「Windows 7上VritualBox Ubuntu10.04への SADインストール手引き」

2011/04/08

佐藤洋一yoichi.sato@j-parc.jp

目次：

１．準備段階（VirtualBoxとUbuntu設定）

１－１　Virtual Box上で仮想マシンを構成 (< 30 min)

１－２　Ubuntuのダウンロード (~15 min)

１－３　VirtualBoxでのUbuntuインストール (2-3 hours)

２．SADインストール

２－１　SADインストール準備のためのUbuntu環境整備

２－２　SADのダウンロード、インストール、動作確認

やり方その１、２、３

**１．準備段階（VirtualBoxとUbuntu設定）**

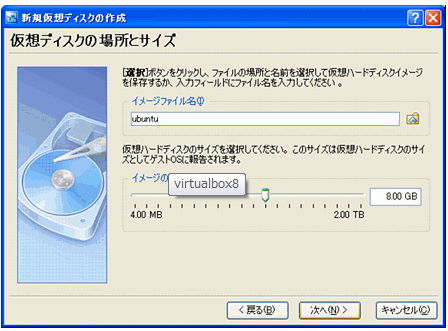
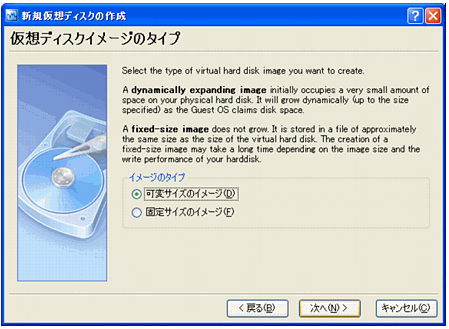
WindowsへのVirtualBoxとUbuntuインストールは

<http://linux.owlnet.jp/virtualbox01.html>　~ <http://linux.owlnet.jp/virtualbox03.html>

を参考にしました。ただし少し古いので、ここでは流れの紹介にVirtualBoxの版による相違点と注意点の指摘を加えます。

１－１　Virtual Box上で仮想マシンを構成（所要時間30分以下）

<http://www.virtualbox.org/wiki/Downloads>でOSに応じたVirtualBoxをダウンロードし、実行します。新規（N）アイコンをクリックして仮想マシンを構築します。仮想マシン名、OSタイプ（Ubuntu）、メモリサイズ、ディスクイメージサイズを設定します。



**64GB**

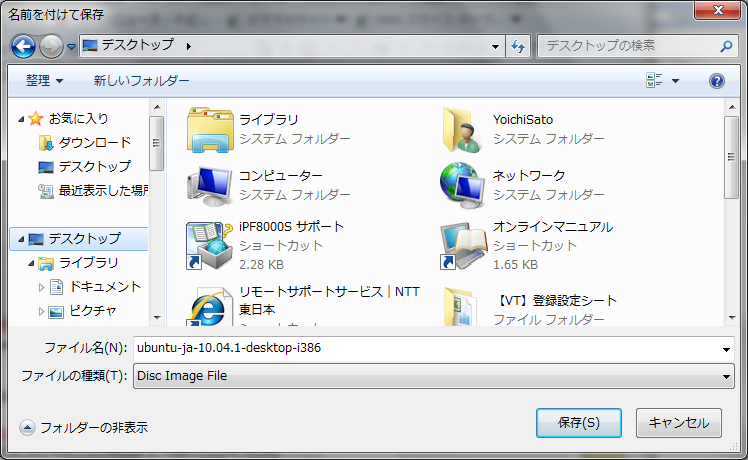
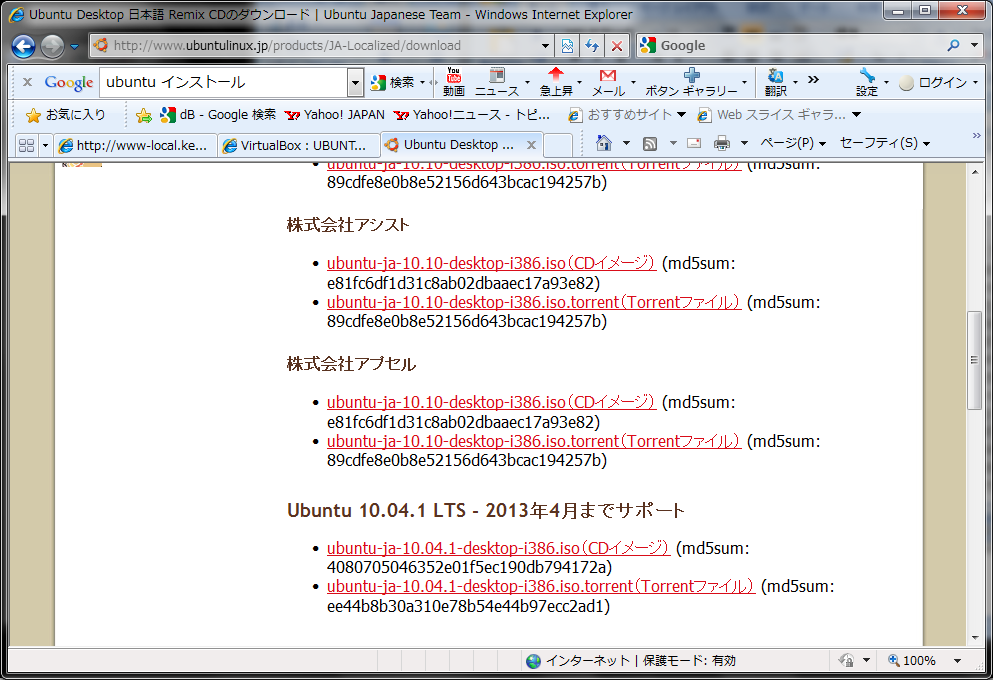
コメント：

ディスクイメージサイズは、固定サイズか可変サイズ（使用領域だけを確保する）を選べます。固定サイズの方はアクセスが早いです。ハードディスク領域に余裕がなく、とにかく節約したい場合は可変サイズが良いと思います。ただし注意点があります。可変サイズを選んだとしてもここ（右上図）で設定する「ディスクサイズ」そのものは後日変更が効きません。この「ディスクサイズ」は可変サイズの最大値を規定します。8GBとかでケチると後でサイズを増やしたい時に面倒です。以下では新規仮想マシンをメモリ1024MB、HD 64GB（固定サイズ）とします。

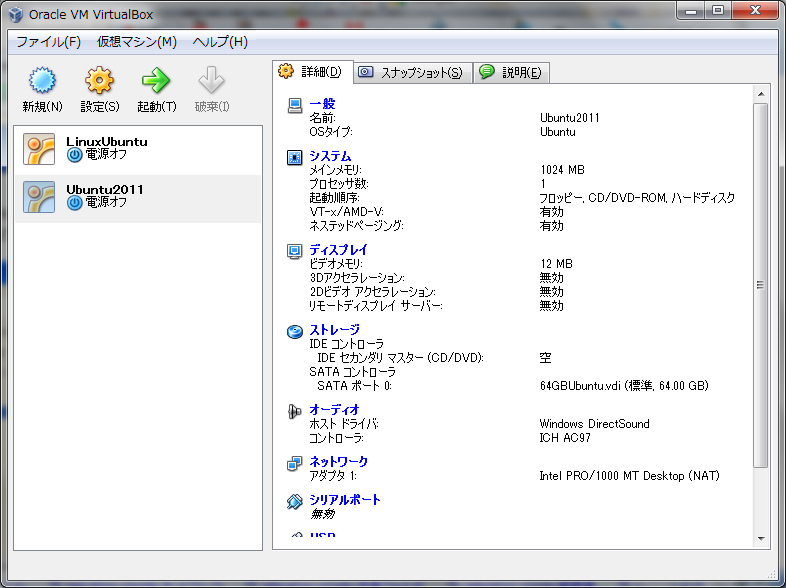
１－２　Ubuntuのダウンロード（所要時間　10~15分）

このVirtualBox仮想マシンにubuntuをインストールします。CD/DVD-ROMのISOイメージファイルをダウンロードしてCD-ROMを焼かずに済ませる方法を採用します。

<http://www.ubuntulinux.jp/products/JA-Localized/download>にあるisoイメージファイル（[ubuntu-ja-10.04.1-desktop-i386.iso（CDイメージ）](http://cdimage.ubuntulinux.jp/releases/10.04/ubuntu-ja-10.04.1-desktop-i386.iso)679MB）をデスクトップに保存します。Long term supportということで最新版10.10ではなく10.04にしておきます。

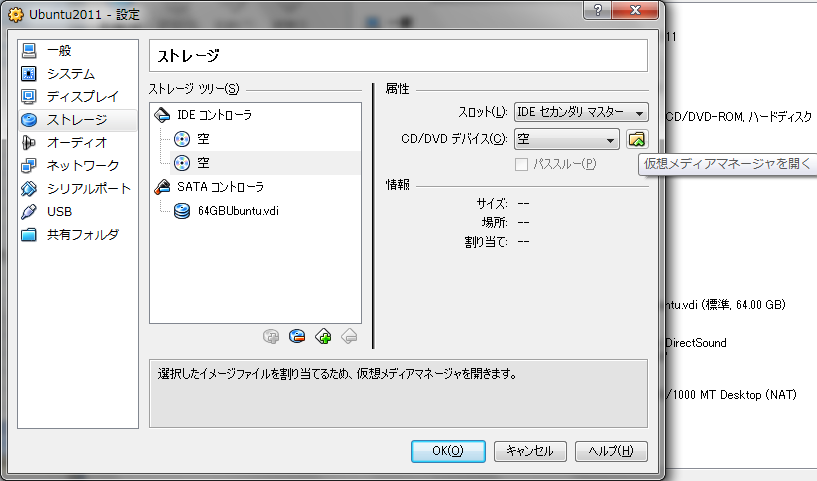
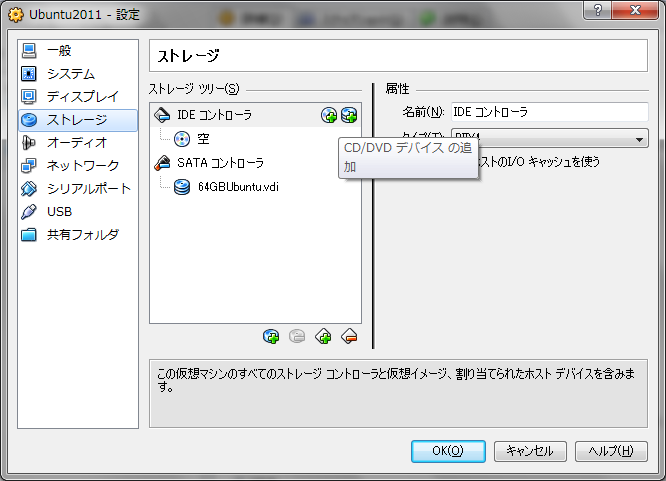


再びVirtualBoxに戻りUbuntuインストールに進みます。

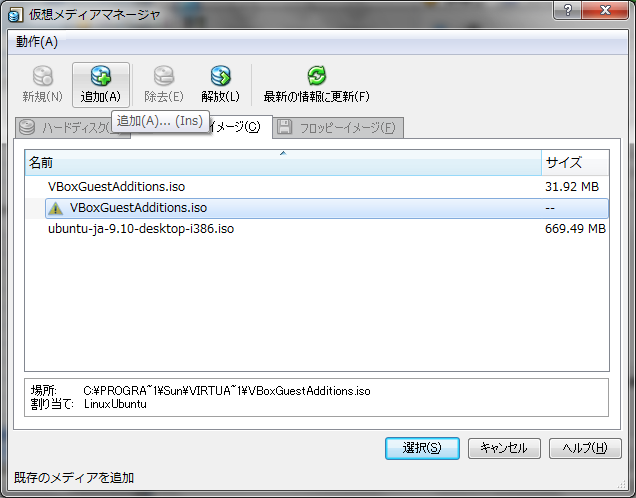
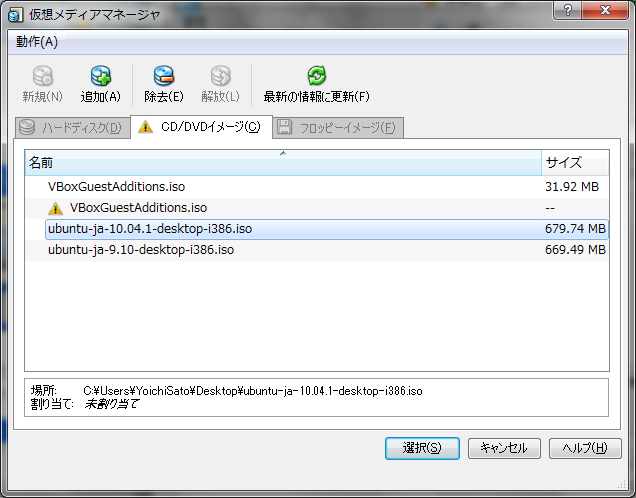


１－３　VirtualBoxでのUbuntuインストール（所要時間　設定1時間（未経験者）+インストール待ち時間OS 1時間弱 アップデートインストール待ち時間25分）

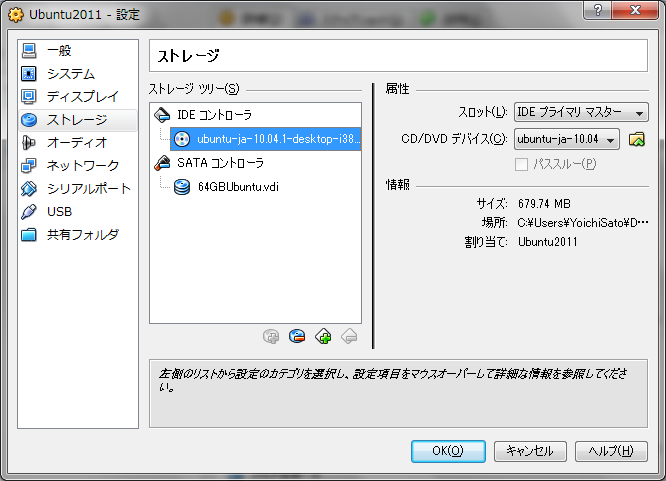
VirtualBoxメインウインドウ内で仮想マシンを選択後、設定アイコンをを選び、ストレージ＞IDEコントローラ＞CD/DVDアイコン経由で仮想メディアマネージャを起動させます。



仮想メディアマネージャ内で、追加アイコン＞「１－２でダウンロードしたubuntu\*.isoイメージファイル」を登録し、選択します。

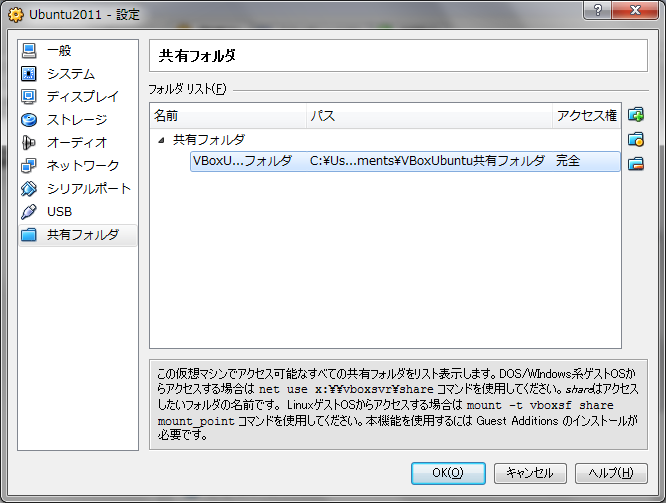
 

設定ウインドウに戻りますが、ここでisoイメージファイルを起動ディスクとするためIDEプライマリマスターに入れて置きます。セカンダリでは起動ディスクになりません。

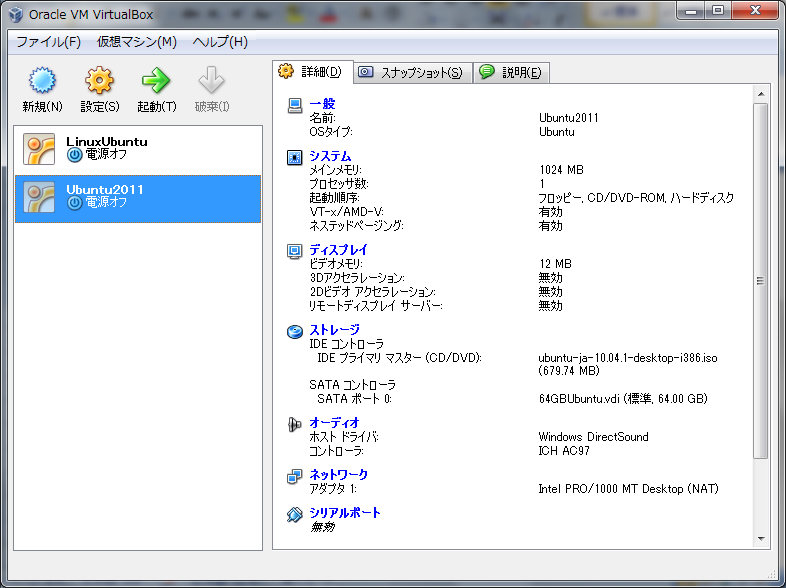


ついでに共有フォルダを設定しておくとWindows上ファイルと仮想マシン間でファイルのやりとりが出来て便利です。共有フォルダの使用法は

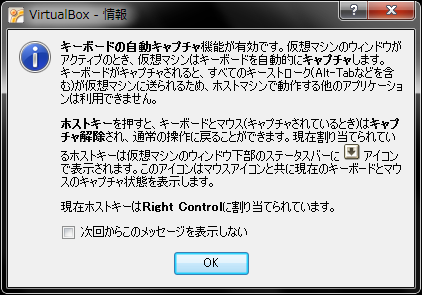
<http://hisway.at.webry.info/200808/article_15.html>などを参照してください。



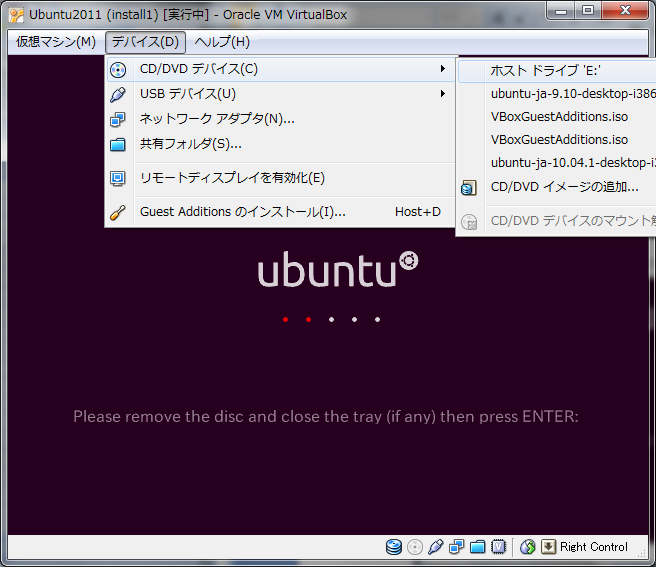
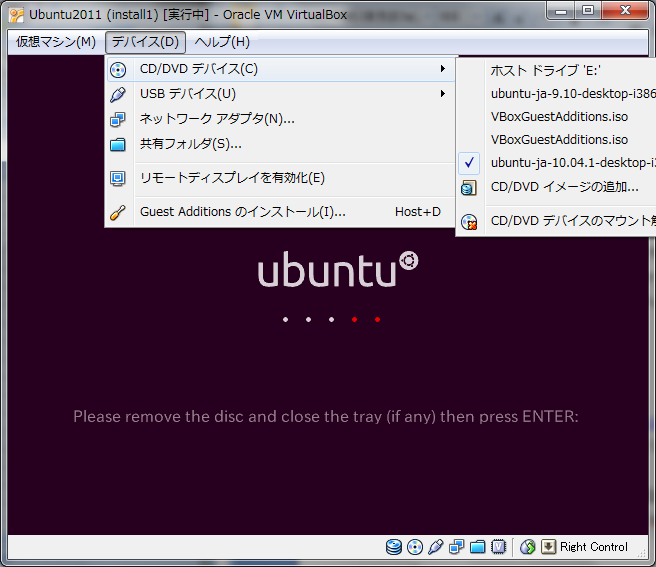
メインウインドウにもどり仮想ディスクを起動させます。



プライマリディスクからUbuntuインストールディスクが起動しこんな表示が出ます。



これの解決策は次項で紹介しますが、今後もubuntuのバージョンアップの度にRight Controlを使うことになります。ここまでいけばUbuntu自体のインストールはガイドに従うだけです（上の表示もUbuntuの文字も出ないようでしたらubuntu\*.isoイメージがIDEプライマリディスクに割り当てられているか、ダウンロードが完了しているかなど確認してください）。インストール終了後はCD/DVDデバイスからubuntu\*.isoイメージを非選択にしてubuntu\*.isoイメージをIDEプライマリマスターから抜きます。

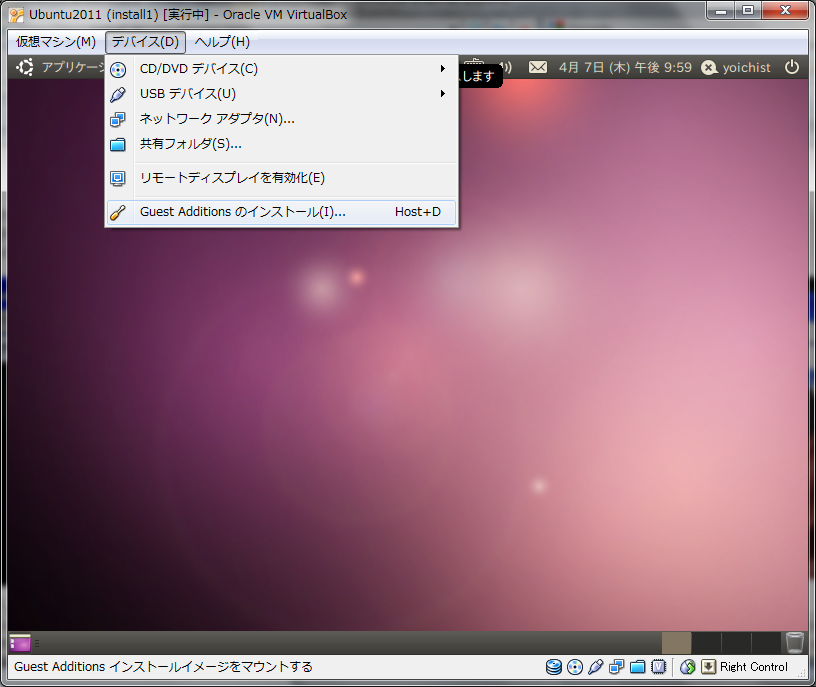


再起動してインストールは終了ですが、ネットワークにアクセスしていろいろ飛ぶ前にセキュリティアップデート（2011/4/7の時点で144個！）を全部済ませます。

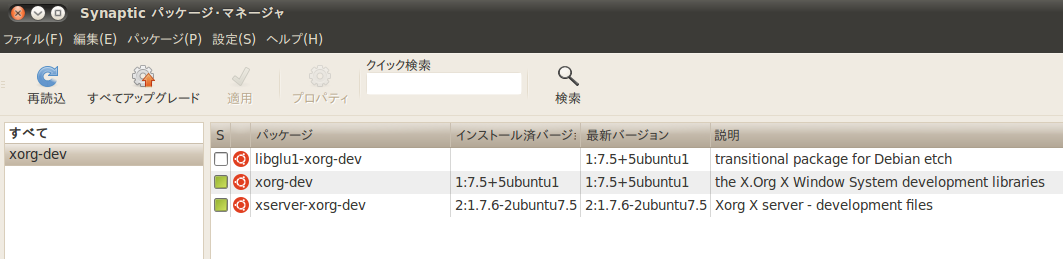
**２．SADインストール**

２－１　SADインストール準備のためのUbuntu環境整備

* 画面拡大、マウス統合のためのGuestAdditionのインストール（VirtualBoxメニュバーのデバイス(D)で選んだ後、Ubuntuメユーバー内の場所からVBOXADDITIONS\*\*を選択し、autorun.shを走らせた後、仮想マシン再起動：所要時間3分）をする。この作業は後日Ubuntuの関連バージョンアップ（カーネルなど）の度に必要となります。



* SADで描画ツールTkinter用のXorg-devのインストール（Ubuntuメニューバー「システム」から「システム管理」を選び、「Syapticパッケージ・マネージャ」を起動する。ここで、”xorg-dev”を検索し、インストールする）ですが、検索の際にはクイック検索ではなく、検索アイコンを使用します。



* その他の一般アプリケーションインストールにはUbuntuメニューバー「アプリケーション」から「Ubuntuソフトウエアセンター」を選び、検索、インストールする。gfortran-4.4は必須。 emacs23, gnuplot, pythonなどはお好みです。

２－２　SADのダウンロード、インストール、動作確認

MacOSX10.6.6 / VirtualBox / Ubuntuでの鎌田さんのインストール解説

<http://acc-physics.kek.jp/SAD/School/SADinstall.htm>

と同じアプローチを試したのですが、ダウンロードまでは問題なく、インストールも問題なく見えたのですが、そのままでは画像機能確認のところでのところでつまずいてしまいました（2-2-C項で紹介）。そこで、山田秀衛さんの御助力で公式版をインストールする方法を紹介します（2-2-A項）。こちらの方は私がSADマシンで走らせていたものが画像も含め同様に動作しました。

２－２－A　やり方その１（公式版による成功例）

SADダウンロード：

Ubuntu内でFirefox（デフォルト）を立ち上げ、SADホームページ

（<http://acc-physics.kek.jp/sad/sad.html>）からSAD source archive

（<http://acc-physics.kek.jp/sad/SADrelease/distfiles/>）

に入り、oldsad-snapshot-2011-04-08.tar.gzをクリック。



端末を開きダウンロードフォルダ内のtarファイルを展開し中に入る。

username@pcname:~$ cd ダウンロード

username@pcname:~/ダウンロード$ tar xzvf oldsad-snapshot-2011-04-08.tar.gz

username@pcname:~ダウンロード$ cd oldsad-snapshot-2011-04-08

以下/home/username/ダウンロード/oldsad-snapshot-2011-04-08　内での作業：

SADTerminal使用時にkey-up/key-downでヒストリを表示するため

$ patch -p0 < contrib/patches/patch-hack:Package:SADTerminal.n

このフォルダに山田秀衛さん作成のパッチと環境ファイル（pp 9-20）

oldsad-snapshot-2011-04-07-gcc44.patchとsad.conf

をコピーした後

$ patch -p1 < ../oldsad-snapshot-2011-04-07-gcc44.patch

sad.confの記述を確認しfortranコンパイラ、フォルダ名を適宜合わせる。

$ emacs sad.conf

*FC=gfortran*

*SAD\_ROOT=/opt/SAD-snapshot-2011-04-08*

ここでは上のまま変えなかったのでgfortranをインストールする。

$ sudo apt-get install gfortran

以上の準備でコンパイルを実行する。

$ sudo make all

$ sudo make install

上手くいけばCanvasが一旦立ち上がりエラーなくインストール終了です。

ここで使ったsad.confではSADは

/opt/SAD-snapshot-2011-04-08/bin/gs

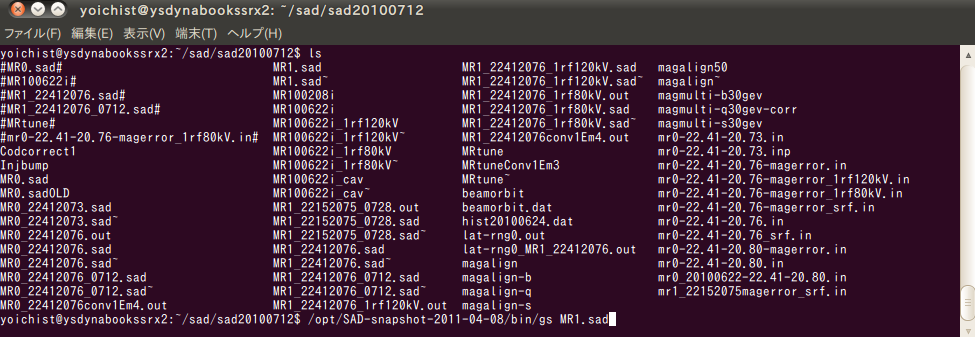
で立ち上がります。

ここで使用したパッチ（pp 9-20）は山田秀衛さんの御好意で提供して頂きました。ご使用に際しては自己責任でお願いします。

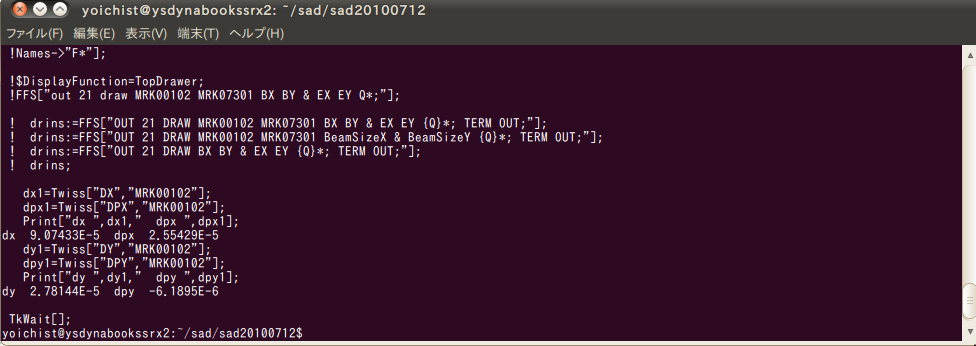
動作確認例：

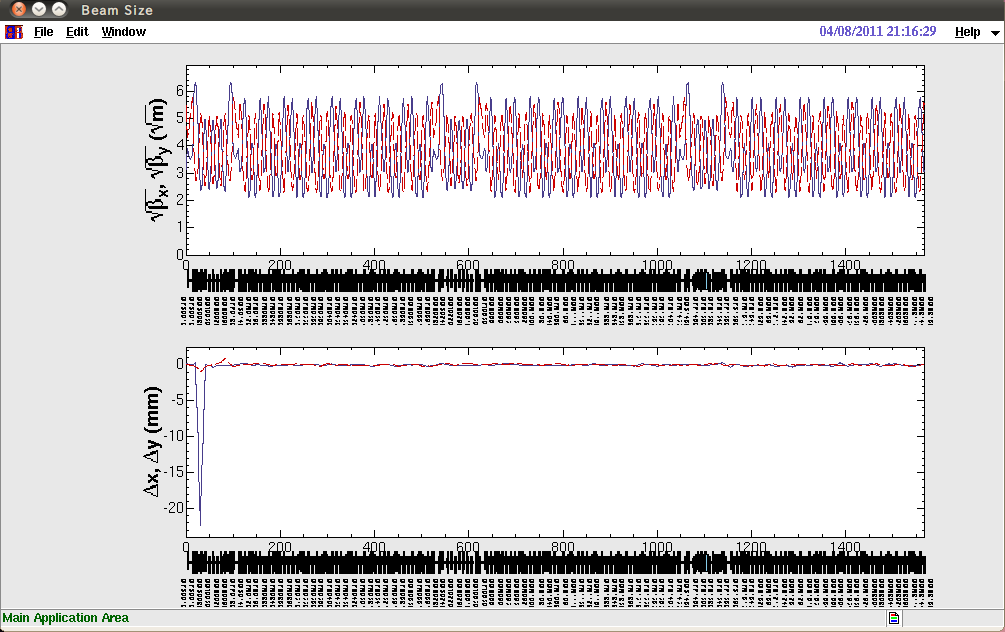
J-PARC MRのOpticsを見たのですが、amsadb.kek.jpの結果とくらべて大まかには違いがないような感じです。

計算開始：

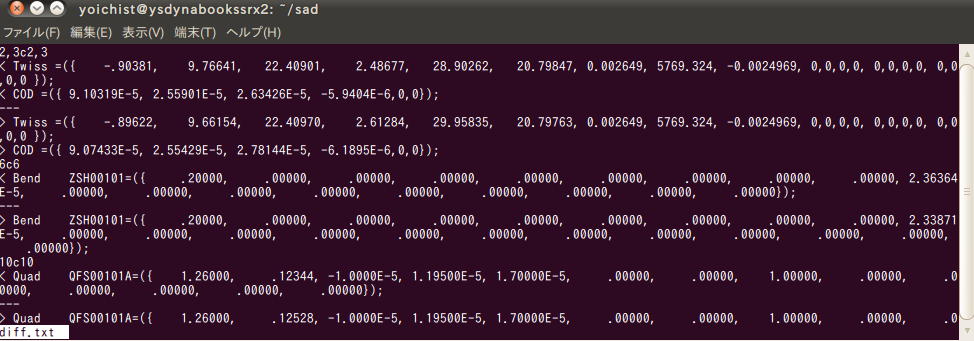


計算終了：





amsadb.kek.jpで計算した時との比較



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sad.conf例（oldsad-snapshot-2011-04-08で使用の場合）山田秀衛さん作成

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*sad.conf例全文\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# New Makefile -- Local Configuration Part Sample

#

## Compiler setting examples

# -- Interl C/Fortran Compiler 7.1 --

#COMPILER=Intel

#FC=ifc

#CC=icc

# -- Interl C/Fortran Compiler 8.x --

#COMPILER=Intel

#FC=ifort

#CC=icc

##SYS\_FOPT+=-vms -ccdefault list

#SYS\_FOPT\_ENABLE\_BACKSLASH\_ESCAPE=-assume bscc

#SYS\_DEPOBJ+=sim/intel\_fortran8.o

# -- GNU Compiler Collection 4.x --

COMPILER=GNU

FC=gfortran

#SYS\_FOPT+=-fbackslash

# -- Debugging source tree --

FOPT=-g -O1

COPT=-g -O1

# -- Debugging memory allocator --

#COPT+=-DDEBUG\_MEMORY

# -- Debugging memory usege --

#COPT+=-DUSE\_ELECTRICFENCE -DUSE\_ELECTRICFENCE\_FREECHECK

#EXE\_LIBS+=-L/usr/local/lib -lefence

## SAD root directory

SAD\_ROOT=/opt/SAD-snapshot-2011-04-08

## Contrib source archive directory

#SOURCE\_ARCHIVE\_DIR=/usr/ports/distfiles

## X11 setup

#

USE\_X11=YES

X11\_PREFIX=/usr

##X11\_INCDIR=$(X11\_PREFIX)/include

##X11\_LIBDIR=$(X11\_PREFIX)/lib

##X11\_LIBS=-lX11

## Tcl/Tk setup

#

USE\_TCLTK=YES

##BUILD\_TCLTK=YES

##USE\_TCLTK\_SHARED\_LIBRARY=YES

##TCLTK\_GUI\_BACKEND=X11|AQUA

TCLTK\_FONT\_SYSTEM=Xft

TCLTK\_VERSION=8.4.19

##TCLTK\_PREFIX=$(SAD\_ARCH\_DIR)

##TCLTK\_INCDIR=$(TCLTK\_PREFIX)/include

##TCLTK\_LIBDIR=$(TCLTK\_PREFIX)/lib

##TCLTK\_LIBS=-ltk$(TCLTK\_VERSION) -ltcl$(TCLTK\_VERSION)

## Python setup

#

#USE\_PYTHON=YES

##BUILD\_PYTHON=YES

##USE\_PYTHON\_THREAD=YES

##PYTHON\_VERSION=2.4

##PYTHON\_PATCH\_LEVEL=1

##PYTHON\_PREFIX=$(SAD\_ARCH\_DIR)

##PYTHON\_INCDIR=$(PYTHON\_PREFIX)/include/python$(PYTHON\_VERSION)

##PYTHON\_LIBDIR=$(PYTHON\_PREFIX)/lib

##PYTHON\_LIBS=-lpython$(PYTHON\_VERSION)

##PYTHON\_PATH=$(PYTHON\_PREFIX)/lib/python$(PYTHON\_VERSION)

## EPICS setup

#

#USE\_EPICS=YES

#EPICS\_PREFIX=/jk/epics/R314-Current/base

#EPICS\_INCDIR=$(EPICS\_PREFIX)/include -I$(EPICS\_PREFIX)/include/os/Linux

#EPICS\_LIBDIR=$(EPICS\_PREFIX)/lib/linux-x86

#EPICS\_LIBS=-lca -lCom -ldbIoc -ldbStaticHost -ldbStaticIoc -ldbtoolsIoc

## for KEK-B EPICS setup

#USE\_KEKB\_EPICS=YES

# --- don't touch ---

ifeq ($(USE\_KEKB\_EPICS),YES)

KEKB\_EPICS\_ARCH=Unknown

ifeq ($(OS\_NAME),Linux)

ifeq ($(CPU\_ARCH),i386)

KEKB\_EPICS\_ARCH=Linux

endif

endif

ifeq ($(OS\_NAME),OSF1)

ifeq ($(CPU\_ARCH),Alpha)

KEKB\_EPICS\_ARCH=alpha

endif

endif

ifeq ($(OS\_NAME),Darwin)

ifeq ($(CPU\_ARCH),PowerPC)

KEKB\_EPICS\_ARCH=darwin-ppc

EPICS\_PREFIX=/proj/epics/R314/base

EPICS\_LIBS=-lca -ldbStaticHost -lCom

COPT+=-I$(EPICS\_PREFIX)/include/os/Darwin

endif

endif

ifeq ($(OS\_NAME),HP-UX)

ifeq ($(CPU\_ARCH),PA1.0)

KEKB\_EPICS\_ARCH=hp700

endif

ifeq ($(CPU\_ARCH),PA2.0)

KEKB\_EPICS\_ARCH=hppa8k

endif

endif

ifeq ($(KEKB\_EPICS\_ARCH),Unknown)

$(error Unknown architecture)

else

USE\_EPICS=YES

EPICS\_PREFIX?=/proj/epics/R313/base

EPICS\_LIBDIR=$(EPICS\_PREFIX)/lib/$(KEKB\_EPICS\_ARCH)

endif

endif # USE\_KEKB\_EPICS==YES

# End of File

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以上sad.conf例\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

oldsad-snapshot-2011-04-07-gcc44.patch 山田秀衛さん作成

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

使用法：　% patch -p1 < ../oldsad-snapshot-2011-04-07-gcc44.patch

内容：　以下の問題を解決するためのパッチ

(1) libtai-0.60のコンパイルに失敗する

- gcc 4.3以降ではunistd.h, stdlib.h等を明示的に#includeしないとコンパイルに失敗する。

- main()が型なしで定義されているのが気持ち悪い

- sad.confの書き方ではlibtaiを使わずにコンパイルできるっぽい。必須なの? 違うの?

(2) src/sim/sad\_signal.h

- コンパイルオプションの-D\_XOPEN\_SOURCEと#define \_XOPEN\_SOURCE = 1が競合してエラーになる。

\*\*\*\*\*\*\*\*\*\*\*oldsad-snapshot-2011-04-07-gcc44.patchファイル全文\*\*\*\*\*\*\*\*\*\*

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-check.c oldsad-snapshot-2011-04-07/files/libtai/patch-check.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-check.c 1970-01-01 09:00:00.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-check.c 2011-04-07 16:24:44.130647703 +0900

@@ -0,0 +1,18 @@

+--- check.c.orig 1998-10-14 01:52:20.000000000 +0900

++++ check.c 2011-04-07 15:27:02.319458427 +0900

+@@ -11,7 +11,7 @@

+ char out[101];

+ char x[TAI\_PACK];

+

+-main()

++int main()

+ {

+ struct tai t;

+ struct tai t2;

+@@ -46,5 +46,5 @@

+ }

+ printf("\n");

+ }

+- exit(0);

++ return 0;

+ }

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-easter.c oldsad-snapshot-2011-04-07/files/libtai/patch-easter.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-easter.c 1970-01-01 09:00:00.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-easter.c 2011-04-07 16:25:00.646462328 +0900

@@ -0,0 +1,27 @@

+--- easter.c.orig 1998-10-14 01:52:20.000000000 +0900

++++ easter.c 2011-04-07 15:28:02.181642074 +0900

+@@ -1,11 +1,12 @@

+ #include <stdio.h>

++#include <stdlib.h>

+ #include "caldate.h"

+

+ char \*dayname[7] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" } ;

+

+ char out[101];

+

+-main(argc,argv)

++int main(argc,argv)

+ int argc;

+ char \*\*argv;

+ {

+@@ -23,8 +24,8 @@

+ caldate\_frommjd(&cd,day,&weekday,&yearday);

+ if (caldate\_fmt((char \*) 0,&cd) + 1 >= sizeof out) exit(1);

+ out[caldate\_fmt(out,&cd)] = 0;

+- printf("%s %s yearday %d mjd %d\n",dayname[weekday],out,yearday,day);

++ printf("%s %s yearday %d mjd %ld\n",dayname[weekday],out,yearday,day);

+ }

+ }

+- exit(0);

++ return 0;

+ }

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-leapsecs.c oldsad-snapshot-2011-04-07/files/libtai/patch-leapsecs.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-leapsecs.c 1970-01-01 09:00:00.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-leapsecs.c 2011-04-07 16:26:21.976897483 +0900

@@ -0,0 +1,18 @@

+--- leapsecs.c.orig 1998-10-14 01:52:20.000000000 +0900

++++ leapsecs.c 2011-04-07 15:25:30.509191308 +0900

+@@ -9,7 +9,7 @@

+

+ char line[100];

+

+-main()

++int main(void)

+ {

+ struct caldate cd;

+ struct tai t;

+@@ -24,5 +24,5 @@

+ fwrite(x,TAI\_PACK,1,stdout);

+ }

+

+- exit(0);

++ return 0;

+ }

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-leapsecs\_read.c oldsad-snapshot-2011-04-07/files/libtai/patch-leapsecs\_read.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-leapsecs\_read.c 2007-12-11 21:03:46.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-leapsecs\_read.c 2011-04-07 16:27:12.921386040 +0900

@@ -1,6 +1,16 @@

---- leapsecs\_read.c.orig Fri Jun 16 22:44:31 2000

-+++ leapsecs\_read.c Fri Jun 16 22:45:17 2000

-@@ -18,7 +18,7 @@

+--- leapsecs\_read.c.orig 2011-04-07 16:23:10.950431620 +0900

++++ leapsecs\_read.c 2011-04-07 15:35:10.708419552 +0900

+@@ -2,7 +2,8 @@

+ #include <sys/stat.h>

+ #include <fcntl.h>

+ #include <errno.h>

+-extern int errno;

++#include <stdlib.h>

++#include <unistd.h>

+ #include "tai.h"

+ #include "leapsecs.h"

+

+@@ -18,7 +19,7 @@

int i;

struct tai u;

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-nowutc.c oldsad-snapshot-2011-04-07/files/libtai/patch-nowutc.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-nowutc.c 1970-01-01 09:00:00.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-nowutc.c 2011-04-07 16:27:38.015501673 +0900

@@ -0,0 +1,32 @@

+--- nowutc.c.orig 2011-04-07 16:24:00.316433907 +0900

++++ nowutc.c 2011-04-07 15:31:32.811624883 +0900

+@@ -10,11 +10,11 @@

+

+ char x[TAIA\_FMTFRAC];

+

+-main()

++int main()

+ {

+ if (leapsecs\_init() == -1) {

+ fprintf(stderr,"utcnow: fatal: unable to init leapsecs\n");

+- exit(111);

++ return 111;

+ }

+

+ taia\_now(&now);

+@@ -23,7 +23,7 @@

+ taia\_tai(&now,&sec);

+ caltime\_utc(&ct,&sec,(int \*) 0,(int \*) 0);

+

+- printf("%d-%02d-%02d %02d:%02d:%02d.%s\n"

++ printf("%ld-%02d-%02d %02d:%02d:%02d.%s\n"

+ ,ct.date.year

+ ,ct.date.month

+ ,ct.date.day

+@@ -33,5 +33,5 @@

+ ,x

+ );

+

+- exit(0);

++ return 0;

+ }

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/patch-yearcal.c oldsad-snapshot-2011-04-07/files/libtai/patch-yearcal.c

--- oldsad-snapshot-2011-04-07.orig/files/libtai/patch-yearcal.c 1970-01-01 09:00:00.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/patch-yearcal.c 2011-04-07 16:28:05.048445663 +0900

@@ -0,0 +1,24 @@

+--- yearcal.c.orig 1998-10-14 01:52:20.000000000 +0900

++++ yearcal.c 2011-04-07 15:28:43.551156183 +0900

+@@ -1,4 +1,5 @@

+ #include <stdio.h>

++#include <stdlib.h>

+ #include "caldate.h"

+

+ char \*montab[] = {

+@@ -16,7 +17,7 @@

+ , "December"

+ } ;

+

+-void main(argc,argv)

++int main(argc,argv)

+ int argc;

+ char \*\*argv;

+ {

+@@ -63,5 +64,5 @@

+ printf("\n");

+ }

+

+- exit(0);

++ return 0;

+ }

diff -ruP oldsad-snapshot-2011-04-07.orig/files/libtai/set-0.60 oldsad-snapshot-2011-04-07/files/libtai/set-0.60

--- oldsad-snapshot-2011-04-07.orig/files/libtai/set-0.60 2008-04-03 21:20:44.000000000 +0900

+++ oldsad-snapshot-2011-04-07/files/libtai/set-0.60 2011-04-07 16:28:10.960204324 +0900

@@ -3,10 +3,16 @@

patch-caltime.h

patch-leapsecs.3

patch-leapsecs.h

-patch-leapsecs\_read.c

patch-tai.h

patch-tai\_now.c

patch-taia.h

patch-taia\_now.3

patch-taia\_now.c

patch-tryclock\_gettime.c

+patch-check.c

+patch-easter.c

+patch-leapsecs.c

+patch-leapsecs\_read.c

+patch-nowutc.c

+patch-yearcal.c

+

diff -ruP oldsad-snapshot-2011-04-07.orig/src/sim/sad\_signal.h oldsad-snapshot-2011-04-07/src/sim/sad\_signal.h

--- oldsad-snapshot-2011-04-07.orig/src/sim/sad\_signal.h 2009-10-03 12:46:55.000000000 +0900

+++ oldsad-snapshot-2011-04-07/src/sim/sad\_signal.h 2011-04-07 15:56:48.970014917 +0900

@@ -4,7 +4,9 @@

#include <sim/sad\_f2c.h>

#ifdef CYGWIN

#else

+#ifndef \_XOPEN\_SOURCE

#define \_XOPEN\_SOURCE 1

+#endif

#include <ucontext.h>

#endif

\*\*\*\*\*\*\*\*\*\*\*\*\* 以上oldsad-snapshot-2011-04-07-gcc44.patch \*\*\*\*\*\*\*\*\*\*\*\*\*\*

２－２－B　やり方その２

CVSでの公式版SADをインストールなら

<http://acc-physics.kek.jp/sad/SADrelease/index.html>

を参照して

Home/username/SAD$ sudo apt-get install cvs

Home/username/SAD$

cvs -z3 -d :pserver:anoncvs@www-acc-theory.kek.jp:/Users/cvs/SAD checkout -P oldsad

と続けます。これ以降の設定は未確認ですが2-2-Aと同じにできると思います。

２－２－C　やり方その３

鎌田さんのインストール解説

<http://acc-physics.kek.jp/SAD/School/SADinstall.htm>

と同じアプローチでやってみました。ダウンロードまでは問題なく、インストールも問題なく見えたのですが、そのままでは画像機能確認のところでのところでつまずいてしまいました。もう少しなんとかしようと思うのですが本稿の締め切り期限となってしまいました。以下は現状報告です。

SADダウンロード：

Ubuntu内でFirefox（デフォルト）を立ち上げ、SADホームページ

（<http://acc-physics.kek.jp/sad/sad.html>）からSAD source archive

（<http://acc-physics.kek.jp/sad/SADrelease/distfiles/>）

に入り、oldsad-amorita-snapshot-3575.tar.gzをクリック。

端末を開き

username@pcname:~$ cd ダウンロード

username@pcname:~ /ダウンロード$ ls

oldsad-amorita-snapshot-3575.tar.gz

username@pcname:~/ダウンロード

$ tar xzvf oldsad-amorita-snapshot-3575.tar.gz

username@pcname:~ダウンロード$ cd oldsad-amorita-snapshot-3575

username@pcname:~/ダウンロード/oldsad-amorita-snapshot-3575$

以下/home/username/ダウンロード/oldsad-amorita-snapshot-3575内で

gfortranパッチ当て：

$ sudo apt-get install patch

$ ./contrib/tools/workaround-gfortran:module-equivalence.sh

20秒でパッチ当て終了。

環境設定：

$ cp sad.conf.linux sad.conf

コンパイル（5分）：

$ sudo make all

$ sudo make install

次に確認として

username@pcname:~/SAD/examples/Benchmark$ ~/SAD/bin/gs bench2.sad

は鎌田さんの結果と同じです。

しかしTkinter確認の

username@pcname:~/SAD/examples/Demos$ ~/SAD/bin/gs demo-KBFrame1.sad

はlibraty [SAD/Tkinter]が読めずにハングしてしまいました。

