実践的 PowerPC 活用テクニック

第13回　GDBスタブの移植とGDBによるプログラムのデバッグ

坂井 弘亮

より本格的なプログラムの開発にはデバッグ環境も重要だ．ここではPowerPC用GDBスタブを作成し，組み込みシステム開発評価キット（通称BLANCA）のオプションとして発売されている，CPUカード/PowerPC（MPC5200）へ移植する．また作成したGDBデバッグ環境を使ってサンプル・プログラムを動かす．

（編集部）

組み込みシステムでは，ホスト・パソコンとターゲット・ボードをシリアル・ケーブルで接続してデバッグする「リモート・デバッグ」が一般的です．このためには，ターゲット・ボードにホスト・パソコンで動作するGDBと通信するプログラム「GDBスタブ」を実装する必要があります．

GDBスタブはシリアル通信によってホスト・パソコンと通信し，ホスト・パソコン上で動作しているGDBからの命令（GDBコマンド）を受信して必要な動作を行い，GDBに対して応答します． コマンドの内容はレジスタ値の取得や特定アドレスのメモリの値の読み書きなど，非常に単純なものばかりです．シンボルの解析などの複雑な動作は，ホスト・パソコン上のGDBが行います．このためスタブは非常に小さなプログラムとして実装できます．

たとえば，不正メモリ・アクセスなどの例外発生時には，例外ハンドラからGDBスタブを呼び出し，GDBスタブに処理を渡します（つまり，GDBスタブは割り込み処理の延長で動作する）．GDBスタブは例外発生をシリアル経由でホスト・パソコン上のGDBに通知し，自身はシリアル受信待ちのビジー・ループに入ってGDBからの指示を待ちます．GDBは例外発生をユーザに通知し，ユーザのコマンド入力に応じてGDBスタブに指示を出し，GDBスタブからの応答を受けてユーザに応答を返します．

表1　gdb-6.8に付属するスタブのサンプル・ファイル名と対応CPU

i386-stub.c	i386
sh-stub.c	SH
m32r-stub.c	M32R
sparc-stub.c	SPARC
m68k-stub.c	680x0

1. GDBスタブの実装

● GDBスタブのソース・コード

GDBのソース・コードには，各CPUごとのGDBスタブのサンプルが付属しています（**表1**）．ライセンス条項については各ファイルの先頭部分を参照してください． GDB自体のライセンスはGPL（General Public License）ですが，スタブに関しては組み込みの利用が前提のため，多くは別のライセンス表示になっているようです．

表1でわかるようにPowerPC用のスタブは付属していません．そこでi386用のスタブである`i386-stub.c`をひな型として，PowerPCに移植することにします．

● PowerPC用スタブの作成

まずはPowerPC用スタブとして`ppc-stub.c`を追加します．今回作成したPowerPC用のスタブを**リスト1**に示します．PowerPC対応のための変更点は次の通りです．

- レジスタの定義と保存方法
- 割り込みベクタ番号からシグナル番号への変換方法
- 例外発生時のホスト・パソコンへの例外通知方法
- ステップ実行時のレジスタ設定
- ブレークポイントでのトラップ命令実行

レジスタの種類は`regnames`で定義されます．当然ながらこの定義はCPUごとに異なります．必要なレジスタと定義の順番はCPUごとに決められており，GDBのソース・コードの`regformats`というディレクトリに記述があります．PowerPC用は`reg-ppc.dat`というファイルに記述されているので，それを元にして定義しています．

`computeSignal()`では例外発生時にベクタ番号からGDBに通知するシグナル番号を計算しています．通常はメモリ・アクセス違反などではSIGBUSかSIGSEGVとし

KEYWORD——GDB，GDBスタブ，例外，ベクタ番号，DDD，Insight