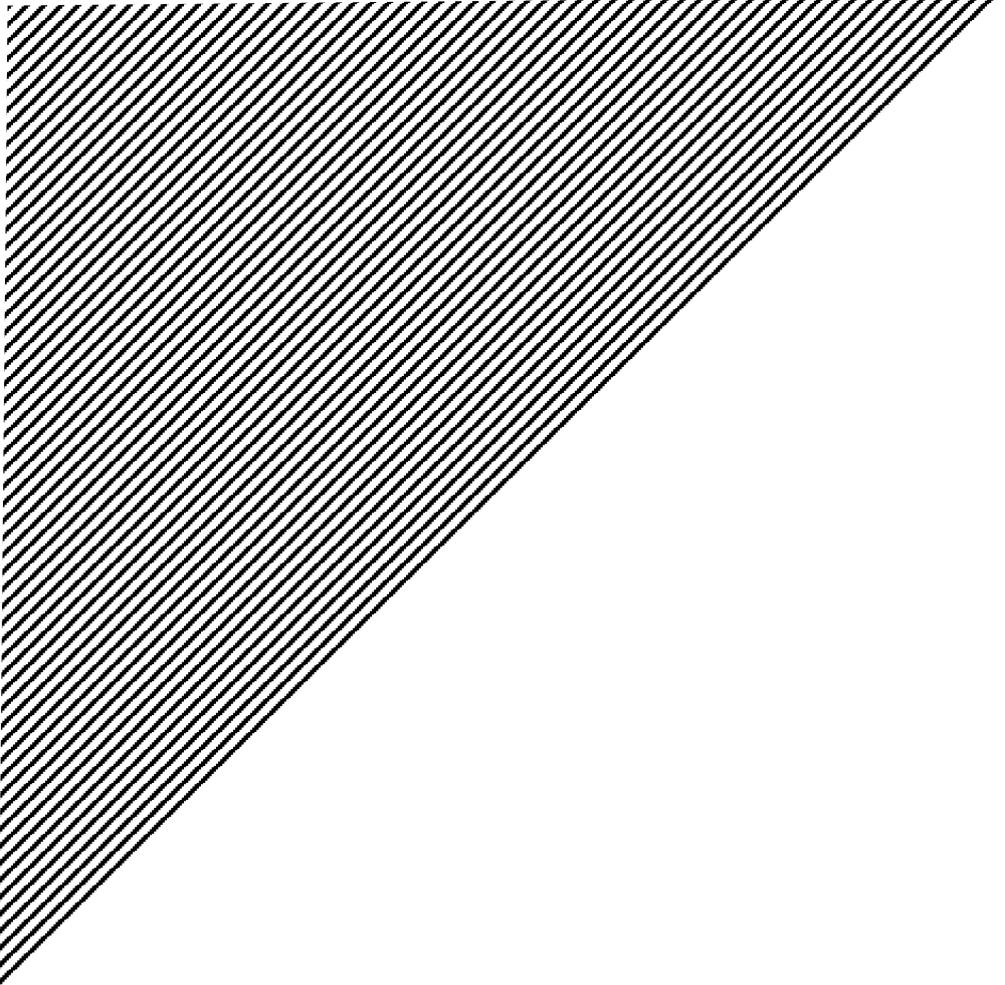


OPERATING SYSTEM MANIACS

OPERATING SYSTEM MANIACS

VERSION 9.0

VERSION 9.0



-Boot Loader-

03 Mac OSX on VMWare on Windows

08 OpenStack Install and Construct

14 Cisco UCS Platform Emulator

18 Cisco IOU/IOL

All Writen by Shoichi tatsukami

Far Northern Other World

Mac OSX on VMWare on Windows

・この記事は？

Windows 上で稼働している VMWare への Mac OSX のインストールについてです。

・サマリー的な

1. VMWare Player を Windows 上にインストール
2. VMWare に胡散臭いパッチをあてて Mac OSX 対応にする
VMWare Fusion 相当にしているものと思われます。
3. Mac OSX をインストール

細かい Version については当方がライセンス付きメディアを所持している
MacOSX Server 10.5、
Mac OS X 10.6 Snow Leopard、
Snow Leopard からダウンロードした OS X 10.11 El Capitan
になります。

これらの Version による差異もあるのでご注意ください。

また、当方のテスト VMWare 環境は 4CPU、メモリ 8GB を割り当てるといってそれなりにリソースが潤沢な環境でテストしています。

割り当てメモリや CPU のリソースが少なければ、インストール後の挙動は当然に重たくなります。
どの程度あれば実用的であるとの判断はできませんが、当面上記程度のリソースを割り当てたところは仮想マシン上でもそれなりに動いております。まあ長期間利用したらわかりませんが。

なお、10.5 Server ではソフトウェアアップデート後に再起動したら設定してた固定 IP でつながらなくなりました。DHCP で取得させたらちゃんとインターネット含め接続できましたが。理由不明です。
また、ネットワークに接続可能な状態でも、「ネットワーク設定」を見ると「ケーブルが接続されていません」という表示になってしまいますが、アップデートもブラウザも可能でした。

CPU については仮想マシン上の話なのでなんともですが、4CPU 割り当ててるんですが、仮想マシン上の MacOSX 上では 2CPU と表示されます。

あんまり細かいところをトレースしてないですがなんかしら理由はあるんでしょう。

Snow Leopard の場合、Yosemite の場合と細かいコメントは追記しますが基本的には 10.5 Server でのインストールです。

(というか本来仮想サーバにインストールがライセンス的に許容されてるのは Server OS だけのはず)

・一応の詳細手順

1. VMWarePlayer のダウンロードとインストール

VMWare Player

をダウンロードして、インストールします。

とりあえず VMWare Player は

Version 12.1.1 build-3770994

での動作を確認しています。

<https://www.vmware.com/jp/>

「ダウンロード」⇒ポップアップしたメニュー内の「無償製品のダウンロード」⇒「Workstation Player」

として選択します。

ダウンロードしたら普通にインストールしますが、ライセンス云々言われますが商用利用するわけではないのでライセンスを入力せず使い続ける、で OK。

インストールの細かい選択肢は端折ります。お好きにどうぞ。さすがに悩むレベルのものはないはず。

2. パッチ(VMWare Unlocker 2.08)をダウンロード

まず

<http://www.insanelymac.com/>

にアクセスします。

「Download」

をクリックしてフォーラムに移動すると、画面右手に

「Create Account」

とあるので、新規アカウントを作成してください。

ユーザー名、メールアドレス、パスワードを入れる標準的なユーザー登録です。

登録後、メールアドレスにレジストメールが来るので、メール内の URL をクリックして

レジスト完了後、ログインしてダウンロードが可能になります。

Download ページの中を検索するか、

「Miscellaneous」⇒「Unlocker」

と辿ってダウンロードしてください。

テストした時点での最新は 2.08 です。

(後述しますが場合によっては 2.07 も必要)

・・・ぶっちゃけパッチなんて言ってますが実質クラックみたいなものです。

あと当方がレジストしたときに自分で入れたパスワードがうまく通らなくてしょっぱなからパスワードリマインダ使うハメになりました。

なんらか怪しいことがあったりしても責任は取れませんのでそのあたりご了承下さい。

3. Mac OSX のインストール

方法は問いませんが OSX を購入してインストールします。

当方はもろもろあって OSX Server10.5 を所持していましたのでそれで試しましたが、

<http://www.apple.com/jp/shop/product/MC573J/A/mac-os-x-10.6-snow-leopard>

から直接 Snow Leopard のメディア購入するのが現状だと一番安価に入手可能かと思います。

税込みでも 2400 円ですし、このくらいはお布施しないと。

パッチを当てた後の VMWare 上の表記を見ると、10.5～10.10 までインストール可能なように表示されています。

Version 表記を見るとわかるのですが、10.x Server と 10.x は別のもので認識されますし、インストール時の注意事項的な部分も別物です。

インストール時の注意点は以下です。

・ISO イメージにしてマウントしたほうがいいらしい

⇒購入したメディアを ISO イメージにし teVMWare からはイメージをマウントしてインストールしましょう。

・新規仮想マシンの作成時

⇒パッチを当てたことで候補に増えている「Apple MacOSX」を選択する。Version はインストールしようと試みるものに合わせた Version に。

・仮想ディスクは単一ファイルにしたほうがよさげ

⇒容量はあらかじめ確保しなくても大丈夫そうです。

・任意のメモリ容量・CPU コア数・サウンドカードを追加を設定

追加のハードウェア設定でメモリ、CPU は当然ですが、サウンドカードも指定したほうがよさげらしいです。すみません、この辺正確な裏を取ったわけではありません。

前述の通りサウンドカードはドライバをインストールしてる記事もあるようですが当方では何もしなくても音は鳴ってました。El Capitan は鳴るのかな？

・ディスク領域の確保

⇒仮想マシンの再生後、ISO イメージから起動してくるが、そのままではインストールディスクが見えないので、ディスクユーティリティでインストールするディスクをフォーマットします。

当方は「消去」タブを選択して領域確保しました。

・インストールドライブの選択

⇒これ以降は特に注意することはありません。

インストールドライブを指定してインストールしましょう。

キーボードを設定したい場合

```
keyboard.vusb.enable = "TRUE"
```

```
keyboard.vusb.idVendor = "0x05AC"
```

```
keyboard.vusb.idProduct = "0x020D"
```

を*.vmx に記載して、起動時のキーボード設定アシスタントに従って設定してください

あと、もろもろ見るとサウンドカードのインストール云々の記事があるところもありますが、当方では何もしないでも音が鳴ってたので気にしてません。

メモリ、CPU 数に糸目をつけなければそれなりに使い物にはなりますが、

実験としては簡単すぎますし、ライセンス的には黒すぎるのであんまり

オススメできるというわけではありませんが、後述する El Capitan の ISO イメージ作成だけはちょっと勉強になりました。

なお、別段 VMWare にクラック気味にインストールしたとてそれで終わりではなく、Linux 上の KVM で MacOSX を動かしてみるという計画も立てているのですが、こちらは当然ながら VMWare よりは敷居が若干高く、ここでは簡単に触れるにとどめます。



少し見難いですが、VMWare 上で起動している様子です。

以下、追加の注意事項です。

・Snow Leopard インストール時の注意

仮想マシンの*.vmx ファイルに、

```
efi64.filename = "efi64-srvr.rom"
```

```
smc.version = "0"
```

を追記

あと実際には CPU によっては

```
cpuid.1.eax = "0000:0000:0000:0001:0000:0110:1010:0101"
```

を追加する必要がある模様。当方は追記してません。

また、[unlocker2.07](#) の firmware フォルダにある

```
efi64-srvr.rom
```

(2.08 には入っていません。別途 2.07 以前を入手する必要があります)

を仮想マシンのフォルダにコピーする必要があります。(もともと、仮想マシンには 10.6 Server しかインストールできないのですがそれをだまらかすためのものの模様)

※っていうかこれたぶんファームウェアっぽいのでそういう意味では真っ黒だなこれ。

・OS X 10.11 El Capitan インストール時の注意

Snow Leopard をインストールし、10.6.8 までアップデート

⇒El Capitan をダウンロード

⇒ダウンロードした El Capitan を ISO イメージに変換

ということを行って、ISO イメージを生成することで、EL Capitan をインストールすることができます。

ISO イメージを Snow Leopard 上で生成するには以下のコマンドで行います。

```
>hdiutil attach
```

```
>/Applications/Install¥ OS¥ X¥ El¥ Capitan.app/Contents/SharedSupport/InstallES  
D.dmg -noverify -nobrowse -mountpoint /Volumes/install_app
```

※実際の入力是一行

```
>hdiutil create -o /tmp/ElCapitan.cdr -size 7316m -layout SPUD -fs HFS+J
```

```
>hdiutil attach /tmp/ElCapitan.cdr.dmg -noverify -nobrowse -mountpoint  
/Volumes/install_build
```

※実際の入力是一行

```
>asr restore -source /Volumes/install_app/BaseSystem.dmg -target  
/Volumes/install_build -noprompt -noverify -erase
```

※実際の入力是一行

```
>rm /Volumes/OS¥ X¥ Base¥ System/System/Installation/Packages
```

```
>cp -rp /Volumes/install_app/Packages
```

```
/Volumes/OS¥ X¥ Base¥ System/System/Installation/
```

※実際の入力是一行

```
>cp -rp /Volumes/install_app/BaseSystem.chunklist
```

```
/Volumes/OS¥ X¥ Base¥ System/BaseSystem.chunklist
```

※実際の入力是一行

```
>cp -rp /Volumes/install_app/BaseSystem.dmg
```

```
/Volumes/OS¥ X¥ Base¥ System/BaseSystem.dmg
```

※実際の入力是一行

```
>hdiutil detach /Volumes/install_app
```

```
>hdiutil detach /Volumes/OS¥ X¥ Base¥ System/
```

```
>hdiutil convert /tmp/ElCapitan.cdr.dmg -format UDTO -o /tmp/ElCapitan.iso
```

```
>mv /tmp/ElCapitan.iso.cdr ~/Desktop/ElCapitan.iso
```

これで、デスクトップ上に ISO イメージが生成されます。ただしサイズは 7GB 超なので注意のこと

OpenStack Install and Construct

・この記事は？

最近、仕事で関わることがありましたので(というか当面関わることになるのですが)、OpenStack について勉強をしていました。

で、LPIC がやってる OpenStack の認定資格である OPCEL に無事合格しましたので、合格記念記事ということで、OpenStack のインストールに関して簡単に書いてみようかと思えます。とはいえ若干 OS って話からは外れるような気もしますが。(まあ OpenStack も略称 OS ですが)

最近はこの手の技術や OSS 関連について時間をかけていろいろやるモチベーションがあまり高まっていなかったため、久々な感じです。

OpenStack は覚えることもやることも多いのですが、なんせ OSS ですから家でもいろいろな試験を試みれるのがいいですね。

・OpenStack とはなんぞや？

OpenStack はオープンソースで開発されているクラウド環境構築用のソフトウェア群です。クラウドの分類のうち、いわゆる IaaS と呼ばれる仮想マシンとストレージ、ネットワークといった、一番低いレイヤーのリソースを提供するクラウド環境が構築できます。

仮想マシンを制御する nova、ネットワークの Neutron、ストレージの Cinder、認証系の Keystone、などなど様々な機能を持ったコンポーネントの集合体です。

細かい話は省略しますが(というかキリがないので)簡素にはじめる方法をざっと書きたいと思えます。

・インストールについて事前整理

本格的に？構築する場合は当然ながら予め構成をきちんと検討しておく必要がありますが、試験環境的に構築する方法で今回は取りまとめます。

細かくきちんと構築するには Packstack というツールを使って、Answer-file という自動応答ファイルに構成を反映したものをきちんと書いた上でツールを実行する必要があります。(というか、冗長構成などを組みたければ Packstack での構築は難しかったです)

で、試験環境を作る際には、とりあえず動かす方針として「All-in-one 構成」といわれる方式で構築します。実運用では、コントロール用のサーバと、仮想サーバを動かすコンピュータノードと呼ばれるサーバに分離するなどの構成をとったり、コントロール用のサーバを A/S 構成にしたりと可用性を高める工夫をしますが、今回はそのあたりは触れません。

また、今回は Libery という Version を入っていますが、Version 話も長くなるので今回は割愛しています。

1. ベース Linux のインストール

実機でも VM でもいいですが Linux をインストールします。

今回は実環境が REHL になる予定なので、なるべく合わせるべく CentOS を選択しています。

また、当方は VMWare(Player)にインストールしています。

2. インストール時の注意

VM 上の場合、VM の「CPU の仮想支援機能」の有効化を行う必要があります。

KVM では下記の Nested KVM の有効化、それ以外はソフトウェアの仮想マシン作成時のオプションを有効にします。

また、KVM で構成したい場合、インストール後に以下の作業を実施する必要があります。

(PackStack が VM 上であることを検知して Qemu で VM を構成しようとするので)

```
/etc/nova/nova.conf
```

```
virt_type=qemu
```

を

```
virt_type=kvm
```

に修正

※資料、Web では「libvirt_type」となっているものがあるが、実際の nova.conf を見たところ「virt_type」となっていた。

※とはいえ別に検証用であれば qemu(ソフトウェアエミュレーション)でもそこまで影響は無いような。

※Nested KVM にする場合は、大元の Host Linux でも

```
/etc/modprobe.d/kvm-intel.conf
```

に

```
options kvm-intel nested=1
```

の記述が必要になりますが(カーネルオプション有効にして再起動)、これは OS の状況、Version 等にもよるようです。

3. インストールに先行して行う作業

ここではまっさらな環境にインストールすることを想定して記載します。

3-1. CentOS のインストール

CentOS は minimal インストールします。ただし必要なものがいくつかあるので、それについては別途インストールします。

minimal インストール時の注意事項はパーティション構成。一般的に自動割り当てで確保されるパーティションのうち、

```
/home
```

に大きい容量が割り振られるので、これを削除して/配下に容量をつけ加えるのが無難。

CentOS の場合、インストール時に日本語化して構わない模様です。

(Ubuntu の場合、コンソールの日本語化がはなはだしいので英語のまま進めるのが無難な模様)

あとはインストール進行中に root パスワードとユーザー作成をして完了

■実環境メモ

・VMWare Player にインストールを実施

VM の構成

メモリ 8GB

HDD 200GB

NIC 1つ

VMWare での仮想支援機構を有効化

ネットワーク IF はブリッジを使用

CentOS を Minimal-ISO を使ってインストール

3-2. インストールした CentOS 上での追加作業

・Static-IP 化

まずは構築時に Static-IP が欲しいので設定を変更。ディストリビューションによって異なると思われるが、それぞれのディストリビューションに応じた方法で設定変更します。

■実環境メモ

CentOS の場合、6.5 あたりから「NetworkManager」が導入されていますが、OpenStack のインストール時には無効化します。(というか有効なままだと OpenStack のインストールに失敗します)

```
systemctl stop NetworkManager
```

```
systemctl disable NetworkManager
```

・interface に個別 IP を振る
ローカル環境の IP アドレスレンジから割り当てています。
アドレスは任意の範囲で調整して下さい。
あと IF のファイル名は環境によって異なると思います。
vi /etc/sysconfig/network-script/ifcfg-ifxxxxxx
BOOTPROTO=static #dhcp から書換
IPADDR=192.168.2.201
NETMASK=255.255.255.0
GATEWAY=192.168.2.1
DNS=192.168.2.1

ネットワークサービスを起動(再起動)する
systemctl restart network

※必須ではないが有用
NetworkManager が入っているため既存?のネットワーク関連コマンド類が入っていません。
(ifconfig とか)
なので入れておいた方がトラブルシューティングなどの際に有用かもしれません。
yum -y install net-tools

3-3 NTP の導入

All-in-one 構成ではあまり関係ありませんが、複数ノード構成の場合確実に NTP にて時刻同期を取っておく必要があります。このため NTP を導入します。
無論まとめてタイムサーバ的なやつにアクセスしてもいいみたいですが。

■ 実環境メモ

実際には Chrony を使って構築しているようですが、当方は All-In-One 環境なので手を抜いて入れていません。

Chrony での設定はインストール後でもできますので、All-In-One の場合はそれでもよいかと。
複数ノードで構成する場合は、ノード間の同期を取る必要がありますので、1 台のコントロール用のサーバをいわゆるパブリックな NTP サーバで時刻同期を行い、残りのサーバはコントロール用のサーバと同期を行うような構成が標準的なようです。

PackStack を利用してインストールをする際には、インストール後に
/etc/chrony.conf

を編集し、
server NTP_SERVER iburst
を追加します。複数の設定も可能です。

※NTP_SERVER は実際に使用する NTP サーバ名もしくは IP アドレス。各プロバイダが提供している NTP サーバ、および公開 JST サーバを指定すればよいかと。

3-4 DNS

試験環境としては必須ではないが必要に応じて DNS についても検討する必要がありますが、今回は割愛しています(あくまでローカル環境での構築のため)
なお、余談ですが OpenStack には DNSaaS ともいえる「designate」というコンポーネントもあつたりします。

4. OpenStack 導入準備

順番としては

4-1 CentOS のアップデート

4-2 RDO リポジトリの設定 ※CentOS では実際は不要

4-3 Openstack-packstack のインストール(Openstack のインストール構成ツール)

■ 実環境メモ

```
>yum -y install centos-release-openstack-liberty epel release yum -y update >yum  
-y install openstack-packstack
```

で、OpenStack インストーラーをインストールし、その後 OpenStack の構成を行います。

なお、

```
4-1 yum -y update
```

```
4-2 yum install -y https://rdoproject.org/repos/openstackrdo-release.rpm
```

```
4-3 yum install -y openstack-packstack
```

という情報もありますが、CentOS の場合は上記にも記載のとおり 4-2 は不要で

```
yum install -y centos-release-openstack-liberty
```

 だけでよいようです。

(packstack 使うなら `yum install -y openstack-packstack`)

理由は簡単で、Extres リポジトリに OpenStack が含まれているのですが、CentOS の場合はそもそもデフォルトで対象になっているため。

RHEL の場合は上記 4-2 の作業が必要。また、自動更新の類は OFF になっていること(通常 OFF かな?)

また、docs.openstack.org を見ると

```
yum install -y python-openstackclient
```

```
yum install -y openstack-selinux
```

をインストールすることになっているが、OpenStack の Version によっては不要なようです。

(厳密には不要というより構成、インストールする中であわせて導入されている模様)

ただし SELinux 設定は以下のように変更しています。

All-In-One 構成の場合などはあまり影響は無いのですが、ネットワーク関連の複雑な設定を行う際に意図したように動かない原因になるようです。

■ 実環境メモ

```
/etc/SELinux/config
```

```
enforce ⇒ permissive
```

5. OpenStack 導入

理想形としては、answer-file を作成して、必要なオプションを設定したのち、その Answer-file を用いてインストールを行うことだと思われる。

・Answer ファイルの生成

```
packstack --gen-answer-file=xxx.txt
```

・Answer ファイルを指定しての packstack の実行

```
packstack --answer-file=xxx.txt
```

※各種コマンドでのオプション指定法。外部ネットワークに繋げるときは

```
--provision-all-in-one-ovs-bridge=n
```

にしておいた方がいい? というような記事もありましたが当方は設定してません。

```
--ntp-servers=xxxxxx
```

ntp サーバは指定できるなら指定したほうがよろしいらしいですが、前述のとおり当方は指定していません。

```
--cinder-volumes-size=xxG
```

Cinder というストレージ領域のサイズですが、下記に別記のとおり、デフォルト設定では Cinder がグループバックスの領域を利用するので、そのサイズ以上にはできないので注意。

ストレージ領域だけでもある程度潤沢に確保したい場合は下記に記載の対応が別途必要になります。

■実環境メモ

簡単に環境を作るには--allinone オプション

```
packstack --allinone
```

⇒Keystone Galnce Nova neutron Swift Cinder Ceilometer がインストールされる

・Cinder の認証がうまく設定されていない

Version が Liberty の場合ですが、そのままインストールすると cinder の認証がうまく動かないので領域確保に失敗するようです。

cinder.conf に以下を追記することで認証は通る模様です。

```
[keystone_authtoken]
auth_uri = http://localhost:5000
auth_url = http://localhost:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = services
username = cinder
password = CINDER_PASS
(CINDER_PASS は answer ファイルの記載パスワード)
```

・Cinder 領域を実際のストレージ領域から確保する

```
CONFIG_CINDER_VOLUMES_CREATE=y
```

↓

```
CONFIG_CINDER_VOLUMES_CREATE=n
```

この設定を y にしていると、ループバック領域を確保してしまいます。

また、これを単に n にしただけだと Cinder 用の Volume(cinder-volumes こちらも設定に記載有)がないので結局ダメなので、「cinder-volumes」という名前で VG を作成してから Packstack を実行する必要があります。

よって、実際には Cinder を有効に使うには

1. Answer-File を生成

1. Answer-File の「CONFIG_CINDER_VOLUMES_CREATE」を書き換え

2. 「cinder-volumes」という名前で VG を作成

3. Packstack --answer-file={answer-file 名} を実行する

という順番で実行し、その上で、

4.cinder.conf に前述の[keystone_authtoken]以下を追記する

という手順を行う必要があります。



OpenStack は Horizon というコンポーネントが Web GUI を提供していますが、そこにアクセスするとログイン画面が現れます。

ここからコントロールパネルにログインし、仮想サーバの作成、ネットワークの構成、ストレージの構成等を行うことができます。もちろんコマンドライン操作も可能。パスワードはユーザーディレクトリの keystoneadmin という環境設定ファイルに記載されています。



・おわりに

OpenStack についての説明を全然してないのでまったくご存知ない方には何がなんやらという感じになってしまいました。そのうち OpenStack 別冊でも作ってみようかしら。

なお、最終的には OpenStack 上で Windows Server と MacOSX を動かしたりできないかなあなどいうことも模索していますが、折角取った資格なので、まじめにそちらの勉強も引き続き進めたいと思っています。

Cisco UCS Platform Emulator

・この記事は？

別段 OpenStack からの流れというわけではありませんが、CISCO の UCS 環境を仮想環境上に構築するためのものです。

1. CISCO UCS ってなんぞや

CISCO UCS は

「Cisco Unified Computing System」

の略称ですが

「ようするに何なのよ？」

というのが実は非常にわかりにくいです。製品名としてはサーバ郡ということのようなのですが、ソリューションのようなものも含めているようでもあり、でも実際には UCS サーバ製品がカタログに載っていたり。

日本で UCS を提供している某代理店的なところに書いてあった以下の情報がややわかりやすかったので一部引用しますが、

「ブレードサーバ、ネットワーク、ストレージ、仮想化技術を統合することで、データセンター全体が単一のシステムとして動作」

する、となっています。

ただ、いわゆるクラウド的なものを構築するだけでなく、デスクトップ統合なども含めているらしく、「Unified」の一言にいろいろ詰め込んでいるようです。

で、正直言うと当方も「わからないでもないが全貌がよく見えない」CISCO UCS ですが、プラットフォームエミュレータが CISCO サイトからダウンロードできるようなので、試してみようかなと思った次第です。

2. 入手

ドキュメントと VM のダウンロードページ：<https://communities.cisco.com/docs/DOC-37897>
(Cisco.com の無料のユーザー登録が必要です)

ドキュメント、VM イメージがダウンロードできます。ただし VM イメージについてはレジストが必要です。

CISCO 社のサイトですすでに登録されている方はそのままダウンロード可能です。

なお、英語ドキュメントですが導入部分については設定変更等含め全てダウンロードできるドキュメント内に記載してあるので、設定等についてはこのドキュメントを見れば済む話なのですが、多少なり書いておきたいと思います。

3. 導入

導入部分は難しいところはまったくありません。

CISCO のサイトからダウンロードした zip ファイルを解凍して、必要に応じてディレクトリを移動し(VMWare の仮想マシンの管理場所に移すとかを想定)そのうえで、「UCSPE.vmx」を VMware で起動すればとりあえず問題なく起動するはずで。

初回起動は、CISCO UCS プラットフォームエミュレータの解凍を行うので多少は時間がかかりますが、そう問題にはならないはずで。

4.起動後の設定

起動後の設定変更周りは多少触れておいたほうがいいかと思うので簡単に記載します。

・アクセス関連

アクセス方法は、大雑把に言って 3 つあります。

1. Web GUI
2. VMWare コンソール
3. ssh(CLI クライアント)

Web GUI はインストール(というか初回起動)完了後に示される IP アドレスを Web ブラウザで開けばアクセスできます。(パスワード制限等なし)

2 は起動直後のプロンプトにも表示されますが、

ユーザー名 : config

パスワード : config

です。

3 は厳密には VMWare コンソールと ssh 接続時のコマンドラインクライアントアカウントのようなものです。

ユーザー名 : cliuser

パスワード : cliuser

にて接続可能です。

※アカウント名のごとく、Linux のアカウントというより、CISCO UCS の CLI 操作アカウントのようです。

・IP アドレスの変更

VMWare で起動した場合は指定されたアドレスをそのまま使っても(テスト環境です)問題は少ないかと思いますが、各自のローカルな IP アドレス状況に合わせる場合に変更が可能です。

VMWare 仮想端末のコンソールの

cisco-ucspe login:

プロンプトに、上記のユーザー名、パスワード「config」でログインします。

ログインすると次ページにあるような画面になりますので、

Select:

プロンプトに

(a) View & Configure Network Settings

を選択し、(プロンプトに対しては"a"だけ入力)、その後の

Change Settings? :

プロンプトに"y"と入力して設定に入ります。

Use DHCP?

プロンプトに対しては"n"と入力します。

以下、IF 数分繰り返します。

Enter IP address

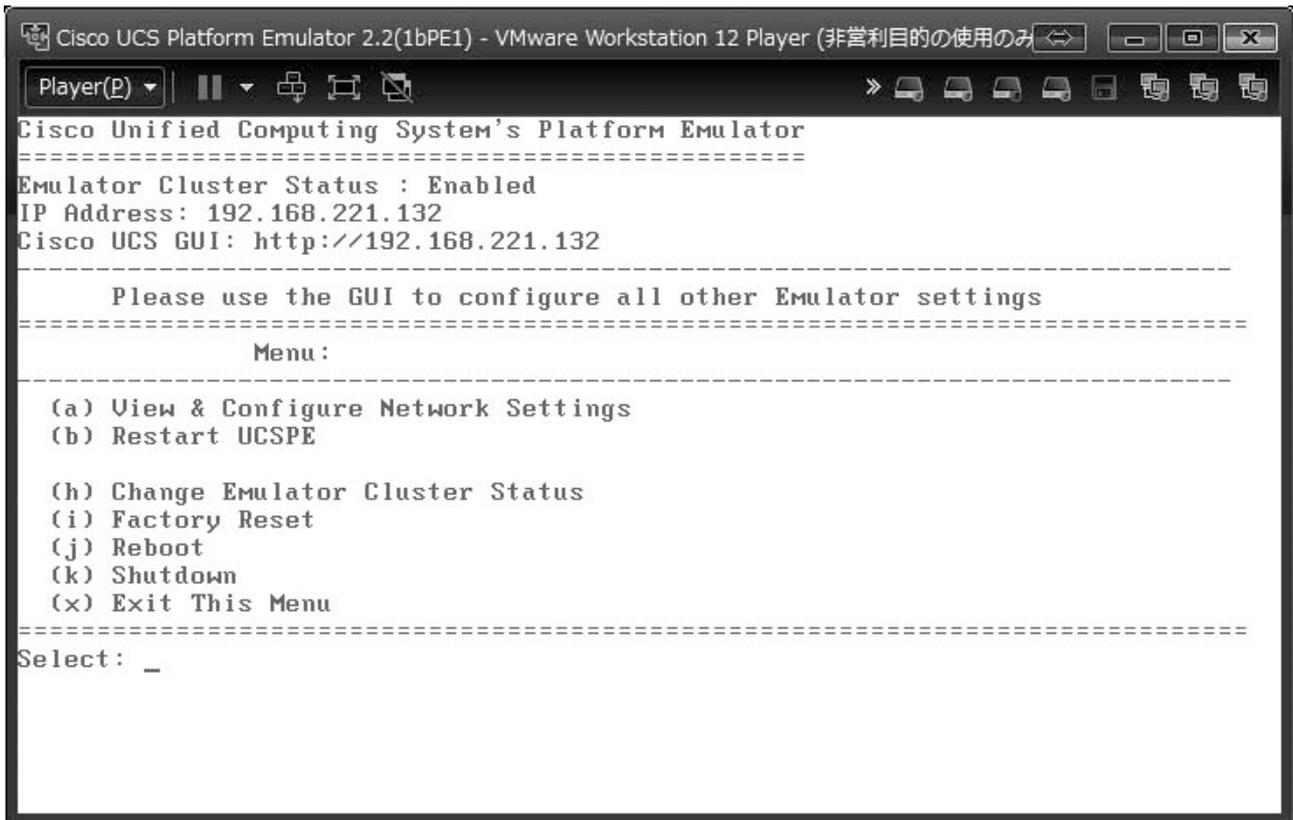
プロンプトに固定 IP を入れて下さい。

Enter netmask

にはネットマスクを

Enter default gateway

にはデフォルトゲートウェイを入れて下さい。



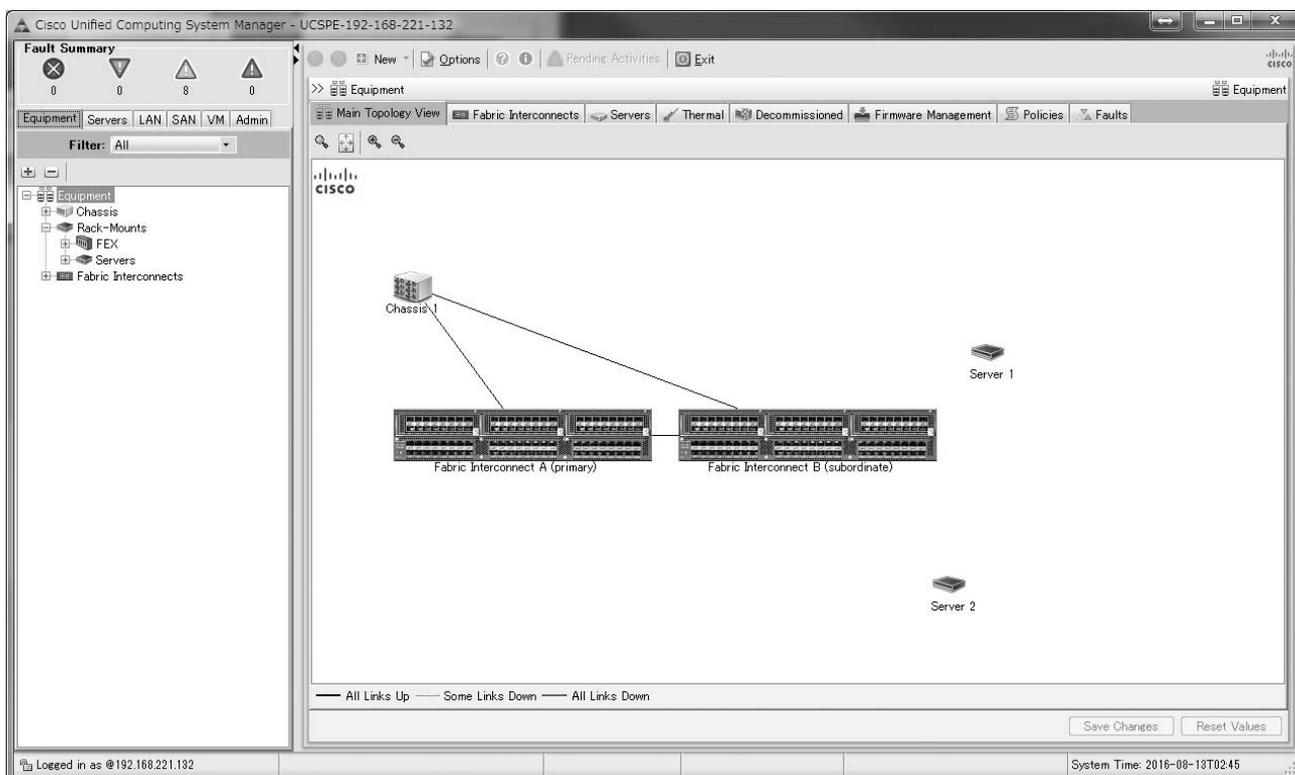
UCS の操作は起動時の画面に表示される IP アドレス(192.168.221.132) をブラウザで開いて、UCS Manager に接続して行います。なお、UCS Manager の Top 画面は開けますが、それ以降の動作は `ucsm.jnlp` をダウンロードして CISCO UCS Manager を起動するので、Java がインストールされている必要があります。



起動時のスプラッシュスクリーン



画面が起動したところ



で、ここから先ですが、ドキュメントの類をまだまだ読み込んでませんのでここまでです。申し訳ない。

VMとあわせてダウンロードできる

Cisco UCS Platform Emulator User Guide version 2.2(1bPE1).pdf

等、しっかり読み込まないとなかなか大変かと思います。

CISCO IOU/IOL

・この記事は？

Cisco IOU/IOL(IOS on UNIX / IOS on Linux)についての記事です。

CISCO の NW 機器をエミュレートさせる場合、Dynamips で Cisco IOS をエミュレートさせ、これを GNS3(Graphical Network Simulator)で取り扱うのが一般的ですが、この場合、L2/L3(というか Catalyst Switch)はエミュレートできなかつたようです。

(すいません、現状は実はまだ把握しきれませんが、過去はそうでした)

Cisco IOU/IOL(IOS on UNIX / IOS on Linux)では、スイッチのエミュレートもできてコマンド操作も可能となります。

ただ、IOU はそもそも CISCO 社内のソフトウェアであり、ライセンス的な問題は最初の記事の MacOSX ON VMWare の比ではありません。

一応グレーゾーンの極みなものであることはご認識いただいた上の記事となります。

2.入手

グレーなためか、現状は肝心の IOU L2/L3 のイメージファイルというものが非常に入手しづらいようです。あちこちググって探せ、とどこのサイトにも書いてあります。

当方もあちこちググって探したクチです。

準備するものは以下です。

1. GNS3 IOU の VM.ova のファイル

⇒これは Sourceforge にありますし、<https://github.com/GNS3>～以下からダウンロードもできます。

https://sourceforge.net/projects/gns-3/files/IOU%20VMs/GNS3%20IOU%20VM_1.3.3.ova/download

※当然ですが実際には 1 行 Github のほうが過去 Version もまとめてダウンロード出来るので良いかもしれません。

2. CISCO IOU L2 と L3 のイメージファイル

i86bi-linux-l2-adventerprisek9-15.1a.bin

i86bi-linux-l3-adventerprisek9-15.4.1T.bin

3. CISCO IOU ライセンスファイル

iourc.txt

4. 任意の仮想マシン

※VirtualBox を使っている解説を載せているサイトが多いですが、VMWare Player に変換して読み込んでも大丈夫そうです。

5.GNS3

<https://www.gns3.com/software/download>

ユーザー登録が必要です。

3.導入

まず、ダウンロードしてきた GNS3 IOU_VM を仮想マシンで起動します。

基本的には単に起動できれば OK っぽいですが、アダプタ設定はホストオンリーアダプタでないとかダメそうなので、そのようになっていることを確認して下さい。

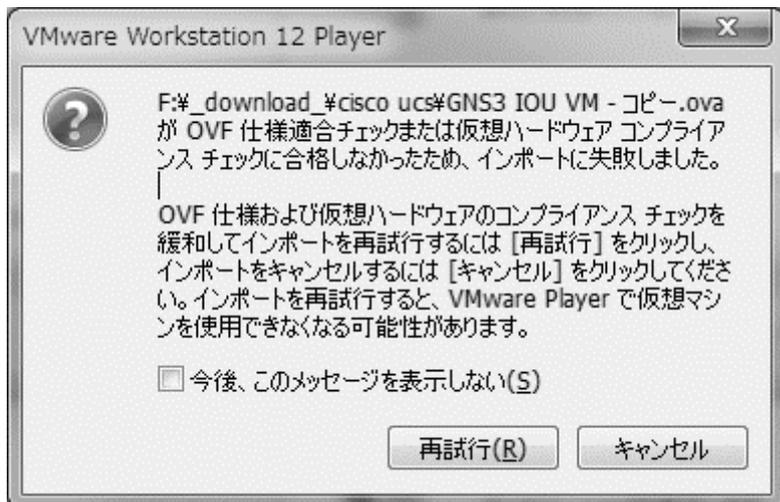
VMWare の場合、

「仮想マシンを開く(O)」

から、対象の.ova ファイルを開き、変換して下さい。

なお、以下のダイアログが出ますが、「再試行」ボタンで続行して下さい。

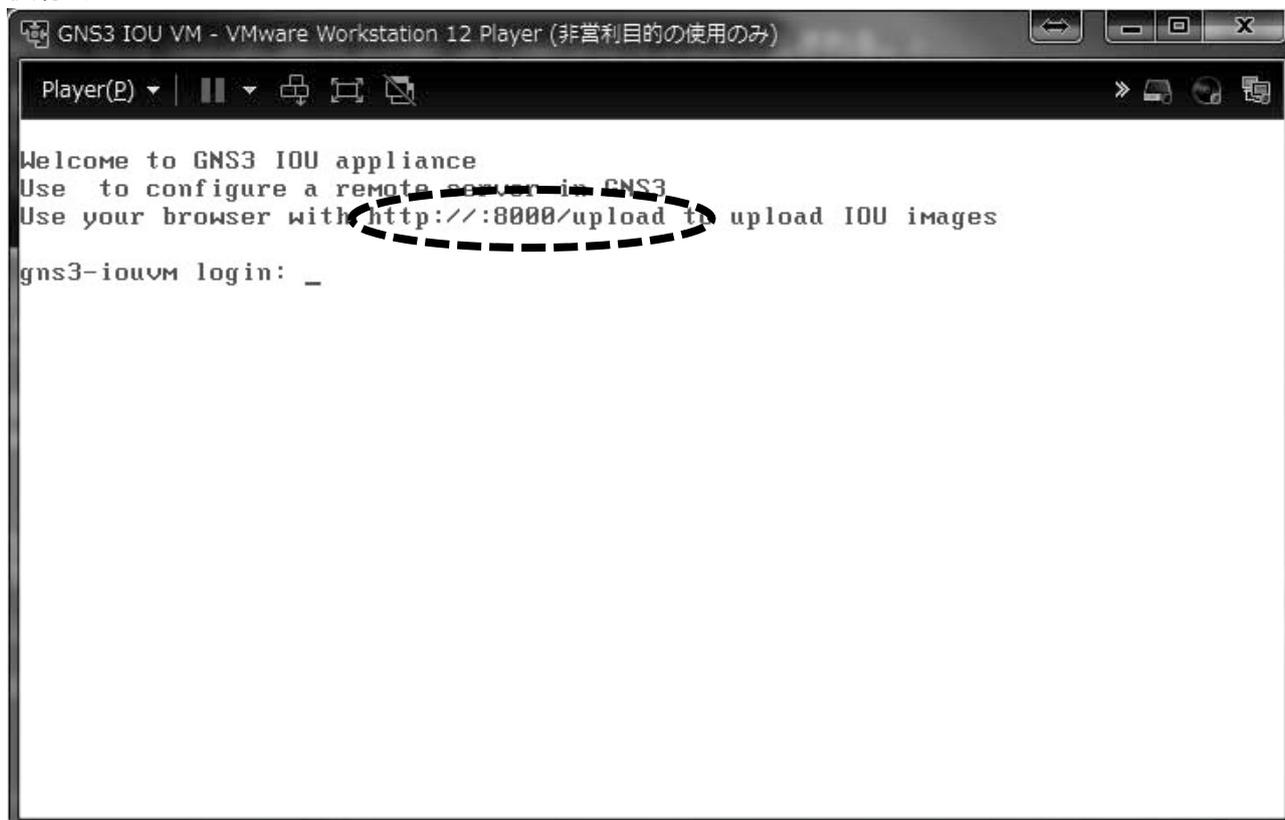
ファイルが壊れるのが不安な方はバックアップを取った上で実施するのを忘れずに。



インポートが完了したら、起動します。

なお、下記起動画面(白黒反転しています)の点線で囲んだ箇所には本来は IP アドレスが表示されるはずなのですが、当方の環境では表示されませんでした。VMWare だからなのかは不明です。

ifconfig で IP アドレスを確認した上で接続を試みたところつながりましたので、一応動作上の問題は無い模様です。



ログインして IP アドレスを確認します。

ユーザー : root パスワード : cisco

でログインできます。

ifconfig を実行して IP アドレスを確認し、

Select & Upload an image for GNS3

File path: 選択されていません
File type:

Powered by GNS3 1.3.0

http://<IP_ADD>:8000/upload

へブラウザで接続します。

かなり拡大してありますが、接続すると上記のような画面になりますので、ここから CISCO IOU のイメージファイル 2 つをアップロードします。

アップロードすると

Files on

/home/gns3/GNS3/images/IOU/i86bi-linux-l2-adventerprisek9-15.1a.bin
のようにアップロード先が表示されるので控えておきます。

Select & Upload an image for GNS3

File path: 選択されていません
File type:

Files on

/home/gns3/GNS3/images/IOU/i86bi-linux-l2-adventerprisek9-15.1a.bin

Powered by GNS3 1.3.0

これで、IOU 側の設定はとりあえず完了です。

・GNS3 のインストール

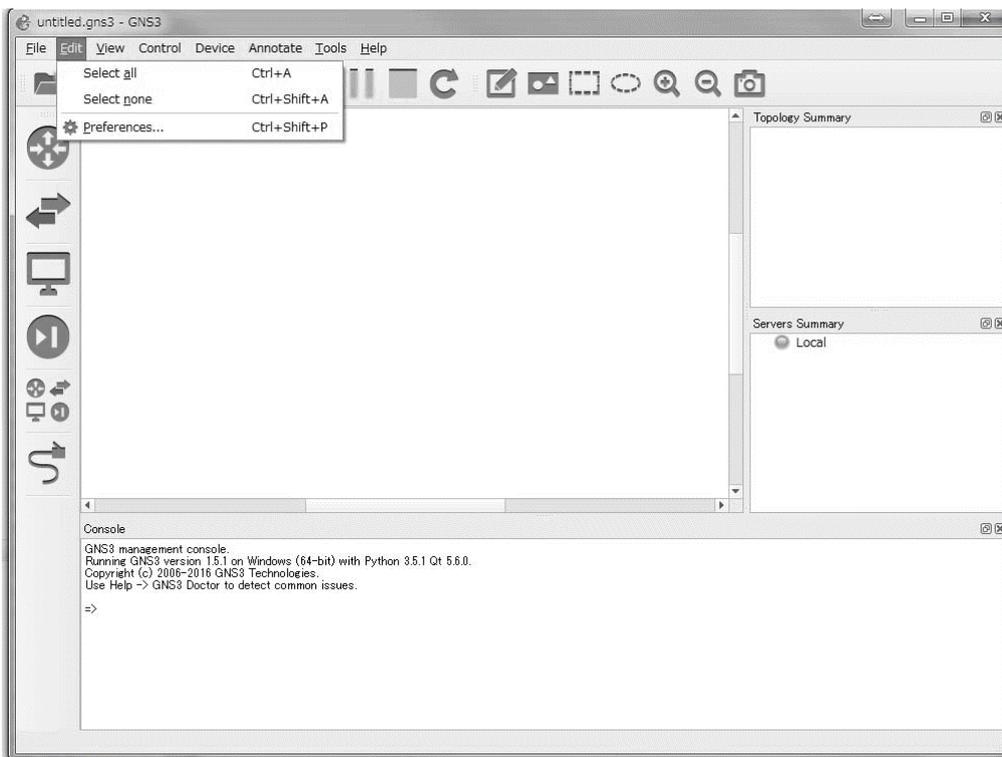
GNS3 をダウンロードしてインストールします。Windwos 環境を念頭においてますのでとくに解説しません。

ただ、ダウンロード可能な IOU VM イメージと同じ Version でないと接続が上手くいかないようです。

今回は Version 合わせとして

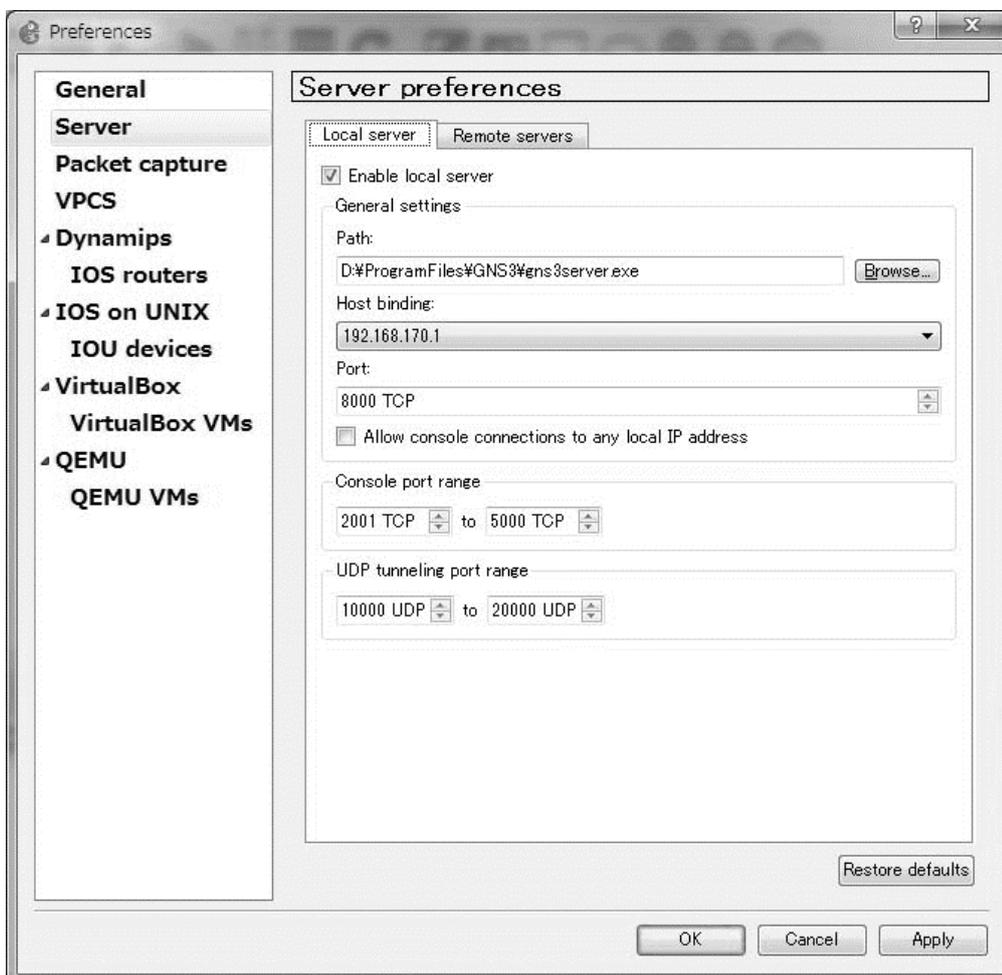
GNS3-1.3.0-all-in-one

を使っています。

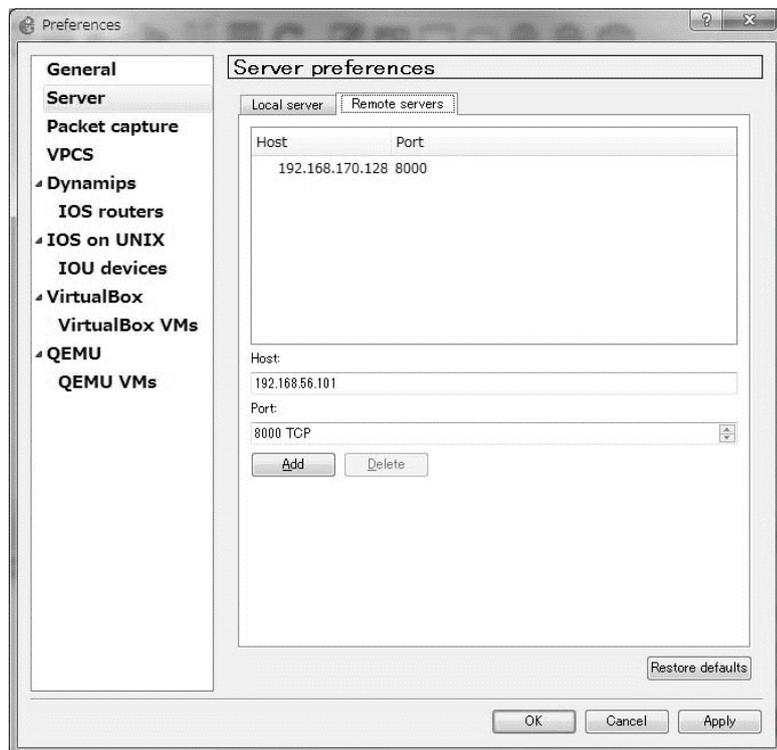


Edit ⇒ Preference を開きます。

Server ⇒ Local Servers タブを開き、「Host binding」に、ローカル側のインターフェースを入力します。当方では 192.168.170.1 でしたが各自の環境で異なります。



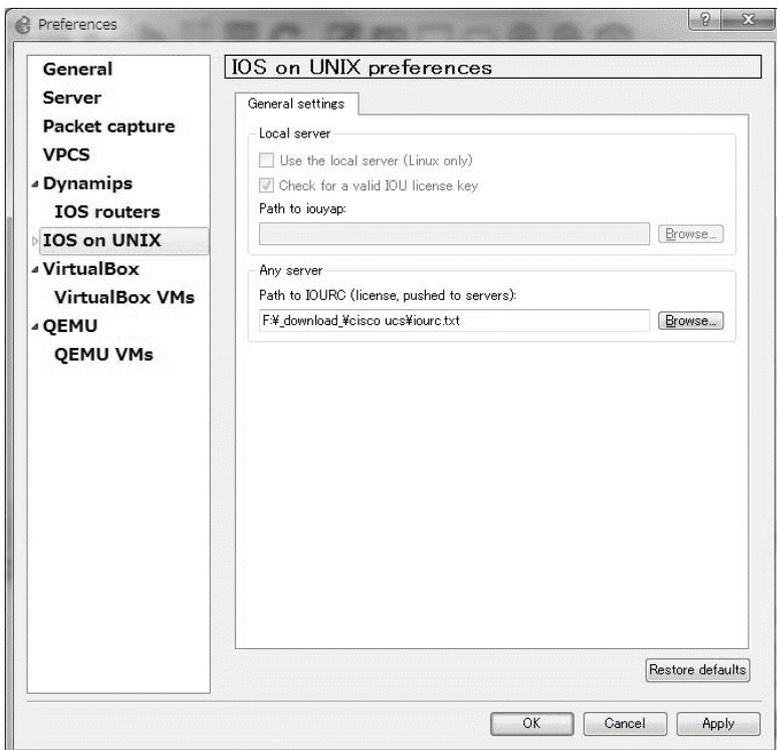
ポート番号はデフォルトで 8000 になっていると思います。



続いて Remote Servers の設定を行います。

Host に VM の IP アドレス、Port はデフォルトの 8000 のままです。

入力したら「Add」を押して一覧に目的のものが表示されることを確認して下さい。

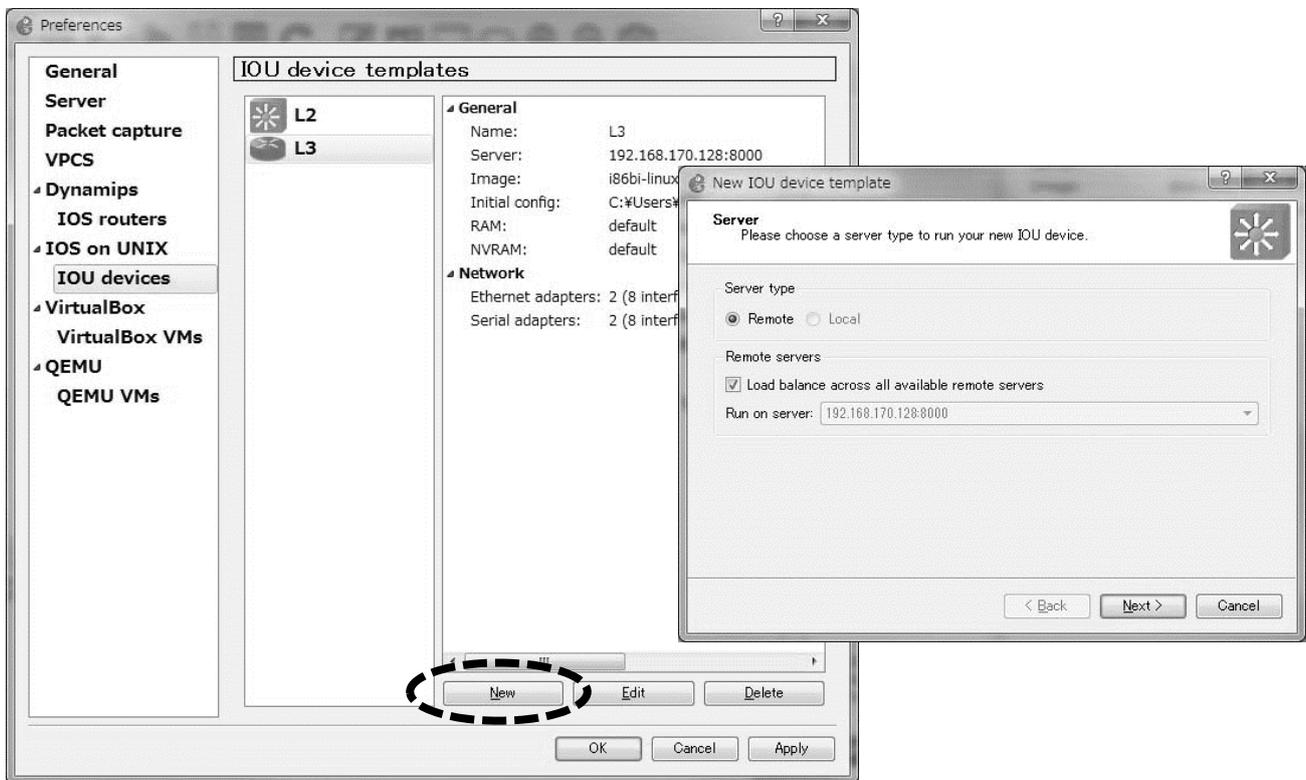


続いてライセンスファイルを登録します。

IOS on UNIX を選択して、Path to IOURC に Browse からライセンスを読み込みます。

続いてイメージパスを設定します。IOU Devices の New ボタンを押すと、Remote Server パスが設定されているので、Next を押します。

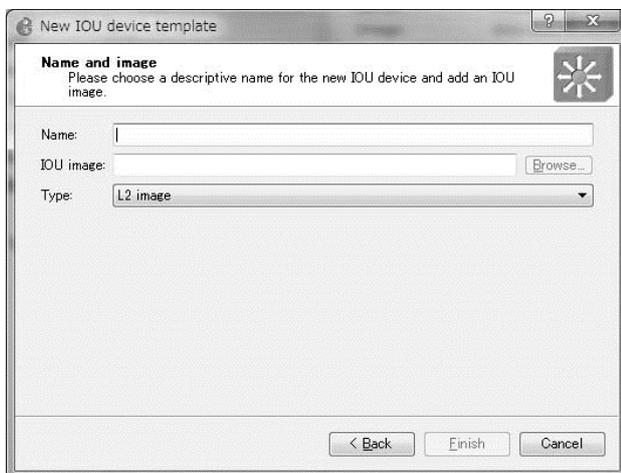
※画像はすでに登録が完了している状態です。



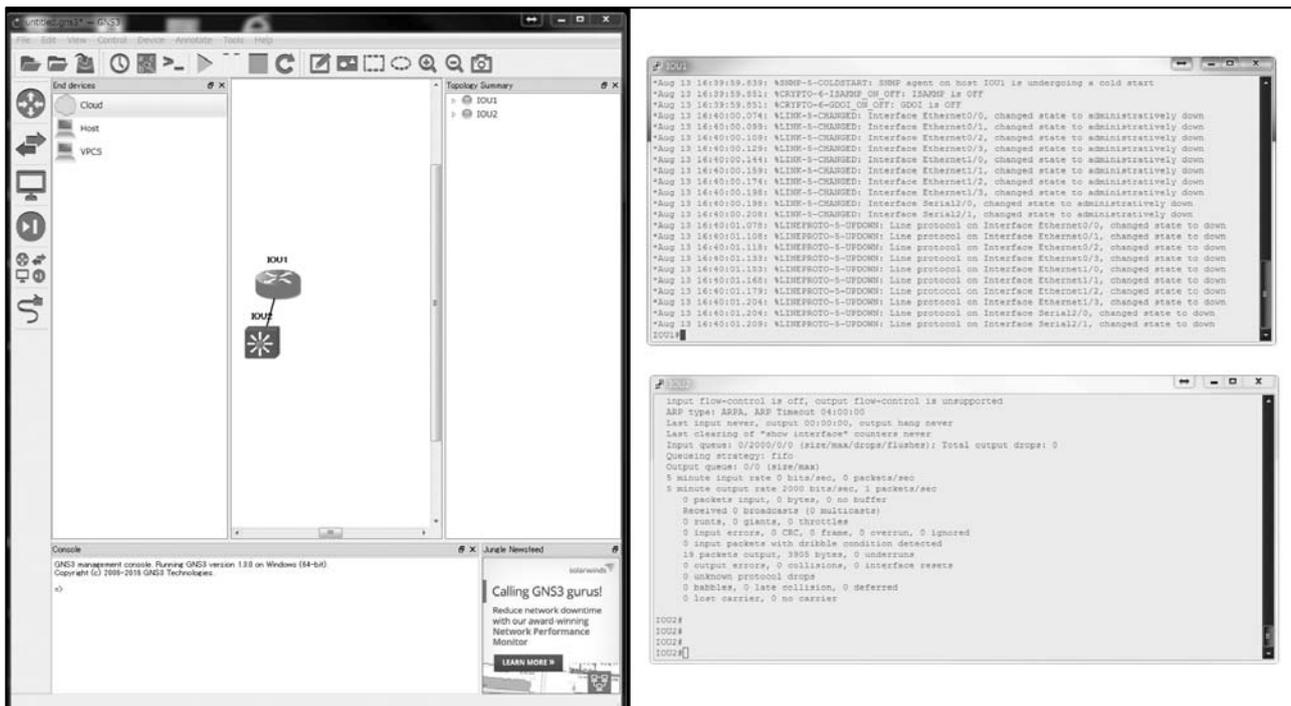
アラートがでますが入力するよう促しているだけです。

Name に任意の名称、IOU image に先ほど IOU VM に登録した際のパスを入力します。

Type はそれぞれイメージに合ったものを選んで下さい(L2/L3)



ここまで設定すれば、GNS3 の GUI から操作が出来ます。



画像が小さくて見づらいますが丸いのがルータ、四角いのがスイッチとして表示されており、間を結線状態にしています。右側のターミナルはそれぞれに接続したところのスクリーンショットです。GNS3の画面左端のアイコンをクリックして表示される機器のアイコンをドラッグ&ドロップで配置できます。ただサーバ周りの配置等は細かいところまではつめていませんが、とりあえずルータ、スイッチ間は結線してみたりログを見たりすることが出来ました。

・おわりに

最新のGNS3は1.5.1ですが、これに対応するIOU VMのイメージがあるのかどうか、といったことが時間が足りずに詰めきれませんでした。IOUもUbuntu、Debian、CentOS、もとのIOUであるところのSolarisといろいろあるようなのですがこちら(そもそもグレーゾーンのものというのがありますが)情報が十分そろわず、記事に起こすほどにはまとめられませんでした。反省です。

Far Northern Other World (Fnow) マイナーOS関連 活動紹介 ・活動コミュニティ

Mixi :

http://mixi.jp/view_community.pl?id=550392

Otaba :

http://otaba.jp/?m=pc&a=page_c_home&target_c_commu_id=2373

Filn :

<http://filn.jp/communities/3726>

GoogleGroups :

<http://groups.google.fi/group/osmanix>

えー、実際のところもろもろあってMixiとGoogleGroupsくらいにしききちんとアナウンスできていません。

毎号の原稿募集、その他アナウンスを行うのが中心ですが、MixiではいくつかマイナーOS系の話題を扱うトピックがあります。

まだコンテンツが豊富とは言い難いですが、当方のマイナーOSネタサイトは以下になります。

Far Northern Other World マイナーOSコンテンツ

<http://fnow.org/>

昨年末に活動をようやく再開したレベルなので、まだまだですがなんとかもう少し時間を確保したいです。

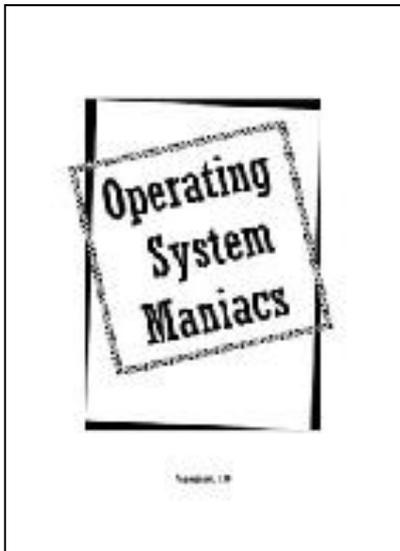
・参加イベント等

コミックマーケット、及びコンピューター関連の即売会が開催されれば、極力参加しています。

マイナーOS本作成以外の活動ももう少しやってみたいとは思っていますが、良い企画がなかなか思いつかず、基本的に本書の頒布と、マイナーOSデータベースの更新が主要な活動となっています。

既刊のご案内

Operating System Maniacs Version 1



2006年夏：Ver1.0 2008年夏：Ver1.1 発行

A5サイズ/オフセット/44P 300円

主なコンテンツ：

【悲運の正統UNIX】UNIXWAREを語る (りろ@涅槃)

SSS-PCを動かしてみる

【悲運の先端OS】OS/2～eComStation (りろ@涅槃)

QNXのインストールと環境設定 (立神梢一)

【マイナーOS】SkyOS Beの遺伝子を受け継ぐ「まぜこぜ」OS (りろ@涅槃)

記念すべき第一号。Webで知り合った、マイナーOSの同志、りろ@涅槃氏にもご寄稿いただき、それなりのボリュームになった。当方はSSS-PCとQNXのインストール、設定記事を執筆。

実際の表紙は、用紙：アトモスブラック インク：薄い茶 となっています。

Operating System Maniacs Version 2



2007年夏：Ver2 発行

A5サイズ/オフセット/44P 300円

主なコンテンツ：

ChorusOSを動かしてみる (立神梢一)

PS2LinuxをDVD-ROM無しで起動する (立神梢一)

マイナーOS求む (立神梢一)

NEC国産OSへの鎮魂曲 (りろ@涅槃)

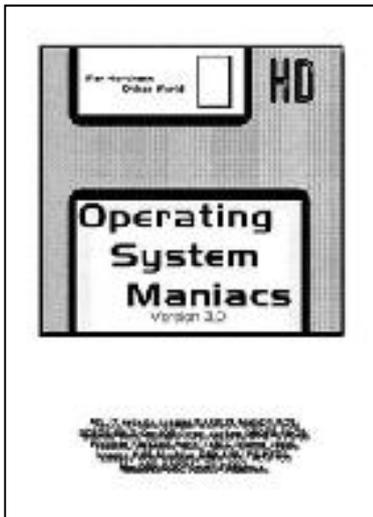
OSとしてのVMWare ESX Server (りろ@涅槃)

長い間の悲願であった、Chorus OSの導入記事と、いまさら感が漂いますが、PS2Linuxの導入記事を掲載しています。

りろ@涅槃氏にも前号に引き続きご寄稿いただきました。

実際の表紙は、用紙：色上質サーモン、インク：赤 となっています。

Operating System Maniacs Version 3



2007年冬：Ver3 発行

A5サイズ/オフセット/44P 300円

主なコンテンツ：

Intel MacminiでマルチOS/マルチブートの試み (りろ@涅槃)

PS2 Linuxをいじってみる ちよとだけ後日談 (立神梢一)

PTOSと過ごした80年代 (りろ@涅槃)

マイナーOS駆け足レビュー (立神梢一)

(AELIX/artasia/asagao/BARBUX/blairOS/BOS/BOZOS/BRIX/ContOS/coron/Darkos/DROPS/FDOS/Freedos/FRITZOS/Hanoi/Idioma/jxOS/knasos/KOS/Moubius/QNX/UUU)

本号から駆け足レビューを開始。とにかく数を紹介したいとの考えからでした。

実際の表紙は、用紙：色上質水、インク：藍 となっています。

Operating System Maniacs Version 4



2008年夏：Ver4 発行

A5サイズ/オフセット/52P 300円

主なコンテンツ：

グランドピアニストにもっと巧く演奏させよう (ZETA Live CD) (SHINTA)

OS9000/80386 DemoFloppyをPCで動かしてみる (立神梢一)

マイナーOSを実用化する (りろ@涅槃)

マイナーOS駆け足レビュー PartII (立神梢一)

BugOS/POSYS/proolix/Roadrunner/runt/scaraOS/SMUGLYOS/Sh

agOS/SORA/spoon/SUNRISEOS/threx/Triangle/ZOFTOS/μnOS

BeOS系アプリケーションの開発等で活躍しているSHINTA氏からご寄稿いただき、

いつもより増ページとなりました。りろ@涅槃氏も皆勤でご寄稿いただき、ありがたい限りです。

実際の表紙は、用紙：アトモスレッド、インク：あずき となっています。

Operating System Maniacs Version 5



2008年冬:Ver5 発行

A5サイズ/オフセット/32P 300円

主なコンテンツ：

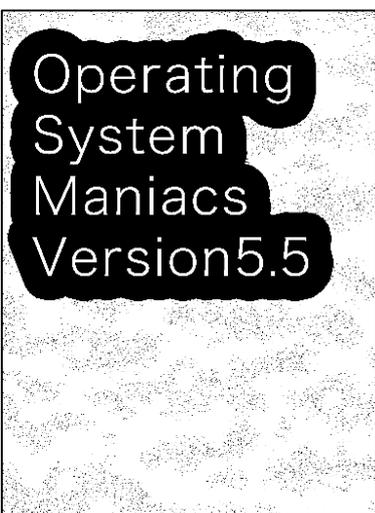
90年代の幻影を求めて AmigaOSの香りを今に伝えるSyllableを試す(りろ@涅槃)

マイナーOS駆け足レビュー PartIII

bgbsystem/MDF/PJOS/RadiOS/SARTORIS/Tesoro OS/tijOS/tizozos/tOfuS/Torison/trion/TYROS/viridis

このころから時間が取れなくなってきて、ちょっとページ減しています。ネタはゲップするほどあるんですが、こなす時間が無いのがつらいですね。。。表紙は色上質特厚若草、インクは黒、本文はさくら/インクは黒となっています。

Operating System Maniacs Version 5.5



2009年12月30日:Ver5.5 発行

A5サイズ/コピー/24P 300円

主なコンテンツ：

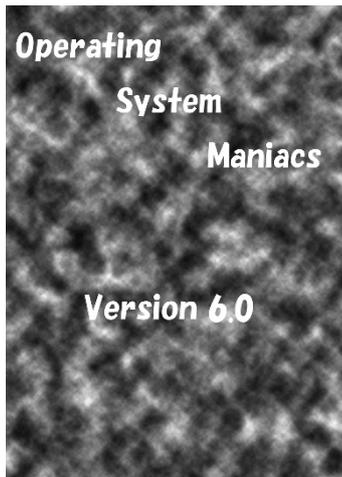
マイナーOS駆け足レビュー Part4

VSTa、MikeOS、DS Linux(2009年夏のコミックマーケット時にペーパーとして配布したものの再掲)

オフセット印刷で発行できずついにコピーになっています。

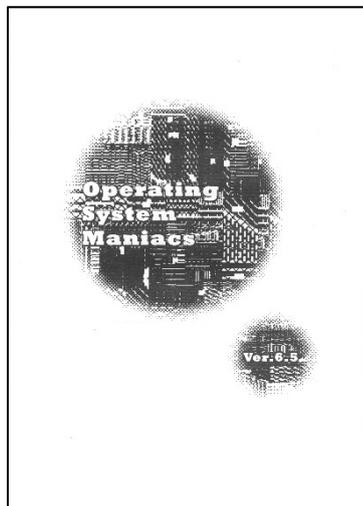
Operating System Maniacs Version9.0

Operating System Maniacs Version 6.0



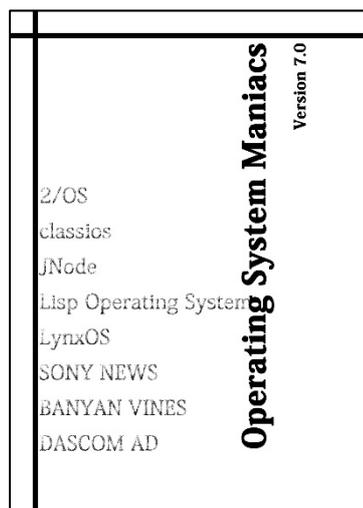
2010年夏:Ver6.0 発行
A5サイズ/コピー-/24P 300円
主なコンテンツ：
マイナーOS駆け足レビュー Part5
AROS、HAIKU、HOUSE

Operating System Maniacs Version 6.5



2011年夏:Ver6.5 発行
A5サイズ/コピー-/32P 200円
主なコンテンツ：
マイナーOS駆け足レビュー Part6
BlueBottle、MINIX3、infortus-os

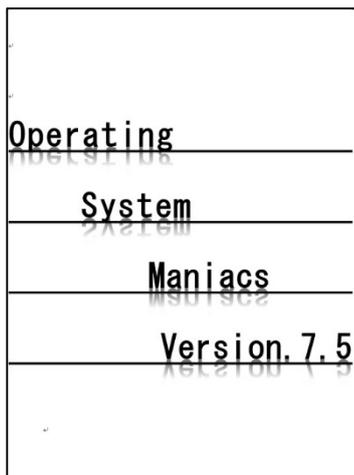
Operating System Maniacs Version 7.0



2014年冬:Ver7.0 発行
A5サイズ/コピー-/36P 300円
主なコンテンツ：
マイナーOS駆け足レビュー Part7
2/OS、classios、JNode、Lisp Operating System、LynxOS
マイナーOSコレクション
SONY NEWS、BANYAN VINES、DASCOM IntraVerse WebStarter
東芝 日本語UX/386 基本部、
EPSON PC-286シリーズ用日本語Disk BASIC

Operating System Maniacs Version9.0

Operating System Maniacs Version 7.5



2015年7月:Ver7.5 発行
A5サイズ/コピー-/24P 300円
主なコンテンツ：
マイナーOS駆け足レビュー Part7.5
Symobi
BANYAN VINES

Operating System Maniacs Version 8.0



2015年8月:Ver8.0 発行
A5サイズ/コピー-/28P 300円
主なコンテンツ：
Historical UNIX
AIX1.3
BANYAN VINES
Coherent
Minor Operating System Collection
AIX6.1、7.1、BSD/OS、Digital UNIX/Tru64 UNIX
IRIX OS/400

DVD版

Version1～8.0 + おまけファイル等で構成されています。

基本的にはPDF版の各Versionと、それぞれで実験したOSのファイルを同梱しております。

既刊は定期的にデジタルデータとしてご提供しております。

イベントによっては気まぐれに再版していく場合もあります。

原稿募集要項

立神梢一

Operating System Maniacsでは、常時、原稿を募集しています。

マイナーOSの導入/環境構築記事

旧UNIX系OSの紹介記事

マイナーOSの紹介記事

異種アーキテクチャへのLinux/BSD系OSなどの導入/環境構築記事

現在/過去のOSに関連する業界動向

特定OSへの偏愛を吐露するコラム

等々。。。

ただしマイナーOSと銘打っている以上、できるだけ、以下のOSについては触れないつもりでいます。

1. Windows 特に2000以降
2. x86用のLinux 特に有名ディストリビューション
3. MacOS 特に9.x及びX

無論、たとえばマイナーなアーキテクチャへの導入関連や、マルチブート環境、エミュレータを使ってのお遊び、などであれば大歓迎です。

あるいは組み込み用OSをいじるケースなどでは、母艦がWindowsということもあるでしょうし、そういうケースは当然ながら特に問題ありません。

実際にはそんなに厳密な決めではないのですが、一応マイナー系OSということ意識するために上記のようにしています。

また、旧Versionで面白いことが出来るケースもあるかなと思い、特に1と3については上記のようにしております。我もと思う方は、是非ご参加ください。

また、原稿ということでもなくとも、情報提供、レビューのリクエスト、素材の提供等も歓迎いたします。特に以下のものについては、入手困難なこともあり、大歓迎いたします。

- ・ハードウェアはない、あるいは導入予定はないが、古いOSのインストールメディアを所持している。
⇒たとえば、EWS-UX、NetWare、SCO UNIX、、、など
- ・逆に、OSは持っていないが古いハードウェアを所持している。
⇒SONY NEWS、EWS4800、BeBOX、、、など

その他、NWアプライアンスの搭載OSについての考察や、ポーティング情報等も歓迎いたします。

また、原稿と言うことでなくとも、記事内容に間違い等がございましたら、ご指摘いただけますと幸いです。

主にWordでのレイアウトまで含めた原稿か、テキスト+画像でこちらでレイアウトする形になるかと思います。(画像挿入位置は個別で指定)

ただし、ページ構成上の都合や、全体のフォント、サイズ合わせ等で若干の変更をさせていただく場合があります。(その場合は変更後の確認をお願いした上で変更します)

詳細は奥付、または活動コミュニティなどへお願いいたします。

編集後記。。。

今回は前回以上にやっちゃってますが、一応理由というか、仕事で使うOpenStackの資格試験(OPCEL)の受験勉強および受験をしたこともあって、あまり時間が取れませんでした。

今回はクラウドやら仮想寄りの話をメインにしていますが、基本的に最新技術を使って意味のある？ことをするというよりも基本迷走気味にやってたいので、KVMでMacOSX動かそうとかか引き続きなるべく変なことをしていきたいです。

前回は宿題的に書いてて全然実行できてませんが引き続き。。。

・KVM上へのMAC OSXインストール

相当に面倒くさそうですが、KVMにインストールした上でOpenStackで扱うとかへんなことができれば面白いなあと。

・Historical Windows

⇒Windowsを1.0から延々インストールしてみる

・スーパーマルチブート実験

⇒とにかく手を変え品を変え1台のPCにどのくらいOSはインストールできるのか？

自分でも取り扱ってますが仮想化全盛の時代にどのくらい意味があんのかというところですが。

・いまさらDOS用GUI

⇒DOSの上で動作するGUIやウインドウマネジャー的なものを特集してみる

もう今さらを通り越していますが。

・マイナーOS環境構築実験

⇒BluebottleやHAIKU、QNX、などメジャーではないOSでサーバを立ててみたりなど。

特に使うこと自体が珍しいBleuBottle、Plan9あたり積極的にいじりたいんですが。

・エミュレータ特集

⇒PDP-11とかOctaneとかあと何があったか。。まあとにかくメジャーなものではなくてもう少しマニアックな？エミュレータを取り上げてみる。

一応今回の方向性に近いといえば近い方向なのですが。

・Google CodeにあるマイナーOS実験

⇒任天堂 esOpearting Systemもいじってませんいろいろ探していじってみたいです。

ただどうしてもLinux、Solaris系が多いんですよ。。まあ当たり前ですが。

もちろん、最近できてませんがマイナー、Oldオペレーティングシステムのインストールとか収集とかも続けていきたいです。

Historical Unixネタもまだ結構な量あるので、すくなくともOpenStepやらXenixやらUnixWareあたりは多少なりいじっておきたいです。

また、方向性が全然違うので苦戦しそうなのですが、OpenVMSもいじっては見たいと思っております。

立神梢一

奥付

Operating System Maniacs Ver8.0

発行：「Fnow」

〒107-0052

東京都港区赤坂1-4-4 富士野ビル501

佐藤誠之(立神梢一)

印刷

秋葉原製作所

2016/8/14発行 (コミックマーケット90)

「Operating System Maniacs」

Far Northern Other World