

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!

カーネル内部
とことん可視化計画

Linux 応答時間の実力

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 海老原 祐太郎

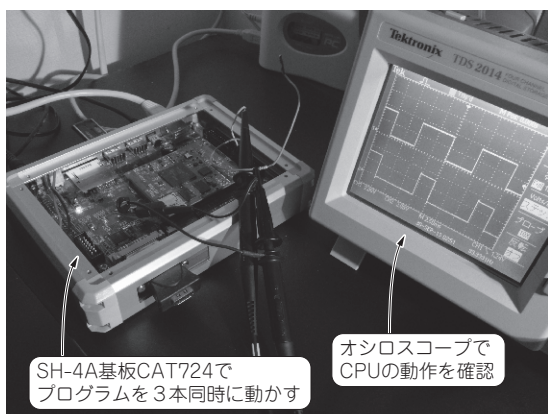


写真1 実験…まったく同じプログラムを3本同時に動かしてタスク切り替えを見てみる

今月はマルチタスク OS のキモであるタスク切り替え(コンテキスト・スイッチ)を実験します。

- **実験1:** 周期10msごとにタスク切り替えが行われることを確認するタイマ割り込み。何もせずに無限ループだけを行うプログラムを作り、3本同時に動かす
- **実験2:** USBメモリからデータを16Kバイト読み出し/読み捨てプログラムを作り、I/O負荷をかけながら3本同時に動かす。不定期的な割り込み要因などがあるとタスクはバチバチ切り替わる
- **実験3:** タスク切り替えのタイマ割り込み周期10msが変更できることを確認する。実験1と同様の実験を行う
実験中のようすを写真1に示します。

● 実験用ハードウェアはSHボード

実験に使用するボードは前回同様にSH-4A(ルネサス エレクトロニクス)を搭載した写真1のCAT724です。SH-4Aを題材としますが、PCアーキテクチャやARMといった他のアーキテクチャでもほぼ同様です。アーキテクチャによって異なる部分は注記します。カーネルのバージョンは3.0.4です。

リスト1 メモリ・マップ確認用プログラム…hello.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

char buffer[256]; /* .bss */

int main(){
    int x=0;
    char *p;
    printf("hello world. pid=%d\n",getpid());

    p = malloc(1024*1024); /* 1Mバイト */

    while(1){
        printf("main=%08lx, printf=%08lx, x=%08lx,
                malloc=%08lx, buffer=%08lx\n",
                main, printf, &x, p, buffer);
        sleep(1); /* 1秒スリープ */
        /* 無限ループ */
    }
    free(p);
    return 0;
}
```

予備実験

Linuxのメモリ・マップを可視化

● 超シンプルなプログラムでメモリ配置を確認

とても簡単なプログラムを使って、Linuxのメモリ・マップ(配置)を見てみましょう。

▶ ステップ1: hello.cプログラムを作成

リスト1のhello.cを使います。hello.cは、

- main() しかない
- printf() を呼び出している
- 広域変数として256バイトのbuffer[] 領域がある
- malloc() で1Mバイトのメモリを確保している
- int xはローカル変数
- 各変数や関数のアドレスを表示する
- プログラムは1秒のスリープをはさんで終了しない

開発用のパソコンで以下のコマンドでクロスコンパイルを行います。

```
$ sh4-linux-gnu-gcc hello.c -O2 -o hello
```

生成されたファイルの各セクション・ヘッダの抜粋