CellSDK21

シミュレータの起動

/opt/ibm/systemsim-cell/bin にパスを通した状態で、

```
/opt/ibm/systemsim-cell/run/cell/run_cmdline
/opt/ibm/systemsim-cell/run/cell/run_gui
```

で起動。コマンドライン版なら、プロンプトの "systemsim % " に続けて

```
systemsim % mysim go
```

で Linux が起動してくる。

プログラムのコンパイル

たとえば、

■ PPE

```
/opt/cell/bin/ppu-gcc -lspe ppe.c
```

■ SPE

```
/opt/cell/bin/spu-gcc spe.c
```

実行

シミュレータに作成したバイナリイメージをアップロードして実行する。
なので、まずは、起動したシミュレータ上のプロンプトで

```
callthru source バイナリ > シミュレータ上でのファイル
```

として、バイナリを転送。
転送したバイナリは、すべて実行ビットを立てておいてメインな PPE バイナリを実行する。
SPE のバイナリも実行ビットを立てておかないと、spe_open_image で失敗する

昔の話

IBM から Cell プロセッサ (CBE) 向けの開発キット群（以下 CBE 環境）が公開されました。
フルシステムシミュレーター、コンパイラ、ライブラリ、
SPU 向け GNU の開発キット、Linux カーネル拡張、サンプルプログラムがあります。

インストール

■ Linux のインストール

対象としている OS が FedoraCore4 なので、おとなしく FedoraCore4 をインストール。

CBE 環境を使用する際には、ファイルイメージを
loop バックしてローカルにマウントしたりもするので、SELinux を無効に。
また、CBE 環境では、tk が使われているので、インストールしておきます。
私は、「開発者向けワークステーション」（だったかな？）とかを選択しました。

■ CBE 環境

ここから必要なファイルをダウンロードします。
インストールは、

```
sh install.sh
```

で終了。ramdisk のところで、mount に失敗した場合は、
SELinux が有効になっていないか確認してみましょう。

動かしてみる

展開したディレクトリから

```
cd systemsim-cell-release/run/cell/linux/
```

と移動して、

```
mount -o loop sysroot_disk /mnt
../run_gui
```