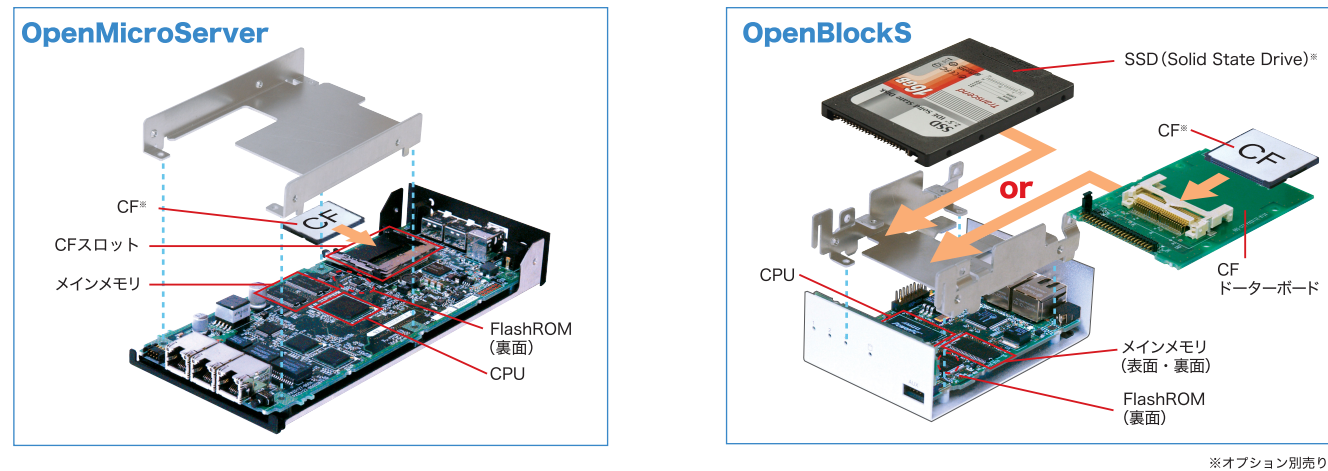
アプライアンスのベースとなるLAMP環境の実装チュートリアルを5ページで紹介しています。

※ふらっとホームのウェブサイトで、その他のシステム実装チュートリアルもご覧いただけます。

内部構造

マイクロサーバシリーズは、熱流体解析による緻密な部品配置を行い、かつ発熱量の多いCPUやNIC部分には筐体全体へ直接廃熱するヒートシンクを実装しています。これにより、高効率な廃熱を達成し「ファンレス」「半密閉構造」を実現しています。

また、ストレージ以外の全ての機能(チップ)をメインボードに直接実装しているため、CPUやメモリ等の組み合わせによる相性の問題や、コネクタによる接触不良が発生しません。数MB(OpenMicroServer:3.64MB、OpenBlockS:2MB)のソフトウェアであれば、ストレージを用いずオンチップで完結した、高耐久かつ高信頼なシステムの構築が可能です。



※オプション別売り

■エージング試験 (OpenMicroServer・OpenBlockSとも無故障連続稼働中)

各モデルとも、連続エージング試験を継続しており、いずれも無故障での連続稼働時間を更新し続けています。

【試験内容】

- ・CPU/メモリ負荷:RAMディスク上のファイルをtarおよびgz圧縮し、/dev/nullに書き込み
- ・ネットワーク負荷:全てのネットワークポート(3ポート)から常にFTPによるデータ通信実行
- ・HDD未搭載

(2008年5月1日現在)

対象機器	OMS-AL400/128 × 10台	OBS266/128/16R × 10台
温度環境	43~50°C	35~40°C
連続稼働時間	819日/19,656時間※1	338日/18,088時間※1※2

※1 設置場所の法定点検による停電、ファームウェアバージョンアップなどのメンテナンス作業を除く
 ※2 エンジニアリングサンプルでの検査期間を含む

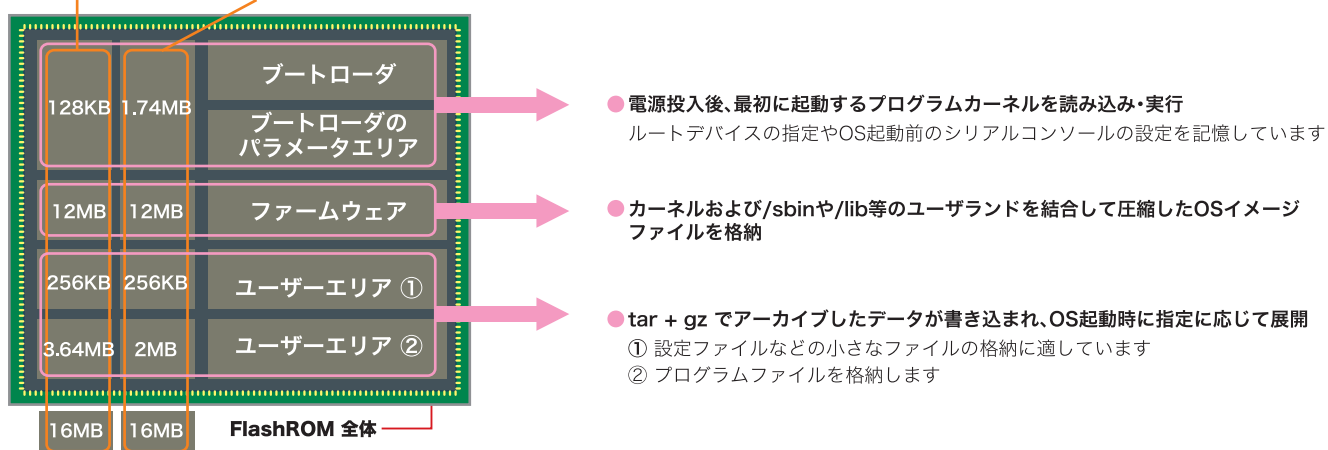


FlashROMとは

FlashROMとは、マイクロサーバの起動に必要なソフトウェアが書き込まれている、基板に実装された不揮発性メモリです。マイクロサーバシリーズは、通常のサーバと異なりFlashROMからカーネルが実行されます。

FlashROMはハードディスク等と書き込み方式が異なるため、独自の"flashcfg"コマンドを用いてファイルを保存します。

OpenBlockS OpenMicroServer



OSについて

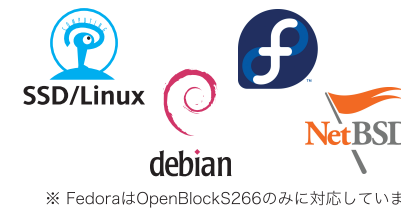
マイクロサーバシリーズに搭載される「SSD/Linux」は、当社が開発・維持・配布を行う、オープンソースの独自ディストリビューションです。SSD/Linuxは、限られたハードウェアリソースで十分に性能を発揮できるように最適化され、Linux標準コマンドを維持しつつ小型化されたアプリケーション開発に特化したOSです。

初期化実行(ファクトリリセット)時の設定やルートデバイスサイズの変更等、カスタムファームウェア開発を行う事も可能です。

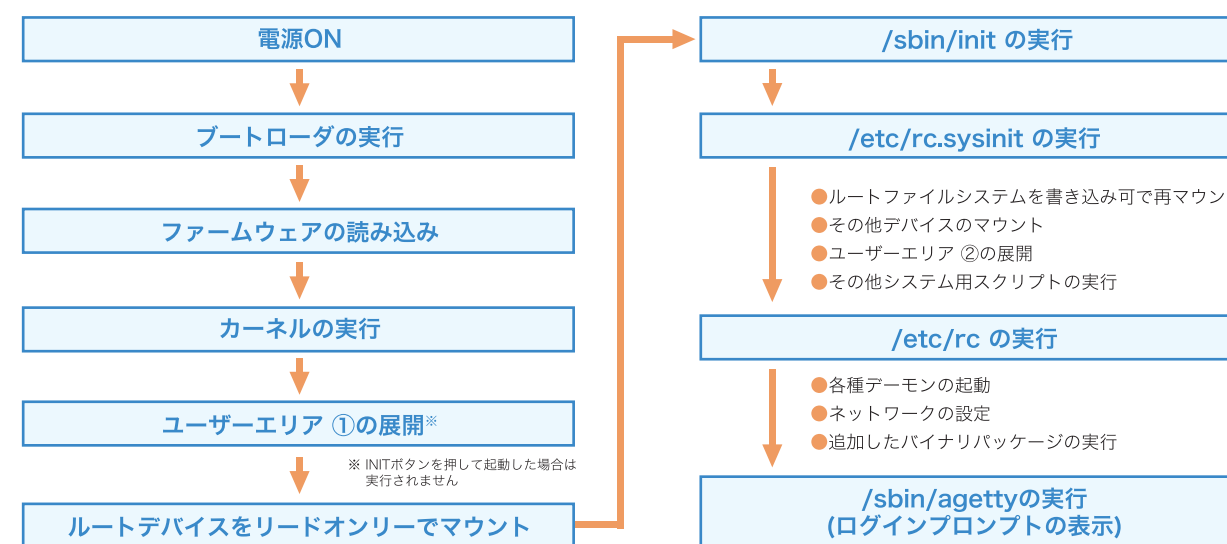
当社WEBサイト「オープンラボラトリ」では、最新のSSD/Linuxおよび開発環境(セルフ/クロスコンパイル環境)に加え、アプリケーションの実装手順やコンパイル済みアプリケーション等の情報を日々更新し、無償で公開/配布しています。

最新の情報は: オープンラボラトリ <http://www.plathome.co.jp/support/labo/>

マイクロサーバシリーズは、Debian, Fedoraに対応しています。OSイメージを展開し、CFやSSD(Solid State Drive)をルートデバイスとして設定することで利用可能です。Linux以外にもNetBSDに対応しており、インストールガイドの提供も行っています。



起動プロセスの詳細



ルートデバイスによるメインメモリ割り当て容量の違い

	ルートデバイス	
	メインメモリ(標準) /dev/ram	CFやSSD(Solid State Drive)※ /dev/hda
FlashROM	カーネル 約2MB ユーザーエリア ①256KB + ②2MB	カーネル 約2MB 未使用
メインメモリ	使用可能領域 96MB ルートデバイス 32MB (RAMディスクとして)	使用可能領域 128MB
ストレージ	—	ルートデバイス 128MB~

ルートデバイスをメインメモリとした場合の図内容量はOpenMicroServer時です。OpenBlockSの場合は、「ユーザーエリア ①256KB ②3.64MB」「ルートデバイス 26MB」です。

コラム: CFの書き込み回数制限について

マイクロサーバは、大容量の記憶デバイスを持たないため、CFをストレージとして使用する場合があります。CFに用いられているNAND型のフラッシュメモリは、1ブロックあたりの書き込み回数制限があることから耐久性を心配されることもありますが、書き込み回数制限の影響を回避する書き込み均等化機能が搭載され、HDDの100万回の書き換え能力に匹敵します。さらに、想定使用容量の2倍の容量を持つCFを選択すれば、ブロックに対するアクセス確率は1/2となり、実質の耐久性は2倍となります。

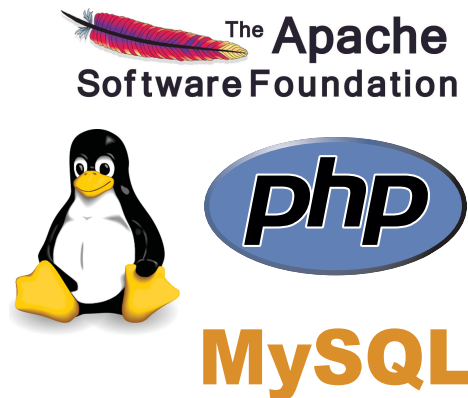
特集:実践OpenMicroServer即運用術

- LAMP環境の実装

OpenMicroServer(以下OMS)は壊れにくい・耐環境性に優れるという面から監視・制御系のアプリケーションとして多く利用されていますが、近年の環境保護対策での省電力・省資源化に対するニーズに応じ、従来型のラックマウント形サーバー装置からの置き代えもよく見られるようになりました。特に、CPU負荷量の少ないDHCPやDNS・小規模ルーター的な使い方はもとより、LAMPシステム(Linux/Apache/MySQL/PHP)を利用した小規模なWebアプリケーションも、十分OMSで利用が可能です。

こうした用途について、以下よりOMSを即稼動するための実践を特集します。

なお、本特集はOMSと同シリーズのOpenBlockS(以下OBS)にも対応可能です。(OMSの部分をOBSと読み代えてご利用ください。)



OMSのファームウェアを最新にする

SSD/LinuxはファームウェアとしてOMSのFlashROMに圧縮保存されていますが、工場出荷バージョンはお客様のご購入時期により若干古いバージョンの場合があります。特にSSD/Linuxはメンテナンスを定期的に行っており、Linuxに何らかのセキュリティホールが見つかったり、いち早くリリースできる体制で臨んでいます。ご使用前にファームウェアを最新のものに更新しておくことをおすすめします。

■OMSにrootログインする

OMSのコンソールポートとPCのシリアルポートを添付のRS-232Cケーブルで接続しOMSを起動します。

PC側ではTera Termなどの通信ソフトを使います。通信ソフトではシリアルポートの設定を以下の様にします。

```
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none
```

通信ポートの設定が正しく行われているとOMSは以下の通りメッセージを出力します。

```
LinuxServer login:
```

OMSの工場出荷状態からrootログインします。工場出荷時のPasswordはrootです。

```
LinuxServer login: root
Password: root
SSD/Linux 0.4-20051218/2.6.12 #3 Sun Dec 18 19:11:21 JST 2005

No mail.
```

ログインに成功したら次へ進みます。

■ネットワークアクセスを準備する

ここでのネットワーク設定は、ファームウェアを更新するためのもので仮の設定です。また、この作業にはインターネットアクセス環境とDHCP環境が必要です。先ずOMSのETHER-0コネクタにネットワークケーブルを差し込みます。次にDHCPクライアントを起動します。

```
# dhclient eth0
Internet Systems Consortium DHCP Client V3.0.3
: 省略
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 172.16.12.10
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 172.16.12.10
bound to 172.16.12.62 -- renewal in 1542 seconds.
```

このように、IPアドレスが取得できれば準備完了です。

■OMSのファームウェアをバージョンアップする

まず、ぶらっとホームの公式ページで最新版のファイル名をWebブラウザで確認します。(OMSの場合 <http://www.plathome.co.jp/support/labo/oms400/>) (OBSの場合 <http://www.plathome.co.jp/support/labo/obs266/>)

公式リリースファームウェア (RELEASE)

弊社公式のファームウェアです。通常用途で運用する場合、最新版にアップデートしてください。

0.4-RELEASE-20080322 リリースノート (31 Mar 2008) **NEW**

サイズ(byte)	md5sum
IPv4 対応	d1 899641 912560ac86ab95fe30c
IPv4, IPv6 対応	a2e15d8eb9e16706b27e1784a07

0.4-RELEASE-20071208 リリースノート (07 Dec 2007)

サイズ(byte)	md5sum
IPv4 対応	8885392 b04617 42a5435237373c40ae7
IPv4, IPv6 対応	9205996 27165a75ae010c74a8396e51d9

0.4-RELEASE-20071020 リリースノート (23 Oct 2007)

サイズ(byte)	md5sum
IPv4 対応	8829316 78e34990a62c1db2d8528f664c0f
IPv4, IPv6 対応	9163392 b065c6b77e1a6502c041a443fe4

WEBに表示されたファイルをマウスで右クリックし、プロパティを表示します。このプロパティのアドレス部分をコピーします。次にOMS上でftpコマンドによりダウンロードします。ftpコマンドの後のファイル名はプロパティからコピーした文字列をペーストした内容です。(Tera Termの場合ペーストはAlt+vです。)

```
# ftp ftp://ftp.plathome.co.jp/pub/OMS400/0.4-RELEASE-20080322/mipse1-obs1550.ipv6/installation/zImage.initrd.treeboot-product
Connected to download.plathome.co.jp.
220 ProFTPD 1.3.0 Server (ftp.plathome.co.jp) [202.32.197.43]
331 Anonymous login ok, send your complete email address as your password.
:
: 省略
:
100% |*****| 9213 KB 999.19 KB/s
00:00 ETA
226 Transfer complete.
9434268 bytes received in 00:09 (998.95 KB/s)
221 Goodbye.
```

ダウンロードが終わるとカレントディレクトリに以下のようにファイルができます。

```
# ls
zImage.initrd.treeboot-product
```

このファイルをflashcfgコマンドでFlashROMに書き込みます。

```
# /usr/sbin/flashcfg -f zImage.initrd.treeboot-product
Load boot image to FlashROM
#####
#####
#####
done
```

ここでOMSを再起動します。

```
# reboot
INIT: Sending processes the TERM signal
省略
```

コンパクトフラッシュ(CF)をOMSのメインストレージにする

OMSのデフォルト設定ではメインメモリーから32MB(OBSは26MB)分がRAM disk領域として確保され、ここへOSや小型のアプリケーションを配置しています。しかし、LAMPなどのアプリケーションには、メインメモリーやストレージ領域を少しでも多く確保するために、CFの増設が適しています。

■CFを初期化しパーティションを作成する

OMSをシャットダウンしACアダプタを抜いてください。

この状態でOMSの外装フタを外します。

CFはOMSの基板上にスロットがあるので、ここへ差し込みます。

準備が出来たらフタを閉めてOMSを起動しrootログインします。

次にfdiskを起動します。

(ここでは、CFのデバイス名はhdaとなっていますが、CFの代わりに外付USBストレージを使用する場合、hdaをsdaと読み替えてセットアップできます。)

```
# fdisk /dev/hda
hda:
```

```
The number of cylinders for this disk is set to 7899.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
 (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help):
```

fdiskが起動するとコマンド入力状態となるので、先ずpコマンドで現在のCF内のパーティションをチェックします。

```
Command (m for help): p
```

```
Disk /dev/hda: 4076 MB, 4076642304 bytes
16 heads, 63 sectors/track, 7899 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	7640	3850528+	83	Linux
/dev/hda2		7641	7899	130536	83	Linux

```
Command (m for help):
```

未使用のCFの場合は、上記のようにパーティションは存在しませんが、何らかのOS領域がある場合は、oコマンドでパーティションを全て削除してください。パーティションが無い状態から、以下の通りに新しいパーティションを作ります。ここでパーティションはシステム領域用とスワップ用の二つを作ります。(スワップ用は必須ではありませんが、必要になった時の事を考えて用意しておきます。)

スワップサイズはメインメモリサイズよりやや多めに用意します。

この例の4GB CFでは、上記の様にシリンダーあたり約516KBを259シリンダーに割り当てるため、システム領域用のエンドシリンダーは7640とします。

パーティションは以下の通りnコマンドで作成します。

なおFirst cylinderとLast cylinderの入力でdefaultを採用する場合はEnterキーのみの入力となります。

したがって、ここでのcylinderの入力は7640と指定している場所だけです。

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
```

```
p
Partition number (1-4): 1
First cylinder (1-7899, default 1):
Using default value 1
Last cylinder or +sizeM or +sizeK (1-7899, default 7899): 7640
```

再起動が完了すると、新バージョンが立ち上がり、バージョンはログインの上の部分に表示されます。

```
Welcome to SSD/Linux 0.4-20080320/2.6.16
eth0: going to full duplex
```

```
LinuxServer login:
```

この表示は2008年3月20日バージョンのものです。

```
Command action
e extended
p primary partition (1-4)
```

```
p
Partition number (1-4): 2
First cylinder (7641-7899, default 7641):
Using default value 7641
Last cylinder or +sizeM or +sizeK (7641-7899, default 7899):
Using default value 7899
```

```
Command (m for help): p
```

```
Disk /dev/hda: 4076 MB, 4076642304 bytes
16 heads, 63 sectors/track, 7899 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	7640	3850528+	83	Linux
/dev/hda2		7641	7899	130536	83	Linux

```
Command (m for help):
```

これでhda1がシステム用として約3.85GB、hda2がスワップ用として130MB設定されました。次にhda1をboot用パーティションにaコマンドで設定します。

```
Command (m for help): a
Partition number (1-4): 1
```

```
Command (m for help):
```

更にhda2をスワップ用としてパーティションIdを82にtコマンドで変更します。

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap / Solaris)
```

```
Command (m for help):
```

変更結果をpコマンドで表示します。

```
Command (m for help): p
```

```
Disk /dev/hda: 4076 MB, 4076642304 bytes
16 heads, 63 sectors/track, 7899 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	7640	3850528+	83	Linux
/dev/hda2		7641	7899	130536	82	Linux swap / Solaris

```
Command (m for help):
```

以上のようにhda1のBootに*マークが付き、hda2のSystemの部分にLinux swapと表示されれば正常です。最後にwコマンドで設定内容を保存しfdiskを抜けます。

```
Command (m for help): w
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
hda: hda1 hda2
hda: hda1 hda2
Syncing disks.
```

以上でCFの準備は完了しました。

■SSD/LinuxをCFに新規インストールする

初期化済みのCFにSSD/Linuxをインストールします。
インストール前に、SSD/Linuxは再起動しているのを、もう一度DHCPクライアントを起動します。

```
# dhclient eth0
Internet Systems Consortium DHCP Client V3.0.3
:
: 省略
:
bound to 172.16.12.62 -- renewal in 1542 seconds.
```

IPアドレスの取得が済んだらsysinstコマンドを実行します。

```
# sysinst

SSD/Linux 0.4-20080321/2.6.16 Installer

Are you sure to install [y/N] ? y
```

インストールを開始するのでyを入力すると、以下のように、初期化済みのCFのパーティションが表示されます。

```
Setup Filesystem

      Use Boot Device      MB   FStype   BSize fsck   Mount Point
-----
0.  *   *   /dev/hda1      3760 ext3      4096 no   /
1.  *   *   /dev/hda2      127  swap          no   swap

x. Exit this menu.

Enter number [0-1 or x] : x
```

この場合、SSD/Linuxのインストール先としてNo.0の領域(hda1)、swapとしてNo.1の領域(hda2)が自動設定されています。

```
Taking mkswap and mke2fs.
Are you sure [y/N] ? y
```

次にファイルシステムを作成するかを聞かれるのでyを入力すると作成を開始します。

```
Setting up swapspace version 1, size = 133664 kB
no label, UUID=86a182d1-2569-4d83-95e6-e02692db7a86
Adding 130528k swap on /dev/hda2. Priority:-1 extents:1 across:130528k
mke2fs 1.39 (29-May-2006)
:
: 省略
:
EXT3 FS on hda1, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
```

ファイルシステムが作成されると追加のディストリビューションを聞いてきます。

```
Select Distributions

0. Generic Kernel      : yes
1. Base                : yes
2. System (/etc)      : yes
3. Compiler            : yes
4. Manuals             : yes
5. Contributions      : no
6. KanshiBlockS Demo  : no
7. Source              : no

x. Exit this menu.

Enter number [0-7 or x] : x
```

デフォルトでyesとなっている項目のとおり、通常のFlashROMバージョンに加えて開発環境(3.Compiler)とマニュアル(4.Manuals)が追加でインストールされます。逆に簡単設定ツール環境(5.Contributions)は、デフォルトインストールから削除されていますが、ここではツールを利用しないのでデフォルトのままとします。また、ソース(7.Source)については必要あればyesにしてください。インストール内容が決まったらxでメニューを抜けます。次にインストールするメディアを聞いてきますが、ネットワークからダウンロードするので、そのままxでメニューを抜けます。

```
Select install media

* Network : eth0

x. Exit this menu.

Enter command [x] : x
```

次はftp設定画面です。
ここでは5.Login Passwordを変更します。
このログインパスワード部分をサーバー管理者のメールアドレスに書き換えます。
ここではメールアドレスを仮にadmin@example.jpとしています。

```
Setup for FTP download

0. FTP server      : ftp.plathome.co.jp
1. Remote pathname : /pub/OMS400/0.4-LATEST/mipse1-obs1550.ipv6/binary
2. Local pathname  : /usr/INST
3. FTP proxy       :
4. Login name      : ftp
5. Login Password  : Your_Mail_Address

x. Exit this menu.
```

```
Enter number [0-5 or x] : 5
Enter password : admin@example.jp
```

```
Setup for FTP download

0. FTP server      : ftp.plathome.co.jp
1. Remote pathname : /pub/OMS400/0.4-LATEST/mipse1-obs1550.ipv6/binary
2. Local pathname  : /usr/INST
3. FTP proxy       :
4. Login name      : ftp
5. Login Password  : admin@example.jp

x. Exit this menu.

Enter number [0-5 or x] : x
```

修正した部分が正しく更新されているのを確認しxで設定メニューを抜けます。
次にftpが開始される前に確認が入るのでyを入力するとダウンロードが始まります。

```
Start fetch binaries and extract.
Are you sure [y/N] ? y

Fetch kern.tgz ...
: 省略
```

ダウンロード後、CFへのシステム展開が始まります。

```
: 省略
./usr/share/man/man9/

Make device files...

Write /etc/fstab.

Modify /etc/rc.conf.

Remove /mnt/usr/INST
Are you sure [y/N] ? y
```

展開が終了するとインストールイメージを削除するかを聞かれるのでyを入力します。
最後にCFにインストールされたroot権限のパスワードを聞いてきます。
(ここではパスワードもrootにします。実際には任意で設定してください)

```
Set root passwd
Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password: root
Re-enter new password: root
Password changed.

Installation Completed.

Good luck...
```

新しいパスワードの確認も含め二回入力し、最後にGood luckと表示されればCFへのインストール作業は完了です。
もし、上手く行かない場合は、rebootして最初からやり直してください。
インターネットへ接続できていない、または、ダウンロード先やファイル名が間違っている場合など失敗することがあります。

■bootをCFに切り替える

先にも述べたようにCF上にシステム構築しても利用されるルートファイルシステムはFlashROM上のもので、SSD/LinuxはFlashROMからbootした後にCFをマウントする動きとなります。
この辺が、他のシステムとは違った動作であることを念頭において下さい。
ではbootをflashcfgコマンドで切り替えます。
起動用のパーティションはシステム構築したhda1になるので以下の通り入力し実行します。

```
# flashcfg -c hda1
Boot device change to hda1
# reboot
```

これで全ての作業が完了しました。
最後にrebootするとCFがSSD/Linuxのメインストレージとなります。

参考:bootをデフォルトに戻す時は以下のコマンドです。

```
# flashcfg -c initrd
```

OMSの基本設定を行う

OMSをサーバーとして運用するためにDNSや固定IPアドレスなどの設定を行います。

■DNSの設定

DNSの設定はDHCPクライアントを起動すると自動的に読み込まれ保存されるので、一度DHCPクライアントを起動します。

```
(none) [1]# dhclient eth0
Internet Systems Consortium DHCP Client V3.0.3
: 省略
:
bound to 172.16.12.62 -- renewal in 1542 seconds.
(none) [2]#
```

この作業だけで完了です。

■ネットワークを設定する

ここではローカルエリアネットワーク利用を前提としますので以下のアドレスで説明を進めます。
(注意:LANの固定IPアドレスはネットワーク管理者に相談してください)

```
IPアドレス      172.16.12.239
ネットマスク    255.255.255.0
ブロードキャスト 172.16.12.255
デフォルトゲートウェイ 172.16.12.1
ホストネーム    lamp.example.jp
```

ホストネームはlampの部分がサーバー名でexample.jp部分がネットワークドメイン名なので、ドメインを持っていない場合はlampだけでかまいません。
もちろんlampの部分も好きな名前を小文字アルファベットでつけてかまいません。
では、編集前のrc.confは以下の通りです。(中間は省略)

```
(none) [3]# cat /etc/rc.conf
: 省略
# Add local overrides below
#
do_contrib_rc=YES
```

修正するのは最後の行で、do_contrib_rc=YES行の先頭に#を付けます。
これはWEBセットアップツールやシリアルセットアップツールでの設定情報を有効にするためのものなので、本特集での説明では外します。
その後にネットワーク設定をviエディタで以下のファイルに追記します。

```
(none) [4]# vi /etc/rc.conf
: 省略
# Add local overrides below
#
#do_contrib_rc=YES
ifconfig_eth0="inet 172.16.12.239 netmask 255.255.255.0 broadcast 172.16.12.255"
default_troute="172.16.12.1"
hostname="lamp.example.jp"
```

編集が終わったら:x!でファイル保存してviを抜けます。
なお、viを使い慣れない人のために、以下に簡単なリファレンスを記します。

```
編集のためのviエディタリファレンス
カーソル移動上 k   カーソル移動下 j
カーソル移動左 h   カーソル移動右 l
文字入力 i を入力後テキスト入力モードでESCキーの入力で終了
文字削除 削除したい文字へカーソルあわせてxを入力
編集を最初からやり直す :e!
編集結果を保存せずviを終了する :q!
編集結果を強制保存(強制はrootのみ)してviを終了する :x!
```

次に自分のIPアドレスを/etc/hostsに書き込んでおきます。
これは、もし参照しているDNSが非稼働状態にあった時、DNSを参照するサービスが起動できなくなる場合に備えるためです。(Apacheなど)
そして、viエディタで/etc/hostsに自分のIPアドレスを追記します。
DNSやアプリケーションの癖によって左右されないように、以下のように入力することを推奨します。

```
ドメイン付きのホスト名の場合
172.16.12.239 lamp.example.jp lamp lamp.example.jp.
```

ドメイン無しのホスト名の場合

```
172.16.12.239 lamp
```

とします。(以下の例はドメイン付きです。)

```
(none) [6]# vi /etc/hosts
: 省略
# RFC 1918 specifies that these networks are "internal".
# 10.0.0.0 10.255.255.255
# 172.16.0.0 172.31.255.255
# 192.168.0.0 192.168.255.255
#
172.16.12.239 lamp.example.jp lamp lamp.example.jp.
```

以上の修正が完了したら:x!でファイル保存してviを抜けます。
次にtelnetとftpを利用できるようにinetd.confをviエディタで修正します。

```
(none) [7]# vi /etc/inetd.conf
: 省略
# These are standard services.
#
#
#ftp stream tcp6 nowait root /usr/libexec/ftpd ftpd
#telnet stream tcp6 nowait root /usr/libexec/telnetd telnetd
: 省略
```

ファイルの中に#ftpで始まる行と#telnetで始まる行があります。
この二行の先頭の#を外します。

```
ftp stream tcp6 nowait root /usr/libexec/ftpd ftpd
telnet stream tcp6 nowait root /usr/libexec/telnetd telnetd
```

以上の修正が完了したら:x!でファイル保存してviを抜けます。
次にLANからのみftpとtelnetを受け付ける設定にします。
hosts.allowをviエディタで編集します。
デフォルトではフルクローズなので#ALL: ALLとなっています。
この行の下に追記します。

```
# $ssdlinux: hosts.allow.v 1.1.1.1 2002/05/02 13:37:06 kanoh Exp $
#ALL: ALL
ftpd : localhost : allow
ftpd : 172.16.0.0/255.255.0.0 : allow
ftpd : ALL : deny

telnetd : localhost : allow
telnetd : 172.16.0.0/255.255.0.0 : allow
telnetd : ALL : deny
```

この例では、どちらもアクセスできる端末は172.16.xxx.xxxとなります。
以上の修正が完了したら:x!でファイル保存してviを抜けます。
次にftpとtelnetで使う時の管理者用アカウントを作成します。
ここではユーザー名をadminとしパスワードもadminとします。(実際には任意で)コマンドラインから以下の通り入力してください。

```
(none) [9]# useradd -g wheel admin
(none) [10]# passwd admin
Enter new UNIX password: admin
Retype new UNIX password: admin
```

これで一通りの設定が済んだのでrebootします。

```
(none) [11]# reboot
```

OMSが再起動するとログインのサーバー名が変わっているはずですが。

```
Welcome to SSD/Linux 0.4-20080320/2.6.16

lamp.example.jp login:
```

再度rootログインしてネットワークの内容を確認してください。
以下のように、設定したIPアドレスなどが反映されていれば完了です。

```
lamp [1]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:0A:85:02:00:28
      inet addr:172.16.12.239 Bcast:172.16.12.255
Mask:255.255.255.0
      inet6 addr: fe80::20a:85ff:fe02:28/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:103 errors:0 dropped:0 overruns:0 frame:0
TX packets:6 errors:3 dropped:0 overruns:0 carrier:6
collisions:0 txqueuelen:1000
RX bytes:11489 (11.2 Kb) TX bytes:0 (0.0 b)
Interrupt:27

lamp [2]# netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
172.16.12.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 0.0.0.0 0.0.0.0 UG 0 0 0 eth0
```

また、telnetとftpも使えるようになっているのでTera Termをtelnetに切り替えadminアカウントでログインできれば正常です。
但し、以降の作業もroot権限で行うので、このままtelnetで作業を継続される方はsuでrootに切り替えてください。

```
lamp [3]# su -l
Password: root
lamp [1]#
```

OMSにLAMP環境を構築する

Linux/Apache/MySQL/PHPの組み合わせで実現できるWebアプリケーション実行環境を「LAMP」と呼びます。アプリケーションはPHPのソースコードレベルで無償配布されているものが多く、機能的にも優れたものが入手でき、実用的です。ここでは、OMSにLAMPの実行環境を構築する手順を解説します。

■パッケージ管理の環境を作る

まずパッケージ管理用のディレクトリを用意します。

```
lamp[1]# mkdir -p /usr/pkg/tmp
lamp[2]# mkdir -p /usr/pkg/var/run
lamp[3]# mkdir -p /usr/pkg/var/log
```

ディレクトリが正常に準備できたかをlsコマンドでチェックしてください。次に追加コマンド周辺をftpでダウンロードします。ダウンロード先の指定ではOSのリリース名に注意してください。(ここでは0.4-20080320ですがuname -aで確認する。)

```
lamp[4]# ftp ftp://ftp.plathome.co.jp/pub/OMS400/pkg/0.4-20080320/bootstrap.tgz
Connected to download.plathome.co.jp.
220 ProFTPD 1.3.0 Server (ftp.plathome.co.jp) [202.32.197.43]
: 省略
100% |*****| 680 KiB 955.62 KiB/s 00:00 ETA
226 Transfer complete.
696781 bytes received in 00:00 (952.64 KiB/s)
221 Goodbye.
lamp[5]#
```

ダウンロードしたファイルbootstrap.tgzを展開します。

```
lamp[5]# tar xvzf bootstrap.tgz -C /
./usr/pkg
./usr/pkg/share
./usr/pkg/share/mk
: 省略
./usr/pkg/var/db/pkg/pkg_install-20071109/+SIZE_ALL
./usr/pkg/var/db/pkg/pkg_install-20071109/+SIZE_PKG
./usr/pkg/etc/mk.conf
lamp[6]#
```

次に環境変数処理を設定します。viエディタを起動しファイルの最後に必要な内容を追記します。

```
lamp[8]# vi /etc/profile
: 省略
# Append any additional sh scripts found in /etc/profile.d/:
for file in /etc/profile.d/*.sh ; do
    if [ -x $file ] ; then
        . $file
    fi
done ← ここがオリジナルの最後で次の行からが追記部分

OS_RELEASE=0.4-20080320
alias pkg_add='PKG_PATH=ftp://ftp.plathome.co.jp/pub/OMS400/pkg/[$OS_RELEASE]/binary/
usr/pkg/sbin/pkg_add'

PKG_TMPDIR=/usr/pkg/tmp
export PKG_TMPDIR

PATH=/usr/pkg/bin:$PATH

if [ "$UID" -eq "0" ] ; then
    PATH=/usr/pkg/sbin:$PATH
fi

export PATH ← ここまで
```

ここでも、OSのリリース名0.4-20080320を間違えないように注意してください。uname -aコマンドで見た最後の部分がリリース名です。ここを間違えるとパッケージのコンパイル環境バージョンが変わってしまいます。編集が終わったら:x!でファイル保存してviを抜けます。そして不要になったbootstrap.tgzを削除します。

```
lamp[10]# rm bootstrap.tgz
```

最後にrebootします。

```
lamp[11]# reboot
```

OMSを再起動してrootログインしてからパッケージの情報を確認してみましょう。

```
lamp[1]# pkg_info
bootstrap-mk-files-20080310 *.mk files for the bootstrap bmake utility
bmake-20080215 Portable (autoconf) version of NetBSD 'make' utility
tnftp-20070806 The enhanced FTP client in NetBSD
pax-20080110 POSIX standard archiver with many extensions
pkg_install-20080313 Package management and administration tools for
pkgsrc
```

この様にパッケージの情報を取得できるようになれば完了です。

■Apache/MySQL/PHPのパッケージをインストールする

pkg_addコマンドでそれぞれのパッケージを追加します。この例では、それぞれのバージョンは以下の通りです。Apache: 2.2系 MySQL: 5.0系 PHP: 5.0系
まずはApacheです。当社FTPサイトからのダウンロードには、少々時間がかかります。

```
lamp[2]# pkg_add apache
tar: ustar vol 1, 1072 files, 12830720 bytes read, 0 bytes written in 9 secs (14
25635 bytes/sec)
: 省略
apache-2.2.8: copying /usr/pkg/share/examples/rc.d/apache to /usr/pkg/etc/rc.d/a
pache
```

次にMySQLですが、パッケージにはserverとclientがあります。

```
lamp[3]# pkg_add mysql-server
tar: ustar vol 1, 1703 files, 54689280 bytes read, 0 bytes written in 36 secs (1
519146 bytes/sec)
: 省略
Please see the manual and the MySQL web site for more instructions.
```

```
lamp[4]# pkg_add mysql-client
tar: ustar vol 1, 97 files, 23684608 bytes read, 0 bytes written in 13 secs (182
1892 bytes/sec)
: 省略
pkg_add: package `mysql-client-5.0.51nb3' was already installed as dependency, n
ow marked as installed manually
```

最後にPHPです。本体以外にも含めて5つのライブラリがあります。

```
lamp[5]# pkg_add php
tar: ustar vol 1, 236 files, 10379776 bytes read, 0 bytes written in 3 secs (345
: 省略
install the lang/pear package.

lamp[6]# pkg_add php5-mysqli
tar: ustar vol 1, 9 files, 161280 bytes read, 0 bytes written in 1 secs (161280
: 省略
Then restart your PHP5-enabled HTTP server to load this module.

lamp[7]# pkg_add php5-mysql
tar: ustar vol 1, 9 files, 92160 bytes read, 0 bytes written in 1 secs (92160 by
: 省略
Then restart your PHP5-enabled HTTP server to load this module.

lamp[8]# pkg_add php5-pdo_mysql
tar: ustar vol 1, 9 files, 51200 bytes read, 0 bytes written in 1 secs (51200 by
: 省略
Then restart your PHP5-enabled HTTP server to load this module.

lamp[9]# pkg_add ap22-php5
tar: ustar vol 1, 9 files, 4136960 bytes read, 0 bytes written in 2 secs (206848
: 省略
AddHandler application/x-httpd-php .php3
```

作業が終わったらインストールされたパッケージを確認します。

```
lamp[11]# pkg_info
bootstrap-mk-files-20080310 *.mk files for the bootstrap bmake utility
bmake-20080215 Portable (autoconf) version of NetBSD 'make' utility
tnftp-20070806 The enhanced FTP client in NetBSD
pax-20080110 POSIX standard archiver with many extensions
pkg_install-20080313 Package management and administration tools for
pkgsrc
perl-5.8.8nb7 Practical Extraction and Report Language
apr-1.2.12 Apache Portable Runtime
expat-2.0.1 XML parser library written in C
apr-util-1.2.10 Apache Portable Runtime utilities
apache-2.2.8 Apache HTTP (Web) server, version 2
mysql-client-5.0.51nb3 MySQL 5, a free SQL database (client)
mysql-server-5.0.51nb1 MySQL 5, a free SQL database (server)
xmlocatmgr-2.2nb1 XML and SGML catalog manager
libxml2-2.6.31 XML parser library from the GNOME project
php-5.2.5nb1 PHP Hypertext Preprocessor version 5
php5-mysqli-5.2.5 PHP5 extension for MySQL 4.1 and later databases
php5-mysql-5.2.5 PHP extension for MySQL databases
php5-pdo-5.2.5.1.0.2 PHP extension for PHP Data Objects (base)
php5-pdo_mysql-5.2.5.1.0.1 PHP PDO extension for MySQL databases
ap22-php5-5.2.5nb1 Apache (apache22) module for PHP5
lamp[12]#
```

以上の様にパッケージ一覧に追加されていればインストール完了です。

■Apacheの実行環境を設定する

Apacheの設定ファイルhttpd.confをviエディタで編集します。

```
lamp[12]# cd /usr/pkg/etc/httpd
lamp[13]# vi httpd.conf
```

viが起動したら以下の行を探してください。

```
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
User www
Group www
```

ここはApacheをWebデザインのユーザーグループに所属させるための設定です。従来型のhtmlドキュメントの実体を階層ディレクトリに配置する方式(静的なWeb)であれば、このままのグループでWebデザイナーメンバーをwwwグループに追加していくこととなります。しかし、LAMPアプリケーションなどの動的なWebシステムでは、htmlドキュメントがデータベース上に保存されます。したがって、Webドキュメントルートの配下には、PHPソースなどを主体としたファイルが保存されるので、一般ユーザーには操作をさせない方が適切です。そのため、ここではApacheのアカウントを以下の通り管理者とし設定を行います。この設定値はOMSの基本設定の項で作成したアカウントです。

```
User admin
Group wheel
```

次は以下の行です。

```
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents, e.g. admin@your-domain.com
ServerAdmin you@example.com
```

このServerAdminにサーバー管理者のメールアドレスを入れます。

```
ServerAdmin admin@example.jp
```

次に以下の行を探します。

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/pkg/share/httpd/htdocs"
```

このディレクトリがHTMLドキュメントのルートになります。必要であればパスを変更してください。

なお、このドキュメントルートの後の作業でオーナーをadminに変更します。次は以下の行を探してindex.phpをindex.htmlの前に追記します。

```
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

次は以下の行で、ここはApacheのエラーログを書き込む場所です。

```
# ErrorLog: The location of the error log file.
: 省略
# container, that host's errors will be logged there and not here.
#
ErrorLog "/var/log/httpd/error_log"
```

変更の必要はないと思いますが、Apacheに何らかのエラーが出た時に参照する場所であることを覚えておいて下さい。デフォルト値は、/usr/pkg からの相対パスで記述されています。(ログディレクトリをまとめる場合は変更してもよい) さらに最後の行までカーソルを移動し以下の二行を追記します。

```
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule> ←ここが最後の行
LoadModule php5_module lib/httpd/mod_php5.so
AddHandler application/x-httpd-php .php
```

以上の修正が完了したら:x!でファイル保存してviを抜けます。

最後にApacheのドキュメントルート(/usr/pkg/share/httpd/htdocs)のオーナーをチェンジします。

これはApacheのアカウントと同様にするため、以下の通りOMSの基本設定の項で作成したadminとします。

```
lamp[22]# cd /usr/pkg/share/httpd
lamp[23]# chown -R admin htdocs
lamp[24]# ls -l
total 24
drwxr-xr-x 2 root wheel 4096 Apr 3 12:08 build
drwxr-xr-x 3 root wheel 4096 Apr 3 12:08 error
drwxr-xr-x 2 admin wheel 4096 Apr 4 15:57 htdocs
drwxr-xr-x 3 root wheel 4096 Apr 3 12:08 icons
drwxr-xr-x 14 root wheel 8192 Apr 3 12:08 manual
```

変更したらlsでhtdocsのオーナーがadminになっているのを確認してください。Apacheの設定は以上です。

■MySQLの初期設定を行う

この例では、MySQLへのアクセスはOMS内のアプリケーションのみに制限するため、ポートはlocalhostのみオープンとします。hosts.allowをviエディタで編集します。

```
lamp[17]# vi /etc/hosts.allow
#
#ssdlinux: hosts.allow.v 1.1.1.1 2002/05/02 13:37:06 kanoH Exp $
#ALL: ALL
ftpd : localhost : allow
ftpd : 172.16.0.0/255.255.0.0: allow
ftpd : ALL : deny
```

```
telnetd : localhost : allow
telnetd : 172.16.0.0/255.255.0.0: allow
telnetd : ALL : deny ← ここまではOMS基本設定で追加した行。
```

```
mysqld : localhost : allow
mysqld : ALL : deny
```

MySQLの行を追加したら:x!でファイル保存してviを抜けます。

■PHPの初期設定を行う

PHPからMySQLを使用できるようにphp.iniをviエディタで編集します。

```
lamp[21]# vi /usr/pkg/etc/php.ini
```

以下の行を探して追記してください。

```
: Windows Extensions
: Note that ODBC support is built in, so no dll is needed for it.
: Note that many DLL files are located in the extensions/ (PHP 4) ext/ (PHP 5)
: extension folders as well as the separate PECL DLL download (PHP 5).
: Be sure to appropriately set the extension_dir directive.
```

```
:extension=php_bz2.dll
: 省略
```

```
:extension=php_xsl.dll
:extension=php_zip.dll
extension=mysql_i.so
extension=mysql.so
extension=pdo_mysql.so
extension=pdo.so
: Module Settings :
: Module Settings :
```

PHPの設定は以上です。

■サーバー起動時にApache/MySQLを自動実行にする

それぞれのサービスがサーバー起動時に自動実行できるよう設定します。自動実行には/etc/rc.confにviエディタで追記します。

```
lamp[20]# vi /etc/rc.conf
#
# $ssdlinux: rc.conf.v 1.6 2008/01/24 06:22:29 todoroki Exp $
#
# see rc.conf(5) for more information.
: 省略
ifconfig_eth0="inet 172.16.12.239 netmask 255.255.255.0 broadcast
172.16.12.255"
defaultroute="172.16.12.1"
hostname="lamp.example.jp" ←ここは基本設定で追加した最後の行

apache=YES
mysqld=YES
```

編集が終わったら:x!でファイル保存してviを抜けます。

ここで一旦rebootして自動起動するかどうかをチェックしてください。

再起動してrootログインを行った後、psコマンドでApacheとMySQLが起動している事を確認します。

```
lamp[21]# ps -e
PID TTY TIME CMD
1 ? 00:00:09 init
: 省略
969 ? 00:00:00 httpd
970 ? 00:00:00 httpd
: 省略
1015 ? 00:00:00 mysqld
1016 ? 00:00:00 mysqld
1033 ttyS0 00:00:00 ps
```

このように、リストの中にhttpdとmysqldが複数確認できた場合は、正常に起動しています。ここで、一通り設定が済んだので実際に動作を確認します。

WebブラウザのurlにOMSのIPアドレス(http://172.16.12.239/)を入れて表示してみます。下記のようにIt works!と表示されていれば正常です。

なお、このhtmlドキュメントの実体は/usr/pkg/share/httpd/htdocs/index.htmlです。



■MySQLのrootパスワードを設定する

次にMySQLへrootでログインします。
(rootとは言っても、OSとは独立したMySQL専用ユーザです)

```
lamp[20]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.51 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

以上のようにrootでログインに成功すればmysql>プロンプトが表示されます。
次にMySQLのユーザーを一覧します。

```
mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | localhost | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | lamp.example.jp | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | 127.0.0.1 | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

rootパスワードはログインの方法により三つのパターンに登録が必要です。
また、user名の無いアカウントも条件によってできてしまうので削除します。
以下の通り順番に入力してください。
ここでのパスワードもrootアカウントにはrootとしています。(実際には任意)

```
mysql> set password for root@localhost=password('root');
Query OK, 0 rows affected (0.00 sec)
mysql> set password for root@lamp.example.jp=password('root');
Query OK, 0 rows affected (0.00 sec)
mysql> set password for root@127.0.0.1=password('root');
Query OK, 0 rows affected (0.00 sec)
mysql> drop user ''@localhost;
Query OK, 0 rows affected (0.01 sec)
mysql> drop user ''@lamp.example.jp;
Query OK, 0 rows affected (0.00 sec)
```

入力し終わったらリストを表示しパスワードが入力されたのを確認します。
(パスワードの部分は暗号化されているので入力の通りには表示されません)

```
mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | localhost | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | lamp.example.jp | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | 127.0.0.1 | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

確認できたら一度MySQLを抜けてパスワード付きでログインし直します。

```
mysql> exit
Bye
lamp[19]$ mysql -u root -p
Enter password: root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.0.51 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

きちんとログインできればrootパスワード設定完了です。
次にデータベースを一覧します。

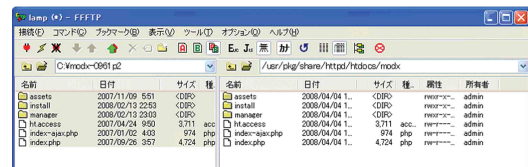
```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.04 sec)
```

このリスト内testはサンプルなので削除します。

```
mysql> drop database test;
Query OK, 0 rows affected (0.04 sec)
```

最後にshow databasesでサンプルtestが消えたのを確認しexitでMySQLを抜けます。

コピーが完了したらローカルとホストのファイル内容が一致しているか確認してください。



ファイル転送が済んだら以下のファイルを作成します。(中身は無し)
manager/includes/config.inc.php

FFFTPではファイル作成が出来ないので、ローカル側にあるconfig.inc.php.blankと言うファイルを右クリックして「名前を変えてアップロード」と簡単にできます。
次にホスト側にコピーされた個々のファイルで属性変更が必要な部分の作業を行います。

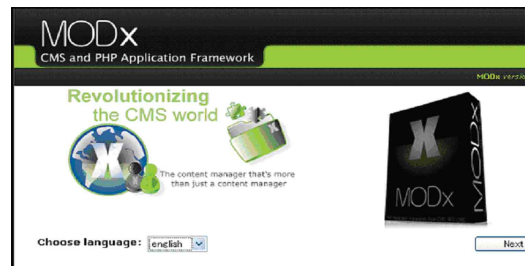
/usr/pkg/share/httpd/htdocs/modxを起点に以下のファイルの属性を変更します。

- assets/cache フォルダの属性を777にする。
- assets/cache フォルダの中にあるファイル全ての属性を777にする。
- assets/export フォルダの属性を777にする。
- assets/images フォルダの属性を777にする。
- manager/incudes/config.inc.php ファイルの属性を777にする。(前で作ったファイル)

全ての属性変更ができたならMODxのインストーラーを起動します。

Webブラウザから以下のurlへアクセスします。

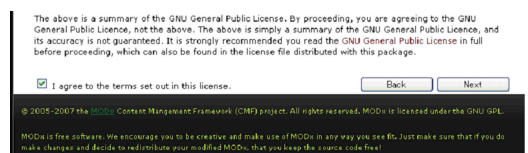
ここではhttp://172.16.12.239/modx/install/です。



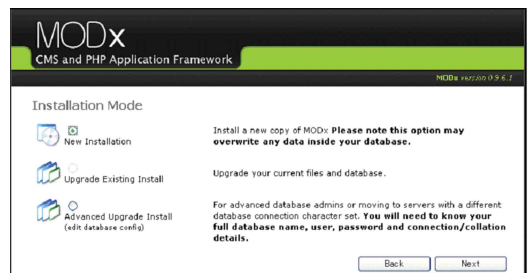
これはインストールのトップ画面です。

このままNextボタンを押し、次の画面ではBeginで進めます。

3画面目でライセンスが表示されるので、確認後 I agree to the terms set out in this license.のチェックボックスにチェックを入れ、Nextボタンを押します。



4画面目はインストールレーションモードを聞いて来るのでNew Installationを選んでNextボタンを押します。

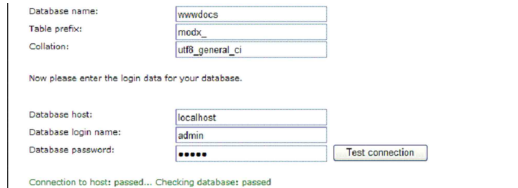


5画面目はデータベース関連の設定です。

これは先ほどMySQLで設定した内容を記載します。

- Database name: wwwdocs ← 先に作成したアカウント用DB名
- Table prefix: modx_ ← デフォルトのまま
- Collation: utf8_general_ci ← デフォルトのまま
- Database host: localhost ← デフォルトのまま
- Database login name: admin ← 先に作成したアカウント
- Database password: admin ← 先に作成したアカウントのパスワード

以上を設定したらTest connectionボタンを押してください。

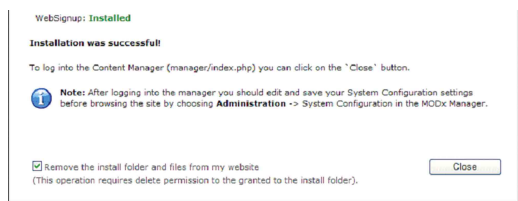


上の画面のように緑色の文字でデータベースコネクションに成功した事が表示されれば正常です。(失敗の場合、赤色の文字でエラーが表示されます)

次に同じ画面の下側にあるMODx管理者登録の項目へ記載します。
これはMODx独自の管理者ですが、ここでもアカウントはadminでパスワードもadminとしています。

管理者のメールアドレスはApacheに設定したものと同じにしています。

- Administrator user name: admin
- Administrator email: admin@example.jp
- Administrator password: admin
- Confirm password: admin



最後に、画面の下段でインストールホルダーを削除するかのチェックボックスにチェックが入っているか確認の上でCloseボタンを押します。

全てのインストール作業が終了するとMODx管理画面へのログインが可能になります。

ここでの管理画面のurlはhttp://172.16.12.239/modx/manager/となります。
MODx用に作成した管理者アカウントでログインしてください。

最初のログインではWebサイトの基本設定画面が表示されます。
ここで、少なくともMODxがサービスするWebサイトの名称設定(site name)および文字コード(Language)を設定してください。

任意のサイトネームを設定後、文字コードはプルダウンメニューからJapanese-utf8を選んでsaveボタンを押せば、管理画面も日本語表示になります。



以上でMODxのインストール作業は完了です。

MODxの利用方法については多くのWebサイトがあるので、そちらをご参照下さい。

■その他PHPなどのWebアプリケーションについて

本稿では最近注目の高いMODxを取り上げましたが、個人サイトで利用者の多いブログ管理システムWordPressや、様々なwikiシステム、その他MODxと近い性能を持つ他のCMSなども容易にインストール可能です。

インストール作業では、データベースへのコネクションにおいて、大半のアプリケーションで同様の設定となります。

また、MySQLに作ったデータベースもアプリケーション間で共有して使えるので、データベースを新たに作る必要もありません。

注意する点は、ApacheやMySQL・PHPのバージョンで動作可能か、使用したいアプリケーションの配布サイトで確認する必要があります。

ぶらっとホームでは、OMS/OBS用バイナリパッケージを、なるべく最新のものとコンパイルし提供しておりますが、アプリケーションの対応状況によっては複数バージョンのバイナリパッケージを用意する場合もあります。

こうした情報はOMS/OBSのサポートサイト「オープンラボトリ」に掲載されていますので、是非ご利用ください。

オープンラボトリ http://www.plathome.co.jp/support/labo/

PHPアプリケーションMODxをインストールする

MODx はコンテンツ管理システム(CMS)として高く評価されているPHPベースWebアプリケーションです。

従来のWeb構築は、PCのローカルでhtmlドキュメントを作成しFTPでアップロードする方法が一般的でしたが、MODxなどのCMSは、Webブラウザ上で複数メンバーが同時にコンテンツ編集できるため、wikiのように手軽なうえ、表現力のあるWebページ作成が可能です。

■MODxで使うデータベースアカウントをMySQLに作成する

ここでは以下の設定で作ります。

- アカウント: admin
- パスワード: admin (実際には任意で)
- データベース: wwwdocs (これは操作可能とするデータベース名)

このデータベースにはhtmlドキュメント類が保存されます。

```
lamp[25]# mysql -u root -p
Enter password: root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.0.51 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> GRANT ALL ON wwwdocs.* TO 'admin'@localhost IDENTIFIED BY 'admin';
Query OK, 0 rows affected (0.01 sec)

mysql> exit
lamp[26] #
```

クエリーの実行が成功したらexitでMySQLを抜けます。

■MODxをダウンロードしOMSへ転送する

MODxのサイトから最新版をPCへダウンロードします。

http://modx.com/downloads.html (2008年4月4日現在 0.9.6.1)

ダウンロードされるファイルはmodx-0961p2.zipです。これを先にPC上で解凍し準備しておきます。

次にftpソフトを立ち上げます。(ここではFFFTPを使って説明します。)

接続先ホストの設定は次の通りです。

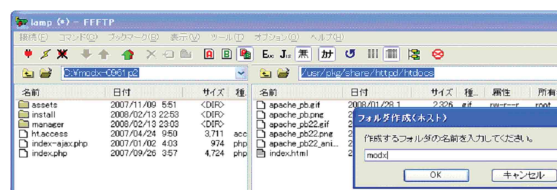
- 設定名: lamp (実際には任意)
- ホスト名: 172.16.12.239 (OMSに設定されたIPアドレス)
- ユーザー名: admin (OMSの基本設定で追加したアカウント)
- パスワード: admin (adminアカウントに設定したパスワード)
- ローカルの初期フォルダ: c:\¥ modx-0961p2 (MODxを解凍したフォルダのパス)
- ホストの初期フォルダ: /usr/pkg/share/httpd/htdocs(Apacheのドキュメントルート)

設定項目を書き込んだら接続します。

接続するとローカルにはMODxの解凍されたディレクトリが表示され、ホスト側にはApacheのドキュメントルート(Apacheの実行環境を設定する項で説明)が表示されます。

ホスト側のドキュメントルートにmodxディレクトリを作成します。

なお、Web公開にMODxだけをメインに使う場合は、modxディレクトリを作らず、ここをwww全体のドキュメントルートとして、MODxを直接置いてかまいません。
この場合、MODxにドキュメントルートの直下でアクセスできます。



ディレクトリを作成したらmodxへ移動してローカルのMODxフォルダの内容を全てアップロードします。

製品仕様

	OpenMicroServer™	OpenBlockS™266
製品名	OpenMicroServer™	OpenBlockS™266
製品画像		
型番	OMS-AL400/128	OBS266/128/16R
メーカー希望小売価格	オープン	
WEB・直販価格	56,400円(税込) (本体価格:53,715円)	45,800円(税込) (本体価格:43,620円)
保証期間	弊社出荷日より3年間(センドバック)	
CPU	RMI Alchemy au1550 400MHz 消費電力:0.5W(Typical)	AMCC PPC 405GPr 266MHz 消費電力:1.5W(Typical)
MEMORY	128MB (PCB基板直接実装 容量変更不可) PC133 SDRAM 512Mbit(64MB) チップ x 2枚	128MB (PCB基板直接実装 容量変更不可) PC133 SDRAM 512Mbit(64MB) チップ x 2枚
FLASH ROM	16MB (ユーザーエリア 約2MB) NOR型 Spansion チップ	16MB (ユーザーエリア 約3.6MB) NOR型 Spansion チップ
ギガビットイーサ (10/100/1000Base)	2ポート Intel E1000チップ (LU82541PI) x 2	なし
ファスティサ (10/100Base)	1ポート CPU au1550内蔵 (PoE受電ポート)	2ポート CPU PPC 405GPr内蔵 x 1 DEC交換Chip(DAVICOM) x 1
Serial	2ポート RS232C仕様(FULL結線) x 1ポート コンソール出力専用(Rx-Txのみ) x 1ポート 形状:RJ-45 x 2 ・RJ-45 - DSub9ピン変換コネクタ x 1 標準添付。 ・2ポートを使用する場合、別売オプション「PH-RD/RH」が 必要。 *RS232Cケーブルは別途ご用意ください。	1ポート RS-232C仕様(FULL結線) x 1ポート 形状RJ-45 x 1 ・RJ-45 - DSub9ピン変換コネクタ x 1 標準添付。 ・別売オプション変換アダプタ「PH-AD/RH」をAUXポートに接続することで、コンソール追加可能。 *RS232Cケーブルは別途ご用意ください。
USB	USB2.0(TYPE B) x 2ポート (NEC「uPD720101F1-EA8-A」チップ)	なし
DIO	Digital I/O x 8bit (IN-4bit/Out-4bit) 2mmピッチピンX8	なし
内蔵ストレージ	CF TYPE-2 (PIOモード4 CFカード)	CF TYPE-1 (DMA/PIOモード CFカード) IDE 2.5" Solid State Drive (UDMA100) *CFとSSD(Solid State Drive)は排他仕様。
スイッチ	INITスイッチ	
表示・警告LED	ステータス x 3、LANアクセス	
外形寸法	本体: 102(W) x 230(D) x 33(H)mm ゴム足0.5mm ACアダプタ: 49.5(W) x 99.5(D) x 25.5(H)mm (突起部を除く) ACアダプタ-本体間ケーブル: 126.5mm ACアダプタ-ACコンセント間ケーブル: 202.5mm	81(W) x 114.5(D) x 38(H)mm ゴム足0.5mm 40(W) x 94.2(D) x 30.5(H)mm (突起部を除く) 152.5mm 200.5mm
本体材質	アルミ合金製	
重量	500g以下	
電源	DC12V/3.0A (専用ACアダプタ付属)	
消費電力	単体時/CF搭載時: 10.1VA(8.1W)以下 / 11.4VA(9.1W)以下 動作時: 9.5VA(7.6W)以下 / 10.6VA(8.5W)以下 待機時: 9.5VA(7.6W)以下 / 10.6VA(8.5W)以下	単体時: 8.7VA(4.5W)以下 / 9.8VA(5.2W)以下 / 9.8VA(5.2W)以下 CF搭載時: 8.7VA(4.5W)以下 / 8.7VA(4.5W)以下 / 8.7VA(4.5W)以下 SSD搭載時: 8.7VA(4.5W)以下 / 8.7VA(4.5W)以下 / 8.7VA(4.5W)以下
電源条件	許容電圧変動: 90V~240V	
動作温度条件	0~50°C (PoE受電時) 0~40°C (ACアダプタ使用時)	0~40°C
動作湿度条件	20~80%Rh *結露しないこと	
保管温度条件	-10~60°C	
保管湿度条件	10~80%Rh *結露しないこと	
温度勾配	15°C/Hour 以下	
湿度勾配	30%/日 以下	
振動 (動作時)	3~100Hzサイン波 往復2分 x 5(各XYZ) : 0.1G以下 2~10Hzサイン波 往復15秒 x 5(各XYZ) : 0.2G以下	
振動 (休止時)	3~60Hzサイン波 往復2秒 x 5(各XYZ) : 0.4G以下	
耐衝撃 (梱包落下時)	レベル3	
静電気ノイズマージン	接触: 5kV 間接: 9kV	
実装部品	PoE受電部分に固形コンデンサを搭載。 (ケミカルコンデンサ非使用)	電解コンデンサの搭載なし。 (すべてチップコンデンサを使用)
MTBF	OMS本体: 420,000時間 付属ACアダプタ: 50,000時間	OBS本体: 1,480,000時間 付属ACアダプタ: 50,000時間
グリーン購入法	適合	
エネルギー消費効率	区分 h 効率 0.011	
標準付属品	・専用ACアダプタ 1個 ・CD-ROM (マニュアル他)1枚 ・保証書・警告用紙各1部 ・RS232C用 RJ45-Dsub9ピン(メス)変換アダプタ 1個 ・RS232C用 10Base-Tストレートケーブル 1本	・専用ACアダプタ 1個 ・フェライトコア 1個 ・保証書・警告用紙各1部 ・SSD(Solid State Drive)装着用絶縁シート 1枚 ・RS-232C用 RJ45-Dsub9ピン(メス)変換アダプタ 1個 ・RS-232C用 10Base-Tストレートケーブル 1本

ソフトウェア仕様

出荷時標準 (括弧内はリリース済の最新バージョン)

製品名	OpenMicroServer™	OpenBlockS™266
ディストリビューション	SSD/Linux STABLE-20051208 0.4-RELEASE-20071208	SSD/Linux STABLE-20070702 0.4-RELEASE-20071020 [0.3-RELEASE-20071020] [0.2-RELEASE-20071110]
カーネルバージョン	2.6.12 (2.6.16)	2.6.16 (2.6.16) , [0.3: 2.4.26] [0.2: 2.4.20]
Glibc	2.3.3 (2.3.6)	2.3.6 (2.3.6) [0.3: 2.3.3] [0.2: 2.3.1]
Gcc	3.3.5 (4.1.2)	4.1.2 (4.1.2) [0.3: 3.3.5] [0.2: 3.2.3]
ファイルシステム	ext2 , ext3	ext2 , ext3
IPプロトコルスタック	IPv4 (IPv4/IPv6)	IPv4 (IPv4/IPv6) [0.3: IPv4] [0.2: IPv4]
標準機能	デフォルトルーティング、スタティックルーティング、iptablesによるパケットフィルタリング、LinuxカーネルモジュールによるIPマスカレード、等	
その他機能	JumboFRAME、タグ付きVLAN	-
その他対応OS	Debian 4.0 etch / 3.1 sarge、NetBSD4.0~	Debian 4.0 etch / 3.1 sarge、Fedora8、NetBSD2.1~

* 弊社ではポリシーとして、出荷時ファームウェアは製品リリース時のバージョンを基本とし、極力変更を避けています。
最新版は弊社WEBサイトよりダウンロードしてご利用いただけます。
* OBS266/128/16R は、2007年8月リリース時に、出荷時標準構成が「SSD/Linux 0.4」をベースに以下の通りとなりましたが、旧バージョン(OBS266、OBS266/128)との互換性を保つ、旧メジャーバージョン「SSD/Linux 0.2」「SSD/Linux 0.3」も弊社WEBサイトよりダウンロードしてご利用いただけます。

適合・準拠規格

■安全設計準拠安全規格適合

国際電気標準会議:IEC950準拠
北米:CSA/UL規格認定
欧州:CEマーク取得
日本:電安法<PS-E> 対応
電波法、電気通信事業法は対象外

■グリーン購入法適合

省エネ法に基づく表示はWEBサイトの各仕様に記載
*OpenBlockS266は、CPUのクロック周波数が低いので規制対象外

■妨害電波規格適合/準拠

北米: FCC class A 取得
日本: VCCI class A 準拠

■RoHS指令対応

鉛、水銀、カドミウム、六価クロム、PBB、PBDE 非含有

オプション

製品名	OpenMicroServer™	OpenBlockS™266
専用コンパクトフラッシュ※1 (OMS-AL400/128、OBS266/128/16R 専用)	・PH-128M/RH (128MB) ・PH-256M/RH (256MB) ・PH-512M/RH (512MB) ・PH-1GB/RH (1GB) 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】	・PH-2GB/RH2 (2GB) ・PH-4GB/RH2 (4GB) ・PH-8GB/RH2 (8GB)
ラックマウントキット	OMS-RM4/RH WEB・直販価格 28,140円(税込) (本体価格:26,800円) 19"ラック・1UにOMS x 4台搭載可能 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】	OpenBlockS266用ラックキット(RoHS対応) OBS-RM5/RH 28,140円(税込) (本体価格:26,800円) 19"ラック・1UにOBS x 5台搭載可能 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】
シリアル変換アダプタ	PH-RD/RH (クロス結線) WEB・直販価格 1,980円(税込) (本体価格:1,886円) 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】	PH-AD/RH (クロス結線) WEB・直販価格:9,975円(税込) (本体価格:9,500円) 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】
Solid State Drive	- 内蔵搭載不可(USB接続デバイスをご使用ください)	PH-16GB/SSD(Solid State Drive) PH-32GB/SSD(Solid State Drive) ※価格はお問合せ下さい。 【保証期間: 弊社出荷日より1年間(センドバック)】 【RoHS対応】

※1 旧機種(OBS266、OBS266/128)ではご使用できません。ご注意ください。旧機種専用コンパクトフラッシュは別途お問合せ下さい。

OEMについて - 自社ブランド化をトータルにサポート

貴社ブランドとしてOEM販売を行う場合でも、ぶらっとホームによるトータルなサポートをご利用いただけます。

■カバーデザインの変更

アプライアンス製品化の際には、ケース天面の製品ロゴを、貴社の製品名やロゴに変更することも可能です。

■製品個装箱の変更

無地または指定デザインの箱へ変更可能です。

■キッティング作業

作業内容例:
・CFカードの搭載
・ソフトウェアのインストール(ソフトウェアインストール済のマスターファイルをご提供いただきコピーします)
・設定変更(内蔵FLASH ROM起動 → CFカード起動への変更など)
・動作確認(装置やソフトウェアの起動確認)

※価格やロット等、詳細は営業担当までお問い合わせください。

USA版

型番:OMS-AL400/128 -USA、OBS266/128/16R-USA

ファームウェア、ドキュメント、電源ケーブル※等が北米仕様に変更されています。

詳細は営業担当までお問い合わせください。

※最大入力電圧120V Aタイプ