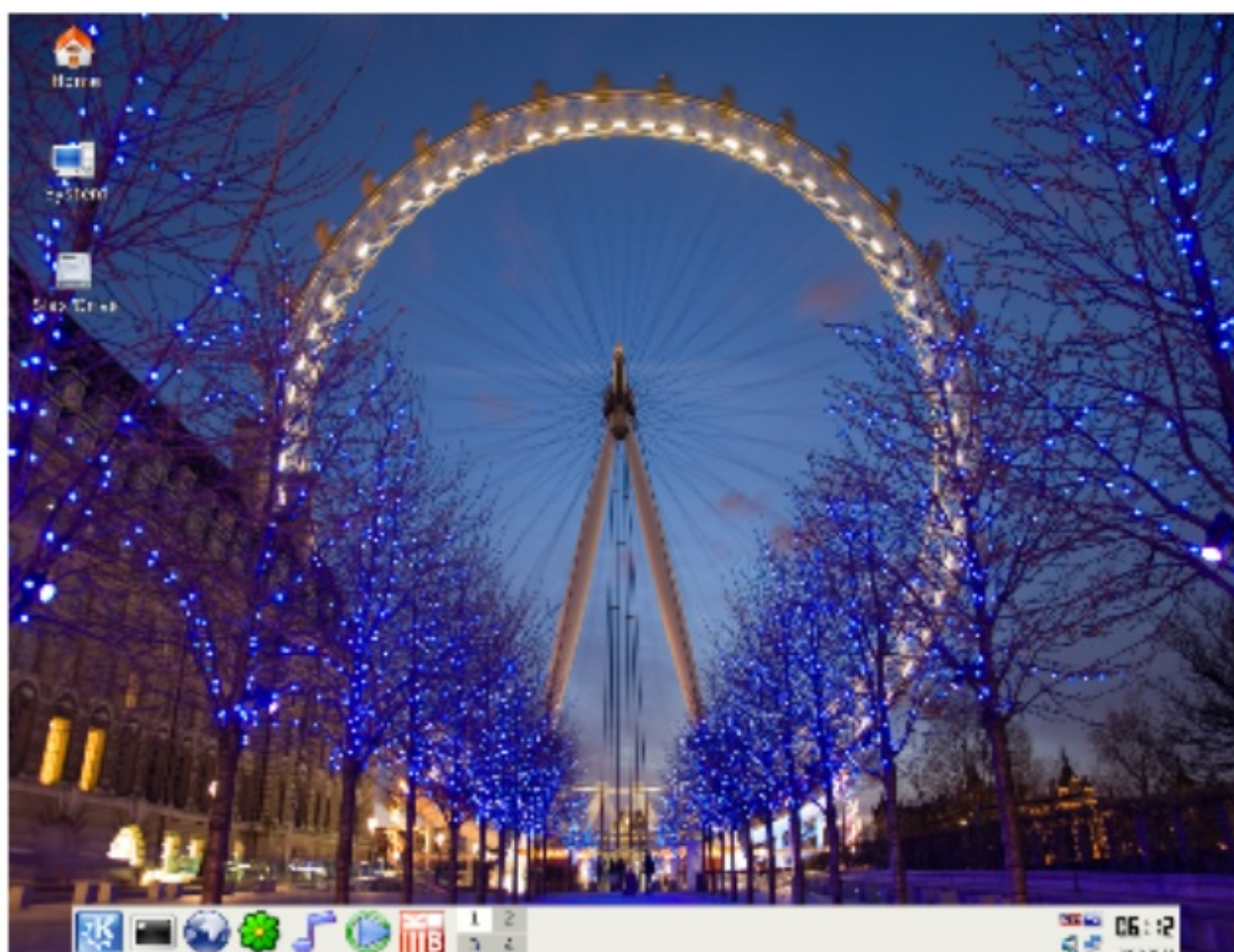


cool-da slax

だってクールだ、slax

スラックス バージョン6 の使い方

slax-linux





1. パソコン本体に、できるだけデータを残さない方法として使う。  
オフィスで、朝、デスクの引き出しからUSBペンドライブを取り出して、起動。  
帰りには、USBペンドライブをデスクにしまって鍵をかけておく。  
デスク上のパソコンには、データはないので、ほぼ安全です。  
あなたが帰ったあと、興味本位で中を見たりする気味の悪い人からも、データを守れます。  
クラウドのようなネット経由は、実際、全社でやり始めたら、  
朝夕は、ノロノロになってしまうのでしょうか。  
FC2では、明け方はアップロードですごく混雑して困る時があると云います。  
ネットを経由しないでUSBペンドライブを利用する方法は、  
ペンドライブの管理をしっかりすれば、良い方法ではないでしょうか。  
USBペンドライブを、ちょっと目を離れたすきに引っっこ抜いて、  
持って行ってしまふようなケースは、どうすればいいのでしょうか。  
人事の選考基準は、人間性が最重要という傾向を、さらに増してもらふ以外に、  
対処法はないように思えます。
2. 専門システムとして、特定の仕事に特化したスラックスを簡単に作れる。  
アップルを使って行われてきた、専門システム的な使い方、例えば音楽用であれば、  
Pro Toolsを中心にして、関係のないものは、出来る限りはずしてしまうことで、  
性能の維持、動作の安定をはかるといふような使い方を、スラックスで、試していきます。
3. スラックスなら管理も簡単。  
例えば、私たちは、デスクトップにはなるべくものを置かないように努めて来ました。  
トップが重くなると、描画性能に影響が出るからです。  
スラックスでは、USBペンドライブのslax/changesフォルダーにデータはコピーされます。  
slax/changes/Desktopの中身が空に近いと、安心というわけです。  
このように、ストレートで管理もやり易い工夫がされているように思います。  
ハードディスクにインストールしたアプリケーションをきれいに削除するのは、  
一般的に相当な知識を必要とします。  
スラックスでは、インストールは、lzmファイルをslax/modulesフォルダーに置きます。  
アンインストールは該当するlzmファイルをゴミ箱します。

スラックス バージョン**6.1.2**を使う。

---

実際、スラックスでカスタムアイソCDを作る際の作業は、相当に単純化されています。

ここで、大事なご注意。

ここで取り上げているスラックスは、バージョン6.1.2です。

サイト上に存在する記事には、どのバージョンを対象にしているのか、分からない場合があります。

例えば、\*.moファイルに関する記述のあるもの、これは、廃止されていますので、お読みになるだけにしてください。

configsaveコマンドは、もう見つかりませんので、廃止されていると思います。

USBペンドライブをお使いになる場合、changesファイルは、USBペンドライブにちゃんと設定されますので、特に操作をするケースはありません。

USBのロットがついてないパソコンでもない限り、HDDにインストールするのは、ポケットに入ってしまう気楽なOSというイメージが薄れてしまうので、

ここでは取り上げませんが、古いパソコンを再利用するような時、

スラックスは役に立つでしょう。

日常の作業が出来る程度の状態、プラス日本語で、サイズはおよそ250MB。

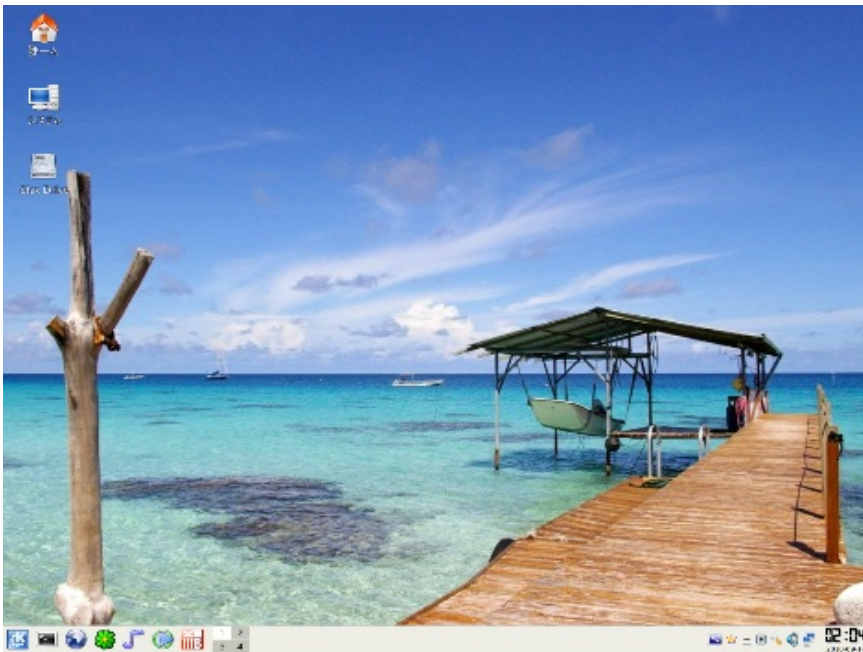
アプリはすべてモジュール化されているので、

アップデートに追われるということはありません。

ホームで作られた前回までのデータは、changes/root フォルダに貯蔵されています。

ホームのデータをしっかり管理していれば、小さなハードディスクでも、

きっと何とかなるでしょう。



## 作業の段取り。

---

1. スラックスのサイトには、コアモデルとして、200メガのイソファイル、ターファイルが用意されています。
2. これに組み込んで使うモジュール、これは、しっぽに.lzmがついているファイルで、サイトに3000以上用意されています。これをお好みでダウンロードして使います。
3. 下記のサイトには、日本語モジュールを組み込んだイソファイルが存在します。  
<http://sourceforge.jp/projects/slax/downloads/44899/slax-ja-6.1.2-2.iso/>  
にて、このファイルをダウンロードして、CDに焼きます。
4. USBペンドライブをフォーマットしたら、CDを立ち上げ、CDの中身、ブートとスラックスフォルダーをUSBペンにコピー。
5. コンソールからUSBペンのブートフォルダーに行き、bootinst.shを実行。
6. modulesなどを空にしたブート専用のCDも作っておきます。  
これがあると、USBからのブートに対応していない、少し旧式のpc、efi対応のインテルマックなどからも、スラックスUSBを利用できます。



これより具体的な作業に入ります。必要な部品は

---

以下のような部品を用意します。

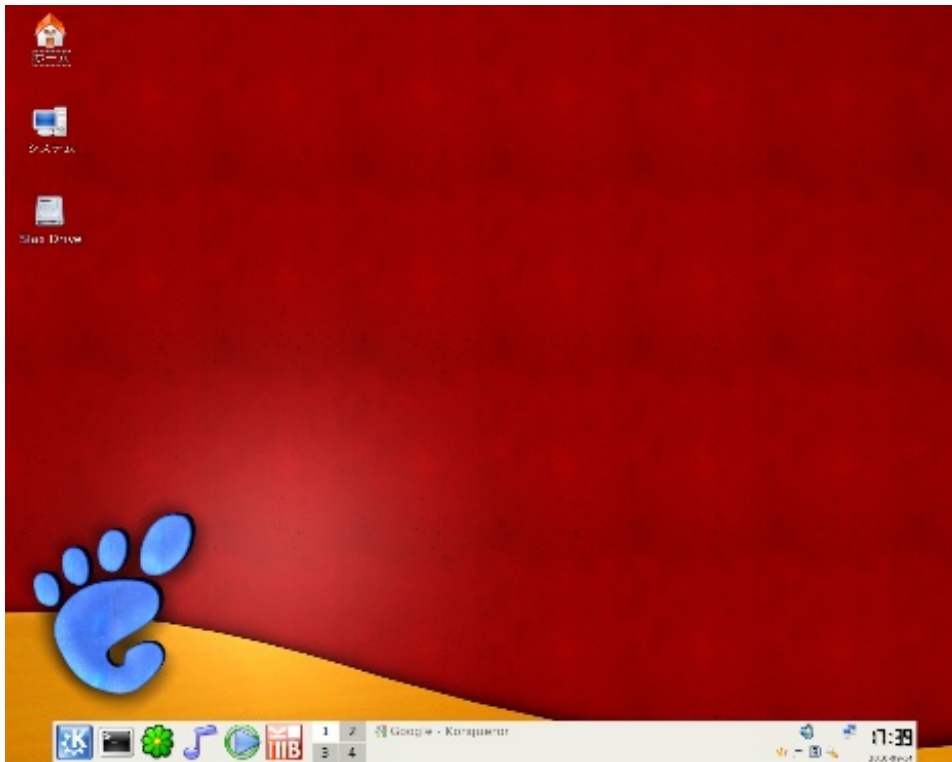
1. 未使用のCD、3～4枚。
2. USBペンドライブメモリ(4GB)。

ホーム(/root)での作業などは、すべてUSBメモリに保存されます。

となると4ギガは、少ないと思われるかも知れません。

USBペンは、複数持っているとう便利です。

今回はコンビニにて、4ギガのメモリを買いました。



## ステージ1 スラックスをCDに焼く。

---



スラックスの日本語コアモデル、slax-ja-6.1.2-2.isoをそのままCDに焼く。

ここでは、

1. アイソファイルをダウンロードして、CDに焼くという行程に問題がないか確認する。
2. CDから起動、終了が問題なく行われるかの確認。
3. スラックスコアモデルの動き、使い方をチェックします。

アイソファイルをCDに焼く方法が分からない場合:

インテルマックでは、ディスクユーティリティを使います。

1. ホーム以下に、iso\_filesのようなフォルダーを用意、そこにアイソファイル、ここでは、slax-ja-6.1.2-2.isoを、入れておきます。
2. Applications -> Utilities とナビゲート、Disk Utility.app を、ダブルクリック。
3. Disk Utilityのメニューから、File -> Open Disk Image をクリック。  
ナビゲートして、iso\_files/slax-ja-6.1.2-2.isoををクリック。  
Openボタンが青色になるので、プッシュ。  
ディスクユーティリティの左側に、slax-ja-6.1.2-2.iso の名前が反映されるので、これをクリック。  
ツールバーのバーンアイコンがクッキリするので、これをクリック、  
トレイが突然オープンするので、未使用のCDをのせて、トレイを軽く押す。  
出来上がれば、トレイがひらく。
4. 油性のサインペンなどで、書き込みのない外周の部分に、タイトルなど、書いておく。
5. SLAXの中身が開いているので、ホームから、三角マークをプッシュして終了する。

Windows 7では、ISOイメージ・ファイルの書き込み機能は、システムの標準機能となっています。

1. Windowsエクスプローラなどを開く。
2. CD/DVD-Rに書き込みたいISO/IMGイメージ・ファイルを右クリックすると、メニューの先頭に、ディスクイメージの書き込み、と出るので、クリック。
3. 案内に沿って操作します。

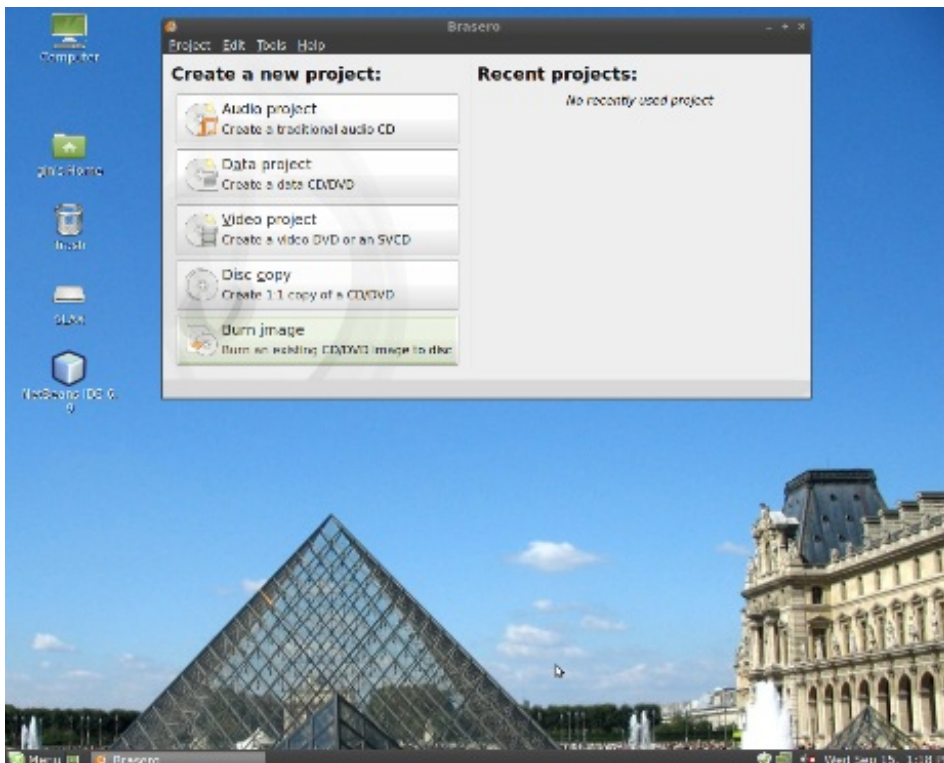
リナックスでは、ブラセーロでCDを焼く。

---

# use-brasero

リナックス(ubuntu, LinuxMintなど)では、Brasero disk burnerが使いやすいと思います。

1. Downloadsのようなフォルダーに、slax-ja-6.1.2-2.isoを、入れておきます。
2. Menu-> all -> Brasero にて、ブラセーロを立ち上げ、Burn imageをセレクト。
3. "Click here to select a disk image"をクリック。
4. ナビゲートして、slax-6.1.2.isoをセレクト。
5. ターミナルを立ち上げ、eject リターンで、トレイがオープンします。
6. 新しいCDを入れると、ダイアログが出る場合、ポップダウンから、do nothingをセレクト。
7. Burnボタンを押します。
8. アイソCDが出来上がれば、トレイが開きます。



# slax-live

## CDからの起動

出来上がったアイソCDをトレイに置いて、再起動します。

その時、インテルマックでは、Cキーを押せばなしにします。

ゴソゴソとCDの準備が進行している雰囲気を感じられれば、離します。

この操作は、わずかですが、慣れを必要とし、それで、ストレスを感じるかもしれません。

そんな時は、rEFItというプログラムを使います。

1. <http://refit.sourceforge.net/> にて、rEFIt 0.14 (6.5M Mac disk image)をダウンロード。

2. ダウンロードしたパッケージをダブルクリック。

3. インストラクションに従ってインストールする。

次回から、起動時に、rEFItが登場するので、

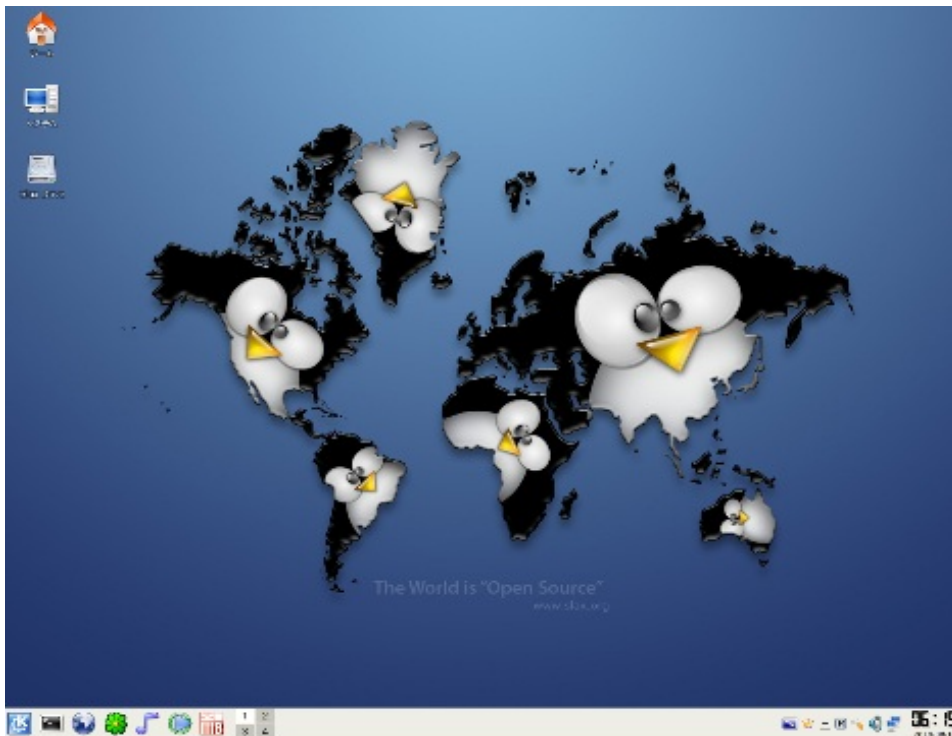
アローキーにて、アイコンを移動させて、

足にCDをつけた、太ったペンギンを選んで、リタン。

スラックスCDからブートされます。

4. 何もしない場合は、OS Xが選択され、ブートされます。

リナックス、ウインドウズでは、アイソCDをトレイに置いて、再起動すれば、アイソCDからスラックスが上がってきます。







プラットホームに関係なく、安定したフォーマット、パーティションの作業が行える方法、それは、GPartedのライブCDを利用する方法です。

操作の方法が、プラットホームに関係なく、全く同じというのも魅力です。

<http://sourceforge.net/projects/gparted/files/gparted-live-stable/>

にて、gparted-0.6.2-2.isoを入手、CDに焼き、立ち上げます。

このとき、USBペンドライブをスロットに差しておきます。

1. 最初に画面がでますが、リターン、もしくは、なにもしなければ、作業が始まります。

2. Don't touch keymapと赤い字がでましたら、リターン。

3. 言語の選択、15といれ、リターン。

4. [0]となっているので、GUIモードとなりますので、単純にリターン。

これにて、出てくるウィンドウは、HDDの状況ですので、

GParted -> デバイスを選択 -> /dev/sdcをセレクト。

フィールドの/dev/sdc1をセレクト。

パーティション -> フォーマット -> fat32 をセレクト。

タスクバーのApplyを、選択します。

続いて、パーティション -> フラグを編集 をセレクト。

bootにチェックマークをつけます。

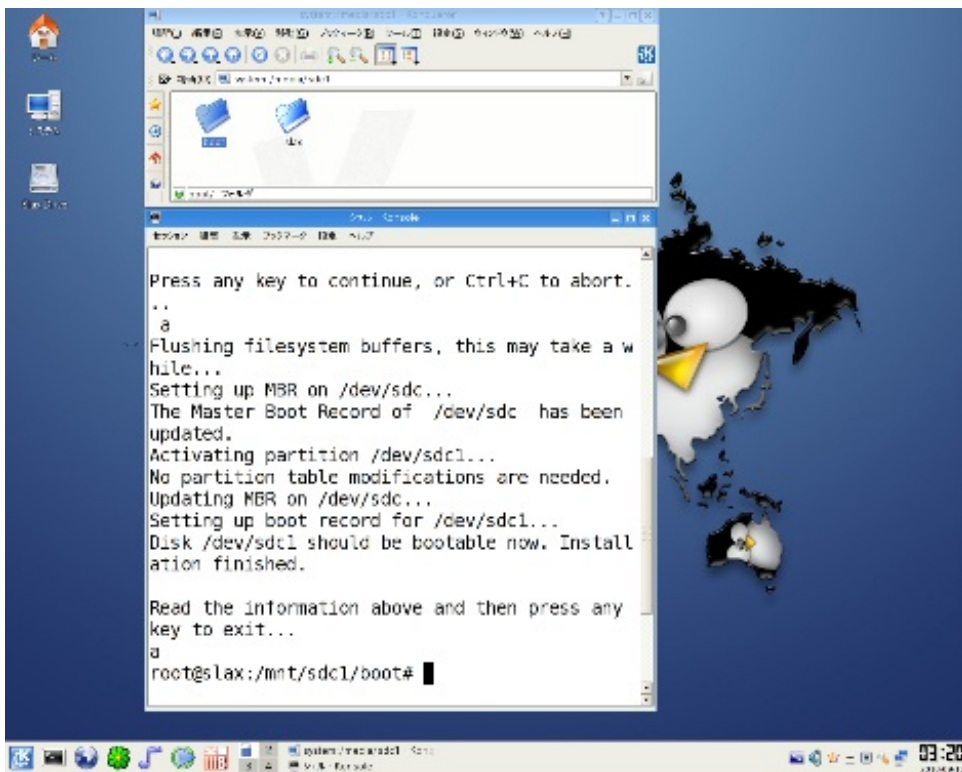
これにて作業は終了です。

左上のExitをダブルクリック、ダイアログのRebootをセレクト。

OKをクリック、もとのOSに戻ります。

## ステージ4 スラックスをUSBペンにインストール。

1. アイソcdをトレイに入れ、再起動、スラックスを立ち上げます。
2. システム->記憶メディア->SLAXをダブルクリック。  
SLAXはオレンジ色のcdのアイコンです。  
すると、中に、boot、slaxと、2つのフォルダーが見える。
3. システム->記憶メディア->SLAX\_JAを開く。  
すると、ファイルは空で、場所は、system:/media/sdc1 と表示されている。
4. CDにあるboot と、slaxフォルダーを、sdc1にドラッグ、コピーします。
5. Konsoleを立ち上げ、今作ったsdc1/bootフォルダーを、  
コンソールにドラッグ、ダイアログからcdを選びます。  
bootに移動したので、ls とタイプすると、  
bootinst.shというシェルスクリプトがあるのが見えます。  
そこで、このスクリプトを実行します。  
# ./bootinst.sh  
途中、入力を促される場面では、aをタイプして、リターンします。  
これで、USBペンドライブへのインストールは終了です。  
# /sbin/rebootにて、元に戻ります。



# myusb-boot-cd

インテルマック、ちょっと古い世代のPCなど、USBからのブートをサポートしていない場合があります。

そこでブート専用のCDを作って、ブートはこれで行い、立ち上がったらUSBペンにつなげます。これがあると、USBのスロット、CDのトレイがあるパソコンであれば、スラックスUSBを使える確率が高まります。

方法その1。

1. <http://www.mediafire.com/file/g50hznzeztm/slax-boot-usb-from-cd-6.1.2.iso>

にて、slax-boot-usb-from-cd-6.1.2.iso をダウンロード。

2. このアイソファイルをCDに焼きます。ファイルサイズはおよそ9メガバイト。

方法その2。

1. 作業用のフォルダー、myusb\_bootを作る。

2. ステージ4で、スラックスをインストールしたUSBペンドライブ、名前はSLAX\_JAとしてありますが、これをダブルクリック、中にあるbootとslaxをmyusb\_bootにコピーします。

3. slax/baseを開き、baseの中にあるモジュールをすべてゴミ箱に捨て、空にします。

4. livecd.sgnをゴミ箱に捨てます。

livecd.sgnは、slaxフォルダーの中、真ん中あたりにあります。

5. myusb\_bootフォルダーをUSBペンにコピーします。

USBペンには、すでにbootとslaxがあるので、それに並んで、myusb\_bootが置かれることになります。

これよりslaxからmyusb\_boot.isoを作成する作業に入ります。

1. スラックスCDを立ち上げます。

2. Homeをダブルクリックして開きます。

3. システム/記憶メディア/SLAX/myusb\_bootを開いて、myusb\_bootフォルダーをHomeにコピー。

4. Konsoleを立ち上げ、`# cd /root/myusb_boot/slax`に移動。

5. slaxの中に、make\_iso.shがあるので、

これを使ってアイソファイルを/tmpに、myusb\_boot.isoという名前で作成。

```
# ./make_iso.sh /tmp/myusb_boot.iso
```

6. Homeから左上にある、上向きのアローをクリック、次のウィンドウで、tmpをクリック。

そこに、myusb\_boot.isoが出来ているので、これを、USBペンドライブにコピーします。

7. これで作業は終了です。`# /sbin/reboot`にて、再起動します。

8. USBペン(名前はSLAX\_JA)を開き、myusb\_boot.isoをCDに焼きます。

## ステージ6 myusb\_boot CD テスト走行。

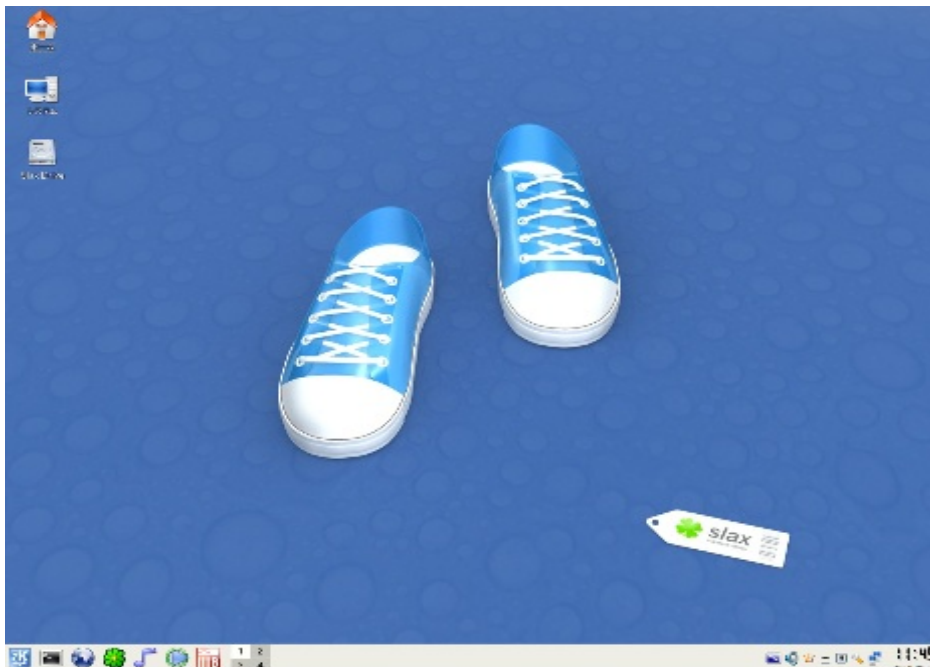
---

1. スラックスUSBペンをスロットに差し、
2. myusb\_bootのCDをトレイに置き、立ち上げます。  
わたしたちが、特に何かするということはありません。  
しばらくすれば、スラックスが立ち上がっています。  
myusb\_bootのCDには、モジュールは入っていないので、  
USBペンを使って立ち上がってきたわけです。

出来上がったスラックスUSBペンは、使ってみたいモジュールを  
USBペンドライブのmodulesフォルダーに置いておくことで、  
機能はどんどん拡大していきます。

それでは、手始めに壁紙の変更を試みましょう。

1. <http://www.slax.org/modules.php>に出かける。
2. サーチにwallpaperと入れ、リターン。
3. SLAX Wallpaper というのが出てくるので、  
read moreをクリックして、依存ファイルがないか確認する。  
Slax Core 6.1.2はすでにインストールされているので、  
SLAX Wallpaper本体のみをダウンロードします。  
ファイルサイズは120KBです。
5. このslax-wallpaper.lzmを、システム/記憶media/SLAX\_JA/slax/modulesの中にコピーします。  
これで次回から壁紙は、このように変更されます。



## ステージ7 壁紙の保管場所。

---

OSが立ち上がってきた時の、見た感じはとても重要です。

複数の壁紙を用意しておけば、気分に応じて変化が楽しめます。

特に変更する気持ちがなければ、このステージはスキップして下さい。

1. <http://art.gnome.org/backgrounds/>にて、背景をさがします。

このサイトに用意された壁紙は、わりとセンスが良いので、  
きっと何か見つかることでしょう。

ノームのマークなどついていますが、気にせず取ってきます。

30枚で約8メガになりました。

2. フォルダーを、以下の順に作ります。

`usr/share/wallpapers`

取ってきたJPEG、PNG、GIFファイルなどを  
このwallpapersフォルダーに収納しておきます。

3. ひとしきり取り込みましたら、`usr/share/wallpapers`というフォルダーを  
`SLAX_JA/slax/rootcopy`の中に置きます。

フォルダーは、`slax/rootcopy/usr/share/wallpapers`のように並びます。

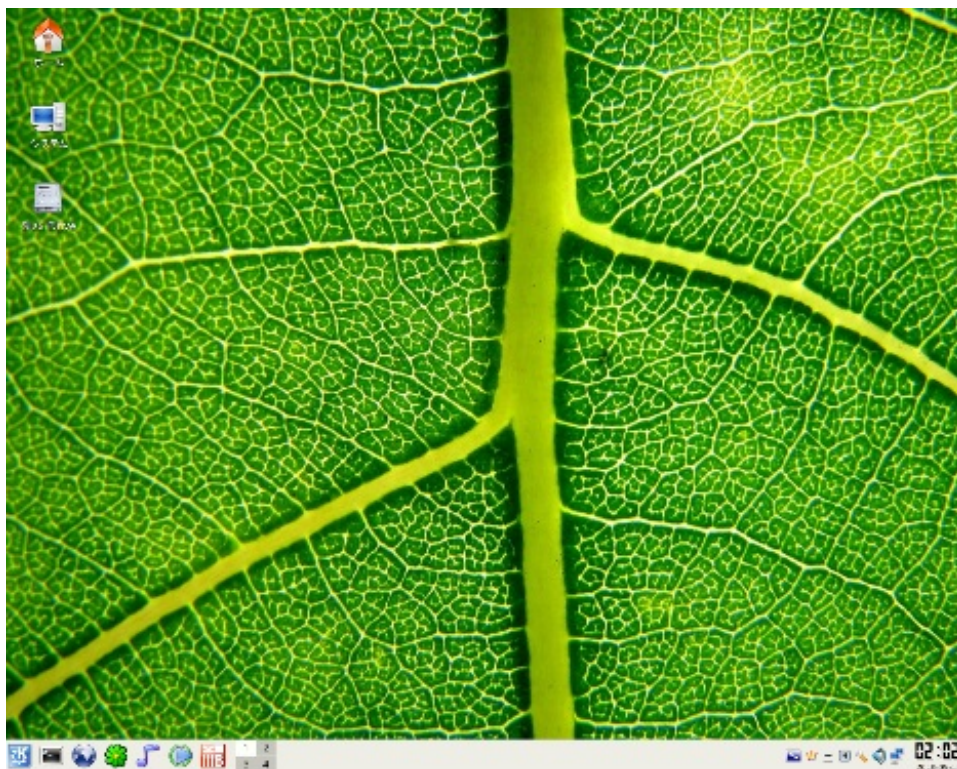
これで、次回から壁紙の変更ができるようになります。



## ステージ8 壁紙を変更してみる。

---

1. ステージ7を参考に、SLAX\_JA/slax/rootcopy/usr/share/wallpapers に、使いたい壁紙を入れておく。
2. スラックスUSBを立ち上げる。
3. Kメニュー -> コントロールセンター -> 外観&テーマ -> 背景 に行く。
4. 画像(P) にポップダウンメニューがある、これで見に行く。
5. これだと思ったら適用ボタンで、壁紙は変更される。



## ステージ9 スクリーンセイバーをセットする。

1. デフォルトでは、スクリーンセイバーはセットされていません。

そこで、スクリーンセイバーのモジュールを組み込みます。

2. <http://www.slax.org/modules.php> に出かけ、サーチに、screen saver と入れてみる。

3. Kscreensaver 3.5.9-3 が、出てきました。

read moreにて、依存ファイルを調べます。3つともベースファイルで、

これらはインストール済みなので、本体のみ引いてくれば良いと分かります。

4. ダウンロードボタンを押して、kscreensaver-3.5.9-3.lzm をダウンロード。

5. システム 記憶メディア SLAX\_JA/slax/modules に、kscreensaver-3.5.9-3.lzmをコピーします。

これで、次回から、スクリーンセイバーの設定が可能となります。

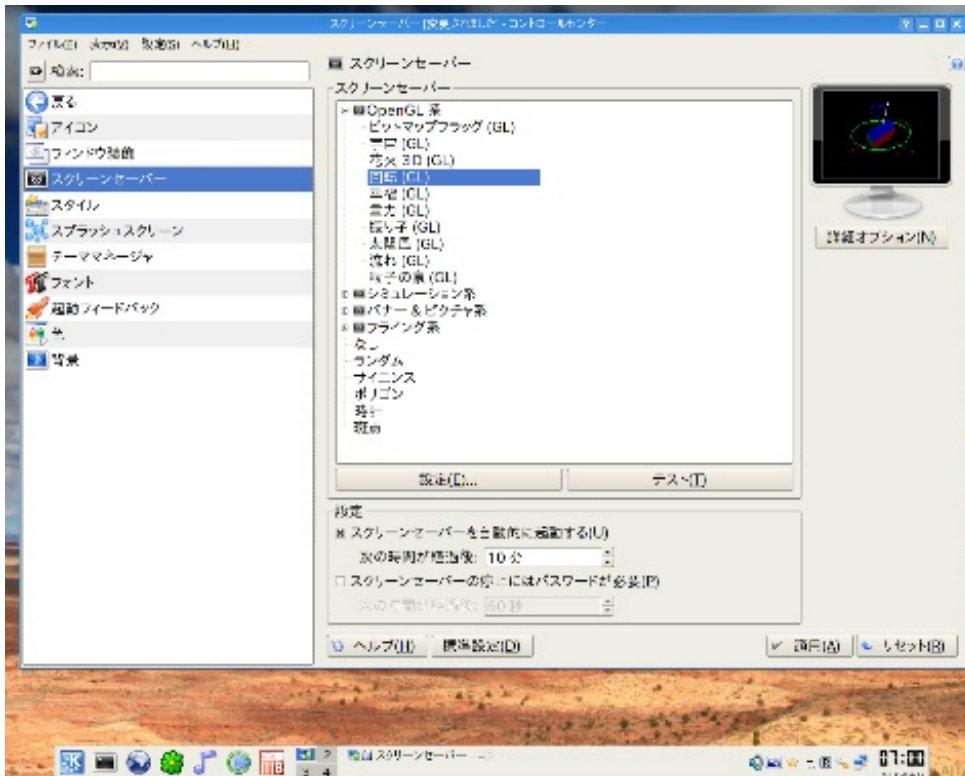
設定は、

1. Kメニュー -> コントロールセンター -> 外観&テーマ -> スクリーンセイバー に行く。

2. リストの項目をセレクトするとプレビューが動きだす。

3. 確定したら、設定を調整して、適用をクリック。

4. 画面を閉じます。





スラックスUSBペンでは、アップルで長年使われてきた、いわゆるプロシューマー的な使い方、それは、特定の仕事に特化するため、それ以外のプログラムは出来る限り抜いてしまうという方法をとることができます。

これにより、ハイスピード、軽量化、性能の劣化を防ぐなど、よいこと、期待できます。

### カスタム スラックス その1

2010年時点で、C言語を始めるキット。

Rubyからプログラムを始めて、Cも練習しようと思った人は、賢い考えの持ち主ですが、ライブラリが少ないのに驚いて、練習はするけど、実際に使うところまではいかないといえます。

C言語固有のライブラリーではないため、

教科書では、紹介されないとはいえ、Cには、立派なライブラリが存在します。

それがglibです。

ジーリブは、それ自体でひとつの完結した世界を演出することができるので、たいていの作業は、

`#include <glib.h>` これでいけます。

専門的に考えると、これは、Cのダイアレクト(dialect)、Cの方言なのかもしれません。

ライブラリー付きで練習し、力がついてきたらライブラリーなしでやってみる。

これが、2010年以降のCの始め方、クールです。







1. モジュールをさがしに行きます。  
http://www.slax.org/modules.php に出かけます。
2. 軽量IDE、ジニーを導入しておきます。  
サーチに、geanyと入力。  
Geany 0.18 をクリック、ダウンロード。
3. ジニーはvteがあると、アプリの下部に端末を表示します。  
サーチにvteと入れます。  
vte-0.10.25をクリック、ダウンロード。
4. コンソールからテキストを操作するエディターとして、viが組み込まれています。  
ここでは、ナノも入れておきます。  
viのやり方を忘れてしまっても、ナノがあればだいじょうぶ。  
サーチにnanoと入力、nano-2.2.4-1.lzmをダウンロード。

今回、加えるパッケージは、全部で3本となりました。

1. geany-0.18-i686-1cca.lzm
2. nano-2.2.4-1.lzm
3. vte-0.10.25-i386-1.lzm

この3本をUSBペンドライブのslax/modulesにコピーします。

これでモジュール追加作業は、終了です。

次回から、使えるようになります。

## ステージ12 ジニーの設定とテスト。



ジニーは、軽快なイメージのIDEで、窮屈な感じがしないので、使い易いと評判です。

ユーザーは、手作業でメイクファイルを用意する必要がなく、ビルドボタンを押せば、Cなら、実行可能なファイルが出来上がります。

先に、vte部分の設定を行います。

1. Kメニュー -> 開発 -> IDS(Geany)にて、立ち上げます。

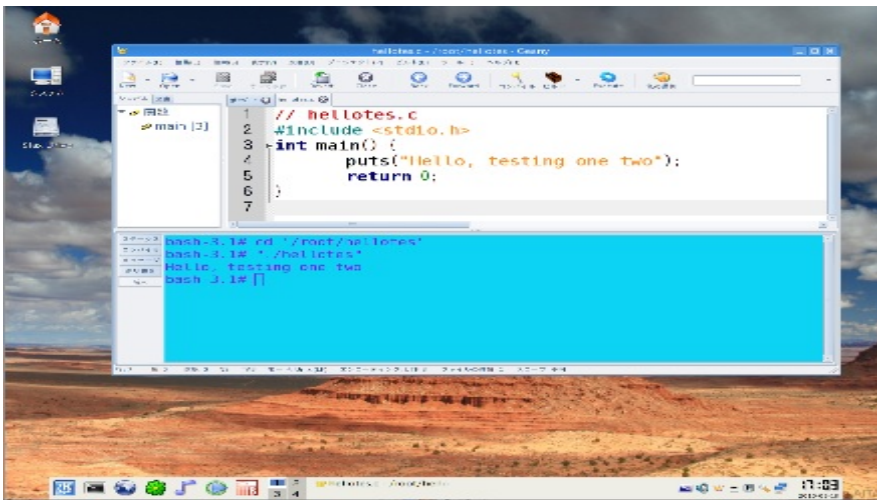
2. 編集 -> 設定 に進み、一番最後の項目、端末に行きます。

端末のフォントを、DejaVu Sans Mono Book 14 のように変更。

vteでプログラムを実行、にチェックマーク。

起動スクリプトを使用しない、にチェックマーク。

以上で、vte部分の設定は終了です。



ジニーを使ってみましょう。

使って見ましょう。

1. ホームにhellotesのようなフォルダーを作っておきます。
2. ジニーで、Newすると、テキストフィールドが現れるので、そこに

```
// hellotes.c
```

とタイプ、これをコピーしたら、ファイル-> Save As から ナビゲート、hellotesにセーブします。

この時点でセーブしておけば、シンタックスカラーが機能します。

残りの部分を書き加えます。

```
#include <stdio.h>
```

```
int main() {
```

```
    puts("Hello, testing one two");
```

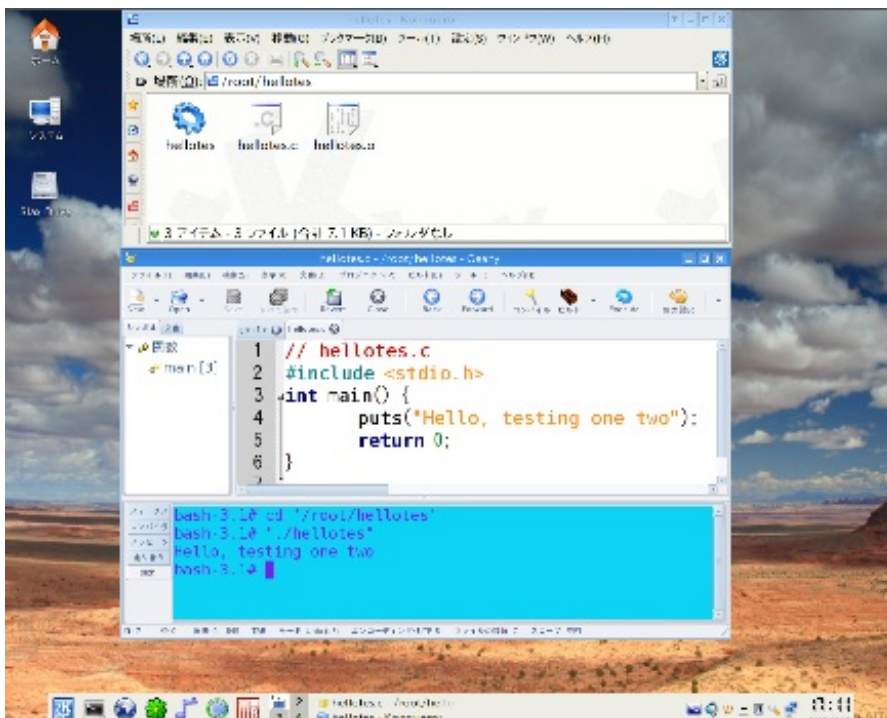
```
    return 0;
```

```
}
```

3. ツールバーのビルドを押してみます。コンパイル完了と出ましたら成功です。
4. ツールバーにあるExecuteボタンを押します。

実行結果は、vteにプリントされます。

端末の移動などもジニーがやってくれるので、便利です。



## ステージ 1 3 ジニーでglibを使う。



ジニーでジーリブを利用したプログラムを作る場合、1回目に、以下の操作を行います。

1. メニューのビルド → インクルードと引数を設定 をクリック。
2. コンパイルのフィールドに `gcc -Wall `pkg-config glib-2.0 --cflags` -c "%f"` をコピーアンドペーストする。
3. ビルドのフィールドに、`gcc -Wall `pkg-config glib-2.0 --cflags --libs` -o "%e" "%f"` をコピーアンドペーストする。

2回目以降は、この操作をしたプログラムをどこかに移動して、ジニーが読み込めなくなってしまうと、この指定は継続するので、自動的にになります。

1. ghelloという名前のフォルダーを作っておきます。
2. ジニーを立ち上げ、Newします。
3. ファイル名を、ghello.c にて、別名で保存します。
4. テキストフィールドにプログラムを書き込みます。

```
#include <glib.h>
```

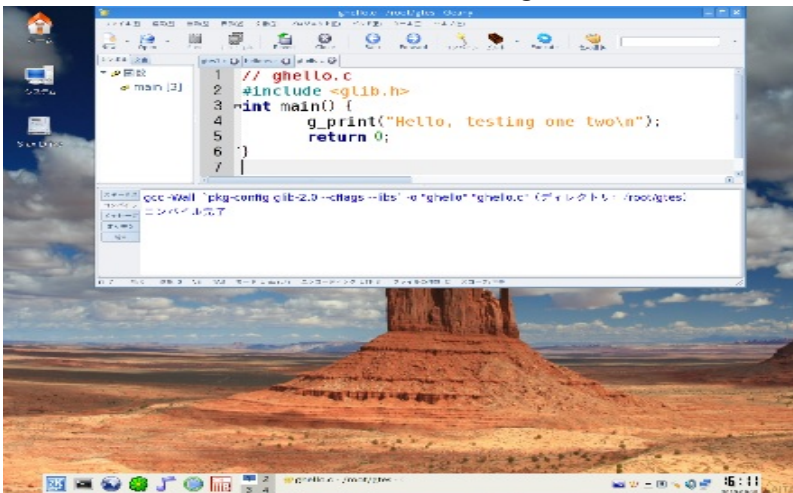
```
int main() {
```

```
    g_print("Hello, testing one two\n");
```

```
    return 0;
```

```
}
```

5. もしも、インクルードと引数を設定、をまだやってなければ、ここで行います。
6. ビルドをクリック。
7. コンパイル完了と出ましたら、Execute をクリック。
8. 端末に切り替わり、Hello, testing one two がプリントされます。





スラックスのtoolsフォルダーには、

スラックウェアのパッケージをlzmに変換する、txz2lzmと、  
デビアン deb ファイルをlzmに変換する、deb2lzmが存在します。

ここでは、deb2lzmを使ってみましょう。

deb2lzmを使う場合、dpkgというプログラムが必要になるので、これを入手します。

1. はじめに、<http://www.slax.org/modules.php>にて、dpkgで、検索すると、  
debian package tools (dpkg)が出てくるので、これをダウンロードします。

ファイル名は、debian-pkgtools-1.13.25-qx.lzm

USBペンドライブのslax/modulesフォルダーに収納したら、

# /sbin/reboot にて、再起動。

2. <http://packages.debian.org/stable/> に行き、\*.debパッケージをダウンロードします。

ここでは、ターミナルで使うelvis-tinyというエディターを取り込むことにしました。

Editors->elvis-tiny(1.4-22)にて、i386用をダウンロードします。

依存ファイルとして、

libc6 >= 2.7-1

libncurses5 >= 5.6+20071006-3 がリストされています。

これらが、lzmで用意されていないか、

<http://www.slax.org/modules.php> にて、チェックします。

3. サーチに、libc6 と入れてみます。

libc6 2.9-25 が存在しますので、ダウンロードします。

libc6の依存ファイル、glibc-locales-2.3.5-11tr もダウンロード。

libncurses でサーチ、

libncursesw5 5.7 が存在しますので、これをダウンロードします。

libncursesw5は、ワイドキャラクター対応ということなので、これでもOKかもしれません。

ここで取ってきたモジュールは、

1. libc6-2.9-25.lzm

2. glibc-locales-2.3.5-11tr.i586.lzm

3. libncursesw5-5.7.lzm

これらをUSBペンドライブの、slax/modulesフォルダーに収納します。

デビアンサイトからは。

---

4. 結局デビアンサイトからは、

1. elvis-tiny\_1.4-22\_i386.deb

2. libncurses5\_5.7+20081213-1\_i386.deb

この2本を取り込みました。

2のファイルは抑えとして、確保しておきます。

5. elvis-tiny\_1.4-22\_i386.debをホーム以下に、

deb\_pkgのようなフォルダーを作り、そこに置きます。

コンソールにて、cd deb\_pkgしたら、

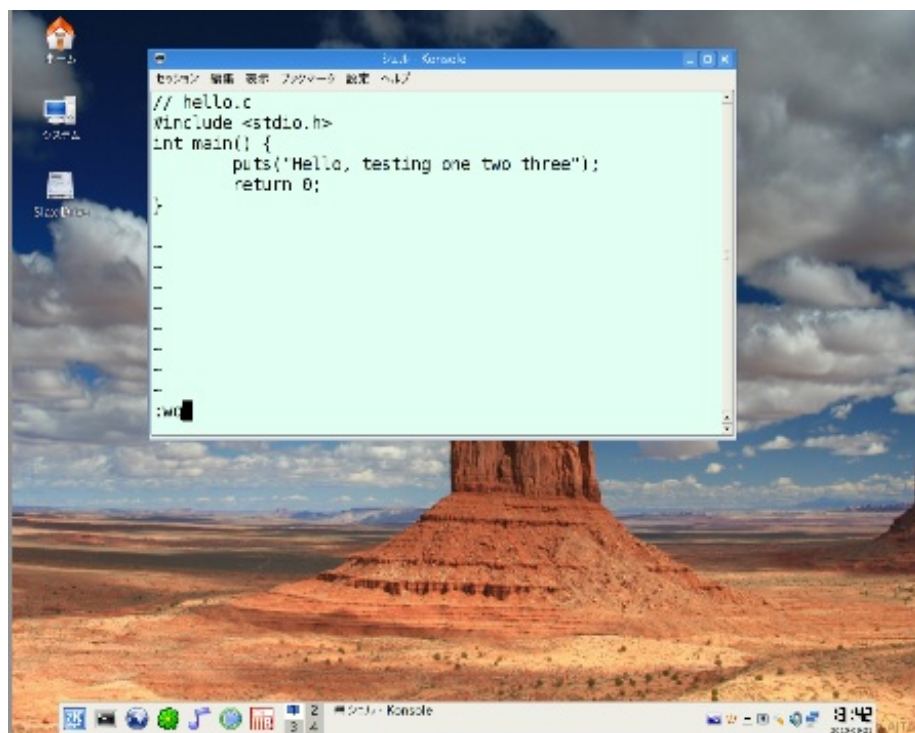
# deb2lzm elvis-tiny\_1.4-22\_i386.deb elvis-tiny1.4-22.lzm とします。

6. 出来上がったelvis-tiny1.4-22.lzmを、

USBペンドライブの、slax/modulesフォルダーに収納します。

7. 次回から、コンソールにて、

# elvis-tiny hello.cのようにして、利用します。



## ステージ 1 5 時刻表示をUTCからJSTに変更する。

時計は、デフォルトでは、UTC(GMT)を表示しています。

UTC+9 =JST です。

タイムゾーンの設定は、コントロールセンターからではなく、

右下の時刻表示の場所から行うところが、すこし分かりにくいので、ご説明しておきます。

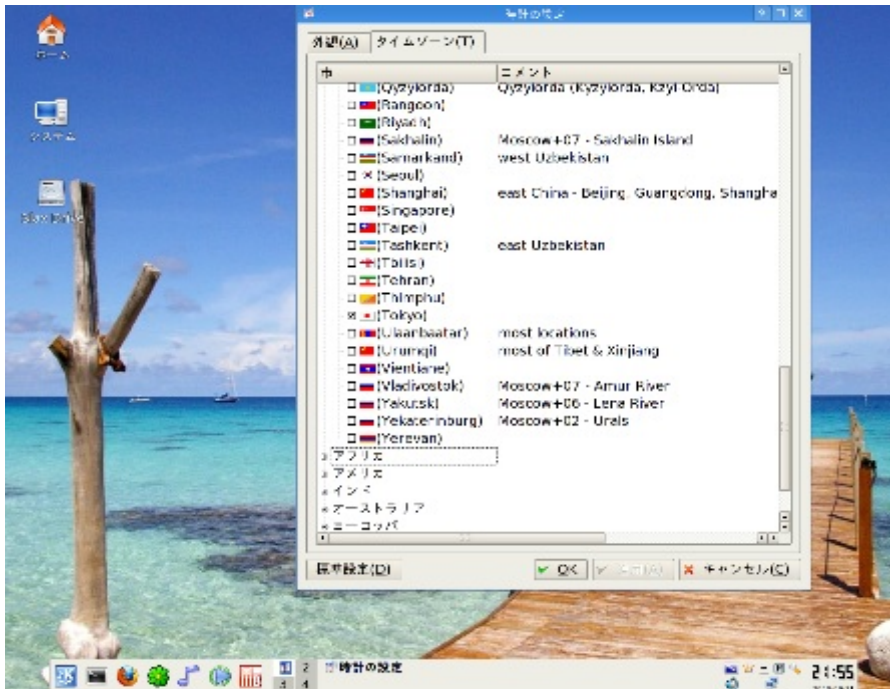
1. パネルの右下、時刻の辺りで右クリック。

タイムゾーンを表示にカーソルをあてると、

タイムゾーンを設定、が出ますので、クリック。

アジアの+をクリック、Tokyo まで送り、

Tokyo にチェックマークをつけます。



2. パネルの右下、時刻の辺りで右クリック。

日付と時刻を調整、をクリック。

時計の針を9進め、適用します。

3. しかし、このままでは、次回には、

また、UTCに戻ってしまうので、timecfg.lzmを使います。

4. <http://www.slax.org/modules.php> に出かけ、サーチに、timecfg と入れます。

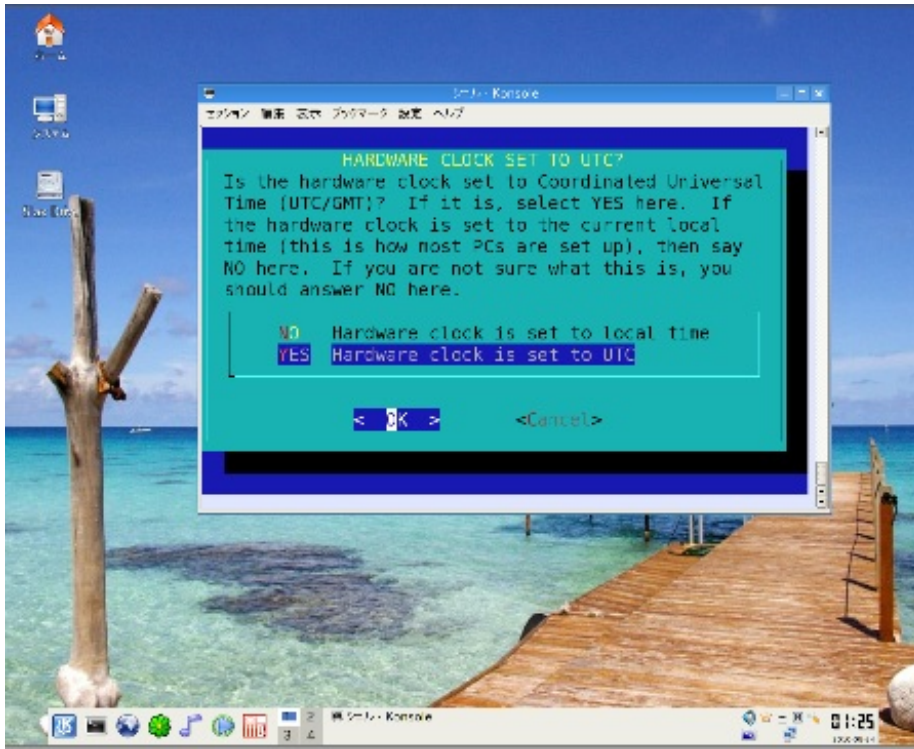
すると、timecfgが出てくるので、ダウンロードします。

このファイル、timecfg.lzmを、/slax/modulesに収納したら、

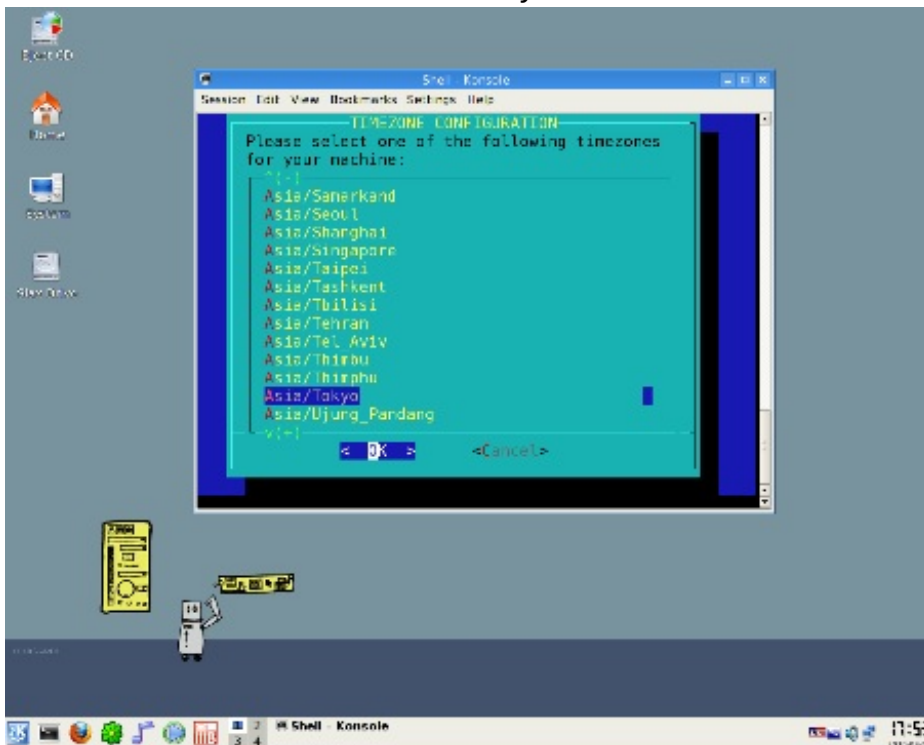
# /sbin/rebootします。

タイムコンフィギュを設定する。

5. スラックスが立ち上がってきましたら、コンソールにて、#timecfgとします。  
画面の下の列、YESをアローキーにて選択、OKします。



続いて、タイムゾーン設定の画面が出るのですが、時として出ないことがあります。しかし、先に1で、タイムゾーンの設定ができていますので、問題ありません。もし、画面が出ましたら、Asia/Tokyoにセットします。



以上で時計のJST表示の設定は完了です。  
次回から時計は、JSTを表示してくれます。





スラックスCDには、デフォルトで、ruby-1.8.7-p174が入っています。  
そこで、もう少し新しいターファイルを使って、lzmを作ってみましょう。  
リナックスでは、ソースコードからビルドする時、以下の3ステップを行います。

```
# ./configure  
# make  
# make install
```

今、この方法をスラックスUSBペンで行ったらどうなるでしょうか。  
そのまま、使えるようになります。

スラックスでは、モジュールにしないで、通常のリナックスと同じように、  
インストールして利用する、という方法を使うことも、実は可能です。  
ただし、モジュール化してしまえば、圧縮もされ、取り回しもクールです。

1. <http://www.ruby-lang.org/ja/downloads/>にて、ruby-1.8.7-p302.tar.gz を入手。  
myrubyというフォルダーを作り、そこにしておく。
2. # tar xvf ruby-1.8.7-p302.tar.gz とタイプ、リターン。
3. # cd ruby-1.8.7-p302 とタイプ、リターン。
4. # configure-for-slax とタイプ、リターン。
5. # make とタイプ、リターン。
6. # mkdir /tmp/myrb187 とタイプ、リターン。これで、tmp以下にmyrb187というフォルダーが  
作られる。
7. # make install DESTDIR=/tmp/myrb187 とタイプ、リターン。  
myrb187の内部に階層構造状に、ルビーがインストールされる。
8. # dir2lzm /tmp/myrb187 /root/myruby/ruby-1.8.7-p302.lzm とタイプ、リターン。  
出来上がったlzmファイルをUSBペンのslax/modulesに収納します。  
7の操作で、フォルダーを用意しなければ、# make installで、  
ルビーは、/usr/local以下に展開されます。この場合、lzmファイルの出番はなくなります。  
/tmp/myrb187は、rootに移動させて、中の状況を観察しておきます。
9. # mv /tmp/myrb187 /root

以上で、ソースから、lzmファイルにする作業は完了です。

もし、/usr/local以下にインストールしたものを削除する時は、

```
# cd /root/ruby-1.8.7-p302 とタイプ、リターン。  
# make uninstall とタイプ、リターン。
```

この時、何が削除され、何が削除されなかったは、myrb187の中身を見ると分かります。

# wrap-up

これで、イントロの巻はおしまいです。

2章は、カスタムスラックスUSBの実践特集の巻、となります。

ここで、スラックスでの終了操作を確認しておきましょう。

もっとも堅実な方法は、コンソールからの操作です。

終了は、`# /sbin/poweroff` もしくは、`# /sbin/halt`

再起動は、`# /sbin/reboot` となります。

メニューのログアウトからですと、

時として、ブラックアウトして待たされるという場面に遭遇します。

慌てず騒がずがいちばんですが、不安になるほど遅いこともあります。

コンソールからの操作では、比較的順調に作業を行ってくれるので、おすすめです。

