

課題 No.6 の解答例

1. sleep システムコールと同じ働きをする mysleep 関数

```
//
// mysleep.c
// (テストドライバ(myslpmain.c)は出題時に公開済みです)
//
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include "mysleep.h"

static void alHandler(int n) { }           // 必要以上に見えないように static

unsigned int mysleep(unsigned int seconds) {
    if (seconds==0) return 0;              // alarm(0)は予約解除(誤動作防止)
    sig_t org = signal(SIGALRM, alHandler); // 以前のハンドラを保存
    alarm(seconds);                        // SIGALRMを予約
    pause();                               // シグナルを待つ
    unsigned int r = alarm(0);              // SIGALRMの予約を確実に解除
    signal(SIGALRM, org);                  // ハンドラを復元する
    return r;                              // 残り時間を返す
}

/* コンパイル結果, テスト結果, 本物との違いを書く
% make
cc -Wall -std=c99 -o myslpmain myslpmain.c mysleep.c <-- エラー・警告なし
% ./myslpmain
0秒スリープ開始
スリープ終了(残り時間0秒)                <-- 0秒スリープ動作確認OK
1秒スリープ開始
スリープ終了(残り時間0秒)
2秒スリープ開始
スリープ終了(残り時間0秒)                <-- 0秒以外のスリープもOK
3秒スリープ開始
スリープ終了(残り時間0秒)
これから10秒スリープします.
その間に ^Cを入力すると残り時間を表示して終了します.
^C                                         <-- シグナルで終了した場合でも
残り時間5秒                               残り時間を正しく返している
%
バグ: alarm を使用中に mysleep を使用するとタイマーが解除される
*/
```