ハードを隠ぺいしたお任せOSを組み込みで使う可能性を探る

実験リサーチ！ Linux応答時間の実力

カーネル内部とことん可視化計画

第6回 先に立ち上げてスリープが常識… プロセスの起動にかかる時間を調べる

海老原 祐太郎

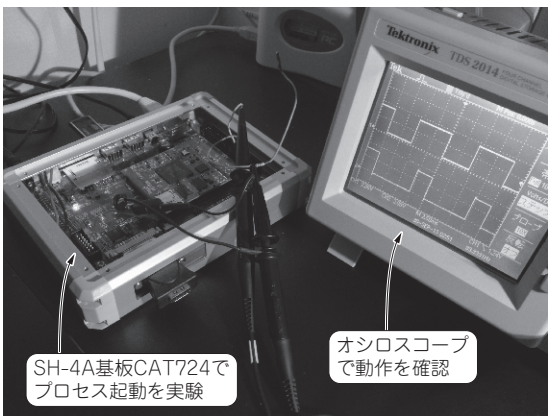

写真1　プロセス起動の実験を行いオシロスコープで測定した

今回は，Linuxのプロセスを実行するとどのくらいの処理時間が必要かを調べてみます．

● プロセス実行の3ステップ

プロセスの実行は，以下の3ステップで行います．

▶ステップ1：**mmap**

プロセスは，バイナリ・ファイルをmmap()することでメモリに貼り付け（マッピング）られます．これが親プロセスになります．

▶ステップ2：**fork()**

fork()システム・コールで親プロセスのメモリ・マップを複製して子プロセスを起動します．fork()直後の子プロセスは親プロセスのコピーで，物理メモリ・ページは共有しています．

▶ステップ3：**exec()**

exec()システム・コールで新しいプロセスのバイナリ・ファイルをmmap()します．

ステップ2までは仮想メモリに領域を確保するだけなので処理時間は短くて済みますが，exec()ではファイルI/Oが発生します．プロセスの起動に時間がかかるのはこれが理由です．

システムが複数のプロセスに分かれる場合，なんらかのイベント要求があってからプロセスを起動させると遅延が発生します．そのため最初にプロセスを起動しておき要求があるまではスリープさせておきます．これをpre-forkと呼びます．例えばWebサーバなどではCGIのプロセスを事前にpre-forkして応答性を改善します．

実験内容

● プロセスの起動時間を測ってみる

Linuxでは新しいコマンドを実行する場合はfork()して子プロセスでexec()します．exec()は新しいファイルを実行し現在のプロセスを置き換えます．

今回は，このfork()とexec()の実行時間を測ります．fork()時点では物理的なメモリ・コピーなどは発生しないため比較的高速です．一方でexec()はファイルI/Oが発生します．具体的には新しい実行ファイルをmmap()でメモリに貼り付けます．

今回は，以下の内容でプロセスの実行ファイルがメモリのどこにマッピングされているか，子プロセスの処理時間がどのくらいかを2段階で実験します．実験のようすを**写真1**に示します．

- 実験1-1…mmmap()で貼り付けたメモリの中身を見る
- 実験1-2…仮想メモリ・アドレスから物理メモリ・アドレスを求める
- 実験1-3…物理メモリの中身を覗いてみる
- 実験1-4…プロセスがmmap()でどこに貼りついているかを調べる
- 実験2-1…fork()で子プロセスにコピーされた中身を調べる
- 実験2-2　子プロセスの実行時間を調べる

● 実験に使うハードウェア

実験に使用するボードは，SH-4A（ルネサス エレクトロニクス）を搭載したCAT724＆EB724Aです．PCやARMといった他のアーキテクチャでもほぼ同様です．カーネルのバージョンは3.0.4です．本稿で紹介

第1回　電源投入からカーネルが起動するまでの動作（2013年11月号）
第2回　ハード制御で需要！割り込み処理を時間で終わらせる三つのしくみ（2013年12月号）
第3回　マルチタスクOSのキモ！タスク切り替えの基本動作（2014年1月号）