

RISC-Vアセンブラで読む リアルタイムOSディスパッチャ

ひばり

アセンブラってなに？

- アセンブラ（アセンブリ言語）は、CPUが直接実行する機械語命令（命令コード）とおおよそ1:1に変換できるプログラミング言語
- ソフトの一部はアセンブラでしか書けないため隅から隅まで理解しようとする必要になる
- 高級言語（CやPythonなどのプログラミング言語）と異なりCPUアーキテクチャが異なればアセンブラの命令セットも大きく異なる
→複雑なCPUを選んでしまうとアセンブラも難しくなる

```
.globl trap_entry
.type trap_entry,
@function
.align 2
trap_entry:
    addi sp, sp, -REGBYTES
```

アセンブラの用途の例

- スタートアップルーチン
 - 高級言語で作られたコードを実行開始するまでの準備
- OSの一機能であるディスパッチャ
 - CPUが多くのタスクを同時並行的に実行しているときに実行中のタスクを止めて別のタスクへ切り替える処理
→今回はここに注目する
- 高級言語で書いてコンパイラに任せるより自分でアセンブラで書いた方が高速な処理が組める場合

OSのディスパッチャがやること

- いままで動いていたタスクの途中経過を記憶（待避）する
- 途中経過を復帰することで中断していたタスクを再開する
 - あるタスクからOSが呼ばれた（例えばタスク起動）ことでタスクの切り替えが必要になったときや、一定時間経過をタイマ割り込みで通知されたときに動く
 - 各タスクの途中経過の多くはメモリ（タスクごとのスタック領域を含む）に入っているが、CPU内のレジスタ（全タスクで共用する）に入っている途中経過もあるためディスパッチャは主にCPUのレジスタの値を待避・復帰する
 - 今回の例では待避先はタスク毎に用意されたスタック領域

組み込みOSの実装をとりあげた資料

- 坂井弘亮著「12ステップで作る 組込みOS自作入門」が有名
 - 2010年出版、ターゲットはH8/3069F
 - この本で作るOSを動かせるボードは現在も秋月電子（秋葉原）で購入可能
- でももう少し新しいCPUを実例にしてOS実装が知りたい
- 仕様が細かいところまで公開されているCPUがよい
- RISC-V用のOSを読んでみた！

とりあげたRISC-V搭載ボード

- Sipeed製Maix M1 Dock
 - DockのほかにMaixduinoやMaix BitなどもCPUまわりは同じ
 - スイッチサイエンスやShigezone（秋葉原）で3千円前後
 - CPUまわりはKendryte K210を利用
 - 今回は使わないが以下の演算回路も搭載している
 - KPU（ニューラルネットワーク処理）
……画像認識などNNを使った人工知能に使える
 - FFT（高速フーリエ変換）
 - APU（多チャンネル音声信号処理）

Sipeed Maix M1 Dock

Sipeed M1

Kendryte K210

デュアルコア
RISC-V CPU

KPU

FFT

APU

SRAM

タイマ

カメラ入力

LCD出力

各種I/O

電源

フラッシュ
ROM

USBシリアル

マイク

DAC・スピーカ

LED

スイッチ

各種コネクタ

K210のCPUやメモリ周り

- デュアルコア64ビットRISC-V
 - 命令セットはRV64GC (RV64IMAFDC)
 - I：基本命令セット
 - M：乗算および除算
 - A：アトミック命令
 - F：単精度浮動小数点
 - D：倍精度浮動小数点
 - C：圧縮命令
 - 動作周波数は400MHz……最近はこれでも非力と言われてしまう
 - 割り込みコントローラはPLICとCLINT（後述）
- RAMは8MB（汎用6MB+AI用2MB）……価格のわりに多い

RV64GC仕様のRISC-V CPUがもつレジスタ

- レジスタとは、大容量のRAMよりも高速に読み書きできるCPU内部の小さなメモリ領域のこと
- ① 汎用レジスタx0~x31
 - タスクの途中経過が入るのは30個
 - x0は0固定、x3は立ち上げ時に値を決めて変更しない
- ② 浮動小数点レジスタf0~f31
 - こちらは32個すべてにタスクの途中経過が入る
- ③ コントロールアンドステータスレジスタ
 - ほとんどはOSのみが使用する
 - mepc（割り込み発生前はどこを実行していたか）とfcsr（浮動小数点演算の設定）にはタスクの途中経過が入る

読んでみたOS

- Kendryte K210 FreeRTOS SDKに含まれるFreeRTOS
 - FreeRTOSとは
 - オープンソースの組み込みOSのひとつ
 - さまざまなCPUに移植されている
 - K210向けFreeRTOSはCPUがデュアルコアであることにあわせてOSの一部APIにプロセッサIDを指定可能にするような拡張が入っている

RISC-V用OSを読むときの注意点

- RISC-V仕様をどこまで実装しているかはさまざま
 - 32bitなのか64bitなのかでCPUレジスタのサイズ変わる
 - 命令セットをどこまで実装しているかで
ディスパッチャで扱うCPUレジスタの数が変わる
 - どの割り込みコントローラを使うかもさまざま
- 「どのRISC-Vでも動くOS」は難しい
ここから先はあくまでもK210の場合の事例

読んでみた結果

- タイマ割り込みにはじまる
ディスパッチ処理アセンブラコードで使われている
アセンブラ命令はたったの21種類
- 以下の2つの資料のレジスタ一覧と命令仕様を見れば
ほとんどの処理は理解できた
 - Patterson氏, Waterman氏「RISC-V原典」
 - しばっち氏訳「RISC-V命令セットマニュアル」
 - どちらも日本語で読めるのが嬉しい

読んでいてちょっと難しかったところ

- M1やK210のドキュメントには
割り込みコントローラCLINTに関するメモリマップが
載っていない
 - CLINTのタイマ割り込みを使っていた
 - CLINTで自分自身にソフトウェア割り込みを入れていた
 - CLINTはRISC-Vの仕様に載っていない
- 同様にCLINTを使う64ビットRISC-Vである
SiFive製Freedom U540の資料が詳しかったので
そちらを参考にした

読んでいてちょっと難しかったところ

- アセンブラコードに「frsr」命令と「fssr」命令があるが命令仕様を探してもそんな命令は見当たらない
→2013年の仕様書の改訂履歴と新しい仕様書の改訂履歴を比較したところ、現在のfrcsrとfscsrであることが分かった

まとめ

- CPU仕様に関する資料が入手しやすいため、
「C言語は分かるけどアセンブラになると分からない」
という方へのアセンブラ入門としても
RISC-Vはおすすめできる
→ソフト設計者がCPUを学ぶ題材になる
- CLINTに関する部分を知るには
RISC-Vの一実装であるSiFive製コアのマニュアルに
頼らないといけない
→このあたりの仕様の標準が定まると
RISC-V向けソフトを増えていきそう