

KOZOS の ARM 系プロセッサへの移植

ささのたかよし (@uaa)

概要

「12ステップで作る
組込み OS 自作入門」で使われた
H8 用 KOZOS を、
Interface 誌 2009 年 5 月号付録の
ARM 基板に移植し、動作させた。

動作デモ

- H8 用と同じように動くはず

移植元 / 移植先の SoC

- H8/3069F (Renesas)
 - 20MHz の H8/300H コア
 - ROM 512KB, RAM 16KB
- LPC2388 (NXP)
 - 72MHz の ARM7TDMI コア
 - ROM 512KB, RAM 64KB

KOZOS の構成

- ブートローダ
 - ボードの初期化
 - シリアルポートから XMODEM で OS 本体をロード
- OS 本体
 - シリアルポートを使用したコマンド処理
 - OS の各種サービス

機種に依存しない部分

- ブートローダ
 - コマンド処理
 - XMODEM 処理
 - ロード後処理
- OS 本体
 - コマンド処理
 - OS の各種サービス

機種に依存する部分

- ブートローダ
 - ボード初期化
 - シリアルドライバ
- OS 本体
 - 割り込みサービス
 - タスクディスパッチャ
 - シリアルドライバ

CPU の違い

- 命令セット
- レジスタの本数
- 割り込み / 例外の処理方法
- CPU の動作モード

周辺ペリフェラルの違い

- 何もかもが違う
 - 割り込みコントローラ
 - シリアルコントローラ
 - メモリ配置

移植における厄介な点

- ディスパッチャの作成
- オブジェクトサイズの問題
- ライセンスの問題

KOZOS のディスパッチャ

- KOZOS の要となる部分

```
void dispatch(kz_conext *context)
```

- `context` に保存されたタスクコンテキストを復帰する
- タスクコンテキストの内容
 - CPU レジスタ
 - 汎用レジスタ・ステータスフラグ
 - PC(Program Counter)

	ARM	H8/300H
SP+0	<i>SPSR_xxx</i>	ER0
SP+4	<i>R0</i>	ER1
SP+8	<i>R1</i>	ER2
SP+12	<i>R14_xxx(PC)</i>	ER3
SP+16	R2	ER4
SP+20	R3	ER5
SP+24	R4	ER6
SP+28	R5	CCR:PC
SP+32	R6	
SP+36	R7	
SP+40	R8	
SP+44	R9	
SP+48	R10	
SP+52	R11	
SP+56	R12	
SP+60	R14_sys	

※R13:SP

※R7:SP

H8 プロセッサの特徴

- CPU に動作モードの概念が無い
- レジスタバンクは無い
- 復帰アドレス・ステータスレジスタはスタックへ

ARM プロセッサの特徴

- CPU は複数の動作モードを持つ
 - USR/SYS, SVC, IRQ, etc...
- CPU の動作モードによってレジスタバンクが切り替わる
 - R13_xxx: スタックポインタ
 - R14_xxx: リンクレジスタ
- 復帰アドレスはリンクレジスタへ
ステータスレジスタは SPSR(Saved PSR) へ

オブジェクトサイズの問題

- kozos.elf は 44KB 程度
 - OS 本体 : 8KB
 - 残りは全部ゴミ (0x00)
- ブートローダのバッファは 32KB
- Sレコード形式を使用することでこの問題を回避

ライセンスの問題

- ~~μT-Kernel~~ を ~~LPC2388~~ に移植した際に書いたコードを流用
 - 移植元 / 先は μT-Kernel ライセンス
 - 修正差分は public domain
- 修正差分が μT-Kernel ライセンスに汚染される危険性
 - 「大丈夫だ、問題ない」

まとめ

- 実は辻褃さえ合えばどうにでもなります
- 移植元 / 移植先の両方を知ることが大事
- LPC2388 版 KOZOS はこちらで配布中
 - <http://bit.ly/d49Lna>

御清聴ありがとうございました