

ライブラリか

## 1. GNU フリー・ソフトウェアによる クロス開発ツール群

GNU フリー・ソフトウェアの中で,クロス開発環境向けの ソフトウェア・パッケージは図1のようなものがあります.

GCC 高級言語処理コンパイラ・コレクション GCC はご存じのとおり,コンパイラ・コレクションです.通 常,C言語コンパイラをセットアップします.GCC は高級言語 (本章ではC言語を使用)からターゲット・プロセッサ向けのア センブラを出力します.

binutils バイナリ・データ処理ツール群

binutils はアセンブラ,リンカ,逆アセンブラ,オブジェク ト形式変換,アーカイバなど,バイナリ・データを扱うための ソフトウェア・パッケージです.アセンブラは,オブジェクト・ ファイルと呼ばれる中間バイナリ・データを出力します.リン カは,オブジェクト・ファイルを合体させ,最終的な実行形式 と呼ばれるプロセッサが実行するためのバイナリ・データを出 力します.逆アセンブラは,オブジェクト・ファイルや実行形 式からアセンブリ言語に逆変換をかけます.アーカイバはライ ブラリを作成する際に用いられ,複数のオブジェクト・ファイ ルを集め,\*.aという形式のライブラリ・ファイルを作成する ためのコマンドです.

newlib C言語標準ライブラリ

C言語を用いる魅力のひとつは,やはり,文字列操作や入出 力操作といった複雑な処理が標準関数として定義されているこ とが挙げられます.組み込み機器において,ユーザへの文字列 出力や,ユーザからの入力を処理することは必須の事がらです. その際に,基本的な文字列操作関数などから組み上げていくこ とはたいへんな労力を要します.

C 言語では, このような基本関数は標準化されています. GNU フリー・ソフトウェアの中にも, この C 言語の標準関数 ライブラリが提供されています.代表的なものとしては,glibc と newlib があります.glibcは Linux などの OS 向けに用いら れるのが一般的です.そのため,ここではより組み込み関連の ユーザを意識した newlibを用います.newlibは RedHat 社が 提供しており,組み込み開発に向けた C 言語標準ライブラリを 提供しようという試みで開発が始まったライブラリです.

## 2. C言語開発環境をセットアップする

まずは,前述のC言語開発環境をセットアップしてみましょう.ここでは,Cygwin上にGNUクロス環境をセットアップします.Cygwinのセットアップの方法については,本誌のWebサイトから1.5.18をダウンロードし,インストールしてください<sup>(1)</sup>.

本来は, Cygwin 向けの GNU クロス環境のソフトウェアは ソース・コードでの配布なので, コンパイルを行わなければな らないわけですが, すでにコンパイルされたものを先ほどの URL に置きました.ここではそのパッケージを使ってセット アップします.

まずは, Cygwin コンソールを起動します.そして,以下の



ed ch r CCLF	ned char TPS	struct { /* Bit	insigned char BF Brood char BF	igned char MI	TTA Shar	union { /* T Ishort WORD	truct { /* Byt	igne char H signe char L		gned char IO	gned char IO	d char BYTE:	struct { /* Bit	ned char TTG	ned c ar TCIE	ter unar TGIE	ned char TGIE	hed char TGIE	gned nort T	char wk7[8	nad short TC	d char BYTE;	+ruct /* Bit	jned char TCF	unsignea cna ined char TCF	ne ' char TGF	ar TGF
--------------	--------------	-----------------	--------------------------------------	---------------	----------	-----------------------------	----------------	-----------------------------	--	--------------	--------------	--------------	-----------------	--------------	---------------	---------------	---------------	---------------	-------------	------------	--------------	--------------	--------------	---------------	-------------------------------	---------------	--------

sh-elf-addr2line.exe	sh-elf-gdbtui.exe	sh-elf-size.exe
sh-elf-ar.exe	sh-elf-gprof.exe	sh-elf-strings.exe
sh-elf-as.exe	sh-elf-insight.exe	sh-elf-strip.exe
sh-elf-c++filt.exe	sh-elf-ld.exe	tcl84.dll
sh-elf-cpp.exe	sh-elf-nm.exe	tclpip84.dll
sh-elf-gcc-3.4.5	sh-elf-objcopy.exe	tclsh84.exe
sh-elf-gcc.exe	sh-elf-objdump.exe	tk84.dll
sh-elf-gccbug	sh-elf-ranlib.exe	wish84.exe
sh-elf-gcov.exe	sh-elf-readelf.exe	
sh-elf-gdb.exe	sh-elf-run.exe	

図2 /usr/local/sh-tools/binのファイル

コマンドで/usr/local に上記のパッケージを展開します.

\$ cd /usr/local

\$ tar zxvf sh-tools.tar.gz

これで,/usr/local/sh-tools以下に SH 向けのクロス開 発環境ができました.

Cygwin コンソールには,デフォルトでは/usr/local/shtools/binにパスが通っていないので,以下のコマンドで, パスを通しておくとよいでしょう.

\$ export PATH=\$PATH:/usr/local/sh-tools/bin さらに/usr/local/sh-tools/binをにコマンドで見てみる と,図2のようなコマンドが出てきます.

## COLUMN 1

第4章~第6章で用いるプログラム・データ

第4章~第6章で用いるプログラムのソース・コードに関 しては,本誌のWebサイト(http://www.cqpub.co.jp/ download/)から手に入れることができます.

まず, yamagiwa\_code.tar.gz をダウンロードします. そのファイルをCygwinのホーム・ディレクトリにコピー します.Cygwinをデフォルト・インストールの場合, C:¥cygwin¥home¥ユーザ名がホーム・ディレクトリになり ます.

そして, Cygwin コンソール上・以下のコマンド・ライン を実行します.

\$ tar zxvf yamagiwa.tar.gz

そうすると, practice1, practice2, practice3, monitor, gdb-stub, cq\_sh7144\_libというディレク トリができあがります.それぞれの中にコンパイル済みの オブジェクト・ファイルと実行形式, ソース・コード, メ イク・ファイルが入っています.メイク・ファイルを使っ てコンパイルできるように,それぞれのディレクトリの中 の構成を変更しないようにしてください.

第4章では,practice1,practice2,practice3, cq\_sh7144\_libを使います.第5章では,monitorと cq\_sh7144\_libを使います.第6章では,gdb-stubを使 います.

- これらのうち,ここでは,
- sh-elf-gcc( コンパイラ )
- sh-elf-as(アセンブラ)
- ●sh-elf-ld(リンカ)
- sh-elf-objdump( 逆アセンブラ)
- sh-elf-objcopy(バイナリ形式変換)
- sh-elf-ar(アーカイバ)
- をおもに使います.

環境が正しくセットアップされているかどうかを確認するために,後ほど解説する練習問題をコンパイルしてみましょう.

まず, **コラム**1に従って, practice1フォルダをホーム・ ディレクトリに用意してください. Cygwin でのホーム・ディレ クトリは, デフォルトの状態でCygwin がセットアップされてい る場合, C:¥cygwin¥home¥ユーザ名になっています. このフォ ルダに先ほどのダウンロード・ページから yamagiwa\_ code.tar.gzをダウンロードして,ホーム・ディレクトリに 展開してください.すると, practice1, practice2, monitor, gdb-stub, cq\_sh7144\_libというの五つのフォルダができ あがります.

まずは, practice1フォルダの中身を使って実験します. 以下のコマンドで実行形式を作ってみます.

\$ cd practice1

\$ /usr/local/sh-tools/bin/sh-elf-gcc -c

-m2 practice1.c

\$ /usr/local/sh-tools/bin/sh-elf-as

```
startup1.s -o startup1.o
```

\$ /usr/local/sh-tools/bin/sh-elf-ld

-Map practicel.map -T memory.def

startup1.o practice1.o -o practice1

以上がエラーなく実行され, practice1.mot が作成されてい れば正しく SH向け GNU クロス開発環境がセットアップされて います.

もし,上記のコマンドの中でエラーが出る,CygwinのDLL が異なるといったCygwin自体のエラーが出る場合,**コラム**2 で示す手順でソース・ファイルからのコンパイルを行ってみて ください.