

TCP/IP プロトコル・スタックの内部処理

坂井 弘亮

TCP/IPプロトコル・スタックはユーザのアプリケーション・ソフトウェアとEthernetデバイス・ドライバの間の処理を行います。

既存のものをそのまま使うのが主流ですが、内部の構造や処理を理解しておけば、ネットワークの通信速度が上がらない場合やトラブルがあった場合でも原因が推測しやすくなります。

そこで本稿では、筆者オリジナルの学習用マイコンOS「KOZOS」のTCP/IPプロトコル・スタックを題材に、内部構造や各タスクの役割分担、実際のプログラムを紹介します。

体について主に解説します。

記事の最後では、TCP/IPプロトコル・スタックの内部処理を理解したところで、ブートローダから動かせるように改造します。次章で紹介する「軽量ファイル転送ソフトTFTPでマイコンのネットワーク起動を試す」の準備になります。

本稿で解説する範囲

今回題材にするTCP/IPプロトコル・スタックは図1のような内部構造になっています。Ethernet, IP, TCPと役割によってプログラムが分かれています。

このうち、ethernetタスク、ipタスク、tcpタスクと、それらの間でやりとりするデータ・パケット格納用の構造

表1 実装されている機能

分類	機能
起動	ブートローダ (シリアル/ネットワークによるファームウェアのロードとELF形式の実行ファイルの起動が可能)
管理	マルチタスク管理, メモリ管理, 割り込み管理機能
	シリアル通信 タイマ管理
ネットワーク	EthernetとTCP/IP通信 Webサーバ, TFTPクライアント
デバッグ	GDBによるリモート・デバッグ シミュレータによるPC上での動作
その他	ROM化

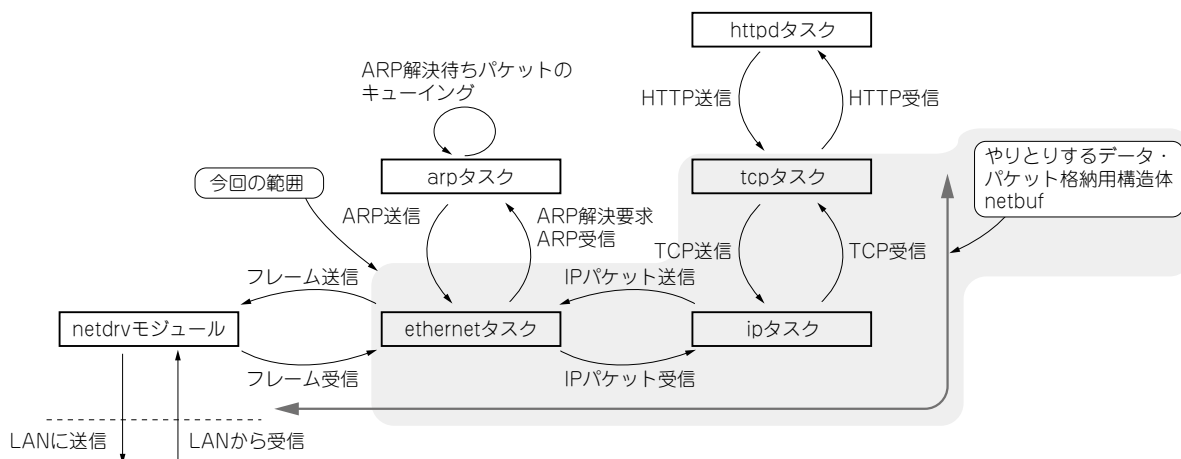


図1 本稿で解説するプログラム
HTTP通信のためのタスクの構成