情報処理 2 - 前期第 10 回課題

柴田健琉

提出日：2026 年 06 月 29 日
2026 年 06 月 23 日

1　はじめに

この課題のプログラムは以下の環境での動作が確認されている:

- OS: Arch Linux, Linux 6.18.36-1-lts x86_64
- CC: GCC 15.2.0
- CFLAGS: `-g -O1 -Wall -Wpedantic`

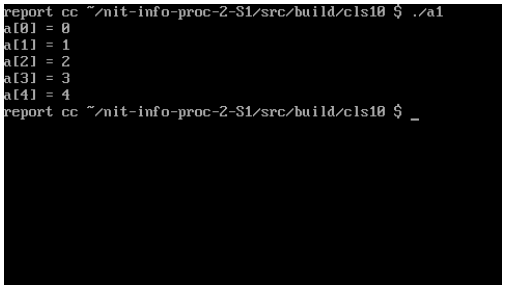
2　課題 1

教科書の演習 10-4 にて整数型配列の内容を配列の添字と同じにする関数 `set_idx` 関数を作成する.

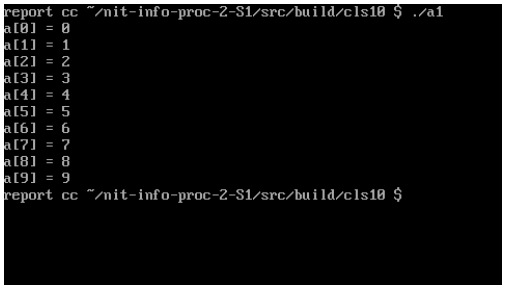
課題 1 のプログラム

```c
#include <stdio.h>

#define A_LEN 5

void set_idx(int *v, int sz) {
    for (int i = 0; i < sz; i++) {
        *(v + i) = i;
    }
}

int main(void) {
    int a[A_LEN];
    set_idx(a, A_LEN);
    for (int i = 0; i < A_LEN; i++) {
        printf("a[%d] = %d\n", i, a[i]);
    }

    return 0;
}
```

2.1 実行結果



(a) A_LEN = 5



(b) A_LEN = 10

図1: 課題 1 の実行結果

2.2 処理の流れ

2.2.1 main 関数

1. 要素数 10 の配列 a を初期化
2. set_idx 関数を引数に配列 a と要素数を設定して呼び出す
3. 配列 a の各要素を標準出力に表示する

2.2.2 set_idx 関数

1. ループカウンタ i を 0 に初期化
2. ポインタ v から i * 4 バイトを加算したメモリアドレスにループカウンタ i を書き込む
3. ループカウンタ i をインクリメントする
4. ループカウンタ i が sz になるまで 2 を実行

2.2.3 main 関数

1. 配列 a を A_LEN の長さで宣言
2. set_idx 関数を引数に a と A_LEN を持たせて呼び出す
3. 配列の内容を標準出力に表示する

2.3 書く際の難点

とくに問題なく素直に書けた.

2.4　想定されるバグ

ヌルポインタの検出がないので引数 v にヌルポインタを渡すとヌルポインタ参照でセグメンテーションフォルトを引き起し, プログラムがクラッシュする.

3　課題 2

配列を昇順に並び換えるプログラム List 8-5 の `bsort` 関数をポインタで書き換える.

課題 2 のプログラム

```c
#include <stdio.h>

#define ARR_LEN 10

void bsort(int *v, int n) {
    int t = 0;
    for (int i = 0; i < n - 1; i++) {
        for (int j = n - 1; j > i; j--) {
            if (*(v + j - 1) > *(v + j)) {
                t = *(v + j);
                *(v + j) = *(v + j - 1);
                *(v + j - 1) = t;
            }
        }
    }
}

int main(void) {
    int a[ARR_LEN] = {0};

    printf("Input␣%d␣integer␣entries:\n", ARR_LEN);
    for (int i = 0; i < ARR_LEN; i++) {
        printf("#%d:␣", i+1);
        scanf("%d", &(a[i]));
    }

    bsort(a, ARR_LEN);

    printf("\nResult:\n");

    for (int i = 0; i < ARR_LEN; i++) {
        printf("a[%d]␣=␣%d\n", i, a[i]);
    }

    return 0;
}
```

3.1 実行結果

```
report cc ~/nit-info-proc-2-S1/src/build/cls10 $ ./a2
Input 10 integer entries:
#1: 9
#2: 4
#3: 6
#4: 8
#5: 7
#6: 1
#7: 3
#8: 2
#9: 10
#10: 5

Result:
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
a[4] = 5
a[5] = 6
a[6] = 7
a[7] = 8
a[8] = 9
a[9] = 10
report cc ~/nit-info-proc-2-S1/src/build/cls10 $ ./a2
Input 10 integer entries:
#1: 366
#2: 234
#3: 974
#4: 278
#5: 437
#6: 379
#7: 469
#8: 879
#9: 137
#10: 352

Result:
a[0] = 137
a[1] = 234
a[2] = 278
a[3] = 352
a[4] = 366
a[5] = 379
a[6] = 437
a[7] = 469
a[8] = 879
a[9] = 974
report cc ~/nit-info-proc-2-S1/src/build/cls10 $
```

図2: 課題 2 の実行結果

3.2 処理の流れ

3.2.1 bsort 関数

1. 値の一時保持用の変数 i を 0 で宣言・初期化する
2. ループカウンタ i の for ループ内で n - 1 回繰り返す:
   （a） ループカウンタ j の for ループ内で n - 1 から i まで繰り返す:
      i. もし, (v + j - 1) が指す値が (v + j) が指す値より大きければ, 交換する

3.2.2 main 関数

1. 長さ ARR_LEN の配列 a を 0 で宣言・初期化する
2. 標準入力から配列 a に各要素の値を取り込む
3. bsort 関数を引数に a と ARR_LEN を持たせて呼び出す
4. ソートの結果を標準出力に表示する

3.3 書く際の難点

既存の関数を配列表記からポインタ表記にするだけなのであっさりと書けた.

3.4 想定されるバグ

ヌルポインタの検出がないので引数 v にヌルポインタを渡すとヌルポインタ参照でセグメンテーションフォルトを引き起し, プログラムがクラッシュする.