

## 2010 年度 New グループ新 B4 課題

牛尾 裕  
平成 22 年 4 月 5 日

### 1 はじめに

本資料は，2010 年度の New グループ新 B4 への課題について説明する．

### 2 課題一覧と新人研修の流れ

新人研修は以下の流れで行う．新 B4 は以下の課題をこなすこと．

(課題 1) 電卓プログラムと仕様書の作成

(課題 2) Linux カーネルのコンパイルとインストール

(課題 3) Linux カーネルへのシステムコールの実装

< x86 CPU についての勉強 >

(課題 4) インラインアセンブリの利用

(課題 5) 仮想アドレス 物理アドレスへの変換

< x86 CPU についての講習 >

### 3 期限

各課題について自身で設定し，New グループ宛てにメールする．最終期限は，5 月の連休明け (5 月 6 日) までとする．

### 4 実験環境

実機上で作成する．実験用計算機は，202 号室の DELL マシンを 1 人 1 台に割り当ててるのでそれを利用する．また，使用 OS は NewOS 実験機と同じ Fedora 10 とする．各自インストールディスクからインストールすること．

## 5 各課題の詳細

課題 1 は、プログラムと仕様書完成後、各自指導教員にみていただくこと。課題 2 と課題 3 は、手順書を作成し、New 打ち合わせに提出すること。課題 4 と課題 5 は、動作を先輩に確認してもらうこと。

### (課題 1) 電卓プログラムと仕様書の作成

< プログラムについて >

printf とシステムコール以外は使用禁止とする。

< 仕様書について >

作成したプログラムについての仕様書を作成する。仕様書には以下の内容を記述すると良い。

- (1) 概要
- (2) 機能
- (3) 動作環境
- (4) 使用方法
- (5) エラー処理と保証しない動作

### (課題 2) Linux カーネルのコンパイルとインストール

### (課題 3) Linux カーネルへのシステムコールの実装

使用する Linux カーネルは、NewOS と同じ Linux kernel 2.6.31 とする。公式サイトからダウンロードし使用すること。

なお、課題 3 について、作成したシステムコールに対して glibc を適応させる作業は必須としない。必要に応じて各自で行うこと。

### (課題 4) インラインアセンブリの利用

以下の 2 つのプログラムを作成すること。

--- 1 つ目 ---

```
int main(void)
{
```

```
    int a, b, c = 0;
    int *p = &c;
```

```
    a = 1;
    b = 2;
```

```
    /*****
```

```
    * (拡張) インラインアセンブリを用いて, "*p = a + b"を行う.
```

```

*****/

printf("*p = %d\n", *p);
}

--- 2 目 ---
int main(void)
{
    int a[3] = {1, 2, 3};
    int b = -1;

    /*****
     * (拡張) インラインアセンブリを用いて, "b = a[1]"を行う .
     *****/

    printf("b = %d\n", b);
}

```

#### (課題 5) 仮想アドレスから物理アドレスへの変換

以下のプログラムを作成すること .

```

--- システムコール部 ---
unsigned int user_syscall(int *virt)
{
    unsigned int phys;

    /*****
     * 'virt' の示す (仮想) アドレスを物理アドレスに変換する .
     * 変換結果は, 'phys' に格納する .
     *****/

    /*****
     * 'phys' に格納された物理アドレスをカーネル空間における
     * 仮想アドレスに変換する (注: 変換不可能な場合もある) .
     * 変換後, 変換した仮想アドレスにアクセスし, int 型の値を得る .
     * 得た値を printk により出力する .
     * (printk は, カーネル内で利用できる printf のようなもの .
     *   ただし, メッセージの出力先が異なる)
     *****/
}

```

```

        return phys;
    }

    --- アプリケーション部 ---
    int main(void)
    {
        int a;
        unsigned int phys;

        /*****
        * ユーザ AP で定義した変数 'a' の (仮想) アドレスをシステムコールに
        * 引数として渡し, 返り値から物理アドレスへの変換結果を受け取る .
        * その後, 結果確認のために, 'a' のアドレスと 'phys' の値を 16 進数で
        * 画面に出力する .
        *****/

        return 0;
    }

```

## 6 Tips

課題を解くために知っておくべきことを以下に示す .

- (1) インラインアセンブリについては, [1] を参考にすること .
- (2) x86 CPU における仮想メモリ空間は, 4GByte(32bit) である . 64bit ではまた別 .
- (3) x86 CPU におけるアドレス変換の仕組みについては, 「IA-32 インテル R アーキテクチャ ソフトウェア・デベロッパーズ・マニュアル (下巻)」の「3.7. 32 ビット物理アドレス指定を使用したページ変換」の節を参照すること . (マニュアルは [2] に置いてある)
- (4) Linux における仮想メモリは, 0-3Gbyte までがユーザ空間, 3-4Gbyte がカーネル空間である .
- (5) Linux におけるユーザ空間のアドレス変換は, 4K バイトページングである .
- (6) Linux のカーネル空間において, 3Gbyte ~ 3Gbyte+最大 896MB の領域は, 物理メモリの 0 ~ 最大 896MB の領域へストレートマッピングされている .
- (7) アドレス変換における多くのフラグは無視して構わない . (ユーザ空間の仮想アドレスは, 必ず 4K バイトページングであるとして良い ページディレクトリのエントリからページングタイプを判断する処理は不要) (ただし, どんなフラグがあるのかは確認しておくこと)
- (8) 仮想アドレス リニアアドレス
- (9) 本資料だけでは全てを補足しきれないため, 分からないことは先輩に聞く .

## 参考文献

- [1] “gcc のインラインアセンブリに関して , ” [http://www.mars.sannet.ne.jp/sci10/on\\_gcc\\_asm.html](http://www.mars.sannet.ne.jp/sci10/on_gcc_asm.html)
- [2] “TwinOS 開発支援ページ , ” <http://newgroup.swlab.cs.okayama-u.ac.jp/TwinOS/index.html>