

2018年度 計算機システム(演習)
補足
2018.12.18

遠藤 敏夫(学術国際情報センター/数理・計算科学系 教授)
野村 哲弘(学術国際情報センター/数理・計算科学系 助教)

第2回 p.21

add 関数

sample23.c

```
int add(struct node *head, int val) {
```

```
    struct node *new;
```

```
    new = (struct node *)malloc(sizeof(struct node));
```

```
    if (new == NULL) return -1;
```

```
    new->val = val;
```

```
    // head->nextがNULLかチェック
```

```
    new->next = head->next;
```

```
    head->next = new;
```

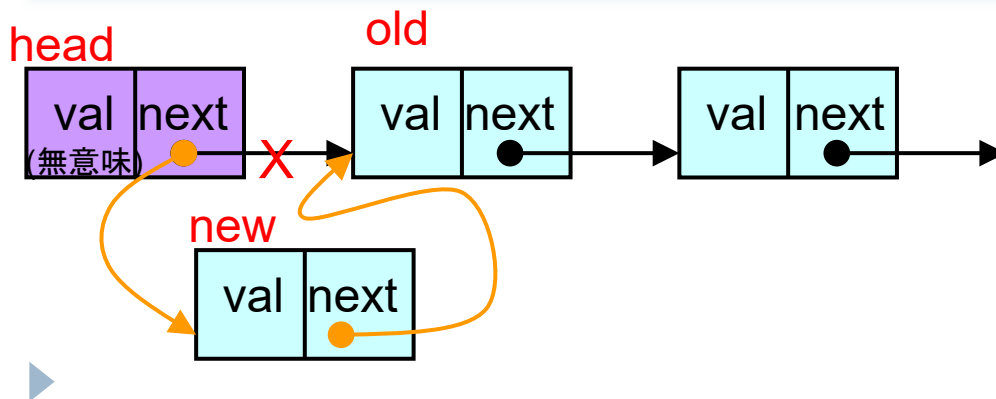
```
    return 0;
```

```
}
```

メモリ確保

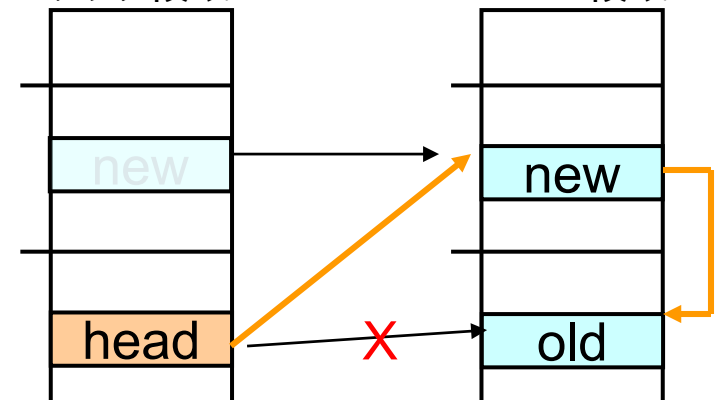
確保されているかしっかりチェックすること

Q: これ要るの?



スタック領域

ヒープ領域



head->next のNULLチェックは必要か

- ▶ A: 要りませんでした、ごめんなさい
 - ▶ スライドの例は単方向リストなので、追加時にhead->nextの中身をいじる必要がなく、NULLでもやることは変わらない
 - ▶ たとえ双方向リストでも、head->nextが(リスト操作関数を使っている範囲で)NULLになりえないのであれば、分岐する必要はない(あえて分岐するとすればエラー処理)

第3回 p.36

課題2: 2つの配列の要素の和

- ▶ これら2つの配列の i 番目と $3-i$ 番目の要素どうしを足し合わせ、表示せよ
- ▶ .wordとして、2つの配列を定義
 - ▶ $A = \{ 1, 2, 3, 4 \}$
 - ▶ $B = \{ 5, 6, 7, 8 \}$
- ▶ 注意点
 - ▶ $A[0] + B[3], A[1] + B[2], A[2] + B[1], A[3] + B[0]$
 - ▶ ループ処理で実装すること
 - ▶ syscallを使ってコンソールに表示すること
- ▶ オプション課題 (課題2-2)
 - ▶ 任意長の配列で同様の動作を実現するプログラムを書け
 - ▶ 配列長の与え方は各自で定義すること

→ 9999

Q: これってどういうこと?

配列長の与え方「各自で定義する」の補足

- ▶ A: 本当に好きに決めてもらって構いません
 - ▶ 本人が「これってアリかよ?」と思うほどに余程変な決め方じゃなければ
 - ▶ 課題の意図としては「配列長が4である」ということに強く依存していないコードを書けますよね という点だけです
- ▶ 例
 - ▶ これを使っていたとしても、これ以外の定義方法でも可

```
A:      .data
        .word 1 2 3
B:
        .word 4 5 6
L: # 配列長
        .word 3
```

```
A:      .data
        .word 1 2 3 -1
B:
        .word 4 5 6 -1
# 配列の終端を-1とする
```

(この例は前回口頭で言いました)

課題締め切り

- ▶ 第01回
 - ▶ 遅れても減点しますが受け付けます
- ▶ 第02回
 - ▶ 12/21 (金)
- ▶ 第03回
 - ▶ 1/8 (火)
- ▶ 第04回
 - ▶ 1/8 (火)