

# 32ビット・マイコン開発

# 多機能マイコンであれこれやりたくなったら, リアルタイムOSがオススメ!

# 付属DVD-ROM内容

# 開発環境のインストールが不要! Windows上で動くエミュレータでソフトウェア開発できる

- ARMマイコン用に構築したソフトウェア開発環境
- 仮想マシンを起動するソフトウェア「VMwarePlayer」
- RTOS [TOPPERS/ASP]および [TOPPERS/JSP]のソース付き

### CQ出版杠

仕事をサッと整理して, CPU 社長をバリバリ働かす

OS は優秀な秘書!

### CPU 社長は仕事がいっぱいで悩んでいた



ロローグ

ハード株式会社の CPU 社長は,抱えている仕 事の多さに頭を悩ませていました.社内会議に メールの返信,見積り書の作成,原稿の執筆, 出張の準備など仕事がたくさんあり,順序よく やればこなせる量ですが,数が多すぎて困って います.

杉本明加

時には, やらなくてはならない仕事をすっか り忘れてしまい, そのたびに大慌てです.

### 仕事を整理してくれる秘書 (OS) を採用した



るときにこなす仕事に分けていきました.

CPU 社長は仕事の段取りを秘書に手伝っても らうことにしました. 秘書は, こう言いました. 「仕事の重要度や緊急度をきちんと整理しましょ う. その管理は私がします. 社長は一つずつ順 番に仕事をこなしてください」.

そして, 社長の机にあった書類や貼ってある 付箋をテキパキと整理していったのです.

期限が迫っているものや社外向けの仕事と いった優先度が高いもの,そういった仕事がな いときにこなす仕事,時間が空いていたり,重 要な仕事がほかの人の都合で止まっていたりす

すると社長は、持っている能力を最大限に発揮して、仕事を全てこなすことができたのです.

この場合,社長(CPU)の仕事がタスクに相当します.タスクが多すぎると,今やるべき仕事は何なのか,いつまでに終わらせる必要があるのかといった管理が大変です.OSは,その管理を引き受けることで CPU がタスクを実行すればよいだけにします.



割り込みで入る仕事は秘書が受け付けをする

秘書は、かかってきた電話や来客を受け付けます。そして、電話を受けて社長に取り次ぎ、社長は仕 事を中断して電話の応対をします。しかし、急ぎの仕事をしているときは社長に知らせず、今こなして いる仕事を終わらせることを優先させます。

OSは、特別なタスクである割り込み処理の管理も行ってくれます。タスクを動かしているときに CPUの外から割り込みが通知された場合、タスクを中断して割り込み処理を先に行う必要があります。 OSが割り込み処理へ移行する部分を担ってくれるため、アプリケーションは処理の本体を作成するだ けでよくなります。

\* \* \*

OS が提供しているマルチタスクという機能は、プログラム全体をタスク(仕事のかたまり)に分割して、複数のタスクを並列に実行させることができる機能です。分割したタスクに優先順位を付けることにより、優先度の高い処理から順に実行させることができます。

本書では組み込みシステムの開発で使われることの多い, OS の持つ機能に時間内に処理を完了させる機能(リアルタイム性)をプラスした「リアルタイム OS」を取り上げます. このリアルタイム OS は, 多機能な 32 ビット・マイコンで広く使われています.

32 ビット・マイコンである ARM マイコンをターゲットとして、リアルタイム OS を使ったプログラ ミング方法と、リアルタイム OS を使うための準備 (移植) について解説します.

すぎもと・めいか 組み込みプログラマ

32 ビット・マイコン開発で使われる

# リアルタイム OSの 機能を理解しよう

石田利永子

単純な処理を行う装置を開発する場合は,OS を用いずにソフトウェアを開発した方が簡単で確 実に処理ができることがあります.しかし,モー タ制御をしながらユーザ・インターフェースの処 理やネットワーク通信を行いたいとなると,複数 の処理を同時に行う必要があるので,それらの処 理を管理するシステムを開発するのは大変です. このようなシステムの開発には,リアルタイム

第

音

OS (以下, RTOS) を使うと非常に効果がありま す.

RTOSには、リアルタイム性が必要な複数の処 理を制御する機能が備わっています.また、ハー ドウェアの違いを OS が吸収してくれる<sup>注1</sup>ので、 ソフトウェア技術者はアルゴリズムの開発のみに 集中できるというメリットがあります.

RTOS が持つ特徴を本章では説明します.

### 1.1 決められた時間内に仕事を終わらせるしくみ

初めに, RTOS の最大の特徴といえる, 時間的 制約をクリアするしくみについて説明します.

#### ● 優先順位を付けてタスクを実行するしくみ →→ スケジューリング・アルゴリズム

RTOS は,スケジューリング [次に処理するタ スク (処理単位)を決める] 機能を持っています. それぞれのタスク処理が時間的制約を満たせるか どうかは,このスケジューリングのアルゴリズム で決まります.

Windows や Linux などの汎用 OS の場合, す べてのタスクが平等に実行できるようにスケ ジューリングされています. しかし, 組み込みシ ステムの場合は, 各タスクはすべて決められた時 間内に処理を終えなくてはならず,そのために早 く処理をしなければならないタスクから優先順位 を決めて実行するしくみが必要になります.

ほとんどの RTOS では、最も優先度の高いタ スクから順に実行するアルゴリズムが採用されて います、そして、優先度の高いタスクの処理が終 わるまで、優先度の低いタスクは実行されません、 また、優先度の高いタスクが実行されると、優先 度の低いタスクは中断されます(図1.1).この切 り替えのことをプリエンプト(preempt)といい ます、さらに、急いでいるタスクが無意味に優先 度の低いタスクに待たされないようなしくみ(優 先度継承、優先度上限プロトコルなど)を備えて いる OS もあります.

注1: 隠蔽するともいう.システムのリソース (資源) は OS が管理することになるので、アプリケーション・プログラムからは、OS に対して決められたコマンドを与えるだけでリソースを利用できるようになる.

VMware Player を使用して, ソフトウェア開発環境を動かす

# 付属 DVD-ROM を使用して 開発環境を構築する

 $\bigcirc$ 

筞

峕

使用するプログラム 01\_VMware Player, 02\_VMware HDD

本書付属の DVD-ROM に収録されているデータを使用して、ソフトウェア開発環境を構築する方法を 説明します.

組み込み開発では、Windowsパソコン上で動くプログラムを開発するのと異なり、開発するター ゲット機器に応じた開発環境を構築する必要があります. ここでは ARM マイコン (LPC2388) が実装さ れている基板をターゲットとして構築したソフトウェア開発環境を起動します.

## 2.1 VMware Player のインストール方法

ソフトウェア開発環境を起動するには、VMware Player がインストールされているパソコンが必要で す. インストール方法を次に説明します.



図 2.1 付属 DVD-ROM の 01\_VMwarePlayer フォルダ

リアルタイム OS を使って, LED を点滅させる

# プログラムをビルドする方法

 $\odot$ 

第

音

使用するプログラム 03 ARM 用 TOPPERSASP, 04 SampleProgram(asp)

杉本明加

本章では、TOPPERS/ASPの概要と、RTOS のプログラミングに必要な開発環境について解説 します. 数ある RTOS の中から、無償で使用で きる「TOPPERS/ASP (Advanced Standard Profile)」という OS を取り上げ、具体的に製品 に組み込むまでの手順を説明します.

TOPPERS とは、Toyohashi OPen Platform for Embedded Real-time System の略で、組み込 みリアルタイム・システム向けのソフトウェア・ プラットホームを開発するという意味です.

# 3.1 今回使用する RTOS 「TOPPERS/ASP」とは

# ● 組み込み向けリアルタイム OS の仕様 …µITRON

μITRON 4.0 仕様は, TRON (The Real-time Operating system Nucleus) 協会が策定した組み 込み向け RTOS の仕様です. 日本国内で使われて いる RTOS の中でも高いシェアを占めています. 利用されている分野は多種多様で, FA (Factory Automation) 機器や AV (Audio Visual) 機器をは じめ, 通信機器や運輸機器など, あらゆる組み込 みシステムに採用されています.

TOPPERS/ASP は、 $\mu$ ITRON (Micro Industrial TRON)の仕様に独自の拡張を行った OS です.したがって、厳密に  $\mu$ ITRON 4.0 仕様に準拠しているとはいえませんが、互換性は高くなっています.

#### ● TOPPERS プロジェクトが開発した

TOPPERS/ASP は, TOPPERS プロジェクト<sup>(2)</sup> で開発された RTOS の一つです. TOPPERS プロ ジェクトは,豊橋技術科学大学(当時)の高田研究 室が中心となって活動を始め、企業や大学の研究 室、各地の工業試験場と連携しながら発展してき ました.法人以外にも個人でプロジェクトに参加 し、開発成果をリリースしているユーザもいます. 代表的な成果物は RTOS です. 日本で使用さ

れる頻度の高い組み込みシステム向け OS である µITRON 4.0 仕様の RTOS と, 自動車システム 向けの OS である OSEK/VDX 仕様の OS を公開 しています.また,それ以外にも TCP/IP のプロ トコル・スタックやファイル・システムも公開し ています.その主な成果物を**表 3.1** と図 **3.1** に示 します.

開発したソフトウェアはオープン・ソースとし て公開されており,実際の製品に搭載されたり, 研究開発などでも利用されています.身近なとこ ろでは,携帯電話やプリンタ,自動車の制御用コ ンピュータなどに採用された例が報告されていま す.また,人工衛星にも搭載されています. 3

6

RTOS のプログラミング・テクニック①

# 時間に合わせた処理を 記述する方法

 $\odot$ 

第

使用するプログラム 04 Sample Program (asp)

峕

本章では,時間ベースの処理を記述する方法を 解説します. どのような処理になるのか,携帯電 話の時計機能を例に考えてみましょう.

携帯電話には時計が付いていて,最近では目覚 まし時計の代わりに携帯電話の目覚まし機能を 使って朝起きている人も多いでしょう.目覚まし がなかったら,早く目覚めるように習慣づけて, 起きる時間を時計を見て確認しなくてはなりませ ん.でも,目覚ましを使えば毎日正確な時刻に起 きられるわけです.

また,カップラーメンを作るときに,携帯電話 のストップウォッチ機能が役に立ちます.

普通の時計で図るのは面倒な上に, 分単位でし

か表示がないと1分程度ずれてしまうかもしれま せん. そんなときにストップウォッチ機能を使え ばきっかり何分たったかを数えて,特定の時間が 経過したことを通知してくれます.

杉本明加

Δ

6

9

両方に共通して言えることは、時間を人が確認 すると大変だということです(図4.1). それを携 帯電話の時計機能に任せれば、あなたは時計を頻 繁にチェックする作業から解放されます.

組み込みソフトウェアでも同じような状況は頻 繁に想定されます.定期的に実行しなければなら ない処理,また時折起こる特定時間を待つ処理を 行う機能が RTOS から提供されています.



図 4.1 RTOS に任せれば、時間管理から解放される

RTOS プログラミング・テクニック②

# タスク間でデータを やり取りする方法

 $\odot$ 

第

使用するプログラム 04\_SampleProgram(asp)

峕

本章では、タスク間でデータをやり取りする方 法を解説します。みなさんは普段どのような方法 を使って、ご家族、友人や仕事上のコミュニケー ションを取っているでしょうか?(図 5.1)

昔なら手紙や葉書でしたし、その後は固定電話 やFAXが普及しました.現在では携帯電話や E-mailやWebチャット、Skypeなどのビデオ・ チャットといったより便利な手段が普及していま す(表5.1).時と場合によって、これらのさまざ まな方法を使い分けているでしょう.プライベー トでは携帯メールで、急ぎのときや重要な話は電 話で、年末年始には年賀状といった使い分けをす るでしょう. 杉本明加

5

仕事であれば、伝える内容が残って複数人でや り取りができる E-mail を使い、文章では詳細が伝 わりづらいやり取りであれば電話や TV 会議など を用いるなど、工夫を凝らしているかと思います.

これは、それぞれのコミュニケーション方法に は一長一短があり、何を、誰に、いつ、どこから / どこにいる相手へ情報を伝えるかを判断して、 どの手段が有効かを判断してのことです。

RTOS の話に置き換えると、人がタスクで、コ ミュニケーション手段がデータをやり取りする方 法にあたります.ソフトウェアにおけるデータの やり取りにもさまざまな方法があり、それぞれ特 徴を持っています.タスク間でやり取りする情報



図 5.1 適切なコミュニケーション方法を選択する必要がある

RTOS プログラミング・テクニック③

# メモリゃ I/Oを管理する方法



第

使用するプログラム 04\_Sample Program (asp)

峕

本章ではメモリや I/O を管理する方法を解説 します.人気飲食店で食事をするためにあなたは 行列に並んでいます.食事を楽しみに待ちながら お店の中と店員さんの様子を見てみましょう.

店員さんは慌ただしく仕事をする一方, 食事を 終えそうなお客さんのようすをうかがっているよ うです.すると二人連れのお客さんが立ち上がり, 会計に向かいました.その後ほどなくしてテーブ ルが片付けられ,テーブルが1人用に並び替えら れた後,行列に並んでいた先頭の人とあなたがお 店の中に招き入れられました.

お店で食事できる人数には限りがあり、お店へ 入るお客さんの人数もバラバラで、いつお客さん 杉本明加

が来るかも決まってはいません。そういった状況 で店員さんはよりたくさんの人に早く食事をして もらえるよう,席を管理しています(図 6.1).お 客さん自身が好き勝手にテーブルに着いたら,店 員さんがいる時のようにはいかないでしょう.

組み込みソフトウェアに置き換えると,限られ た資源であるメモリやI/Oはお店の席数です. 当然大事に使わなければなりませんが,アプリ ケーション(お客さん)による管理では,システ ム構成が変わるたびに開発者がメモリやI/Oの 管理をアプリケーションで行わなくてはなりませ ん.RTOS(店員さん)に任せておけば,そんな煩 雑な管理もバッチリ行ってくれます.



図 6.1 店員 (RTOS) さんが、お客さんの数に合わせて席を作ってくれる

6

簡易タイム・レコーダの製作

複数の OS で動作する プログラムを作成する

使用するプログラム 04 SampleProgram(asp)

 $\bigcirc$ 

峕

これまでリアルタイム OS (RTOS) の基本的な 使い方について説明してきましたが、本章ではそ の集大成として実際に動くシステムを作成してみ ましょう、今まで扱ってきたプログラムに比べる と多少複雑ですが、順を追って説明していくので 十分理解できるものと思います。

現在では、開発される組み込みシステムが大規

杉山明加

模になり,一人でシステム全体を手がけることが 難しくなってきています.しかし,一つのシステ ムを作り上げることで,組み込みシステム開発の 全般を理解し,個々の技術要素への習熟度も高ま ります.したがって,たとえ小さなシステムで あっても一人で完成させるということは貴重な体 験になるでしょう.

# 7.1 作成するアプリケーションの概要

本章で作成するアプリケーションは, FeliCa カード・リーダを使用した簡易タイム・レコーダ です. FeliCa カード・リーダは, さまざまな IC カードからデータを読み出すことができます. 有 名なのは, Edy やおサイフケータイです. 今回は、FeliCaの機能のうち、FeliCaチップ に書き込まれた固有の ID 値 (IDm) を読み取る機 能を使った簡単なタイム・レコーダを実現します (図7.1). 実際, IC カードに対応したタイム・レ コーダは数多く開発されており、さらにタイム・



図 7.1 簡易タイム・レコーダの構成 FeliCa カードの情報を読み出し、パソコンに表示させる.

コンパイラやアセンブラ, リンカの動きを理解する

# ターゲット・ハードウェアで 実行できるコードを生成する方法

中村 建真

Δ

6

8

作成したプログラムを,目的のハードウェア上 で動作させるには,ソース・コードを CPU が実 行可能なコード(機械語)に変換する必要があり ます.さらに組み込み開発では,開発するシステ ムごとに CPU やシステム構成がそれぞれ異なる ため,その環境に合わせた実行コードを生成する ことが求められます.

音

それぞれのシステムに合わせて実行コードを生

成するには、ハードウェアの知識に加えて、開発 ツールの知識が強く求められます。例えば、「リ セット後の実行開始番地は 0x0000 番地である」 というのはハードウェアの知識になりますが、 「0x0000 番地にプログラムを配置する」という のはツールに関する知識になります。

本章では,開発に必要なツールであるコンパイ ラやアセンブラ,リンカについて解説します.

### 8.1 CPU が実行できるコードにコンパイルする

CやC++などの高級言語で記述されたソース・コードを機械語に変換して, CPU上で実行 可能なコードを生成する処理をコンパイルといい ます.そして,コンパイルするツールをコンパイ ラといいます.

#### ● x86 マシン用の実行ファイルを生成

まず,パソコン上の CPU でプログラムを動作さ せてみましょう. **リスト 8.1** は,C 言語の入門とし てよく見かける Hello プログラムです. これは組 み込み用のプログラムではなく, Windows や Linux といったパソコン上で,「Hello, CQ」をテ キスト出力する (ディスプレイに表示させる) プ ログラムです. このプログラムを, 今回の開発環 境である Ubuntu 上で動作させてみましょう. Ubuntu が動作するのは x86 マシンなので, x86 用のコンパイラを使います.

コンパイラには、GCC<sup>注1</sup>を使用します. これ

```
#include <stdio.h>
int main()
{
    printf("Hello, CQ¥n");
    return 0;
}
```

デバッガのシミュレーション機能を 使ってみよう

# GDB を使った プログラムのデバッグ

 $\odot$ 

使用するプログラム 05\_Sample Program (jsp)

作成するプログラムをデバッグするには,大き く分けて二つの方法があります.一つは,printf 文などを適切に配置して内部状態を探りながらデ バッグを行う方法です.もう一つは,デバッガを 使ってブレークポイントやステップ実行,変数の 監視を絡ませながら行う方法です.

デバッガを使った方が、簡単で効率的です.特 に、組み込み機器のようにハードウェアに近いと ころで動作するソフトウェアを開発する際は、デ バッガを使わずにプログラムを作成することは難 しいでしょう.そこで本章では、コマンド・ライ 中村 建真

ン・ デバッガである GDB<sup>注1</sup>を紹介します. GDB は単独でも使えるだけでなく, Eclipse と協 調させることで GUI (Graphical User Interface) デバッガとしても使用できます.

本章では, 第8章で使用した空の main 関数を 持つアプリケーション・プログラムを例題として GDB の使い方を説明します. 使用するプログラ ムは, 本書付属の DVD-ROM の /05\_Sample Program(jsp)/00\_empty の中に収録されてい ます.

### 9.1 GDB のシミュレーション機能を使う

まず、00\_emptyをシミュレータで動かしてみ ましょう、00\_emptyは、空のmain()をARMマ イコン基板に実装するように作成したプロジェク トです、シミュレータには、GDBのシミュレー ション機能を使います、そのため、使用するシ ミュレータが LPC2388 に対応しているかどうか 心配です、しかし、このプログラムでは、ARM コアとメモリの配置以外にハードウェアに依存す る部分がないため、シミュレータが対応していて もいなくても関係ありません.

GDBのシミュレーション機能には、ステップ 実行をはじめとするデバッグ用の機能が付いてい るため、プログラムの動作を観察できます.ちな みに、第8章で使用したシミュレーション・コマ ンドarm-none-eabi-runは実行のみで、ス テップ実行やブレークポイントの機能はありませ ん.そのため、今回のようなメッセージを出力し ないプログラムを実行させても意味がありません. 6

9

注1:デバッグはプログラムの不具合を探し出し、それを修正する作業のこと.また、デバッガはデバッグを支援するツー ルまたはソフトウェアのこと.ブレークポイントは、実行中のプログラムを中断させること、あるいは中断させる個 所を指す.ステップ実行は、プログラムを1行ずつ確認しながら実行させること.GDB (GNU Debugger)は、GCC と同様に Free Software Foundation (FSF) が配布しているフリー・ソフトウェア.

RTOS を移植するための準備作業①

# ARM マイコン LPC2388の 機能をチェックする

 $\odot$ 

第

使用するプログラム 05\_Sample Program (jsp)

TOPPERS/JSP を各種プロセッサに移植する には、TOPEPRS/JSP 自身とプロセッサのアー キテクチャ、そして開発ツールの知識が必要です. いずれも深い知識が必要になるため、新しいプロ セッサに一から OS を移植しようとするとかなり 大変です.

幸いなことに、ARM マイコン基板に搭載され ている LPC2388 の CPU コアである ARM7TDMI の依存部は、TOPPERS/JSP の公式コードとし て配布されています。したがって、LPC2388 へ の移植に関していえば CPU 依存部を移植する必 要はなく、システム依存部だけを移植すればよい 中村 建真

ことになります.システム依存部とは、タイマと シリアル・ポート、割り込みコントローラ、PLL などの機能や、スタートアップ時に行う処理のこ とです.

本章では、LPC2388のシステム依存部を確認 する方法を説明します.LPC2388の各機能につい てはユーザーズ・マニュアル(UM10211 LPC23xx User's Manual)で確認できますが、今回はマニュ アルに載っている各機能を実機でテストする方法 を解説します.使用するプログラムは、本書付属 のDVD-ROMの/05\_Sample Program (jsp) の中に収録されています.

### **10.1** ARM マイコン「LPC2388」の構造

LPC2388の構造は、図 10.1 に示すように ARM 7TDMI-S コア<sup>注1</sup>を中心として、複数のバスに 多数のペリフェラル(周辺機能)が配置されてい ます. バスの本数が複数あるのは、Ethernet MAC や USB OTG のようなバス・バンド幅が必要 なペリフェラルを内蔵しているためです.

これらはバス・ブリッジで接続されており, 互 いのトラフィックを妨げず, また CPU と自由にや りとりができます. ブリッジはプログラマからは 見えず, すべてのペリフェラルが一つのアドレス 空間にマップされているため, プログラミング・ モデルは単純です.

ペリフェラル・レジスタは、すべてメモリ・ア ドレス空間にマッピングされています.つまり、 C 言語からポインタを使ってペリフェラル・レジ スタにアクセスできます.

例えば、インクルード・ファイルで以下の宣言 をしておけば、レジスタへ変数のようにアクセス 6

注1: ARM7TDMI-S は、ARM7コアに、T(16ビット固定長の Thumb 命令を採用)、D(オンチップ Debug 機能)、M(ハードウェアによる乗算機能)、I (組み込み ICE 機能) という四つの機能が追加された CPU を意味する.また、-S はソフト・マクロにより論理合成されたデバイスであることを示す.

ステップアップで移植の手順を理解する

# ARM マイコン LPC2388 に RTOS を移植してみよう

 $\odot$ 

第

使用するプログラム 05\_SampleProgram(jsp)

本章では、LPC2388 に TOPPERS/JSP のシス テム依存部を移植する方法を解説します. CPU 依存部については移植済みコードがすでに存在す るので、システム依存部の移植は比較的容易な移 植作業といえます.

しかし、もともと RTOS の移植は複雑な作業 である上、TOPPERS プロジェクトから移植作業 のためのスタブやテスト・ドライバなどが提供さ れていません。

そこで,移植作業を少しでも楽にするために, 第10章で紹介した試験プログラムやすでに存在 中村 建真

するモジュールなどを活用して、システム依存部 の移植を進めることにします.

ここでは、LPC2388 用 TOPPERS/JSP の SRAM 上で動作する版とフラッシュ ROM 上で動作する 版を作ります.いずれも本書の解説用であり、実 アプリケーションで使用することはお勧めしませ んが、学習するには十分です.

TOPPERS/JSP のソース・ ツリーのルートは 通常は"jsp"にしますが、すでに存在するリリー ス版との混同を避けるために、ここでは"jsp\_ study"とします.

### 11.1 移植に必要な変更点を確認する

今回の移植に際して,各段階における状況を示 すスナップ・ショットは、本書に付属するDVD-ROMの中にCVSレポジトリとして収録してあ ります.各節のタイトルに対応するスナップ・ ショットのCVSタグ名を表記してあるので、 CVSクライアントを使って変更点を追いかけて ください.それぞれの版の差分を取れば変更点が わかります.

CVSは、UNIX で使われているバージョン管 理システムで、オンラインのサーバにソース・ コードを格納し,作業の履歴を保管するものです. 今回は,移植の過程を CVS に記録しながらサン プル・コードを作りました.

サンプル・コードの各スナップ・ショットは, シェルから CVS コマンドで, あるいは Eclipse の CVS 機能でも取得できます.

Eclipse の CVS 機能では, 任意のタグの版の ソース・ツリーを簡単に取得できるほか, ほかの タグとのソース比較もできるので活用されること をお勧めします (図 11.1).

### 簡単なアプリケーションを動かしてみよう

# RTOS で LED を 光らせる 初めの一歩

使用するプログラム 05 Sample Program (jsp)

第

 $\bigcirc$ 

中村 建真

# 12.1 簡単なプログラムをビルドする

● TOPPERS/JSP のサンプル・プログラム TOPPERS/JSP には、あらかじめ単純なサン プル・プログラム Samplel が用意されています. これは、キャラクタ端末を通してユーザと対話す るプログラムです.ユーザは、このアプリケーショ ンにコマンドを送ることにより、TOPPERS/JSP の機能を体験することができます.自分自身でア プリケーションを開発する手始めとして、このプ ログラムをビルドして動かしてみましょう.

Samplel アプリケーションをビルドするために, アプリケーションのディレクトリを用意します. 図 12.1 に, ディレクトリ構成を示します. 親ディ レクトリが workspace になっていますが, この ディレクトリは好きな名前でかまいません. ここに 移植を行った jsp\_study を置きます. そして, jsp\_study と並んで新しく sample1 ディレクト リを作ります. studyの中に降りて,コンフィギュレータをビル ドします.ビルドが成功したら,今度は sample1 の中へ移動して,サンプル・プログラム Sample1 を作ります.

この一連のコマンドの流れを図 12.2 に示しま す. このコマンドの中で呼び出されている ../ jsp\_study/configure は, コンフィギュレー タを呼び出すスクリプトです. パラメータの -C と -S は, それぞれ CPU 依存部とシステム依存 部を指定しています. このスクリプトの使い方の 詳しい説明は, jsp\_study/doc/user.txtを 参照してください.

注意深く見ればわかることですが, configure スクリプトのパラメータはターゲットとなるシス テムを指定しているだけで, アプリケーションの ソース・コードを一切指定していません. この場 合, configure スクリプトは TOPPERS/JSP の ソース・ツリーに含まれている Sample1 アプリ

次に,	TOPPERS/JSP のソース・	ツリー	jsp
-----	-------------------	-----	-----

workspace	
jsp_study	
sample1	

図 12.1 アプリケーション Sample1 のディレクトリ構成



#### このPDFは、CQ出版社発売の「インターフェースZERO No.03」の一部見本です.

内容・購入方法などにつきましては以下のホームページをご覧下さい。

- 内容 http://shop.cqpub.co.jp/hanbai/books/MIF/MIFZ201209.htm
- 購入方法 http://www.cqpub.co.jp/order.htm

定価2,520円 本体2,400円